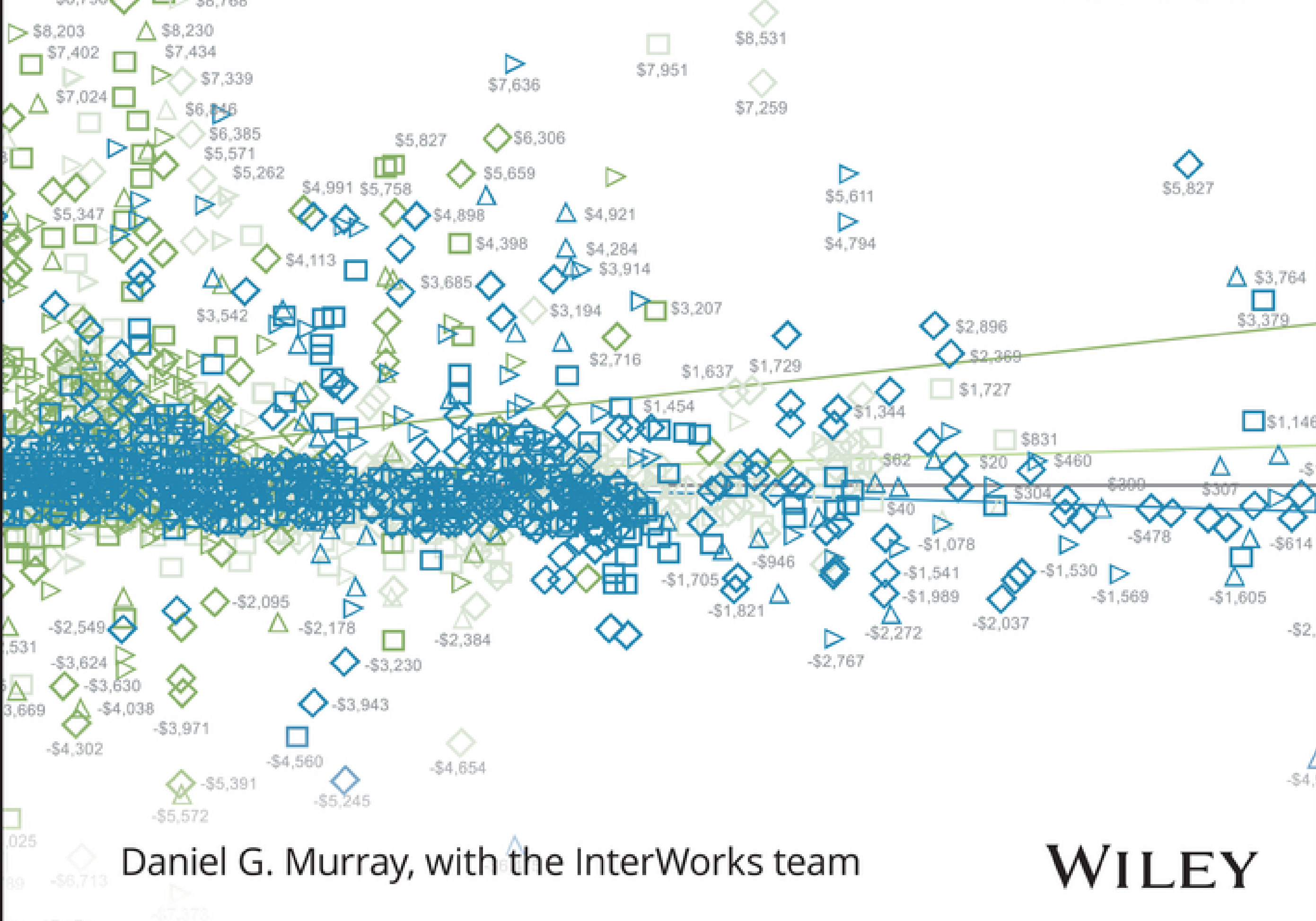
Tableau Your Data!

Fast and Easy Visual Analysis with Tableau Software®

2nd Edition



Daniel G. Murray, with the InterWorks team

WILEY

Tableau Your Data!

Tableau Your Data!

Fast and Easy Visual Analysis with Tableau Software®

Second Edition

Daniel G. Murray

WILEY

Tableau Your Data! Fast and Easy Visual Analysis with Tableau Software®

Published by

John Wiley & Sons, Inc.
10475 Crosspoint Boulevard
Indianapolis, IN 46256
www.wiley.com

Published simultaneously in Canada

ISBN: 978-1-119-00119-5
ISBN: 978-1-119-00121-8 (ebk)
ISBN: 978-1-119-00120-1 (ebk)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

Library of Congress Control Number: 2015952725

About the Author

Daniel G. Murray has over 30 years of professional experience. Dan has seen firsthand the technical revolution in data that led to the creation of Tableau Software. Prior to starting the InterWorks Tableau/BI practice in 2008, he held a variety of increasingly responsible roles in finance, accounting, sales, and operations for a mid-sized global manufacturing company serving the heavy industry and construction markets. During the late 1990s, his employer acquired over 50 companies. Dan's role in 2006 as the CFO/CIO led to an assignment to integrate and create a global reporting environment. Uninspired by the high cost and complicated products available from traditional vendors, Dan discovered Tableau Software through data visualization expert Stephen Few. Less than one month after downloading a trial license of Tableau Software, Dan and his team were able to successfully create a reporting platform for less than 15 percent of the cost and one-tenth the time that traditional vendors had quoted. At this point, it was apparent that everyone needed Tableau—they just didn't know it yet.

Within months after speaking at Tableau's first customer conference, Dan went to friend and founder of InterWorks, Inc., Behfar Jahanshahi, to convince him to allow Dan to form a boutique consulting team focusing on providing the best practices of data visualization and reporting using Tableau Software and any emerging or popular database. Since the publication of the first edition of *Tableau Your Data!* Dan has visited 50 cities across North America and Europe presenting over 70 speeches on data and data visualization.

InterWorks, Inc. is now the premier Gold Professional Consulting Partner for Tableau Software with clients all over the world and over 35 Tableau consultants providing data visualization, database, and hardware expertise to many of the most significant organizations—spanning business, education, and government.

Dan is a 1982 graduate of Purdue University's Krannert School of Business. He and his family live in the metro Atlanta area.

About the Technical Editor

Dick Holm is a successful, longtime entrepreneur with expertise in data analysis, statistics, and product positioning and presentation. Dick has been interested in the graphical presentation of information since the age of four when he learned to write his name in the Minnesota snow. He founded his own business based on presenting visual process information to machine operators. It eventually became a $10 million company. He now spends several hours a day with Tableau Desktop on his screen and a smile on his face.

Credits

Project Editor
Adaobi Obi Tulton

Technical Editor
Dick Holm

Production Editor
Rebecca Anderson

Copy Editor
Nancy Rapoport

Manager of Content Development and Assembly
Mary Beth Wakefield

Marketing Director
David Mayhew

Marketing Manager
Carrie Sherrill

Professional Technology & Strategy Director
Barry Pruett

Business Manager
Amy Knies

Associate Publisher
Jim Minatel

Project Coordinator, Cover
Patrick Redmond

Compositor
Maureen Forys, Happenstance Type-O-Rama

Proofreader
Kim Wimpsett

Indexer
Johnna VanHoose Dinse

Cover Designer
Wiley

Cover Image
Courtesy of Daniel G. Murray

Acknowledgments

When I first approached this edition, I mistakenly thought that writing an updated edition would be easier than writing the original book. I've discovered that writing a guidebook for Tableau is never going to be easy. Tableau's product evolution has been consistently aggressive. During the last two years, Tableau added a slew of new features to Desktop and Server. Consequently, nearly every figure in this book has been updated, and there are over 200 pages of totally new content.

During the last two years, I received lots of feedback from readers of the first edition. That feedback was universally positive and helpful. One individual went out of his way to document improvement ideas by sending me multiple pages of detailed feedback. Dick Holm's feedback was so good I asked him if he would be willing to provide technical editing on this edition. Dick agreed, and his feedback informed every chapter of Part I. I don't want to fail to mention Molly Monsey of Tableau Software. Molly's contribution to the first edition of the book was extensive and invaluable.

This challenging project would have been impossible without the help and support of the InterWorks team. James Wright provided an excellent first draft of Chapter 9, "Designing for Mobile." Kate Treadwell drafted the new Chapter 10 on Story Points. Our resident server wizard, Brad Fair, updated the initial draft for Chapter 11, "Installing Tableau Server." Mat Hughes updated the initial draft of Chapter 12, "Managing Tableau Server." Eric Shairla, Javod Khalaj, and Greg Nelms provided the first draft of Chapter 13, "Automating Tableau Server," which included new code samples for illustrating some of the new capabilities of Tableau's expanding API toolsets.

InterWorks has many wonderful clients, but I would specifically like to thank Donna Costello of Cigna Healthcare for inviting me to speak at one of her Tableau User Group meetings and providing a case study from Cigna's internal user group. As Tableau's first successful partner, we've learned a lot about how to effectively deploy Tableau. InterWorks East Coast Team Leader James Wright recaps key success factors in his case study that shares lessons learned.

After writing the first edition of this book I had the honor of speaking at over 50 Tableau User Groups all over North America and Europe—meeting hundreds of Tableau users. Their enthusiasm reminds me of how I felt about Tableau Software when I first discovered it in 2007. One of the best things about

Tableau Software is the community of people who use the product. Your Tableau Public posts and generosity inspire me. The Tableau Zen Masters are a great example of this sharing. I want to thank all of them for the insights they have shared with the community.

This book wouldn't have been possible without the faith and support of InterWorks CEO Behfar Jahanshahi. Behfar believed in a big idea with a little following in 2007. His strategic and tactical leadership continues to shape our success. His wisdom, insight, and kindness continue to inspire me.

Contents at a Glance

Contents

Introduction: Overview of the Book and Technology

This book aims to provide an introduction to Tableau in the context of the needs of enterprises—large and small. With every Tableau deployment, there are several user constituencies—report designers who are responsible for performing analysis and creating reports, information technology team members who are responsible for managing Tableau Server and maintaining good data governance, and the information consumers who use the output and may want to do their own report creation.

This book's goal is to provide each group with a basic introduction to Tableau's Desktop and Server environments while also providing best practice recommendations that encompass novice, intermediate, and advanced use of the software.

HOW THIS BOOK IS ORGANIZED

There are four distinct sections. Part I (Chapters 1–10) covers the basics of Tableau Desktop and then progresses to more advanced topics including best practices for building dashboards to ensure they are understandable to end users, load quickly, and are responsive to query requests made by your audience. A lot of new and updated content had to be created in Chapters 1–5 because of interface enhancements and added capability in the tool. Chapter 6 includes expanded content related to data discovery and editing content on Tableau Server. Chapter 7 includes new tips and tricks that relate to V9+. Chapter 8's detailed dashboard example has been edited to include an additional action to create a floating and disappearing chart. Tablet computers have become ubiquitous and Tableau has added more capability in its web/mobile platform, so this edition includes a new Chapter 9 on mobile design and consumption with additional content. Chapter 10 on Story Points is also a completely new chapter.

Part II (Chapters 11–13) focuses on Tableau Server, mostly from the perspective of a technology manager responsible for installing, securing, and maintaining the Tableau Server environment. Tableau Server Version 9 is one of the most significant upgrades to Tableau Server ever. Not only has the user interface been redesigned and improved, but the backend processes have been enhanced, and Tableau's API toolset has been expanded. Consequently, every chapter in Part II contains significant content updates and additions.

Part III (Chapter 14) includes a new case study that provides tips for ensuring successful Tableau deployments. Cigna Healthcare created an internal user group aimed at improving employee skills and building enthusiasm around their Tableau deployment.

Part IV (Appendixes A–G) provides additional details on Tableau Software's current product ecosystem, supported data connections, keyboard shortcuts for Windows and the Mac, and recommended hardware configurations, and also includes a detailed Tableau function reference that provides explanations for function syntax and code samples. Some of these sections were included in the chapter content in the first edition. Due to their dynamic nature of the material, it made sense to break them out into separate appendixes so that we could include the most up-to-date information as close to publication as possible.

It's a challenge writing a book on this product line. Tableau Software's research and development spending reached an all-time high last year, and Tableau's management team remains committed to 12–15 month major release cycles. There isn't a lot of time to update a guide that is 600+ pages.

WHO SHOULD READ THIS BOOK?

This book is intended to introduce new users to the features that Tableau Desktop has to offer from the perspective of someone who needs to create new analysis or reporting. It is also intended for staff responsible for installing, deploying, and maintaining Tableau Server.

The chapters related to Tableau Server are more technical because the subject matter assumes that you have a grasp of server terminology and security.

You can read the book sequentially from start to finish. Or, you can skip around and read about a topic of particular interest. Each chapter builds on the previous material, but if you've already mastered the basics of connecting and using the Desktop, you can skip any chapter related to Tableau Desktop and focus on topics of interest. Care has been taken to cross-reference related topics. So if you are the type of reader who wants to skip to a particular topic, breadcrumbs have been dropped so you can efficiently find related topics.

TOOLS YOU WILL NEED

You can read the book without having Tableau Software installed on your computer, but you'll get a lot more from the material if you follow the examples yourself. Tableau provides free trials of the software. Alternatively, you can download Tableau Public for free, indefinitely—all of the book examples related to Tableau Desktop should work on Tableau Public.

WHAT'S ON THE COMPANION WEBSITE?

Tableau constantly updates the Desktop and Server products with multiple maintenance releases and at least one major product release every 12 to 15 months. The book's companion website includes articles related to the releases, sample files related to the book's examples, and also examples related to new capabilities added to the product as Tableau makes them available. The InterWorks team actively tests new Tableau products, so the companion website may also include demonstrations of new visualization types or techniques before they become available publicly.

Wiley also has a website dedicated to the book that you can find at `www.wiley.com/go/tableauyourdata2e`.

SUMMARY

Tableau lowers the technical bar for accessing data from many different data sources. This book should allow you to advance your technical ability and save time deploying Tableau in your enterprise by enabling you to make better decisions earlier in your deployment.

Tableau Your Data!

PART I

DESKTOP

In this part

CHAPTER 1

Creating Visual Analytics with Tableau Desktop

Data graphics should draw the view's attention to the sense and substance of the data, not to something else.

—EDWARD R. TUFTE[1]

The seeds for Tableau were planted in the early 1970s when IBM invented Structured Query Language (SQL) and later in 1981 when the spreadsheet became the killer application of the personal computer. Data creation and analysis fundamentally changed for the better. Our ability to create and store data increased exponentially.

The business intelligence (BI) industry was created with this wave, each vendor providing a product "stack" based on some variant of SQL. The pioneering companies invented foundational technologies and developed sound methods for collecting and storing data. Recently, a new generation of NoSQL[2] (Not Only SQL) databases are enabling web properties like Facebook to mine massive, multi-petabyte[3] data streams.

Deploying these systems can take years. Data today resides in many different databases and may also need to be collected from external sources. The traditional leaders in the BI industry have created reporting tools that focus on rendering data from their proprietary products. Performing analysis and building reports with these tools require technical expertise and time. The people with the technical chops to master them are product specialists who don't always know the best way to present the information.

The scale, velocity, and scope of data today demand reporting tools that deploy quickly. They must be suitable for non-technical users to master. They should connect to a wide variety of data sources. And, the tools need to guide us to use the best techniques known for rendering the data into information.

THE SHORTCOMINGS OF TRADITIONAL INFORMATION ANALYSIS

Entities are having difficulty getting widespread usage of traditional BI tools. A recent study by the Business Application Research Center (BARC, 2009) reported adoption rates are surprisingly low.[4]

In any given BI using organization just over 8 percent of employees are actually using BI tools. Even in industries that have aggressively adopted BI tools (e.g., wholesales, banking, and retail), usage barely exceeds 11 percent.

NIGEL PENDSE, BARC

In other words, 92 percent of the people who have access to traditional BI tools don't use them. The BARC survey noted these causes:

- The tools are too difficult to learn and use.
- Technical experts were needed to create reports.
- The turnaround time for reports is too long.

Companies that have invested millions of dollars in BI systems are using spreadsheets for data analysis and reporting. When BI system reports are received, traditional tools often employ inappropriate visualization methods. Stephen Few has written several books that illuminate the problem and provide examples of data visualization techniques that adhere to best practices. Stephen also provides examples of inappropriate visualizations provided by legacy vendor tools.[5] It turns out that the skills required to design and build database products are different from the skills needed to create dashboards that effectively communicate. The BARC study clearly indicates this IT-centric control model has failed to deliver compelling answers that attract users.

You want to make informed decisions with reliable information. You have to connect with a variety of data sources and may not know the best ways to visualize the data. Ideally, the tool used should automatically present the information using the best practices. Tableau has become a popular choice because it makes industrial-strength reporting, analysis, and discovery accessible to less-technical staff. During the last few years, information technology teams have started to embrace end-user empowerment because it provides a more efficient way to provide information, reduces request backlogs, and provides a toolset for leveraging the knowledge of constrained technical human resources.

THE BUSINESS CASE FOR VISUAL ANALYSIS

Whether your entity seeks profits or engages in non-profit activities, all enterprises use data to monitor operations and perform analysis. Insights gleaned from the reports and analyses are then used to maintain efficiency, pursue opportunity, and prevent negative outcomes. Supporting this infrastructure (from the perspective of the information consumer) are three kinds of data.

THREE KINDS OF DATA THAT EXIST IN EVERY ENTITY

Reports, analysis, and ad hoc discovery are used to express three basics kinds of data.

Known Data (Type 1)

Encompassed in daily, weekly, and monthly reports that are used for monitoring activity, these reports provide the basic context used to inform discussion and frame questions. Type 1 reports aren't intended to answer questions. Their purpose is to provide visibility of operations.

Data You Know You Need to Know (Type 2)

Once patterns and outliers emerge in type 1 data, the question that naturally follows is: Why is this happening? People need to understand the cause of the outliers so that action can be taken. Traditional reporting tools provide a good framework to answer this type of query as long as the question is anticipated in the design of the report.

Data You Don't Know You Need to Know (Type 3)

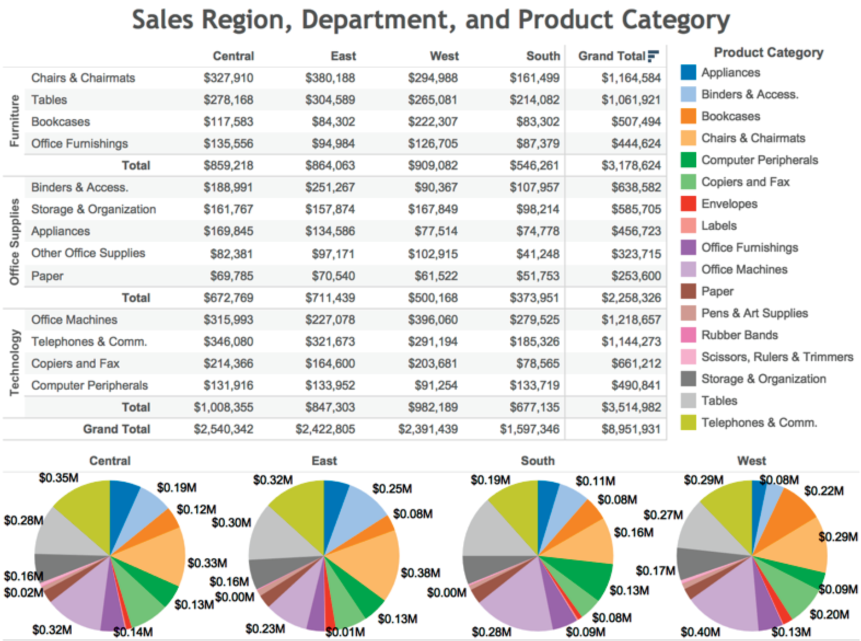
Performing analysis with data in real time while using appropriate visual analytics provides the possibility of seeing patterns and outliers that are not visible in type 1 and type 2 reports. The process of interacting with granular data yields different questions that can lead to new actionable insights. Software that enables quick, iterative analysis and reporting is becoming a necessary element of effective business information systems.

Distributing type 1 reports in a timely manner is important. This requires speedy design and build stages when a new type 1 report is created. To effectively enable types 2 and 3 analyses, the reporting tool must adapt quickly to ad hoc queries and present the data in intuitive ways.

HOW VISUAL ANALYTICS IMPROVES DECISION MAKING

Rendering data accurately is easy to achieve with Tableau, but your knowledge of the best practices enhances the clarity of the information being displayed. The next three figures illustrate how the choice of chart types can make it easier for your audience to see and understand important findings in the data. The goal of these examples is to provide sales analysis by region, product category, and product subcategory.

Figure 1-1 presents data using a grid of numbers (a text table) and pie charts. Text tables are useful for finding specific values. Pie charts are intended to show part-to-whole comparisons. The pie charts compare sales by region and product category.


Sales Region, Department, and Product Category

		Central	East	West	South	Grand Total
Furniture	Chairs & Chairmats	$327,910	$380,188	$294,988	$161,499	$1,164,584
	Tables	$278,168	$304,589	$265,081	$214,082	$1,061,921
	Bookcases	$117,583	$84,302	$222,307	$83,302	$507,494
	Office Furnishings	$135,556	$94,984	$126,705	$87,379	$444,624
	Total	$859,218	$864,063	$909,082	$546,261	$3,178,624
Office Supplies	Binders & Access.	$188,991	$251,267	$90,367	$107,957	$638,582
	Storage & Organization	$161,767	$157,874	$167,849	$98,214	$585,705
	Appliances	$169,845	$134,586	$77,514	$74,778	$456,723
	Other Office Supplies	$82,381	$97,171	$102,915	$41,248	$323,715
	Paper	$69,785	$70,540	$61,522	$51,753	$253,600
	Total	$672,769	$711,439	$500,168	$373,951	$2,258,326
Technology	Office Machines	$315,993	$227,078	$396,060	$279,525	$1,218,657
	Telephones & Comm.	$346,080	$321,673	$291,194	$185,326	$1,144,273
	Copiers and Fax	$214,366	$164,600	$203,681	$78,565	$661,212
	Computer Peripherals	$131,916	$133,952	$91,254	$133,719	$490,841
	Total	$1,008,355	$847,303	$982,189	$677,135	$3,514,982
	Grand Total	$2,540,342	$2,422,805	$2,391,439	$1,597,346	$8,951,931

FIGURE 1-1 *Sales mix analysis using a text tables and pie charts*

Text tables are not the most effective way to make part-of-whole comparisons or identify outliers. Pie charts are a commonly used chart type but are one of the least effective ways to make precise comparisons. This is especially true when there are many slices that are similar in size or very small.

Figure 1-2 employs a bar chart and heat map to convey the same information. Bar charts provide a better means for making precise comparisons. The linear presentation makes it easier to see the relative values. The heat map on the right provides total sales for each product category. The grayscale background color in the heat map highlights the high and low selling items. The blue-orange color encoding in the bar chart provides additional information on profit ratio. More importantly, this color scheme is visible to color-blind people.

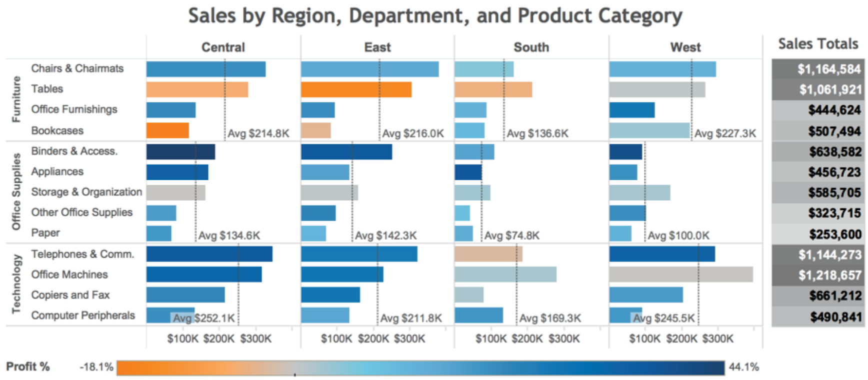


FIGURE 1-2 *Sales mix analysis using a bar chart and heat map*

The bar chart and heat map communicate the relative sales values more quickly while adding profit ratio information with the use of color. The reference lines provide average sales values for each region and department. One could argue that the bar chart doesn't communicate the details available in the text table, but in Figure 1-3 those details are provided via tooltips that pop out when you point your mouse at the mark of interest.

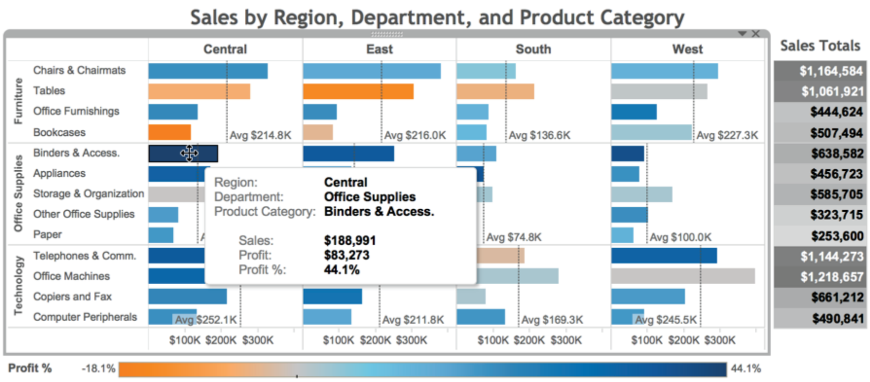


FIGURE 1-3 *Tooltips provide details on demand.*

Appropriate visual analytics improve decision making by highlighting the important outliers and making them easier to see. Interactive analytics make those details come to life by allowing your audience to explore areas of interest.

TURNING DATA INTO INFORMATION WITH VISUAL ANALYTICS

Data that is overly summarized loses its ability to inform. When it's too detailed, rapid interpretation of the data is compromised. Visual analytics bridges this gap by providing the right style of data visualization and detail for the situational need. The ideal analysis and reporting tool should possess the following attributes:

- **Simplicity:** Be easy for non-technical users to master.
- **Connectivity:** Seamlessly connect to a large variety of data sources.
- **Visual competence:** Provide appropriate graphics by default.
- **Sharing:** Facilitate sharing of insight.
- **Scale:** Handle large datasets.

Traditional BI reporting solutions aren't adapted for the variety of data sources available today. Analysis and reporting can't occur in these tools until the architecture is created within the proprietary product stack. Tableau Software was designed to address these needs.

ANALYSIS AS A CREATIVE PROCESS

Tableau's design encourages interaction with data because the information consumer gets immediate feedback. Tableau's CEO, Christian Chabot, talked about this in his keynote address at Tableau's 2014 customer conference in Seattle.[6] Chabot compared data analysis to artistic expression and explained that the way artists go about creating art (through trial, error, and refinement) is informing Tableau's design.

- Encouraging experimentation
- Giving speedy feedback
- Providing an expressive environment
- Giving users control

The most impactful Tableau analyses are discovery stories. Sometimes discoveries translate into millions of dollars of improved profit for Tableau users. In Chapter 6, you'll learn proven discovery techniques that will provide some examples for you to apply, refine, and develop into your own discovery methods.

Each new release of Tableau Desktop improves its reporting, analysis, and discovery capabilities. The next section provides a brief introduction into Tableau's product line.

TABLEAU'S DESKTOP TOOLS

Tableau Desktop is the design tool for creating visual analytics and dashboards. There are two versions: Personal edition and Professional edition. Professional is more popular because it connects to a wider variety of data sources than Personal. If there isn't a connector for the specific data you need to access, you can connect to it via the Open Database (ODBC) standard.

Tableau also provides a free consumption tool called Tableau Reader.

TABLEAU DESKTOP PERSONAL EDITION

Tableau Desktop Personal edition is the entry-level design tool that connects to data sources located in a file on your personal computer. It supports data connections to Excel, Access, text files (`.csv`), OData, Microsoft Windows Azure Marketplace DataMarket, and Tableau Data Extract (`.tde`) files. You can also import workbook files from other Tableau workbooks.

PROFESSIONAL EDITION

Tableau Desktop Professional edition is similar to the Personal edition but offers a much wider range of data connections. In addition to the "In a file" options provided by the Personal edition, you have the ability to connect to a wide variety of database files. Desktop Professional connects to relational databases, columnar-analytic databases, data appliances, NoSQL data sources, web-service APIs, and other ODBC 3–compliant data sources. See Appendix B for a complete listing of connections for the Windows and Mac OS X editions.

TABLEAU FILE TYPES

You can save and share data using a variety of different file types in Tableau. The differences between each file type concerns the amount and type of information being stored in the file. Table 1-1 summarizes different Tableau file types.

When you save your work in Desktop, the default save method creates a workbook (`.twb`) file. If you need to share your work with people who don't have a Tableau desktop license or don't have access to the data source, you can save your work as a packaged workbook (`.twbx`) by using the Save As option when saving your file.

TABLE 1-1 Tableau File Types

FILE TYPE (EXTENSION)	SIZE	USE CASE	INCLUDES
Tableau Workbook (.twb)	Typically small	Tableau's default way to save work. Normally used to save your data connection to your local or database data sources.	Information to visualize data. Tableau metadata. No source data.
Tableau Data Source (.tds)	Typically small	An exported data source that you may connect to frequently. Access to saved, frequently used data sources.	Metadata, including data source type, connection information, groups, sets, calculated fields, bins, and default field properties.
Tableau Data Extract (.tde)	Potentially large	Improves performance. Enables more functions. Allows for offline access.	Source data and workbook metadata as filtered and aggregated during extract.
Tableau Book Mark (.tbm)	Typically small	Sharing sheets between different workbooks.	Information to copy visualizations and the data source(s) used.
Tableau Packaged Workbook (.twbx)	Ranges from small to potentially large depending on the size of the source data	Sharing with Tableau Reader or those without access to the source data.	Extracted data and workbook information to build visualizations.
Tableau Packaged Data Source File (.tdsx)	Potentially large	Sharing with people who don't have access to the original source files, typically via publishing to Tableau Server.	Contains all of the information in a `.tds` file as well as any local file data sources for those who do not have access to the local files.

Tableau Data Sources (`.tds`) are useful when you frequently connect to a particular data source or you have edited the metadata associated with that data source in some way (renaming or grouping fields, for example). Using saved data sources reduces the time required to connect to the data.

Tableau Data Extracts (`.tde`) leverage Tableau's proprietary data engine. When you create an extract, your data is compressed. In versions prior to V8.2, if your data source is from a file (Excel, Access, text), Data Extracts add formula functions that don't exist in those sources—including `Count Distinct and Median`. Beginning with V8.2, the Data Extract is no longer necessary to gain these functions because connections using the newer optimized connection create a local temporary extract automatically.

If you are publishing workbooks via Tableau Server, Data Extracts provide an effective way to separate the analytical load Tableau generates from your source database.

Tableau Packaged Data Source (`.tdsx`) files provide a means for publishing data extract files (`.tde`) or any filed-based data to Tableau Server. The Data Server can then be scheduled to refresh these files at regular intervals automatically.

TABLEAU READER

Tableau Reader is a free version that allow users to consume Tableau Desktop Reports without the need for a paid license. To make files consumable by Tableau Reader, save your workbook as a Tableau Packaged Workbook (`.twbx`) file.

TABLEAU ONLINE HELP

I hope that while reading this book, you have Tableau Desktop running nearby and go to it often to work on examples, explore other options, and access Tableau Online Help (TOH). TOH is an excellent repository of information that is frequently updated by experienced practitioners. If something in this text isn't making sense to you, try searching the TOH for another perspective on the topic.

INTRODUCING THE TABLEAU DESKTOP WORKSPACE

This book is intended as a supplement (not a replacement) for Tableau's online manual. If you are running Tableau Desktop while you are reading, you should try pressing the F1 key in Windows or Shift+Command+? on the Mac and take a look at the online manual. Go to the Tableau Help menu and select the Watch Training Videos option. This will take you to Tableau's training and tutorials website. Watch the introductory video and review the numerous training videos that are sorted by topic. Most of them range from 3 to 20 minutes in length. Using the training videos in combination with this book will jumpstart your Tableau learning and deepen your understanding of more advanced skills.

In the remainder of this chapter, you learn the basics of Tableau's workflow and user interface.

NEW WORKSPACE DESIGN

Tableau Desktop Version 8.2 and earlier versions included a Start page that contained a workspace for connecting to data sources, saving data sources, and saving frequently used workbooks. The Start page also included links to videos and sample workbooks.

The Start page in Tableau Desktop Version 9 has been improved to include more links to training and idea resources. Tableau has also made it easier to search the growing number of data sources that you can access. All of the examples in this text reflect Version 9. The figures were created using an iMac or MacBook Pro unless specifically noted otherwise.

For those readers using the Windows version, the menus, functions, and location of controls are virtually identical. The only differences relate to the ways that Windows and OS X handle menu positioning and keyboard shortcuts. A detailed explanation of the differences between the two operating systems is outside the scope of this text. The biggest difference between the Windows and Mac editions relates to the number of connectors available. The Windows edition has more connectors at this time. See Appendix B for a complete list of connections.

Go to Appendix C to see a complete listing of Windows and Mac keyboard shortcuts. You can also search the Tableau manual to find the latest list of keyboard shortcuts.

USING THE START PAGE CONTROLS EFFECTIVELY

If you are accustomed to working with spreadsheets or other analysis tools, learning Tableau's desktop environment will be a breeze. If you have no familiarity with database terminology or spreadsheets, you can still be using Tableau effectively within a few hours.

Tableau's Start page has been redesigned in Version 9 to provide more convenient access to your data, workbooks, and learning resources for new users. The new design of the Start page should also be more appealing to experienced users because it provides convenient access to Tableau's web content and to interesting public content features such as the Tableau Public Viz of the Week.

THE START PAGE

Open Tableau and you are presented with the Start page displayed in Figure 1-4.

Clicking the small tableau logo displayed in the upper left allows you to toggle between the Start page and the Tableau sheet workspace. Think of the Start page as the control center and the sheet workspace as your building environment. The Start page is divided into three sections from left to right.

- Connect pane
- Open pane
- Discover pane

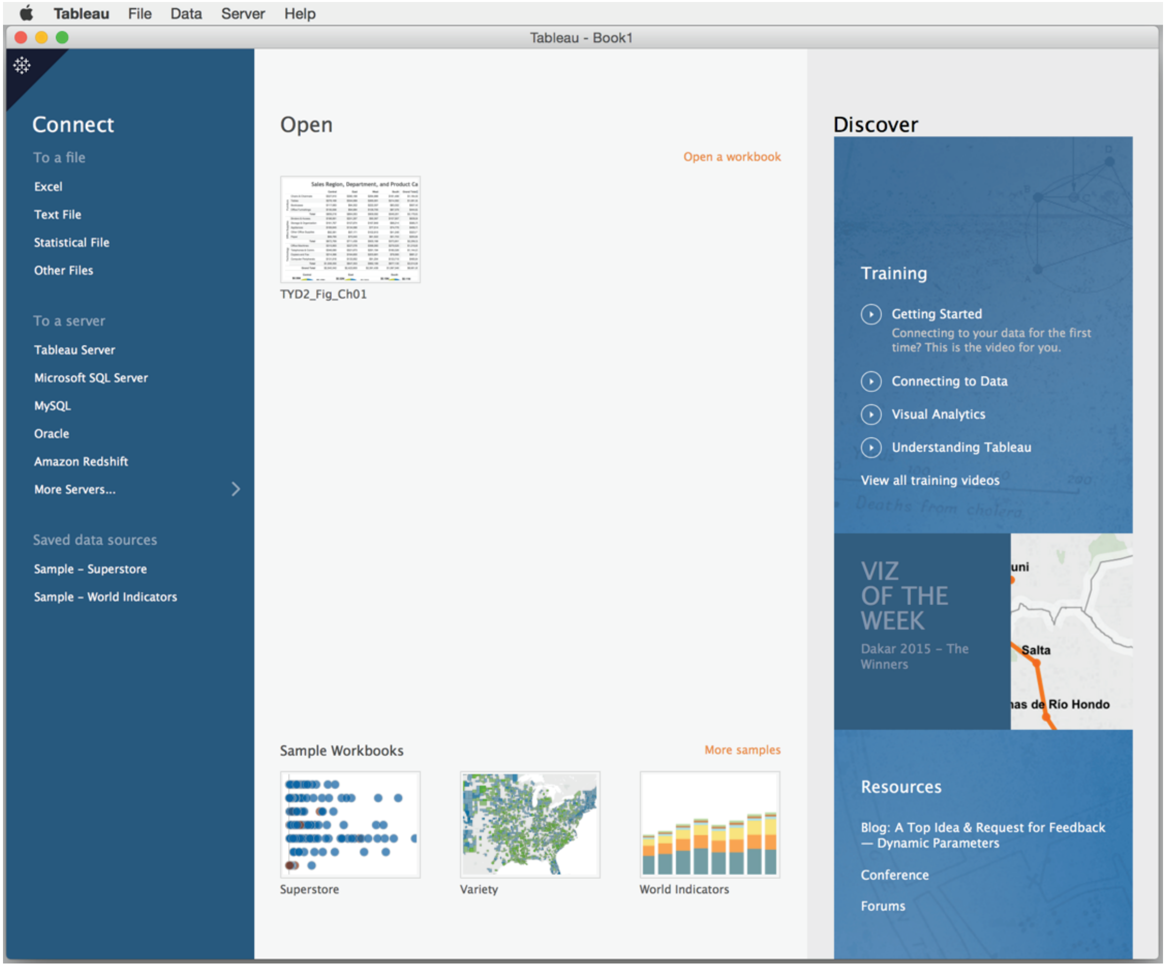


FIGURE 1-4 *Tableau Start page*

In the Connect pane, you connect to several kinds of data sources. The Open pane in the center of the screen displays your nine most recently opened workbooks. It also provides options for connecting to your work files or to sample workbooks provided by Tableau. The Discover pane provides links to training videos, the Tableau Public Viz of the Week, and other links including Tableau's blog, company news, and the popular Tableau user forum.

If you want to follow along in Tableau, select and open one of the saved data sources that are provided. Alternatively, go to this book's companion website (see Appendix F for the address), download the Chapter 1 sample file, and use the File menu option in the Start page to open the sample workbook. This will allow you to see the actual sheets used to create the figures in this chapter.

The Connect Pane

The Connect pane provides convenient access to data stored in files on your computer, data stored in databases, and sample data sources provided by Tableau, as well as to frequently used data sources that you have saved for easy access.

In a File

Look on the top-left side of the Start page right below the Connect heading. This area includes options for connecting files available to you, including Microsoft Excel, Microsoft Access (Windows only), Statistical files, and Text files.

To a Server

You must be using Tableau Desktop Professional to have access to the To a Server connections. If you are using Tableau Desktop Personal edition, you can't connect to data sources in this area. If you are using Tableau Desktop Professional, you should see five connections, starting with Tableau Server. At the bottom of this group of connections is an additional option to connect with more server data sources. Clicking "more servers" will expose more server connection types to a variety of databases and web services.

The order in which server connections appear in this area will change depending on which server connections you use most frequently. In Figure 1-4, Google Big Query appears just below Tableau Server because I've been using that data source recently. The order that connections appear on your connection pane will be different.

Saved Data Sources

At the bottom of the Connect Pane displayed in Figure 1-4 is the Saved data sources area. Tableau allows you to save frequently used data source connections here. This is a big time-saver because you don't have to enter all the connection and security information very time you need to access the saved connections for items that reside in a database on a server. It also provides a convenient way for you to access the files that you use often on your personal computer.

Tableau populates this area with sample data sources that you can use for learning. Any connection that appears has been saved to a folder called My Tableau Repository. This folder is created when you install Tableau Desktop. Don't worry about the details of how these work right now. Be confident that files or connections you choose to save will appear in this area. You learn to connect to and save data sources in Chapter 2.

The Open Pane

If you've just installed Tableau Desktop, the white space to the right of the Connect pane will be empty on your screen. This area is used to display your most recently opened workbooks. In Figure 1-4, there are two workbook images. These are workbooks on my laptop that includes the sheets used to create the content for this chapter and another workbook created from a Google BigQuery database. If you have made any workbooks, they should appear in this space. Once you start building workbooks, you will continuously cycle through the last nine workbooks in this space. You can also pin frequently used workbooks to this area.

Pin or Delete a Workbook from the Save Workbooks Area

In Figure 1-5, you see the options for pinning workbooks to this area or deleting workbooks from the Connect page. When you want to pin or unpin a workbook, hover your mouse pointer over the blue pin icon in the image of the workbook you want to pin, as you can see on the left workbook displayed in Figure 1-5. Clicking the blue pin will keep the workbook displayed in this space. Clicking that blue pin again will unpin the workbook and allow more recent work to displace it in the space.

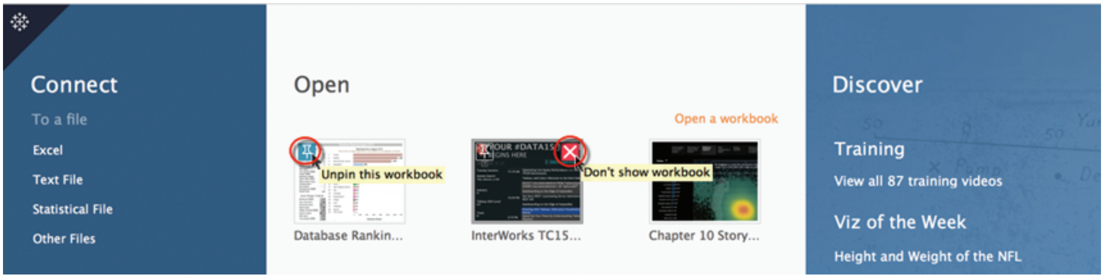


FIGURE 1-5 *Pin a workbook to the Start page.*

To see how to remove a workbook from this space, look at the right image in Figure 1-5. Hovering your mouse pointer over the red X button and clicking that deletes the workbook image from this area. This will not delete the workbook for your hard disk—only the image in the workbook Open pane.

Sample Workbooks

At the bottom of the Open pane in Figure 1-4, you will find the Sample Workbooks area. These are workbooks that are included with Tableau when you install Desktop. They were created by Tableau as a learning resource.

More Samples

Clicking the More samples link at the bottom right of the Open pane shown in Figure 1-4 will take you to a Tableau web page that includes more dashboards and workbooks that you can download and explore. This is a great way to get ideas for your own work.

The Discover Pane

On the right side of the Start page, you will find online learning resources that are provided by Tableau Software. This pane is divided into three sections: Training, Viz of the Week, and Resources.

Training (View All)

The most popular training videos are displayed in this space. Clicking the (view all) option will take you to Tableau's training and tutorials website that includes links to all of Tableaus training resources. These include a large library of free On-Demand training videos, a schedule of upcoming live web training sessions that are also free, and a schedule of Tableau's Public training classes, which are fee-based. In addition, you will find links to quick-start guides, starter kits, knowledge base articles, and other learning resources. The content in this area continues to change and expand.

Viz of the Week

Tableau offers a free hosting service that allows data nerds who also blog to share visualizations. This service is called Tableau Public. The Viz of the Week is selected from the best examples provided by the Tableau Public user community. Clicking the Viz of the Week takes you to the Tableau Public website and the winning visualization. Normally Tableau provides a link to the website or blog that the dashboard appeared. You can also download the workbook to your computer for further analysis. This is a great way to learn.

Resources

At the bottom of the Discover pane, you'll find links to recent Tableau blog posts, Tableau news, the Tableau Conference website, and the excellent Tableau community forums. One of the best aspects of the Tableau world is the community of users who share technique. As Tableau's website has grown in size and scope, it can be challenging to find what you need. The Resources area provides shortcuts to these resources right from your desktop. Keep in mind that you need a live Internet connection to access these files.

Now that you've learned about Tableau's Start page, it's time to dive into some data. In the next section, you learn about Tableau's primary connection and analysis environment—the sheet workspace.

THE TABLEAU DESKTOP WORKSPACE

Figure 1-6 contains a sheet view of a connection to one of the sample files that ships with Tableau Desktop: the Sample-Superstore Sales-Excel dataset. Tableau updates these sample files often. Once again, the companion website (see Appendix F) includes a zip file containing the workbooks used to create the examples in this book. If you want to follow along with these examples, download the file for Chapter 1.

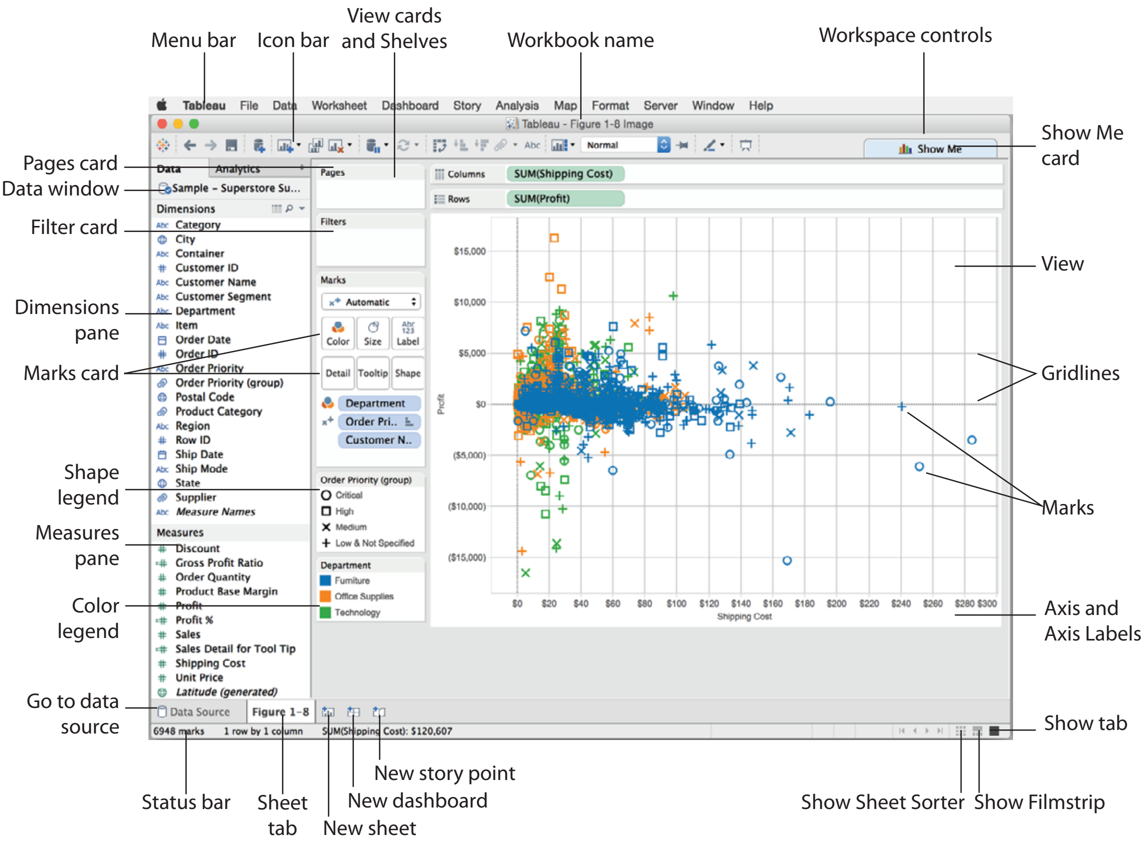


FIGURE 1-6 *Tableau Desktop sheet workspace*

The sheet workspace is where you build views. Tableau Desktop also has a dedicated workspace for dashboard building, which is covered in Chapter 8, and for creating stories, which is covered in Chapter 10. Figure 1-6 shows a sheet containing a scatter plot. The Superstore dataset includes sales data for a retail business.

You can open as many connections as you want in Tableau by clicking the home icon to open the Connect page. There are a several other ways you can establish a data connection, as you will learn in Chapter 2. The remainder of Chapter 1 covers the different parts of the workspace displayed in Figure 1-6.

The toolbar areas at the top of the sheet view contain the menu bar, the icon bar, and the Show Me button.

The Sheet Sorter

At the bottom right of the workspace in Figure 1-6 are three icons. The left-most icon in the group is a cluster of nine squares. Clicking the Show Sheet Sorter icon will display all of the contents of the workbook. Figure 1-7 shows the Sheet Sorter view.

The Sheet Sorter resembles a PowerPoint slide desk sorter view. In the Sheet Sorter, you can reorder worksheets, dashboards, and stories by dragging the images to the desired position. Double-clicking any image in the Sheet Sorter will open that view.

You can also use the Sheet Sorter to preload the contents of your workbook into memory. This is particularly helpful if you are using Tableau to give a live presentation. By preloading everything into memory, transitions between each sheet, dashboard, or story will be immediate. Preload all the workbook contents into memory by right-clicking while pointing anywhere in the Sheet Sorter and selecting the Refresh All Thumbnails option.

Show Filmstrip and Show Tabs Icons

Tableau normally displays tabs at the bottom of the workspace, but you can change that presentation. Immediately to the right of the slide sorter icon in Figure 1-6 are two additional icons—Show Filmstrip and Show Tabs. With these icons, you can toggle between each type of display. Figure 1-8 shows the Filmstrip and Tab displays.

The upper part of Figure 1-8 is the Tab view. On the bottom is the Filmstrip view. Experiment with each method to see how the presentation of the sheets changes. If you hover your mouse over the tabs, an image of the sheet contents is displayed as well.

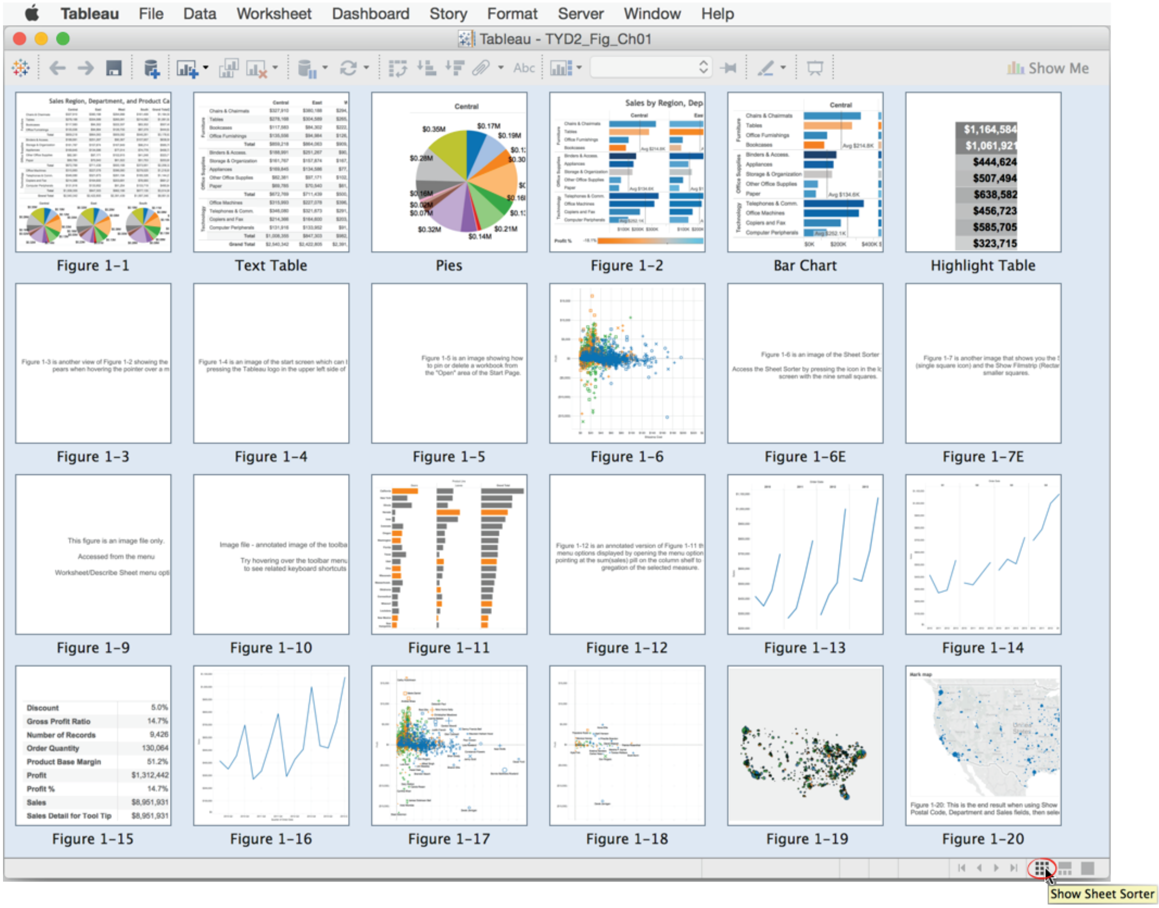


FIGURE 1-7 *The Sheet Sorter*

Return to the Start page by clicking the Tableau logo icon you see in Figure 1-8 in the upper-left side of the workspace. Note that you can also refresh all thumbnails to preload every sheet from the Filmstrip, just as in the Sheet Sorter.

Right-clicking a tab for the Filmstrip icon exposes a context-specific menu option relating to the tab. Notice that you can color-encode tabs from this menu, duplicate the tab, or export the sheet to create a new workbook. Try experimenting with the other menu options exposed when you do this. You'll find that right-clicking any object in Tableau always exposes context-specific menu options.

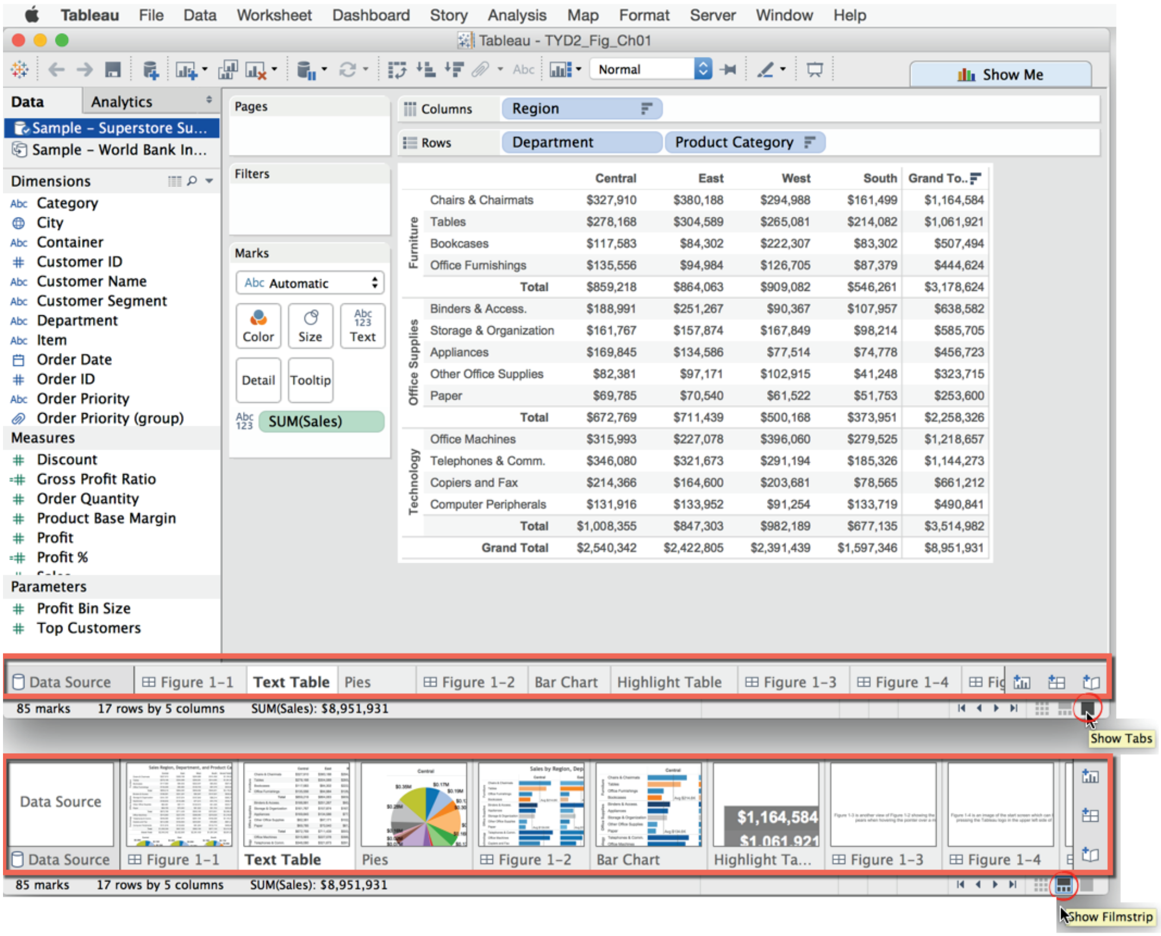


FIGURE 1-8 *Show Tabs and Show Filmstrip*

What You Need to Know About the Toolbar Menu

At the very top of the sheet workspace in Figure 1-6 is the menu. Once again, the appearance of the menu is slightly different on Windows and the Mac, but both Desktop editions contain the same menu options.

As Tableau Desktop has matured, many options that used to be accessible only from the menu have migrated closer to the workspace. Described in the following sections are those options that are accessed primarily or only via the main menu. The text that follows ignores the menu items that are more easily accomplished with other methods or that are very rarely utilized. You can find more details regarding the toolbar in Tableau's online manual found in the Help menu.[7]

File Menu

Like any Windows program, the File menu contains New, Open, Save, and Save As functions. The Export Packaged Workbook option allows you to create a packaged workbook (`.twbx`) quickly. Saving your workbook this way eliminates a couple of clicks versus the more commonly used File/Save As method.

The most frequently used feature found in this menu is the Print to PDF option (a Print menu option on the Mac). This allows you to export your worksheet or dashboard in PDF form. If you can't remember where Tableau places files or you want to change the default file-save location, use the Repository Location menu option to review and change it.

Data Menu

The Paste Data option is handy in a couple of ways. You can use this if you find some interesting tabular data on a website that you want to analyze with Tableau. Highlight and copy the data from the website and then use the Paste Data option to input it into Tableau. Once pasted, Tableau will copy the data from the Windows clipboard and add a data source in the data window.

The Tableau Data Server menu allows you to refresh and append data sources that you have published to Tableau Server. Details related to these features are covered in Chapter 12.

The Edit Relationships menu option is used in data blending. This menu option is necessary if the field names in two different data sources are not identical. It allows you to specifically define the related fields. Chapter 2 covers details related to data blending.

The Replace Data Source is most typically used when you are working with spreadsheet data sources (and the spreadsheet name has been changed) you need to direct Tableau to the new version of the spreadsheet. This feature isn't limited to spreadsheets; it also works for database sources.

Worksheet Menu

Several frequently used features exist in this menu. The Export option allows you to export your worksheet as an image, in an Excel text tables, or to an Access database file. The Actions and Tooltip menu options are covered in detail in Chapter 8. Figure 1-9 shows an example of the output from the Describe Sheet menu option.

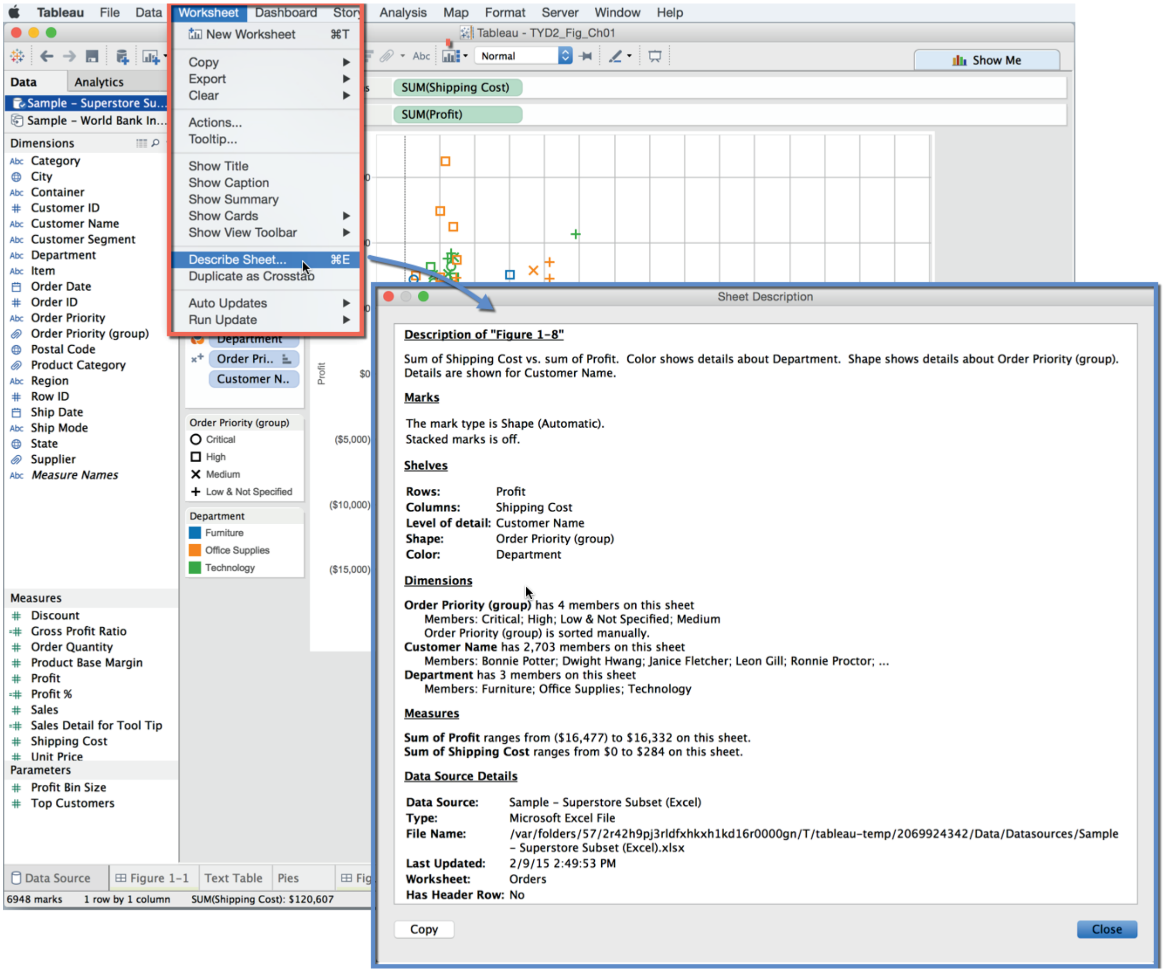


FIGURE 1-9 *Describe Sheet output*

Using the Describe Sheet menu option creates a summary of the sheet metadata that displays summarized information about the sheet contents and source data. Notice that you can copy the output from Describe Sheet and paste it into another application.

Dashboard Menu

The Action menu is a very useful feature that is reached from both the Dashboard menu and the Worksheet menu. Chapter 8 covers the three types of actions in detail.

Story Menu

The Story menu is one way you can start a Story Point view. This feature is covered in detail in Chapter 10.

Analysis Menu

As your skills grow, you'll venture to this menu to access the Aggregate Measures and Stack Marks options. These switches allow you to adjust default Tableau behaviors that are useful if you need to build non-standard chart types. You'll build an example in Chapter 7 that requires the use of these options. The Create Calculated Field and Edit Calculated Field options are used to make new dimensions or measures that don't exist in your data source.

Tableau Desktop Version 9 offers many new features related to calculated values and table calculations. Chapter 4 has been significantly revised to show the new ways you can create calculated values and table calculations.

Map Menu

The Map menu includes options for Background Maps, Background Images, Geocoding, Edit Locations, Map Legends, Map Layers and Map Options.

The Background Maps option is used to alter the base map type that is being used in your sheet view. The None menu option removes the map image from the background but still plots the marks using latitude and longitude for the placement of the data marks. Use the Offline option if you don't have access to the Internet but still want to look at the map view. The offline maps are less-detailed than Tableau's online maps. Select the Tableau Classic map option if you want to add census data to your maps via color-encoded polygon shapes.

Geocoding is used to import custom locations to Tableau. For example, you may want to plot your company office locations on a map. The Edit locations option is used to correct locations in your data that Tableau doesn't automatically recognize. For example, you may have a misspelled city or state name. Selecting the Map Options exposes the Map Options window that will float over the map. The options contained in that menu allow you to control whether the user can pan and zoom, show the map search control or show the view toolbar.

All these options and more are covered in Chapter 5.

Format Menu

You may not use this menu very often because pointing at anything in your workbook and right-clicking gets you to a context-specific formatting menu more quickly. On rare occasions, you may need to alter the cell size in a text table. Do that from the Cell Size menu option. Or, if you don't like the default workbook theme, use the Workbook Theme menu option to select one of the other two options. You learn many tips and tricks about formatting in Chapters 7 and 8.

Server Menu

Use this menu if you need to log in and publish work to Tableau Server. The Sign In menu option is used to enter your server address and security credentials. The Switch User and Switch to Self menu options are used for simulating the login credentials of another user to see how a dashboard or workbook will appear for them. Create New User Filter is used to create filters that will hide information from view that you have included in the data source. Chapter 11 covers these options in detail.

The section in Chapter 11 on options for securing reports describes how to use the menu option to create user filters. This provides row-level security by using a dimension to filter out data from view.

If you are using Tableau Desktop to build dashboards for fun or for a blog post, use the Tableau Public menu. To use this service, you must sign up for a free Tableau Public account. The menu options here allow you to save and manage workbooks saved to your Tableau Public account.

Window Menu

The Presentation Mode option removes all design tools from view, making your worksheet or dashboard data stand out. If you want to share a single sheet with someone else, use the Bookmark/Create Bookmark option to save a `.tbm` file that you can share with another Tableau user. I never use the rest of the menu options in this menu because there are easier ways to accomplish the same things, as discussed throughout this book.

Help Menu

The top section of this menu includes menu options that access Tableau's online manual, training videos, and sample workbooks. All of these options are accessed from the Start page covered earlier in this chapter. Use the Choose Language option to change the default language that Tableau Desktop uses to display text in menus, windows, cards, and content. This also changes how Tableau expresses geographic names in maps and for map search. After you select the new language, Tableau will prompt you to restart Tableau Desktop to effect the desired language change.

The most important feature in the Help menu is contained in the Setting and Performance menu option. The Start Performance Recording option tells Tableau to collect key performance metrics about your workbook. Stopping the performance recording causes Tableau to build a Tableau dashboard of your Tableau workbook performance. This is a critical feature that is covered in detail in Chapters 8 and 12.

If you need to find your product key, the Manage Product Key menu option will display it. Finally, the About Tableau menu option (Windows only) displays the Tableau version that you are running. If you are using a Mac, this information is located in the Tableau menu option. This information is particularly important if your company is running multiple versions of Tableau. You can install multiple releases of Tableau on your computer, but because different versions are not backward compatible, you should confirm that you are using the correct release before you start creating new analysis.

Understanding the Toolbar Icons

The toolbar displayed in Figure 1-10 makes the most commonly needed functions conveniently accessible. The section headings that follow match the pop-ups that occur in Tableau when you point at the icon.

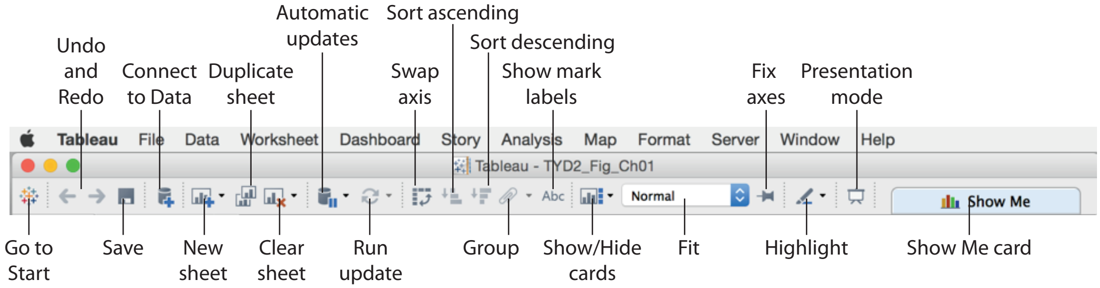


FIGURE 1-10 *Toolbar icons*

Start

Clicking the small Tableau logo on the left side of the toolbar toggles you between the Start page and the workspace.

Undo/Redo

Undo reverses the most recent action in the workbook. Redo repeats the last action you reversed. Use Undo to step back in time to the beginning of your work session in Tableau Desktop. Redo allows you to step forward through your session. If you don't like what you see onscreen, use Undo to go back in time and try again.

Save

Click Save to save changes made in your workbook.

Add New Datasource

This option opens the Connect page where you can create a new connection or open a saved connection from your repository.

New Worksheet, Dashboard or Story

This option creates a new sheet, dashboard, or story in your workbook. You can also add these by right-clicking a Tab or Filmstrip view at the bottom of the work area. In addition, you can add these by clicking the related icons next to the tab or at the bottom of the work area. See the bottom of the view pane in Figure 1-6.

Duplicate Sheet

Make an exact copy of your current sheet. This is handy if you want to experiment with different ways of viewing data without altering the view contained in the original sheet.

Clear Sheet

This option clears the current sheet. Use the drop-down options to clear formatting, manual sizing, axis ranges, filters, sorts, or context temporary files.

Automatic Updates

Use this option to stop auto-updating the view when changes are made. Use the drop-down menu to pause updates for the entire sheet or for quick filters. Suspending auto-updating can be very helpful when you are connected to slow data sources. By suspending updates, you can place many fields into the workspace without delay. Then use the Run Update icon to update the view when you are ready to see the results.

Run Update

When you suspend auto-updates, click the Run Update icon to see the results of your work.

Swap

This icon swaps the fields on the row and column shelves. This will pivot the orientation of your view vertically or horizontally.

Sort Ascending and Sort Descending

These icons allow you to change the order that data is presented based on measures contained in the view. Ascending sorts from smallest to largest values. Descending sorts from largest to smallest values. These sorts persist only during the work session and do not override default sorting defined in the view.

Group Members

Use this feature to create a group by combining selected values. When multiple dimensions are selected, use the drop-down menu to specify whether to group on a specific dimension or across all dimensions.

Show Mark Labels

This option enables you to switch between showing and not showing labels. As you become more experienced with Tableau Desktop, you will find more flexible ways to display marks in your view by using the Marks Card Label option.

Show/Hide Cards

You can reset what cards are included in your sheet view, including title, caption, summary, legends, quick filter, parameters, map legends, the column shelf, the row shelf, pages shelf, measure values shelf, current page, and the marks card.

Fit

You can specify how the view should be sized within the view. Normal is the default viewing option and uses the least amount of screen real estate possible to fully display all of the marks. This may result in vertical or horizontal scroll bars. Fit Width, Fit Height, and Fit Entire View allow you to force Tableau to display the view without scrolling within the specified fit orientation.

Fix Axes

This icon toggles between a locked axis that shows a specific range and a dynamic axis that adjusts the range based on the minimum and maximum values in the view. Tableau's default is to use the dynamic range option, which is the best if you are presenting dynamically changing data. The Fix Axes option should be used with care. It is most appropriately used with static data where the values are known.

Highlight

Highlight turns on highlighting for the selected sheet. Use the option on the drop-down menu to define how values will be highlighted. For example, if you want to highlight based on a specific combination of fields, select the specific fields you want Tableau to combine for highlighting the view. This is particularly helpful when displaying your data in a scatter plot if you want to highlight based on a combination of dimensions.

Presentation Mode

This option toggles the view between Presentation mode and normal work mode. Presentation mode removes all design tools from view so that your audience can focus on the information being presented.

Show Me

This displays alternative ways to look at data. The types of charts available are dependent on the fields already in the view as well as any selections you have made in the data window. Chapter 3 covers this feature in detail.

The Data Window, Data Types, and Aggregation

When you connect Tableau to a data source, it is expressed in the data window. You can connect to as many different data sources as you want in a single workbook. The small icons associated with data connections provide additional details about the nature of the connection. Figure 1-11 shows a workbook with three different data connections.

There are subtle visual clues regarding the exact state of each connection. The blue check mark icon next to the Coffee Chain (Access) data connection indicates that it is the primary active connection for the sheet. The related icon shows two cans with an arrow and a blue check mark. This denotes that the connection is a data extract. A single can without an arrow denotes a direct connection to the data source.

Note the blue highlight on the Sample – Coffee Chain data source. The highlighting is used only when multiple data sources have been connected to the workbook and indicates which data source fields are on display in the dimensions and measure areas.

For example, clicking the Sample – Superstore Subset data source would cause that data source to be highlighted in blue and its related dimensions and measures to be displayed. And orange highlight would appear on the left of the Dimensions and Measures windows along with a small link icon next to any fields that the two data sources have in common. This is used to indicate the potential for data blending, which is covered in Chapter 2.

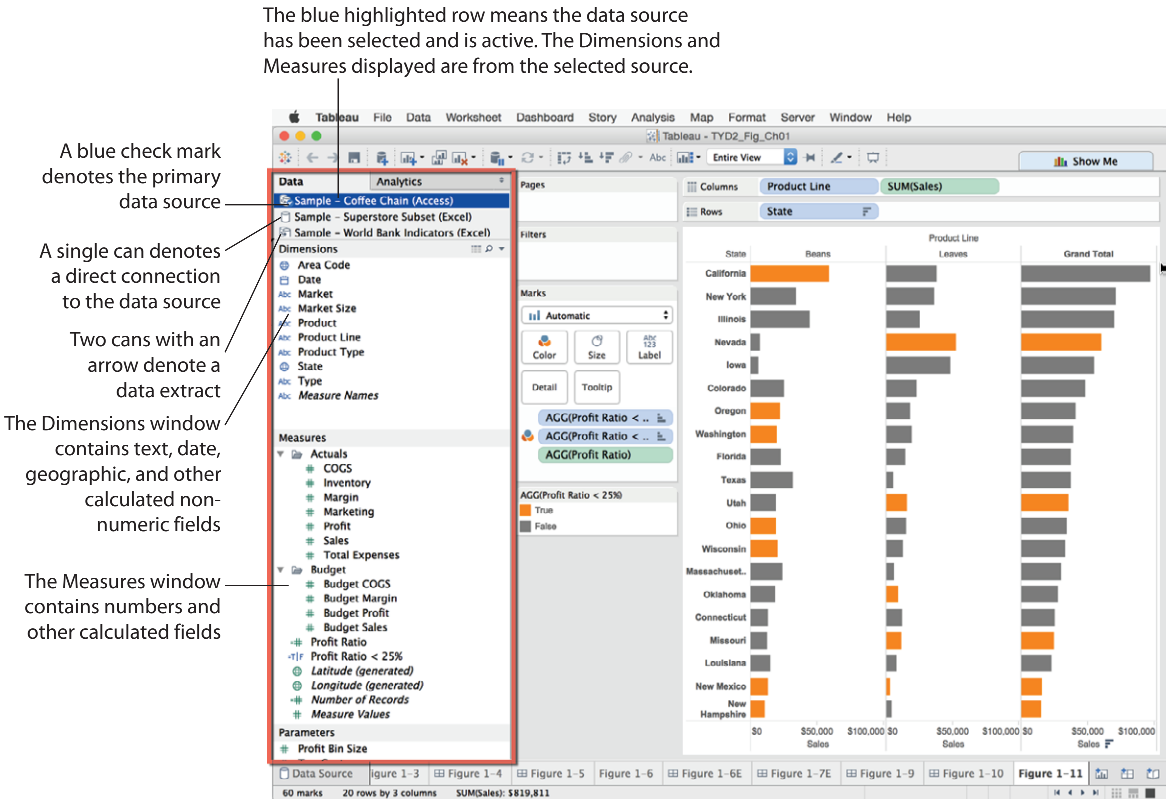


FIGURE 1-11 *Data window*

The Superstore subset (Excel) data connection is a direct connection. This is indicated by the single can icon. The World Bank Indicators (Excel) data source at the bottom of the data window is another data extract, denoted by the icon with two small cans and the arrow. Data extract files copy and compress data from any data source into Tableau's proprietary data engine. Sometimes the compression and performance improvement with extracts can be significant.

When you create data connections, Tableau will evaluate the fields and place them on the dimensions and measures shelves automatically. Tableau normally gets the fields placed correctly. If a field is placed in an area you don't prefer, simply drag the field to the location you desire.

For example, if you connect to a spreadsheet that contains a customer identification number, that field may be placed into the Measures pane. You may want that to be a dimension in your workbook. Dragging a customer identification number from the Measures pane into the worksheet would result in the field being summed. If that field was dragged from the dimension shelf, the customer identification number would behave like a dimension and be expressed in a column or row the same way product line and state are expressed in Figure 1-11.

Data Types

Tableau expresses fields and assigns data types automatically. If the data type is assigned by the data source, Tableau will use that data type. If the data source doesn't specifically assign a data type, Tableau will assign one. Tableau supports the following data types:

- Text values
- Date values
- Date and time values
- Numerical values
- Geographic values (latitude and longitude used for maps)
- Boolean values (true/false conditions)

Focus on the icons next to the fields contained in the Dimensions and Measures shelves in Figure 1-11. These icons denote specific data types. Small globes are geographic features; calendars are dates. A calendar with a clock is a date/time field. Numeric values have pound signs, and text fields are denoted by "abc" icons. Boolean fields have "T/F" icons. An equal sign (=) denotes that field is a calculated value. Explore Tableau's manual for more examples.

Aggregation

It is often useful to look at numeric values using different aggregations. Tableau supports many different aggregation types, including:

- Sum
- Average
- Median
- Count
- Count Distinct
- Minimum
- Maximum
- Percentile
- Standard Deviation

- Standard Deviation of a Population
- Variance
- Variance of a Population
- Attribute (ATTR)
- Dimension

If you aren't a statistician or database expert, refer to Tableau's manual for detailed definitions of these aggregation types. Adding fields into your visualization results in default aggregations being displayed. Tableau allows you to change the default aggregation or just alter the aggregation type for a specific view.

To change the default aggregation, right-click any numeric field and change its default by selecting the menu option (default properties/aggregation). You can also change the aggregation of a field for a specific use in a worksheet. When the term "pill" is used in this context, it means that a field that has been placed on a view card, the Rows Shelf, or a Columns Shelf. In other words, the field is being actively used in the sheet.

Figure 1-12 provides an example. By right-clicking the SUM (Sales) pill on the columns shelf and selecting the Measure (SUM) menu option, you can select any of the aggregations highlighted.

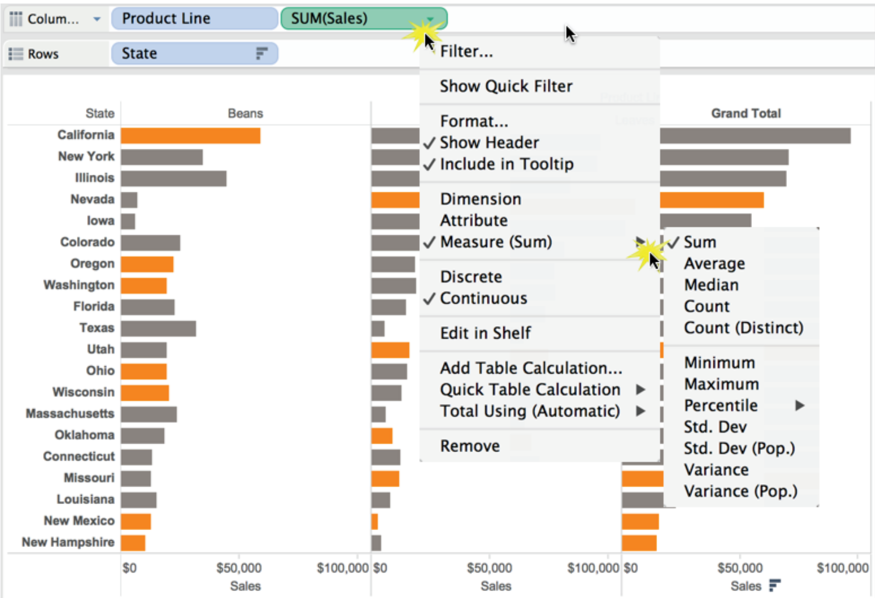


FIGURE 1-12 *Changing aggregation*

The data source used in Figure 1-12 is a data extract of an Access database called Coffee Chain. In Tableau Desktop versions before V9, if you are working with Excel, Access, or a text file source and you don't see the full range of expected options, run a data extract and you'll see that more aggregation options will become available. In Tableau Desktop V9, running an extract to get more aggregation types is not necessary. You will learn about aggregation in Chapter 4.

A Word About Dimension and Attribute

Most aggregates involve mathematical concepts that should be comprehensible if you don't have a statistics degree. Even if you don't understand specifically what standard deviation is, you probably appreciate that it has something to do with variation of data within a set of numbers.

The attribute aggregate (ATTR) type is best understood by looking at examples of it in use. Refer to Chapter 7 and Appendix E for examples.

Building Visualizations with the Row and Column Shelves

Row and column shelves are used to express data on the vertical and horizontal axes in the view. Dimensions and measures can be displayed in any order and on either shelf. Figure 1-13 is a basic time series chart that shows sales trends by year and then quarter.

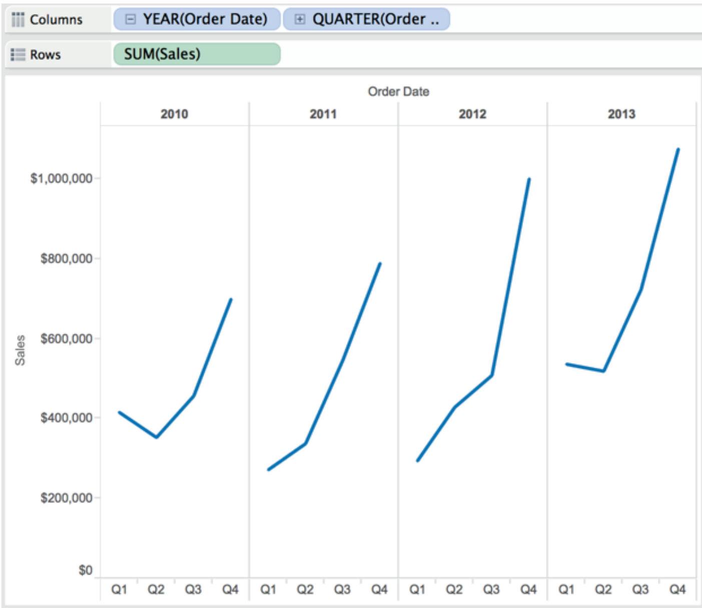


FIGURE 1-13 *Time series by year, quarter*

The time series has breaks in the line because time is discretely broken down by year and then quarter. Figure 1-14 displays the same data, rearranging time by showing quarter first and then year, making it easier to see how sales changed in each quarter.

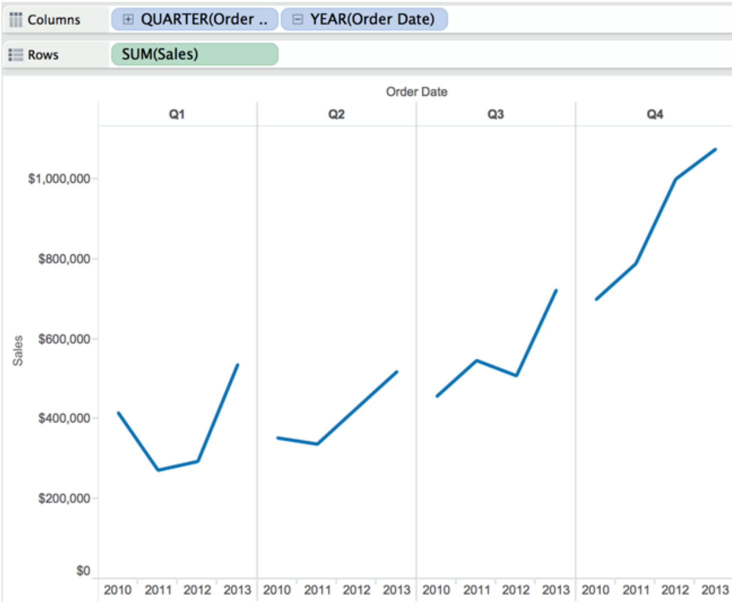


FIGURE 1-14 *Time series by quarter, year*

Dragging the year field (pill) to the right of the quarter pill on the Columns shelf altered the context of the view making it easier to see how sales changed in each quarter over time.

Number of Records, Measure Values, and Measure Names

Tableau automatically adds three fields to every dataset:

- Number of records
- Measure names
- Measure values

Number of records is a calculated value that sums the number of rows in the data source. As mentioned earlier, field icons preceded by an equal sign are calculated values. Measure Names and Measure Values are special fields that allow you to display multiple measures on a single axis. Figure 1-15 was created by double-clicking the Measure Names field and clicking the Swap icon on the toolbar to change the orientation of the chart.

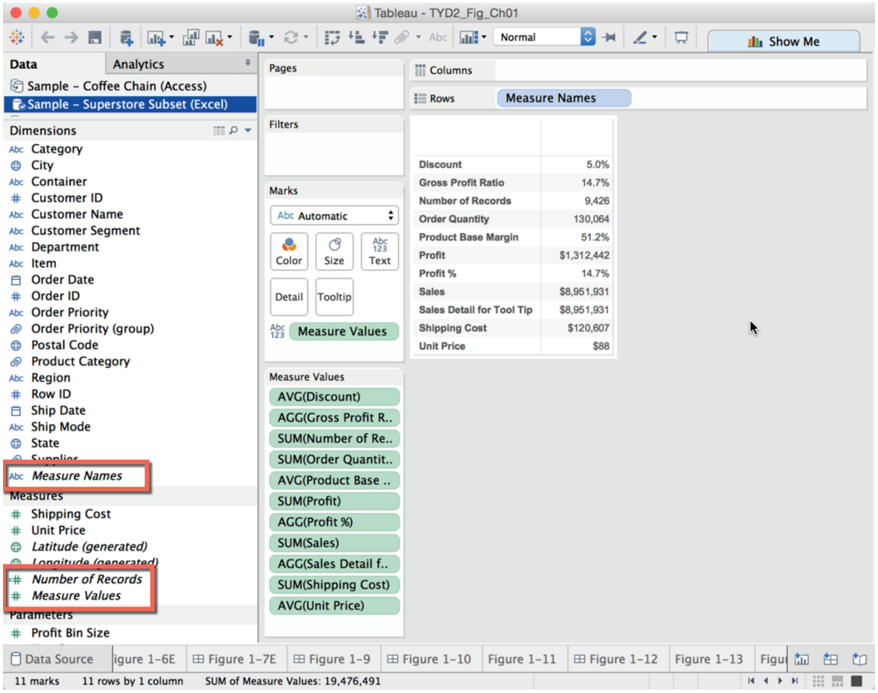


FIGURE 1-15 *Measure Names, Measure Values*

When the Measure Values field is used in a view, a new shelf appears that holds the pills for every measure in the dataset. Selecting Measure Names and Measure Values will automatically display all of the measures in your data source with their corresponding descriptions. You can use the Measure Names pill to filter out specific measures by right-clicking in the pill and de-selecting measures you no longer want to display on the axis.

Understanding Color in Icons and Pills

Have you noticed the color of the pills placed on shelves is either green or blue? Look at Figures 1-13 and 1-14 again. Can you guess what those colors mean? Most people think blue pills are dimensions and green pills are measures. That is a good guess, but the right answer is more subtle. Figure 1-16 displays the time series without any breaks between the years. Notice that there is only one pill on the Column Shelf and it is green.

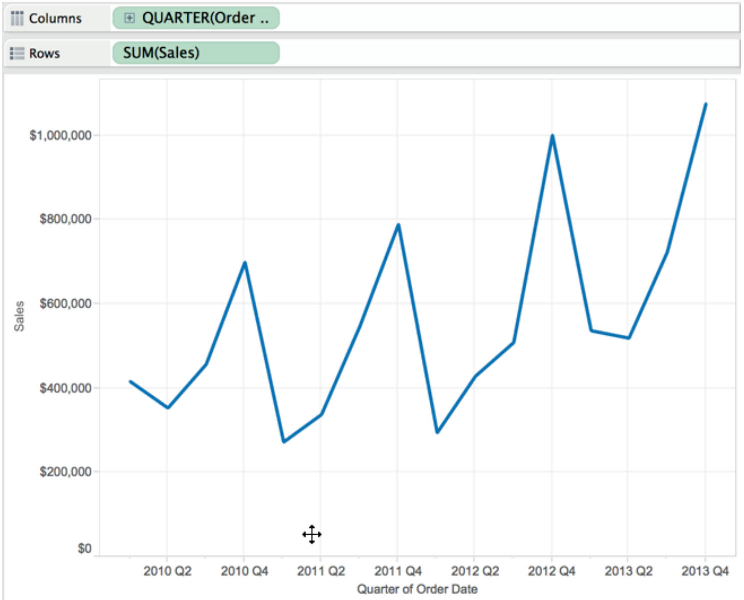


FIGURE 1-16 *Time series with continuous date*

Green denotes continuous and blue measures discrete. When a time dimension pill is green, the data is displayed using an unbroken, continuous line. In Figures 1-13 and 1-14, the time dimension pills are blue. Time buckets are displayed discretely by year and then quarter. Measures aren't always continuous either. Histograms convert normally continuous measures into discrete dimensions by making a new, discrete bin field from a measure. You'll see an example of a histogram in Chapter 3.

Using the View Cards Filter and Highlight Data

The View Card area to the right of the data window in Figure 1-8 is used to apply filters in your sheet and to define the use of color, shape, size, and other visual means to convey information in your visualization. Fields placed on the Pages shelf create an auto-scrolling filter. Fields placed on the Filter shelf create manual filters. Let's focus on the Marks card area first.

The Marks Card Properties

The drop-down selector at the top of the Marks area is used to select the mark type for display. By default, it is defined as Automatic. In this mode, Tableau Desktop will pick the mark type used. You can override that selection by picking any of the available mark types that Tableau supports.

You can use different visual tools to convey meaning in your visualization by placing fields on the desired Marks button. Note that when you hover your mouse over the buttons, additional information will appear:

- **Automatic:** By default, Tableau will supply an appropriate mark type but you can change the mark type used to display the measure by selecting the drop-down menu.
- **Color:** This expresses discrete or continuous values.
- **Size:** This expresses discrete or continuous values.
- **Label:** One or more fields can be expressed as mark labels.
- **Detail:** This disaggregates the marks plotted when dimensions are used and makes the field available in Tooltips.
- **ToolTip:** This makes fields available to ToolTips without disaggregating data.
- **Shape:** This expresses discrete or continuous values.
- **Path:** This is available when the Line or Polygon mark types are selected and your data supports the sequencing of the data. You can path-encode your data using a dimension or a measure.

Multiple fields can be placed on the Color, Label, Detail, and Tooltip buttons. Figure 1-17 displays a scatter plot with color, shape, and size all being utilized to visualize a comparison of profit and shipping cost.

The Column shelf in Figure 1-17 contains shipping costs, making that measure plot horizontally across the page. Profit, on the Row shelf, is displayed vertically. Color is being used to display department, Shape shows order priority (group), and the size of the marks provides information on sales. The customer name is displayed with the Label card, and the average unit price was placed on the Tooltip card, making it available within the tooltip.

The scatter plot in Figure 1-17 is displaying four measures (one inside the tooltip) and three dimensions while displaying the outliers in a way that makes them stand out. Notice the customer names display only when they don't overlap. All of the visual styles were applied by dropping individual fields on the desired Marks card buttons.

Look at Figure 1-17 and focus on the pills contained in the Marks card. Notice that the little icons to the left of each pill provide an indication of how that field is being visualized (what Marks card button it was placed on). The colored circles to the left of Department denote that department is being expressed using color in the view.

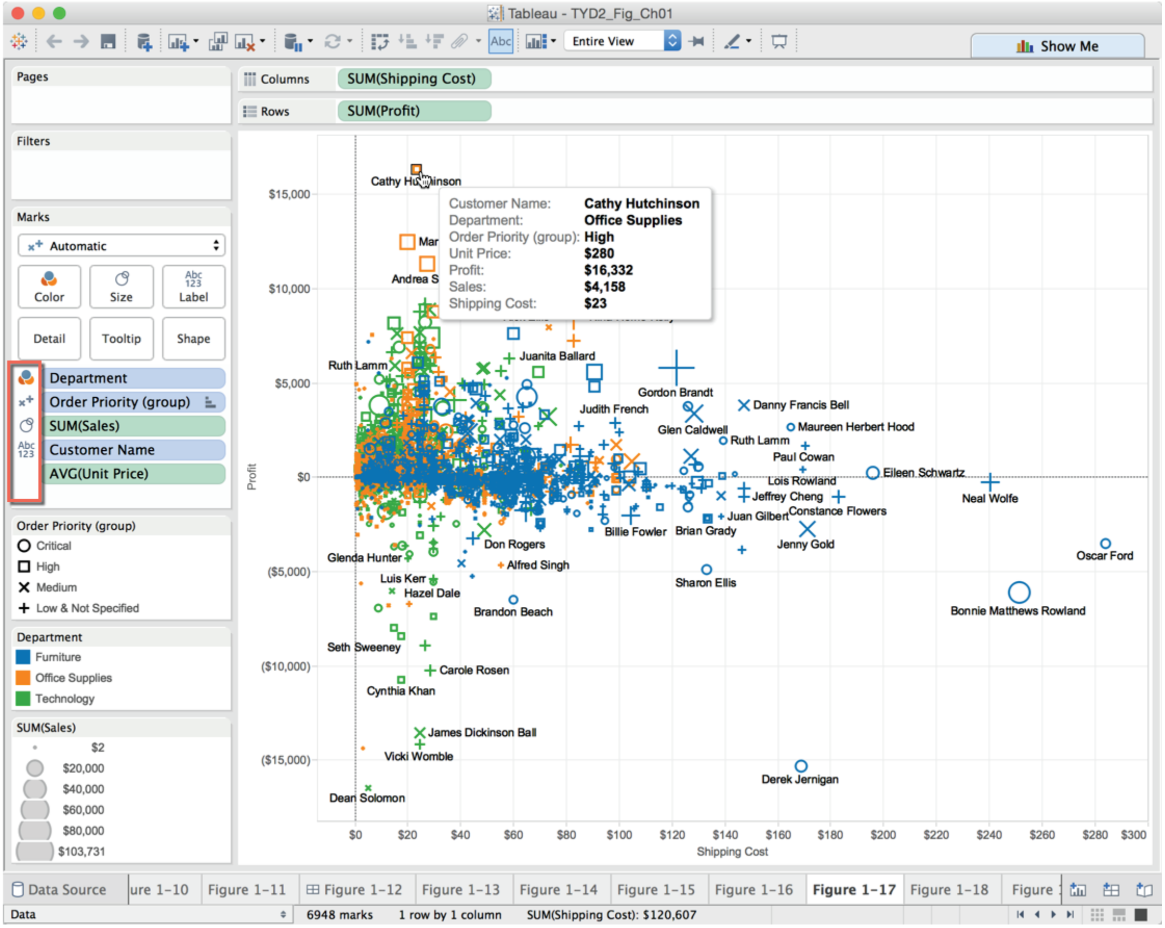


FIGURE 1-17 *Scatter plot*

One way that you can alter how each field in the Marks card is being used is to point at the small icons to the left of each pill, click the icon, and select another display option.

The Pages Card

As mentioned earlier, any field placed on the Pages shelf generates a scrolling filter that will advance through each member of the set of data that is being displayed manually or automatically.

The Pages shelf creates animated visualizations in Tableau Desktop (this feature does not apply to Tableau Server) by stepping through all of the discrete members of the set. For example, placing a date field on the Pages shelf that is expressed as month-year will enable the page filter to increment through each month-year over time.

In Figure 1-18, you see that when a field is placed on the Pages shelf, another supporting shelf can be accessed directly under it that contains a manual field selector; auto-scrolling controls providing forward, pause, and stop control over scrolling speed; and a Show History check box.

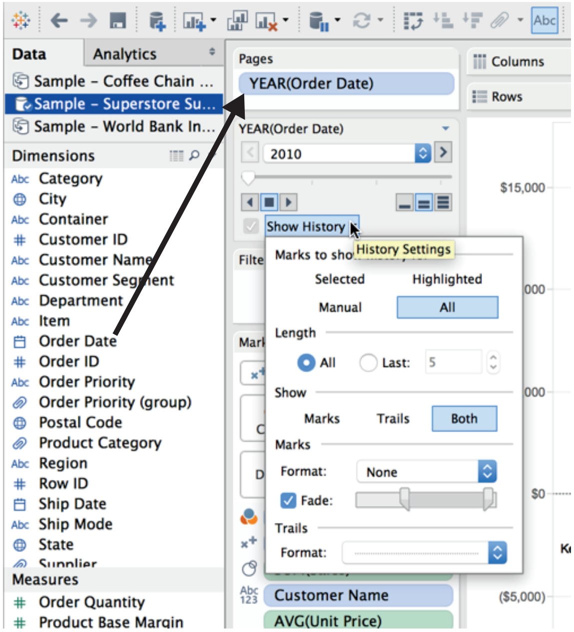


FIGURE 1-18 *Pages shelf*

In addition, any field placed on the Pages shelf will be converted to a discrete field type because each page will be a member of that field.

Checking the Show History menu exposes options that provide selections to control the way history is displayed and how many marks will be displayed while the filter increments through whatever field has been placed on the shelf.

Trails are lines that connect marks sequentially as scrolling occurs. Selections made in the show section of the menu enable you to control whether marks, trails, or both marks and trails are displayed as the auto filter increments. The Marks section provides controls over the color and fade of the marks. The Trails section provides color and line style controls for the trails. Auto-scrolling filters are not supported in Tableau Server, but they can be consumed via Tableau Reader or Tableau Desktop.

The Filter Card

To filter based on any field (dimension or measure), drag the field to the Filters shelf. The available filter types are somewhat dependent on whether the data is continuous or discrete. You can also filter based on a hierarchy of selections that you will see in the filter menu. To expose the filter on the desktop, right-click the field pill and select the Show Quick Filter option. You learn more about quick filters in Chapters 7 and 8.

Using the Status Bar to Understand Visualizations

The status bar appears in the lower left of the worksheet. It provides basic metrics about the number of marks displayed in your visualization. The map visualization in Figure 1-19 demonstrates the value of the status bar.

The map in Figure 1-19 plots pie charts that show sales by postal code and department. Notice the status bar at the bottom left of the worksheet indicating 3,356 marks are in view. The total sales value of the marks is $8,341,246. Each slice in the pies counts as a mark. The status will change if a mark or groups of marks are selected in the worksheet, reflecting the count and value of the selection.

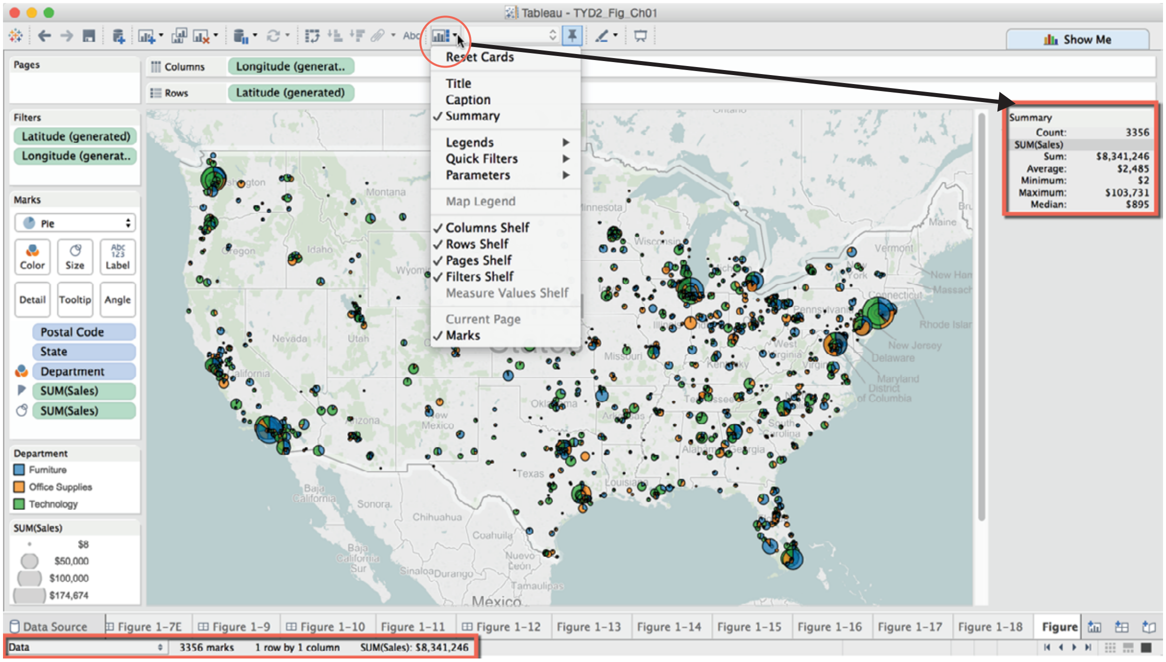


FIGURE 1-19 *Status bar and summary*

The larger summary card in the upper right is optional. Enable it by using the Show/Hide Cards icon circled in Figure 1-19, and then selecting Summary menu option.

Saving Time with the Show Me Button

Using the Show Me button allows you to build visualizations very quickly. If you can decide on the combination of dimensions and measures you want to analyze, Show Me will build your visualization for you. It will place all of the pills on shelves automatically. See how a map similar to Figure 1-19 can be re-created using the Show Me button in Figure 1-20.

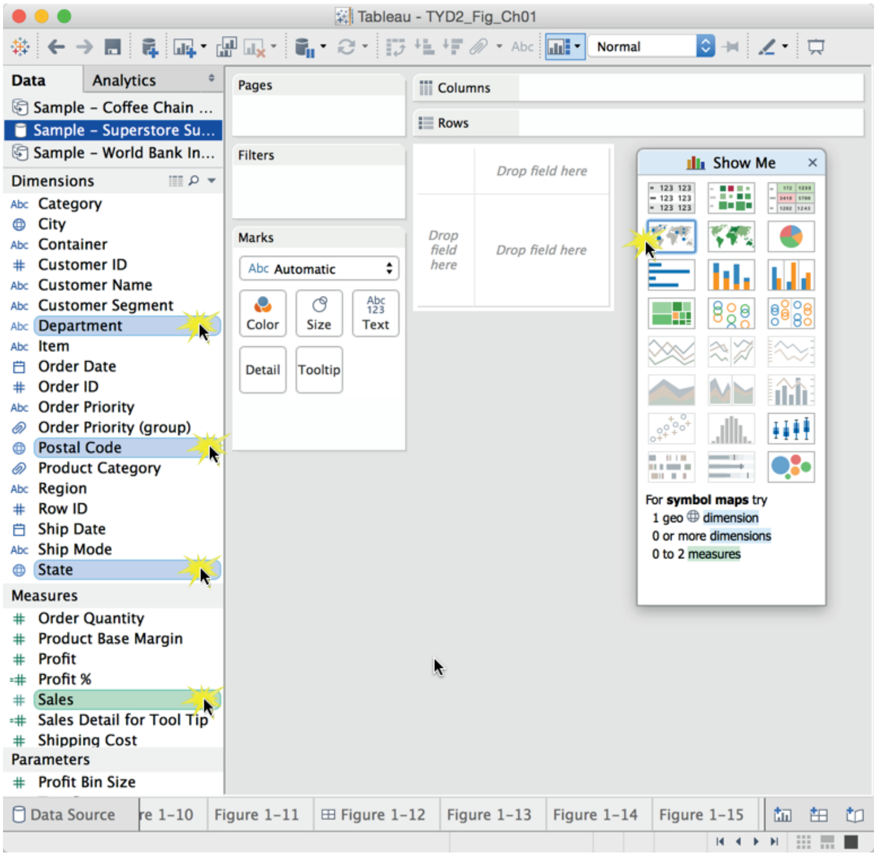


FIGURE 1-20 *Building a map with Show Me*

Using the Sample – Superstore Subset (Excel) data source, multi-select State, Postal Code, Department, and Sales. Then pick the symbol map and hold the Ctrl button (Windows only) down on your keyboard while pointing at and clicking each field. To multi-select fields on the Mac, use the Command key. Your screen should look like Figure 1-20.

Show Me can be dragged to any location on your desktop. The text at the bottom of the Show Me dialog box provides additional feedback on the combination of dimensions and measures that should be selected in order for the chart type to be available. Other highlighted chart styles are also supported by the selections of measures and dimensions. The charts that are grayed-out are not currently available based on the current field selections. Note that the time series charts are all gray because a date dimension hasn't been selected.

The map in Figure 1-19 was created by selecting the map highlighted by Show Me. The marks were styled using the Color Marks card fading the color and putting a black border around the marks. Leaving the Show Me button open allows you to quickly pick many different chart styles and see the results. Show Me is a time-saver and a great way to see how different pill placements can affect the appearance of your visualization.

SUMMARY

Now that you've got a basic introduction to the desktop workspace, in the next chapter you learn a variety of ways you can connect to data and the different kinds of data sources you can connect to using Tableau Desktop.

NOTES

1. Edward R. Tufte, *The Visual Display of Quantitative Information* (Cheshire, CT: Graphics, 2001), 91.
2. Margaret Rouse, NoSQL (Not Only SQL Database), "Essential Guide, Big Data Applications: Real-World Strategies for Managing Big Data," SearchDataManagement, October 5, 2011, `http://searchdatamanagement.techtarget.com/definition/NoSQL-Not-Only-SQL`.
3. Andrew Ryan, "Under the Hood: Hadoop Distributed Filesystem Reliability with Namenode and Avatarnote," `https://www.facebook.com/notes/facebook-engineering/under-the-hood-hadoop-distributed-file-system-reliability-with-namenode-and-avata/10150888759153920`.
4. Stephen Swoyer, "Report Debunks Business Intelligence Usage Myth," `http://tdwi.org/Articles/2009/05/20/Report-Debunks-BI-Usage-Myth.aspx?Page=1`.
5. Stephen Few, *Information Dashboard Design: The Effective Visual Communication of Data*, Berkeley, California, (O'Reilly Media, Inc, 2006), 4.

6. "The Art of Analytics," Keynote by Christian Chabot and Chris Stolte at the Tableau Customer Conference 2014, September 9, 2014. `https://tc14.tableausoftware.com/keynote`.

7. "The Tableau Workspace, Toolbar," accessed December 2, 2014, `onlinehelp.tableausofwared.com/v9.0/pro/online/windows/en-us/help.htm#environ_workspace_toolbar.html`.

CHAPTER 2

Connecting to Your Data

"I think a manager's world is not black and white. It's a world filled with uncertainties and dilemmas. The sort of thing that would leave any neophyte moaning, "What the Hell is this."

GORDON MACKENZIE[1]

It would be nice if all the data you needed to access resided in one place, but it doesn't. Your data is scattered over multiple databases, text files, spreadsheets, and public services. Tableau's ability to directly connect to a wide variety of data sources makes it easier to analyze data residing in different places. At the time of this writing, fifty different connectors are available in Tableau for Windows (twenty-two connectors for the Mac version). You can analyze spreadsheets, public data tools, analytic databases, Hadoop, and a large variety of general-purpose databases as well as data cubes.

WHAT YOU WILL LEARN IN THIS CHAPTER

Tableau Software has made the user interaction easier for data connections, joins, and data blending. A new Connect Page persists in the worksheet view that provides easier access to all of your data connections. Joining tables is now a more visual experience, and the new Data Interpreter provides easy tools for dealing with poorly formatted spreadsheet and text file data sources. The Data Interpreter provides a better way to deal with typical spreadsheet problems. Readers of the first edition of this book will find a lot of new content in this chapter.

We start by introducing the Connect Page in the context of connecting to a local file. Then you will learn about connecting to databases and cloud data sources. After that, we discuss values that Tableau generates when you connect to any data source. You learn the difference between a direct connection to a data source and using Tableau's data extract engine, as well as the advantages of each type of connection.

If you spend a lot of time working with data, you know that data preparation normally takes a lot more time than the actual data analysis. In that context, we will introduce Tableau V9.0's new Data Interpreter by working with some spreadsheet data downloaded from the U.S. Census Bureau. Then you learn how to perform joins between different tables in a database or different tabs in a spreadsheet. Then you learn how to blend data from different data sources in a single visualization.

HOW TO CONNECT TO YOUR DATA

The most fundamental skill in Tableau is connecting to your data. You can connect to local files on your computer, database files on servers, and public data sources in the cloud. In this section, you learn more about the details of the connecting to different types of data sources. Before you start working through connection examples, let's look at one part of the Start page that we didn't open in Chapter 1. Clicking the More Servers option opens the expanded pane of database connection possibilities. Figure 2-1 shows the Windows and Mac editions of Tableau Desktop versions.

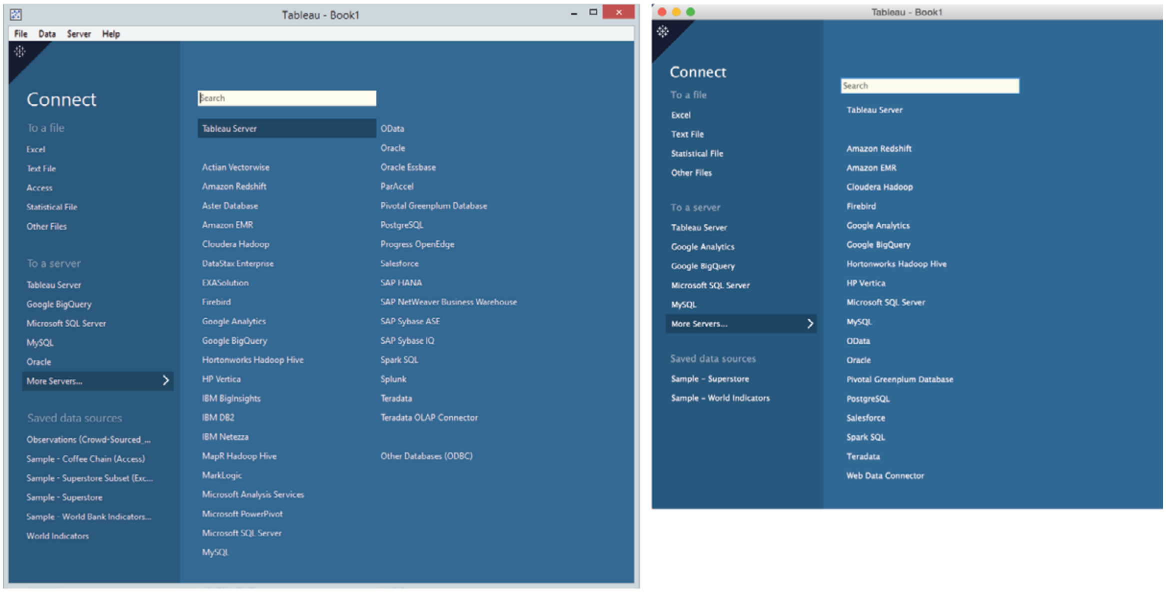


FIGURE 2-1 *More Servers pane*

The exposed list of connections includes databases, data cubes, and cloud-based services. To access a database, you must install the driver particular to the data source. Installation normally takes a couple of minutes. You can find the drivers at `www.tableausoftware.com/support/drivers`.

Frequently used databases will appear in the list of sources included in the main Connect pane below the (To a Server) section. The arrangement of the connections is dynamic and dependent on the frequency of usage. On the right side of Figure 2-1, the Google Analytics connection appears second and then Google BigQuery. That is because I connected to Google Analytics yesterday while in Tableau Desktop for Mac to analyze website activity. I also used Google BigQuery for some additional analysis.

If Tableau doesn't provide a dedicated connector for a database you want to analyze, try the Other Databases (ODBC) option. That connection utilizes the Open Database Connectivity standard.

The saved data sources area you see at the bottom left of the Connect pane displays data sources that you have saved for easy access. Tableau also provides some sample training data sources in this area by default. The exact number and type will depend on the version of Tableau Desktop (Windows or Mac).

You learn how to save a data source later in this chapter. Saved data source files (`.tds`) are found on your computer's hard disk in the data sources directory under the My Tableau repository. If you are logged into Tableau Server, you may also see saved data sources on your server's repository. Next you'll learn how to make a connection to a local file on your computer.

CONNECTING TO DESKTOP SOURCES

Now you will connect to an Excel spreadsheet data source. The files used in this example and in the examples for the rest of the chapter can be found on the book's companion website. See Appendix F, "Companion Website," for the address. Download the Chapter 2 file and put it in a folder on your computer. Open Tableau Desktop and click the Excel option. This exposes the Open window, providing a view of the directories on your computer. It should look like Figure 2-2.

Navigate to the location where you downloaded the Superstore for TYD2 spreadsheet and select the file; then click the Open button you see in the lower-right side of Figure 2-2 to connect the spreadsheet file to Tableau. Doing this establishes the connection. You should now see the Connect Page in Figure 2-3.

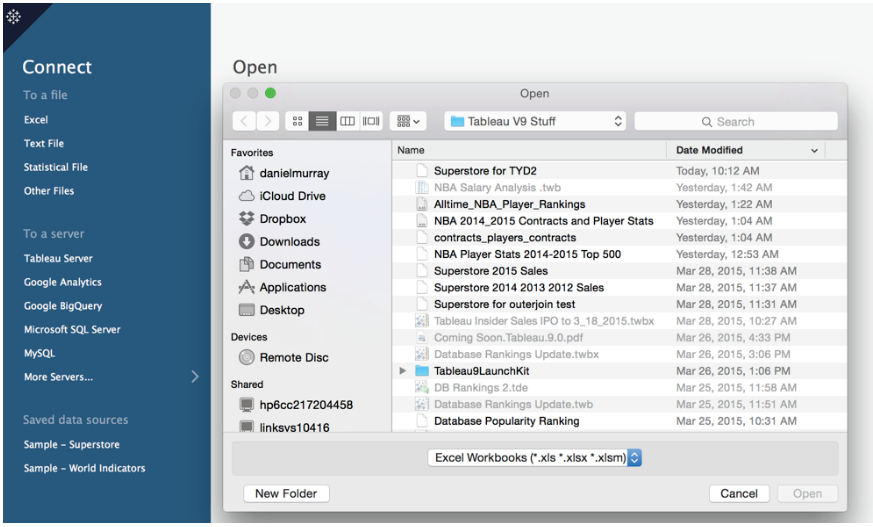


FIGURE 2-2 *Open file window*

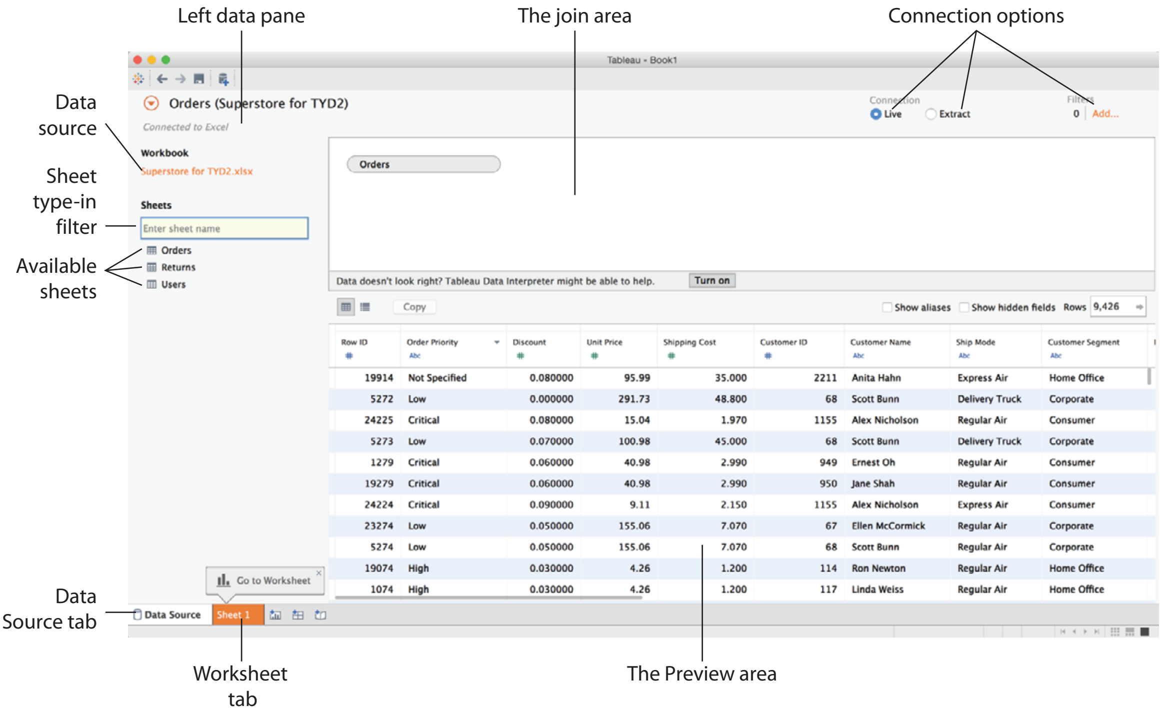


FIGURE 2-3 *The Connect Page*

Look at the left pane. The items below the sheets area are individual tabs that are contained in the spreadsheet. Double-clicking the Orders sheet will cause that table to appear in the join area. You can also drag and drop the Orders sheet into the join area. Doing this establishes a live connection to the sheet in your workbook. Save your work. Let's pause for a moment and go through the contents of the Connect Page in more detail.

UNDERSTANDING THE CONNECT PAGE

The Connect Page replaces the Connect to Data screen and connection window used until the release of Tableau Desktop V8.2 in 2014. It provides a more visual interface for connecting to data and joining tables. It also centralizes other tools for analyzing the contents of the data, performing data extracts, and restructuring data. You will learn how to use the new data cleaning features later in this chapter in the "The Data Interpreter" section.

The Left Pane Area

On the left side of the page, you can rename the data source connection and view the related sheets contained in the data source. At the top left of Figure 2-3, you see the connection currently in view. The connection has been renamed as Orders (Superstore for TYD2). You can rename the connection at the top left of the page by entering your own text and pressing Enter.

Immediately below in the Workbook area you see the filename of the data source. The sheet area contains the individual worksheet tabs from the spreadsheet data source. If the data source is a database, tables contained in the database would be listed. To see the contents of a particular sheet, look at Figure 2-4.

Hovering your mouse over the sheet of interest exposes the View Data icon to the right of the sheet name. Clicking that icon will open a tabular view of the data. This is similar to the Preview area but allows you to see the data before you place the sheet in the join area.

Connection Options

The upper-right area in Figure 2-3 contains connection options. Tableau uses a live connection to the data source by default. Clicking the Extract radio button enables you to exact the data into Tableau's proprietary data engine. Using an extract can significantly improve performance. It also allows you to view database files remotely even if you don't have access to the Internet. Extracting compresses the data and allows you to save the data extract on your computer. I'll cover the details about data extracts later in this chapter.

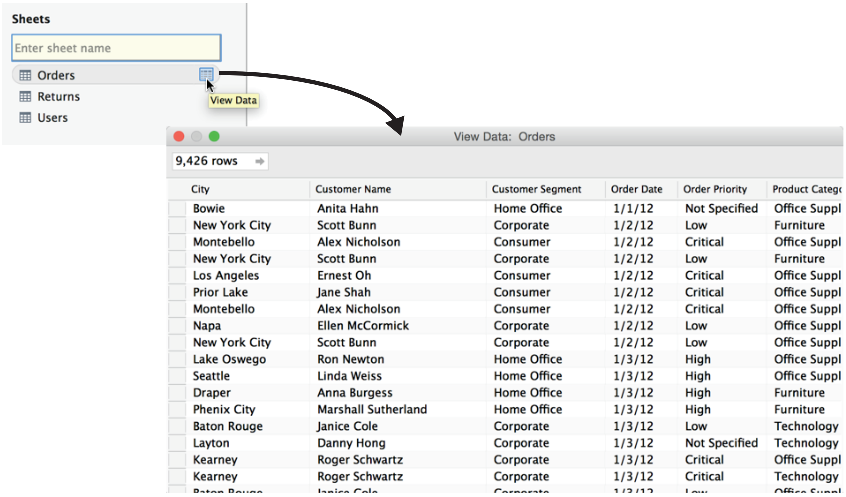


FIGURE 2-4 *Viewing sheet content*

To the right of the connection area is the Filters option. Selecting the Add option allows you to filter the data source. A running tally of the filters applied is also displayed. By filtering a direct connection, you may improve performance by eliminating unneeded data from your analysis.

The whitespace containing the orders sheet in Figure 2-3 is the join area. Starting with desktop V8.2, Tableau made defining joins a more visual experience. This approach makes the concept of joining tables more accessible to new users. You learn about the nuances of joining tables later in this chapter.

The Preview area in the bottom half of the page displays the rows and columns of the sheets placed in the join area. The two small icons at the top left of the Preview area permit you to toggle between the Data Source view visible in the Preview area in Figure 2-3 and the Manage metadata view in Figure 2-5. The Show aliases and Show hidden fields check boxes allow you to display or not display renamed or hidden items in the preview.

The Data interpreter helps you deal with problematic source data. It is turned on by clicking the button above the Preview area as shown in Figure 2-3. The Rows preview box on the top-right side of the Preview area displays the row count contained within the data source. If you are connected to a very large dataset, Tableau will initially limit the row count to 10,000 records. You can enter any number you want as the upper limit in the Rows text box.

Field Name	Table	Remote Field Name
Row ID	Orders	Row ID
Order Priority	Orders	Order Priority
Discount	Orders	Discount
Unit Price	Orders	Unit Price
Shipping Cost	Orders	Shipping Cost
Customer ID	Orders	Customer ID
Customer Name	Orders	Customer Name
Ship Mode	Orders	Ship Mode
Customer Segment	Orders	Customer Segment
Product Category	Orders	Product Category
Product Sub-Category	Orders	Product Sub-Category
Product Container	Orders	Product Container
Product Name	Orders	Product Name
Product Base Margin	Orders	Product Base Margin
Region	Orders	Region
State or Province	Orders	State or Province
City	Orders	City
Postal Code	Orders	Postal Code
Order Date	Orders	Order Date
Ship Date	Orders	Ship Date
Profit	Orders	Profit
Quantity ordered new	Orders	Quantity ordered new
Sales	Orders	Sales
Order ID	Orders	Order ID

FIGURE 2-5 *Manage metadata view*

Saving Data Sources and Workbooks

Saving a data source adds the connection to your Start page at the bottom of the Connect pane. Saving a workbook combines the data source connection metadata and all of the work you have done in one file, the workbook. You will be saving workbooks much more often than data sources. Once you've established saved data connections to the sources you use frequently, you won't have to edit them very often.

Saving a Data Source

There are a few steps required to save a data source. Figure 2-6 shows the menu options required.

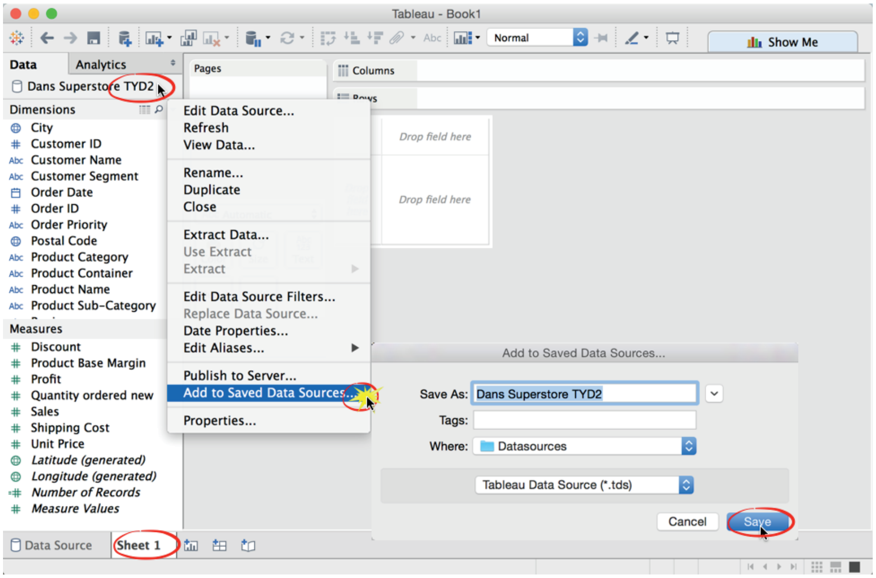


FIGURE 2-6 *Saving a data source*

Clicking the Sheet 1 tab in the lower left of the Data Source page takes you to the worksheet. Once there, point at the data pane displaying the data source. In Figure 2-6, the filename is Dans Superstore TYD2. The following are the steps required to save the data source:

1. Point at the data connection and right-click to expose the menu.
2. Select the Add to Saved Data Sources menu option.
3. Click the Save button in the Add to Saved Data Sources window.

Notice in Figure 2-6 that the data source is being saved under the name assigned earlier. It was saved to the data sources directory on my computer as a Tableau Data Source (`*.tds`) file. Now, whenever I open Tableau Desktop, my Start page will include this data source, as you see in Figure 2-7.

The Start page includes the Dans Superstore TYD2 connection, but my workbook (Tableau – Book 1) is not saved yet.

Saving a Tableau

Before saving the workbook, drag the Product Category field from the Dimensions shelf to the Rows shelf. Then drag the Sales field from the Measures shelf to the Columns shelf. You should now have a bar chart in the view. Figure 2-8 shows one way to save the workbook.

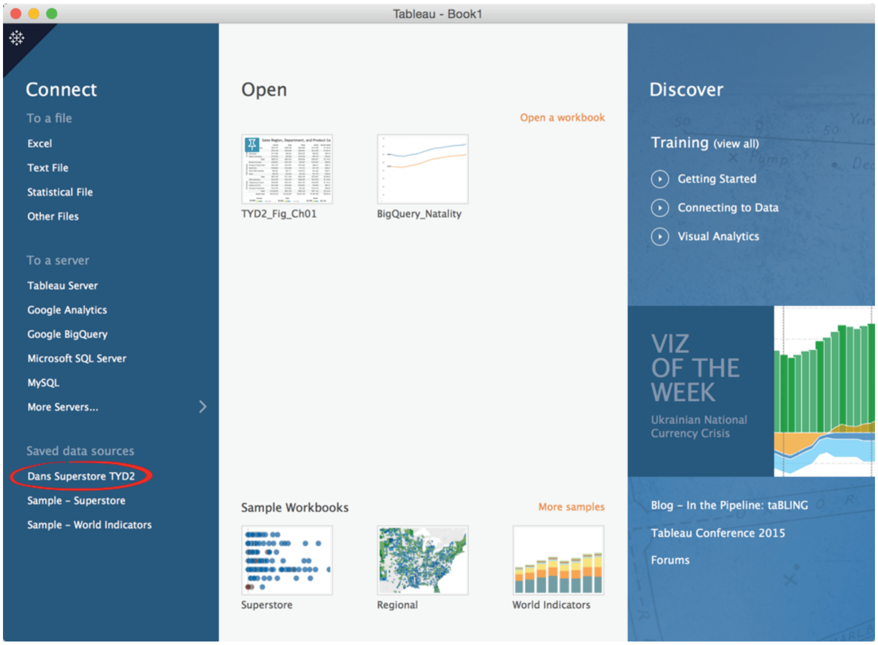


FIGURE 2-7 *Saved data sources*

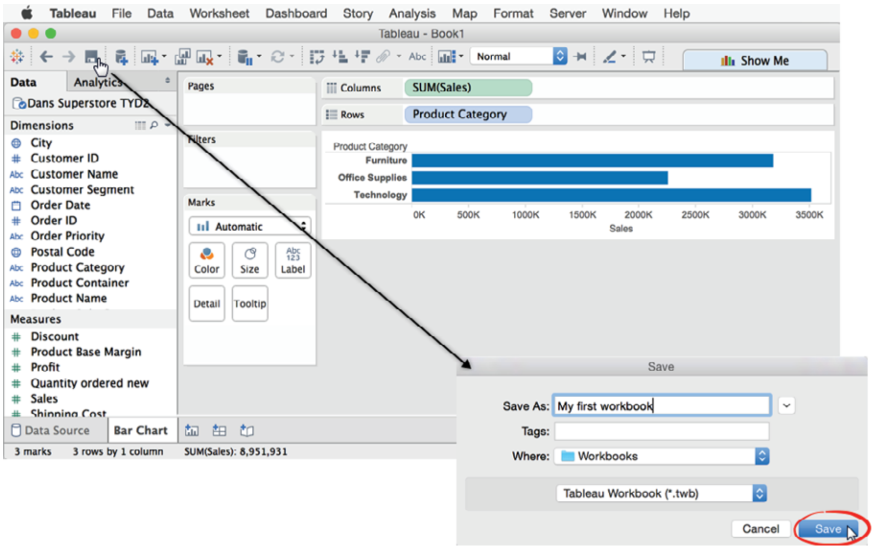


FIGURE 2-8 *Saving the workbook*

You can select File and then Save from the menu or just click the Save icon. Name the file "My first workbook" and then click the Save button, as you see in Figure 2-8. You have now saved your work as a Tableau Workbook (`*.twb`) file. In this scenario, you not only save the connection metadata but the work you've done in your workbook. Next, let's look at how you can connect to a database that resides on a server.

Connecting to a Database

Databases have an additional level of security—requiring you to enter a server name and user credentials to access the data. The username and password you enter are assigned in the database, meaning the security credentials and the amount of access granted are controlled by the database—not Tableau. Figure 2-9 shows the connection window for a MySQL database.

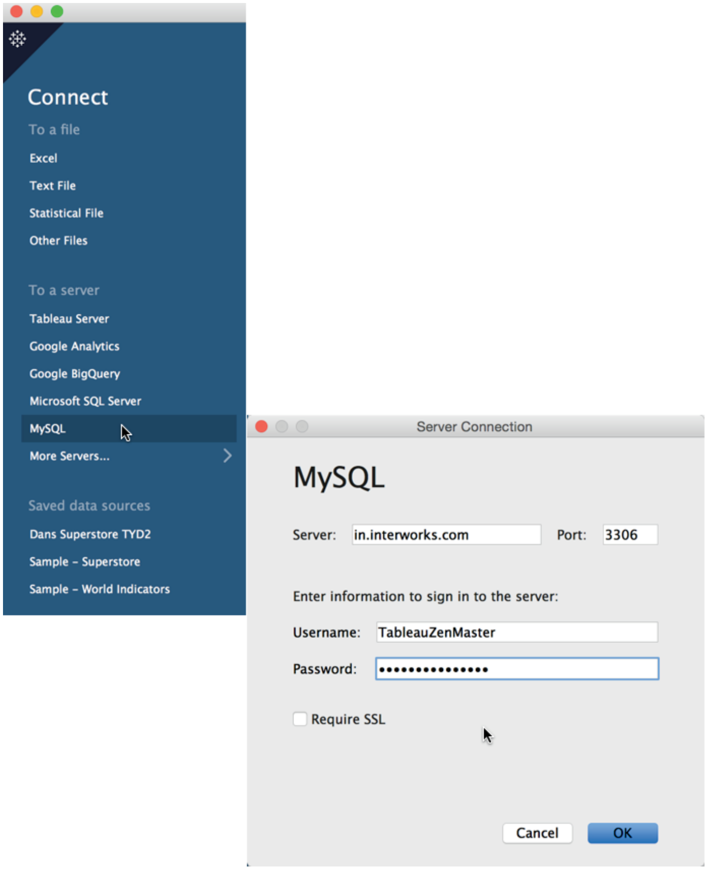


FIGURE 2-9 *Connection to MySQL*

If you need to access a database source and you don't know the server address, your username, and your password, you must request that information from your database administrator. After you receive the requested server connection information, you will access the same Connect Page you worked with in the previous example in Figure 2-9, but you will probably have a much larger number of tables displayed under the sheets section. Finding the tables you need for your analysis is facilitated by the Search filter.

Tableau's manual provides specific details and screenshots for accessing all of the data sources that you can connect to. Figure 2-10 is a screenshot of the online manual.

Because Tableau frequently adds new connectors, the online manual is the best source of information regarding data connections. Go to Help ➢ Open Help ➢ Supported Data Sources in Tableau's online manual to find specific details on connecting to different data sources. Click the Technical Specifications link in the manual to find the latest drivers for each database.

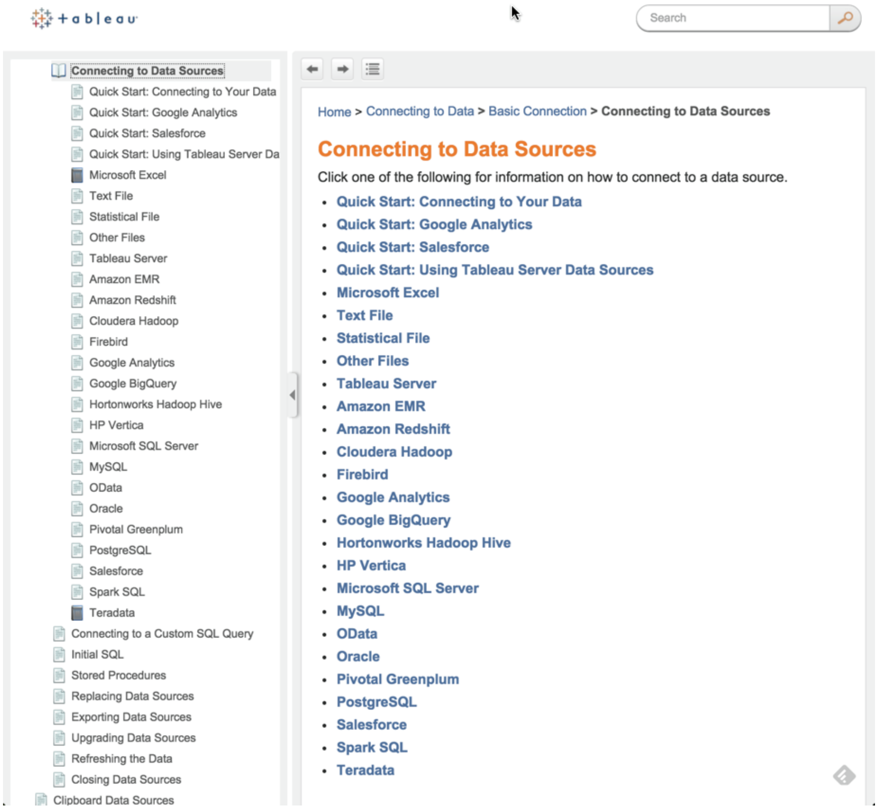


FIGURE 2-10 *Tableau online manual*

Connecting to Cloud Services

The increasing quantity and variety of data available on the Internet falls into three categories:

- Public domain datasets
- Commercial data services
- Cloud database platforms

The United State Census Bureau provides free population and business data. The World Bank provides a wide variety of country data, and many other public data repositories have emerged over the past decade. This data can be used to augment your proprietary data.

There are also a growing number of commercial data sources. Tableau currently provides connectors to

- Google Analytics
- Google Big Query
- Amazon Redshift
- Salesforce.com
- Open Data Protocol (ODATA)
- Microsoft Windows Azure Marketplace

The Google Analytics connector can be used to create customized click-stream analysis of web traffic. Google Big Query and Amazon Redshift connectors allow you to leverage storage and data processing services offered by Google and Amazon. Both enable you to purchase petabyte-scale database processing capacity on demand. There is also a connector for the popular cloud-based CRM tool, Salesforce.com, as well as the related Salesforce data products (Force.com and Database.com). Microsoft supplies data over the web via the Windows Azure Marketplace DataMarket and was the founding developer of OData (Open Data Protocol), facilitating the creation and consumption of REST APIs.

Connecting to BigQuery

Let's look at one of the cloud database platforms, Google BigQuery. The connection screens are displayed in Figure 2-11. You can read all of the details about connecting to BigQuery in the help manual, but the setup requires only two steps. First you log in to your Google account; then you authorize Tableau to access your account.

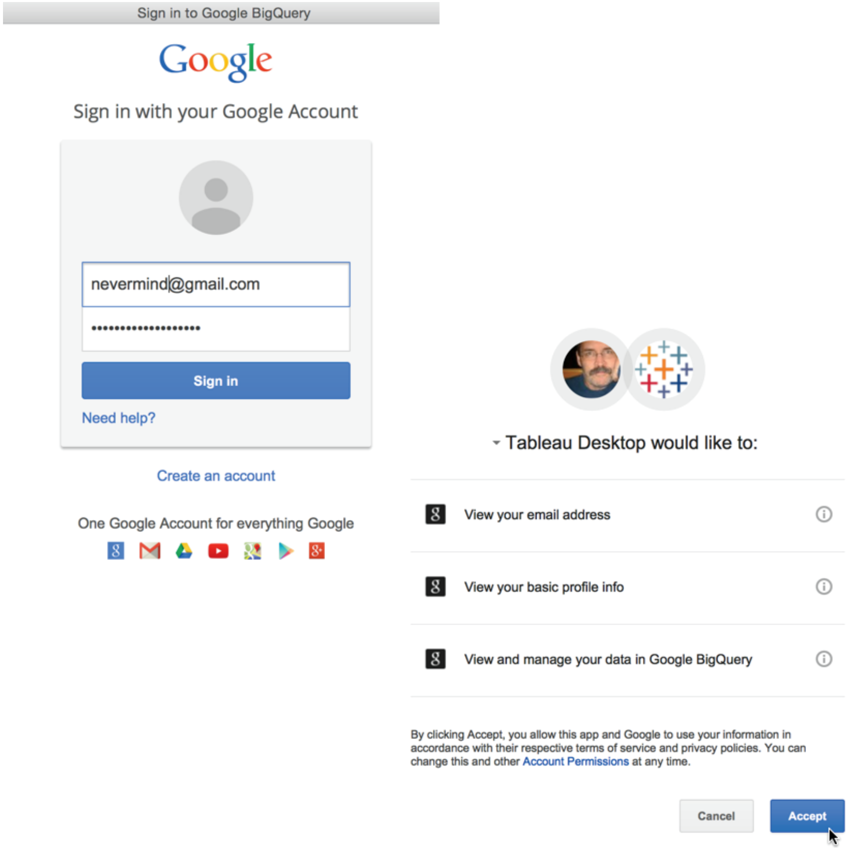


FIGURE 2-11 *BigQuery connection*

I've used BigQuery to build dashboards that have over 100 million records, with less than two-second latency using a standard Internet connection. More businesses are using cloud data services because they are secure and reliable and perform well in many use cases. Tableau Software also provides a free cloud-based service, called Tableau Public, for publishing workbooks and dashboards.

Tableau Public

Tableau Public is the largest deployment of Tableau Server in the world. Thousands of people use it to share dashboards and visualizations with others. Figure 2-12 shows an example dashboard published on Tableau Public that was used in a blog post.

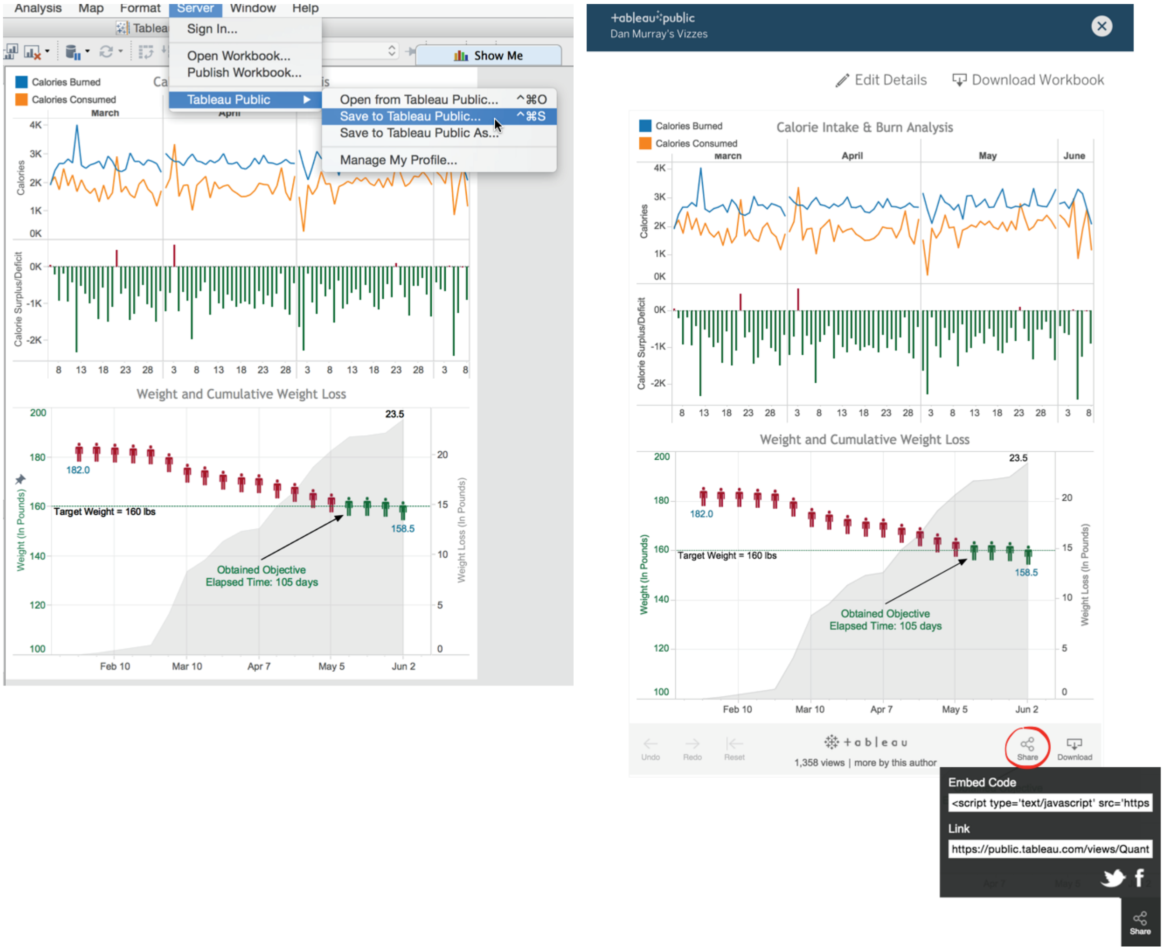


FIGURE 2-12 *Tableau Public*

After you sign up for a free account, two steps are required to share a dashboard on Tableau Public: Build a workbook that includes visualizations and dashboards and then publish the content you wish to share to your Tableau Public account.

The left side of Figure 2-12 shows a dashboard in Tableau Desktop and the menu options for publishing. To publish a workbook, go to the Server menu, select Tableau Public, and then pick the Save to Tableau Public menu option. If you haven't logged into the service, you'll be prompted to log in; then you can define exactly what parts of your workbook you want to publish, along with other options. The right side of Figure 2-12 shows the dashboard in the Tableau Public environment. Note the Share menu option at the bottom right of the image. Clicking that option exposes the embed code and link. If you want to embed the dashboard in a website, you copy that code and paste it into your post. You can also share a link to your published material on Twitter or Facebook.

Tableau has continued to expand the amount of data that Tableau Public users are able to publish. If you don't have access to a licensed copy of Tableau Desktop, download the free Tableau Public desktop version. It works just like the paid desktop license with the notable exception being that the only place you can save your work is Tableau Public. Be careful not to publish proprietary data there as it is available to everyone without restriction.

WHAT ARE GENERATED VALUES?

When you connect to a data source, Tableau creates new fields that make difficult tasks easier. You see these fields in the data pane when you connect to a data source.

Figure 2-13 zooms in on the data pane area. If you want to follow along, the companion website contains the workbook used to build the figures in this section. Open that workbook or connect to one of the sample datasets in your saved data sources.

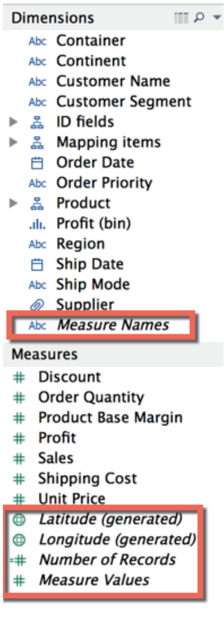


FIGURE 2-13 *Tableau-generated fields*

Measure Names, Measure Values, and Number of Records are always present. If your dimensions include standard geographic place names, Tableau will also generate center-point geocode coordinates.

Measure Names and Measure Values

Measure Names and Measure Values can be used as shortcuts for viewing all the measures in your dataset or to express multiple measures on a single axis.

In Figure 2-14, you can see that two measures are shown, SUM (Profit) and SUM (Sales). These appear as separate columns in the same bar chart. The generated value, Measure Names, is used in the Columns shelf to separate the bars. Measure Names is also used on the Marks card to distinguish color and on the Filters shelf to limit the number of measures shown in the view. Measure Values contains the data, and this is shown as rows as you would expect from this type of bar chart.

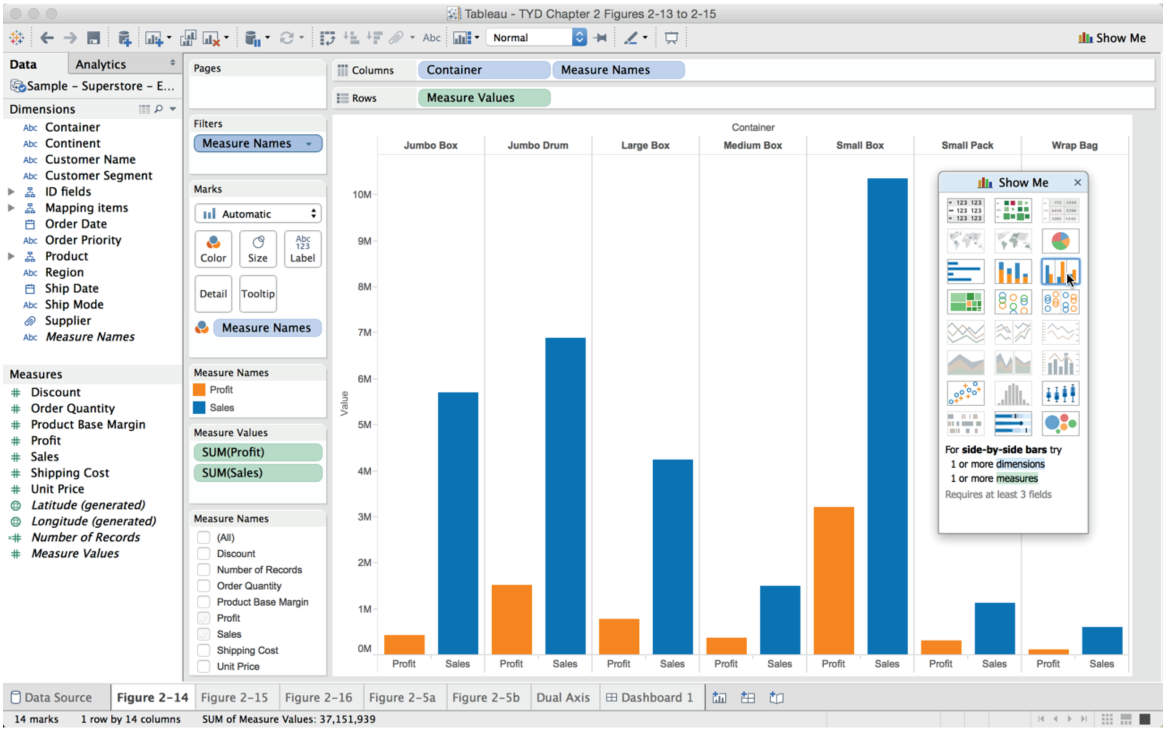


FIGURE 2-14 *Multiple measures on an axis*

The side-by-side bar chart in Figure 2-14 was created by multi-selecting one dimension and two measures. Using Show Me the side-by-side bar chart was selected to create the view. Tableau automatically applied Measure Names to the Columns shelf and plotted both measures on the horizontal axis. The Measure

Names quick filter was exposed by right-clicking Measure Names in the Filters shelf and selecting the Show Quick Filter menu option. Using the Measure Names quick filter, you can add or remove measures from the axis. This style of chart works well only if the measures being plotted have similar value ranges.

There are many ways to combine multiple measures on a single axis. You learn those details in Chapter 3.

Tableau Geocoding

If your data includes standard geographic fields, such as Country, State, Province, City, or Postal Codes—denoted by a small globe icon—Tableau will automatically generate the longitude and latitude values for the center points of each geographic entity displayed in your visualization. If Tableau doesn't recognize a geographic dimension, you can edit the field by right-clicking it and selecting the desired geographic unit. Figure 2-15 shows a map created using country, state, and city and then using Show Me to display a symbol map. The map is filtered using the region field to show only the United States.

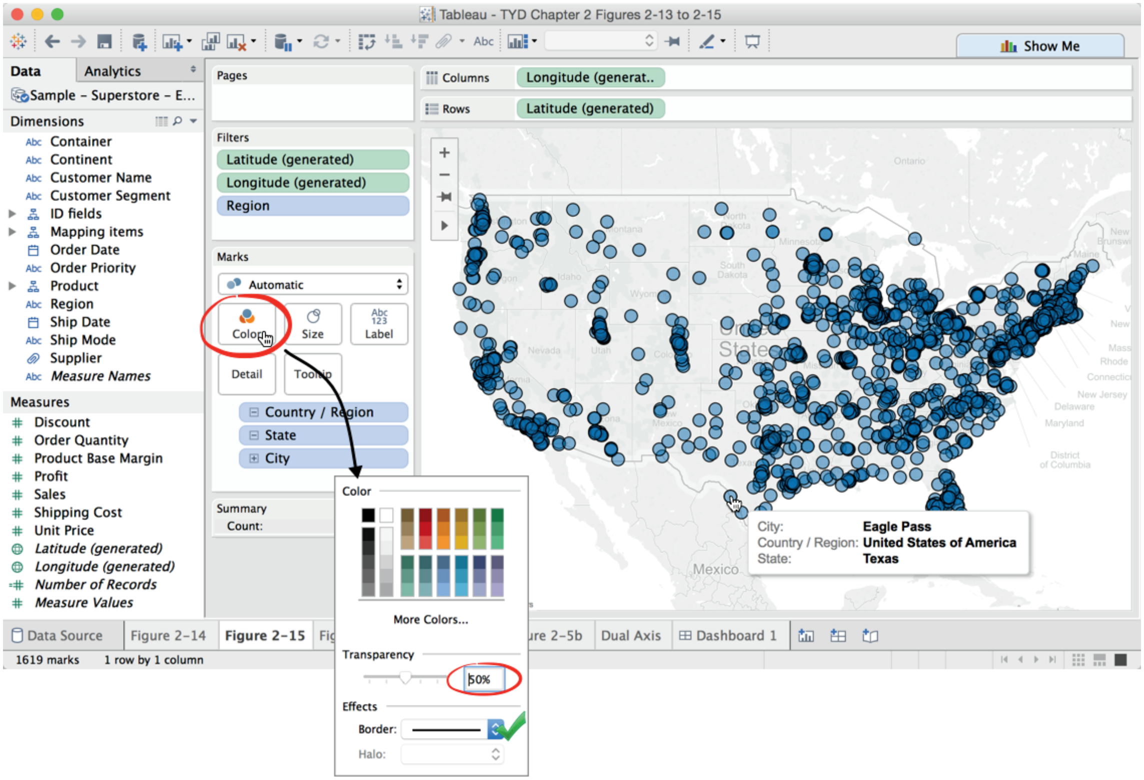


FIGURE 2-15 *Automatic geocoding*

You can edit the color of the marks in the map by clicking the Color button in the Marks card and then selecting the desired color. Figure 2-15 shows the Color dialog box open with Transparency at 50% and a black border. The marks in the map were styled using the Marks card's Color—changing the color transparency to 50% and adding a black border. This makes overlapping clusters of marks easier to see. Hovering over a mark exposes a small dialog box (tooltip) that includes additional details about the mark. The contents of tooltips can be edited by clicking the tooltip button on the Marks card.

The Eagle Pass tooltip shown in Figure 2-15 displays additional facts about the mark selected in the view. Notice that the summary information in the Status bar (lower left) of Figure 2-15 shows you that there are 1,619 marks being plotted in the map. This is the number of cities plotted on the map because city is the most granular standard geographic unit placed on the Marks card.

If Tableau fails to recognize any location, a small gray pill appears in the lower right of the map. Clicking that pill exposes a menu that helps you identify and correct the geocoding. Chapter 5 covers Tableau's mapping capabilities in detail.

Number of Records

The final measure automatically provided by Tableau is a calculated field near the bottom of the Measures shelf called Number of Records. Any icon that includes an equal sign denotes a calculated field. The Number of Records calculation formula is typically the number one. Tableau creates a formula summing the number one to create a count of the rows contained in the data. In the special case of aggregated extracts, it is the number of records that were aggregated into that row. The bar chart in Figure 2-16 displays the record count for each customer segment and the grand total.

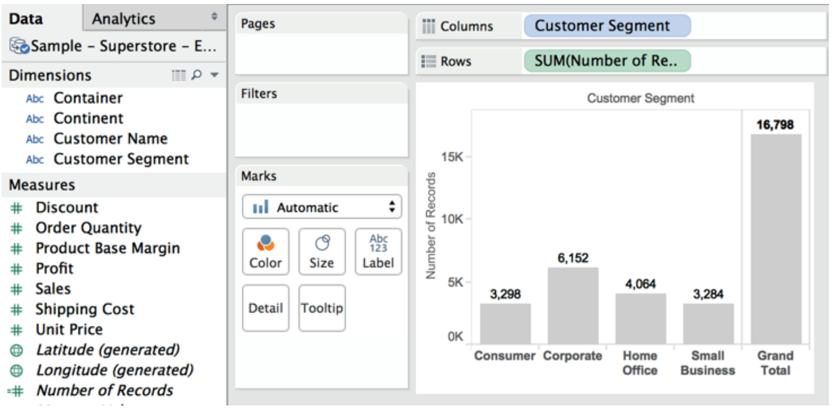


FIGURE 2-16 *Number of Records*

Number of records is a Calculated Value that helps you understand attributes about the data source. It is particularly helpful when you begin to join tables. Monitoring how the record count changes helps you understand data quality issues or design challenges that you may need to address in your visualization.

KNOWING WHEN TO USE A DIRECT CONNECTION OR A DATA EXTRACT

Direct connections allow you work with live data. When you extract data, you import some or all of your data into Tableau's data engine. This is true in Tableau Desktop and Server. Which connection method is the best to use? The answer to that question is dependent on the situation, requirements, and network resources.

The Flexibility of Direct Connections

Connecting to your data source with a direct connection means you are always visualizing the most up-to-date facts. If your database is being updated in real time, you only need to refresh the Tableau visualization via the F5 function key (Command+R on the Mac) or right-click the data source in the data window and select the Refresh option.

If you connect to massive data, the visualization is very dense, or your data is in a high-performance enterprise-class database, you may get a faster response time with a direct connection. Choosing a direct connection doesn't preclude the possibility of extracting the data later. You can also swap from an extract to a live connection by right-clicking the data source and unchecking the Use Extract option.

The Advantages of a Data Extract

Data extracts don't automatically refresh the data. A direct connection provides the latest content of the data source when the connection is made, but using a Tableau data extract can provide a number of benefits:

- Performance improvement
- Additional functions
- Data portability

Performance Improvement

Perhaps your primary database is already heavily loaded with requests. Using Tableau's data engine enables you to split the load from your primary database server to the Tableau Server. Tableau's extract may be updated manually or automatically on a daily, weekly, or monthly basis during off-peak hours. Tableau's server can also refresh extracts incrementally and in time intervals as low as 15 minutes. In many cases, the small time consumed during the data extract update is more than offset by the performance gains.

There are several options available for creating an extract. First, you can aggregate the extract, which will roll up rows so that only the aggregation and fields used are updated for the visible dimensions and measures. Aggregating for visible dimensions when performing a data extract will reduce the amount of data that Tableau is importing. Selecting this option will cause Tableau to summarize the underlying data at the level of detail visible in your view. The appropriate level of fidelity is provided, and the size of the extract file is reduced—which makes the extract file more portable while also improving security by removing details that you don't wish to share.

Extracting incrementally also speeds refresh time because Tableau isn't updating the entire extract file. It is adding only new records. To do incremental extracts, you must specify a field to use as the index; Tableau will refresh the row only if the index has changed, so you need to be aware that changes to a row of data that doesn't change the index field will be excluded from the update.

Another way to speed extracts is to apply filters when extracting the data. If the analysis doesn't require your entire dataset, you can filter the extract to include only the records required. If you have a very large dataset, you will rarely need to extract the entire contents of the database. For example, your database may include ten years of historical data but you may require only one year of history.

Once you have created an extract, you may append another file. This may be a great alternative to custom SQL if you are considering a table UNION (applicable for spreadsheet sources on versions prior to 8.2 only). This technique might be useful if you need to combine monthly data that is stored in separate tables.

Additional Functions

If you are using Tableau Desktop V8.1 or earlier and the data source is from a file (Excel, Access, text), doing an extract will add calculation functions (median and count distinct) that are not supported by the data source. This is no longer required in V8.2 and later.

Data Portability

Extracts can be saved locally and used when a direct connection to your data source is not available. A direct connection doesn't work if you don't have access to your data source via a local network or Internet connection. For example, you may need to supply a dashboard to an executive that will be flying to a remote location. Using a data extract (`.tde`) file provides the user access to a fully functional, high-performance, local data source. Data extract files are also compressed and are normally much smaller than the host system database tables.

In enterprise environments, data governance is an important consideration. If you distribute many data extract files to field staff, keep in mind that you should consider the security of those files. Appropriate safeguards should be in place (non-disclosure agreements) before you provide these files to traveling or remote staff. Consider restricting what the extract includes via filters and aggregating for visible dimensions.

USING TABLEAU'S FILE TYPES EFFECTIVELY

Tableau provides flexible options for the sharing of data and design metadata. This is accomplished through a variety of file types:

- Tableau Workbook (`.twb`)
- Tableau Packaged Workbook (`.twbx`)
- Tableau Datasource (`.tds`)
- Tableau Bookmark (`.twb`)
- Tableau Data Extract (`.tde`)

These files are saved within the My Tableau Repository folder located in My Documents by default.

Tableau Workbook Files

Tableau Workbook files (`.twb`) are the default file type created by Tableau when you save workbooks. These are usually small files because they contain metadata related to your connection, field pill placements for rendering the views along with metadata associated with field aliases, renamed fields, and calculations.

Workbook files do not contain any of the actual data from the database. For this reason, workbook files will normally be small. Every time you open a workbook file that has a direct connection to a live data source, you are looking at the most up-to-date data.

Tableau Packaged Workbooks

To share a workbook with someone who doesn't have access to the data source used to create the workbook, save it as a packaged workbook (`.twbx`).

Packaged workbooks (`.twbx`) bundle the data and metadata into a single file. If you later need to access the original data source file contained within the packaged workbook, right-click the `.twbx` file in an Explorer window, and select the Unpackage menu option.

People without licensed access to Tableau Server or Tableau Desktop can view packaged workbooks using a free tool provided by Tableau Software called Tableau Reader. If you are distributing sensitive or proprietary data, be aware that packaged workbook files are zip files. Packaged workbooks can be unzipped exposing potentially sensitive information. While you can mitigate this by filtering and aggregating the extract when it's created, this isn't a replacement for Tableau Server. The server environment provides robust data security and data governance features. The chapters on Tableau Server in Part II cover security and governance options in detail.

Tableau Data Source Files

Changes made within your data pane (the left side of the desktop workspace) alter the metadata of your connection. Grouping, sets, aliased names, field-type changes, and any other modifications made in your workbook are part of the metadata. Can you share just the metadata with others? The answer is yes. This is done by creating a Tableau Data Source (`.tds`) file.

A Tableau data source file defines where the source data is, how to connect to it, what field names have been changed, and other changes applied in the dimensions and measures shelves. Data source files can be saved locally or published to Tableau Server. This is particularly helpful if you work in a large enterprise. Perhaps you have a small number of database experts that understand your database schema well. They can create the connection, define table joins, group or rename fields, and then publish the data source file for other staff to use as a starting point.

To create a data source file, right-click the filename in your data window; then select the Add To Saved Data Sources option. Data source files are placed in the My Tableau Repository/Datasource folder. Additionally, files placed in that folder are automatically displayed as saved data connections on Tableau Desktop's start page.

Alternatively, you can publish data source files to Tableau server and share them with other staff. This is a great option for sharing workbooks containing complicated database joins that must be done by more technical staff. In this way the knowledge of the most experienced and knowledge staff can be shared with a larger number of analysts. These data source files can serve as a starting point for analysis by operations team members that may know more about the business domain.

Changes made to Data Source files (`.tdsx`) are available to everyone who has been given permission to access the file on Tableau Server. When changes are made to these published connections, everyone who has been granted access to the file will have the latest version.

Tableau Bookmark Files

What if you have a massive workbook (with many worksheets) and you want to share one worksheet only with a colleague? This is done by using a bookmark (`.tbm`) file. Bookmark files save the data and metadata related to a worksheet within your workbook––including the connection and calculated fields.

To create a bookmark file, go to the Window menu bar and look for the Bookmark menu option and select Create Bookmark. The bookmark will become visible when a new Tableau session is started. The file will appear in the Window menu. Opening the bookmark file will initiate the connection and add it to the workbook. Tableau bookmark files are stored in your Tableau Repository in the Bookmarks folder.

Copying Contents

A new feature provides the ability to copy the contents of worksheet from one workbook to a different workbook. You can do this by selecting the tabs you want to copy, right-clicking, and then selecting the Copy Sheet menu option. Then, open a new workbook and paste the contents. This copies all of the metadata and data source connections to the new workbook.

In the remainder of this chapter, you learn the challenges related to establishing data connections. You also learn how to deal with data quality and data structure issues, join tables, and blend data from disparate data sources.

DEALING WITH DATA SHAPING AND DATA QUALITY

Inaccurate data can lead to bad decisions. If your data isn't clean, errors and missing information won't necessarily be easy for knowledgeable information consumers to identify. If you create reports and analysis that other people rely upon, you should do your best to find and correct bad or missing data before you share your workbook. If you choose to share unaudited data, you should disclose potential data quality issues to the people who will be using the information.

Tableau provides tools to help you deal with issues without requiring intervention at the database level. This is helpful if your time is limited or key technical staff are not available to fix the problem.

The best course of action when you find errors is to report them to the IT person responsible for data quality within the database you are using. If you bring together data from many different sources into a spreadsheet, it is a good idea to make notes in your workbook to document the process you went through to build the analysis. This lends credibility to your work.

Quick Data Shaping and Editing in Tableau

There are several different ways you can enhance and shape data while you are building views and stories or dashboards. Whether you are connecting to an enterprise-class database or a spreadsheet, Tableau makes it easy to refine filenames, group data elements, create user-friendly naming of fields and data elements, fix unidentified geographic place names, and deal with unmatched null values.

Renaming

You rename a field in Tableau by right-clicking the field and typing in a new name. Field member names can also be aliased. These changes do not alter the source database. Tableau "remembers" what you renamed without altering the source data.

Grouping

Let's assume that a company name has been entered using all of these variations: A&M, A & M, A and M, A+M. With Tableau, you can multi-select (Control+Select in Windows or Command+Select in the Mac) each of these names—group them—and create a name alias for the ad hoc grouping. So, all the versions of the name appear as one record in Tableau—A&M. This grouping and name alias is saved as part of Tableau's metadata.

Using Name Aliases to Provide Better Descriptions

It is common for database designers to categorize rows contained in a database based on a range of values or some other attribute of interest. This makes it easier to build drill-down reports. Sometimes these codes are not very descriptive to the people accessing the data. For example, the database designer may have created codes (P1, P2, G1, G2) to describe ranges of annual sales values.

In Tableau, you can alias these codes with more descriptive titles. P2 might mean Platinum Level 2, which is used to describe customers with annual sales of $1 million to $5 million. By right-clicking the P2 field name and selecting the Edit Alias option, you can create a more descriptive name (P2 - Annual Rev $2M to $5M). The Name Alias will now be how the P2 description will appear in Tableau.

Geographic Errors

Although Tableau has built-in mapping that works very well, there will be occasions when geographic locations are not recognized. Tableau will warn you by placing a small gray pill in the lower-right area of your map. Click that pill to edit problem locations or filter them out of view. This is also accessible from Tableau's map menu.

Null Values

When you see the word "null" appear in a view, that means Tableau can't match the record. You can filter out nulls, group them with non-null members of the set, or correct the data causing the null. There are many reasons why a null value could result. If you aren't sure how to correct the null, seek assistance from a qualified technical resource.

Data Quality Challenges

Tableau makes it easy to connect to a wide variety of databases. Today's marketplace rewards speed. Being the first to gain insight into something of importance has value. The "single-version-of-the-truth" ethos that pervades the database community is intended to ensure data quality, but it comes at the cost of speed. This is why people with access to a database often get time-sensitive information in spreadsheets. They can't afford to wait. Or, if you are like the majority of people in the world, most of your analysis is done using a spreadsheet.

Spreadsheet data is readily obtainable but comes with higher data quality risk. Poor data quality can be broken down into three major areas:

- Wrong data
- Missing data
- Poor data structure

Using a spreadsheet as a data source places responsibility on you to ensure that the data in the spreadsheet is accurate and complete. If that isn't the case, you have to decide if the data is good enough for your purpose. Tableau can't help you deal with this directly. The main problem with spreadsheets is that the data analysis and presentation layers are in the same place. The format used in spreadsheets doesn't always comply with the best practices of data storage. The structure of the rows and columns in your spreadsheet may not support the analysis you need to do. In addition, the data may be completely unaudited. As the designer, you assume responsibility for wrong or missing data.

For years, Tableau provided an Excel add-on tool called the Data Reshaper. It reduced the time required to deal with data structure issues. Many of the features of the Excel add-in are now built directly into the Tableau's data source page. In the next section, you learn how to use the Tableau Data Interpreter to clean up and reformat spreadsheet data.

THE DATA INTERPRETER

Tableau's Data Interpreter was introduced in V9.0 and is intended to help you deal with poorly formatted and structured spreadsheet data. It includes a variety of tools to help fix problems and address issues with your data source:

- Pivoting columns
- Reformatting data
- Renaming headings
- Splitting cells
- Changing data types
- Hiding unneeded data

Using the Preview Area in the Connect Page in Figure 2-3, you can determine the condition of the data and whether or not reshaping your data is necessary. If the data doesn't look right, see if the Data Interpreter can help by clicking the Turn on option.

The example you'll work through in the next section uses nearly all of the Data Interpreter's toolset. The resulting clean dataset will be used in the data blending section at the end of this chapter.

Preparing a Spreadsheet for Tableau

This example utilizes a spreadsheet with population data from the United States Census Bureau.[2]

A copy of the source spreadsheet (`NST-EST2014-0.xlsx`) is provided on the companion website. Figure 2-17 provides a view of the unedited data.

The spreadsheet includes census information and population projections for April 2010 through July 2014 for the entire country, regions, states, and Puerto Rico. It's formatted for presenting and consuming the data within the spreadsheet and is not ideally formatted to serve as a data source.

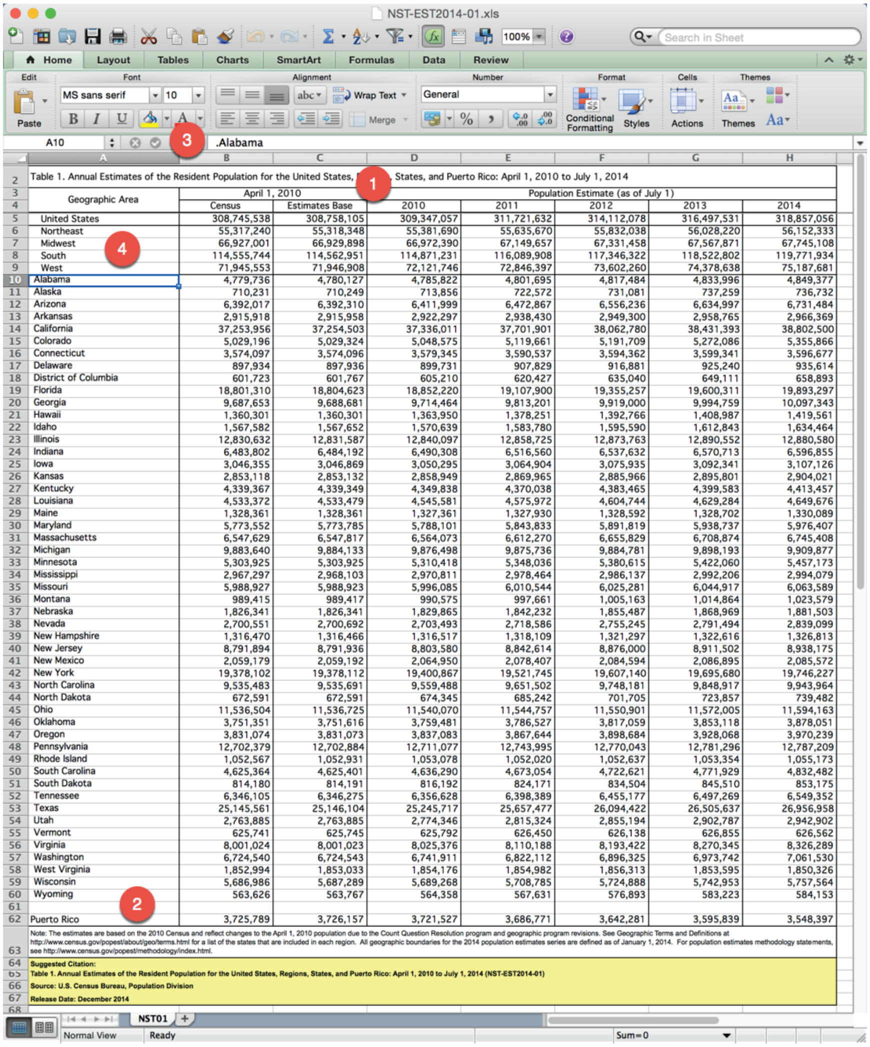


Table 1. Annual Estimates of the Resident Population for the United States, [illegible] States, and Puerto Rico: April 1, 2010 to July 1, 2014

Geographic Area	April 1, 2010		Population Estimate (as of July 1)				
	Census	Estimates Base	2010	2011	2012	2013	2014
United States	308,745,538	308,758,105	309,347,057	311,721,632	314,112,078	316,497,531	318,857,056
Northeast	55,317,240	55,318,348	55,381,690	55,635,670	55,832,038	56,028,220	56,152,333
Midwest	66,927,001	66,929,898	66,972,390	67,149,657	67,331,458	67,567,871	67,745,108
South	114,555,744	114,562,951	114,871,231	116,089,908	117,346,322	118,522,802	119,771,934
West	71,945,553	71,946,908	72,121,746	72,846,397	73,602,260	74,378,638	75,187,681
Alabama	4,779,736	4,780,127	4,785,822	4,801,695	4,817,484	4,833,996	4,849,377
Alaska	710,231	710,249	713,856	722,572	731,081	737,259	736,732
Arizona	6,392,017	6,392,310	6,411,999	6,472,867	6,556,236	6,634,997	6,731,484
Arkansas	2,915,918	2,915,958	2,922,297	2,938,430	2,949,300	2,958,765	2,966,369
California	37,253,956	37,254,503	37,336,011	37,701,901	38,062,780	38,431,393	38,802,500
Colorado	5,029,196	5,029,324	5,048,575	5,119,661	5,191,709	5,272,086	5,355,866
Connecticut	3,574,097	3,574,096	3,579,345	3,590,537	3,594,362	3,599,341	3,596,677
Delaware	897,934	897,936	899,731	907,829	916,881	925,240	935,614
District of Columbia	601,723	601,767	605,210	620,427	635,040	649,111	658,893
Florida	18,801,310	18,804,623	18,852,220	19,107,900	19,355,257	19,600,311	19,893,297
Georgia	9,687,653	9,688,681	9,714,464	9,813,201	9,919,000	9,994,759	10,097,343
Hawaii	1,360,301	1,360,301	1,363,950	1,378,251	1,392,766	1,408,987	1,419,561
Idaho	1,567,582	1,567,652	1,570,639	1,583,780	1,595,590	1,612,843	1,634,464
Illinois	12,830,632	12,831,587	12,840,097	12,858,725	12,873,763	12,890,552	12,880,580
Indiana	6,483,802	6,484,192	6,490,308	6,516,560	6,537,632	6,570,713	6,596,855
Iowa	3,046,355	3,046,869	3,050,295	3,064,904	3,075,935	3,092,341	3,107,126
Kansas	2,853,118	2,853,132	2,858,949	2,869,965	2,885,966	2,895,801	2,904,021
Kentucky	4,339,367	4,339,349	4,349,838	4,370,038	4,383,465	4,399,583	4,413,457
Louisiana	4,533,372	4,533,479	4,545,581	4,575,972	4,604,744	4,629,284	4,649,676
Maine	1,328,361	1,328,361	1,327,361	1,327,930	1,328,592	1,328,702	1,330,089
Maryland	5,773,552	5,773,785	5,788,101	5,843,833	5,891,819	5,938,737	5,976,407
Massachusetts	6,547,629	6,547,817	6,564,073	6,612,270	6,655,829	6,708,874	6,745,408
Michigan	9,883,640	9,884,133	9,876,498	9,875,736	9,884,781	9,898,193	9,909,877
Minnesota	5,303,925	5,303,925	5,310,418	5,348,036	5,380,615	5,422,060	5,457,173
Mississippi	2,967,297	2,968,103	2,970,811	2,978,464	2,986,137	2,992,206	2,994,079
Missouri	5,988,927	5,988,923	5,996,085	6,010,544	6,025,281	6,044,917	6,063,589
Montana	989,415	989,417	990,575	997,661	1,005,163	1,014,864	1,023,579
Nebraska	1,826,341	1,826,341	1,829,865	1,842,232	1,855,487	1,868,969	1,881,503
Nevada	2,700,551	2,700,692	2,703,493	2,718,586	2,755,245	2,791,494	2,839,099
New Hampshire	1,316,470	1,316,466	1,316,517	1,318,109	1,321,297	1,322,616	1,326,813
New Jersey	8,791,894	8,791,936	8,803,580	8,842,614	8,876,000	8,911,502	8,938,175
New Mexico	2,059,179	2,059,192	2,064,950	2,078,407	2,084,594	2,086,895	2,085,572
New York	19,378,102	19,378,112	19,400,867	19,521,745	19,607,140	19,695,680	19,746,227
North Carolina	9,535,483	9,535,691	9,559,488	9,651,502	9,748,181	9,848,917	9,943,964
North Dakota	672,591	672,591	674,345	685,242	701,705	723,857	739,482
Ohio	11,536,504	11,536,725	11,540,070	11,544,757	11,550,901	11,572,005	11,594,163
Oklahoma	3,751,351	3,751,616	3,759,481	3,786,527	3,817,059	3,853,118	3,878,051
Oregon	3,831,074	3,831,073	3,837,083	3,867,644	3,898,684	3,928,068	3,970,239
Pennsylvania	12,702,379	12,702,884	12,711,077	12,743,995	12,770,043	12,781,296	12,787,209
Rhode Island	1,052,567	1,052,931	1,053,078	1,052,020	1,052,637	1,053,354	1,055,173
South Carolina	4,625,364	4,625,401	4,636,290	4,673,054	4,722,621	4,771,929	4,832,482
South Dakota	814,180	814,191	816,192	824,171	834,504	845,510	853,175
Tennessee	6,346,105	6,346,275	6,356,628	6,398,389	6,455,177	6,497,269	6,549,352
Texas	25,145,561	25,146,104	25,245,717	25,657,477	26,094,422	26,505,637	26,956,958
Utah	2,763,885	2,763,885	2,774,346	2,815,324	2,855,194	2,902,787	2,942,902
Vermont	625,741	625,745	625,792	626,450	626,138	626,855	626,562
Virginia	8,001,024	8,001,023	8,025,376	8,110,188	8,193,422	8,270,345	8,326,289
Washington	6,724,540	6,724,543	6,741,911	6,822,112	6,896,325	6,973,742	7,061,530
West Virginia	1,852,994	1,853,033	1,854,176	1,854,982	1,856,313	1,853,595	1,850,326
Wisconsin	5,686,986	5,687,289	5,689,268	5,708,785	5,724,888	5,742,953	5,757,564
Wyoming	563,626	563,767	564,358	567,631	576,893	583,223	584,153
Puerto Rico	3,725,789	3,726,157	3,721,527	3,686,771	3,642,281	3,595,839	3,548,397

Note: The estimates are based on the 2010 Census and reflect changes to the April 1, 2010 population due to the Count Question Resolution program and geographic program revisions. See Geographic Terms and Definitions at http://www.census.gov/popest/about/geo/terms.html for a list of the states that are included in each region. All geographic boundaries for the 2014 population estimates series are defined as of January 1, 2014. For population estimates methodology statements, see http://www.census.gov/popest/methodology/index.html.

Suggested Citation:
Table 1. Annual Estimates of the Resident Population for the United States, Regions, States, and Puerto Rico: April 1, 2010 to July 1, 2014 (NST-EST2014-01)
Source: U.S. Census Bureau, Population Division
Release Date: December 2014

FIGURE 2-17 *Census Bureau spreadsheet*

The data should be an unbroken list with a single header row; that data should be consistent, and the list should be more row-oriented. Ideally, the multiple columns displaying the population counts and projections should be in a single column and a new reference column added that defines type of data being presented in the data column.

Areas 1 and 2 in Figure 2-17 highlight some issues. Focusing on Area 1, you see that multiple headers contain similar measures (population estimates) applied across more than one column. In Area 2 you see a gap in the rows between Wyoming and Puerto Rico. Having each year in a separate column is typical in a spreadsheet, but Tableau will interpret each year as a different measure. That isn't necessarily bad but will not suit our needs. While each column represents a different time period, the facts collected include an actual census and estimated projections using the census as a base.

Less obvious problems exist in this dataset. For example, there are periods inserted in front of the state names that you can't see in column A in Figure 2-17 but are visible in the field displaying Alabama (Area 3). The geographic units presented in column A (Area 4) are not consistent. Country, region, and state are all included in the same column. Let's see how the Data Interpreter can help you address these issues.

Using the Data Interpreter to Prepare the Data for Analysis in Tableau

You will use the Data Interpreter to fix data structure problems. The other issues with the data quality and consistency will be addressed using a Tableau function and filtering. The steps for this exercise include

1. Open Tableau and connect to `NST-EST2014-01.xlsx`.
2. Use the Sheets preview to review the data source.
3. Turn on the Data Interpreter.
4. Rename the headings.
5. Split the period from the heading in the Geographic Area field.
6. Rename the split field as "state."
7. Pivot the year columns.
8. Rename the pivoted fields (state and year).
9. Define the state field data type as geographic (state).
10. Build examples to validate cleanup.

Before you connect to the census data spreadsheet, open it and review the layout and contents. Get familiar with the layout in the NST01 tab. Now open Tableau and connect to the data source from the Start page by selecting the Excel option in Tableau's Start page and picking the `NST-EST2014-01.xlsx` file from the location you stored it on your computer. Alternatively, you can drag and drop the spreadsheet on the Start page. This will open the file in Tableau as well.

Now that you've connected Tableau to the file, notice Tableau displays two sheets. The green symbol indicates that the spreadsheet contains named ranges. The second sheet circled in Figure 2-18 is the raw spreadsheet data. Double-click the second version of NST01.

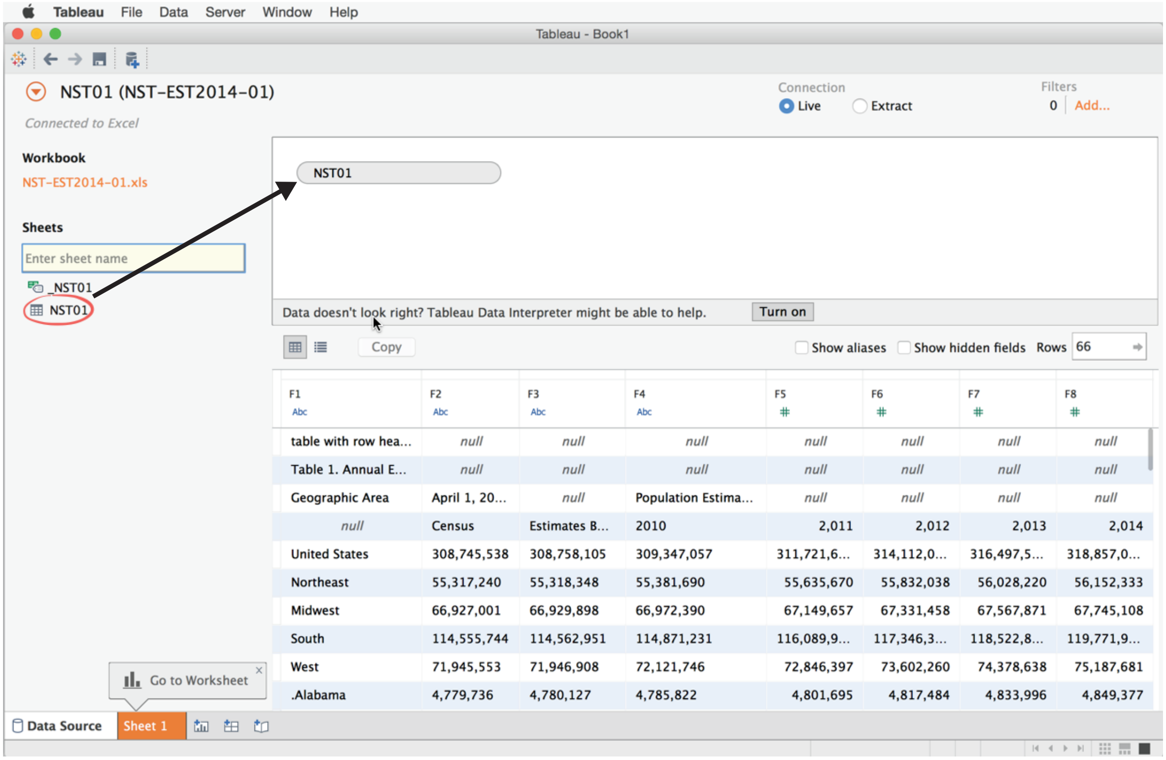


FIGURE 2-18 *Data source connection*

This will add the sheet (table) to the join area and display the contents in the Preview area, as you see in Figure 2-18.

Preview the Sheet Content

Focus on the Preview Area in Figure 2-19. You can see that the headings in the first four rows are a problem. There are lots of null values and missing heading names.

Remove the NST01 table from the join area by clicking the gray pill in the join area and then click the red X to remove it. Or, just drag it off the screen. Then drag the first NST01 sheet with the green icon into the join pane. Notice that this named range version of the connection looks a little better. The range names in the spreadsheet were defined by the person who created the spreadsheet at the Census Bureau. Let's work with the ugly version so that we give the Data Interpreter a bigger challenge.

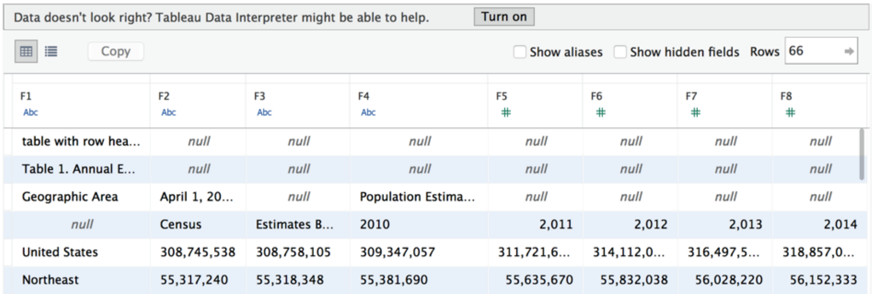


FIGURE 2-19 *Turn on the Data Interpreter.*

Replace the NST01 (named range version) with the second NST01 sheet by dragging it on top of the named range version in the view.

Turn on the Data Interpreter

Turn on the Data Interpreter by clicking the Turn on button, which you can see at the top of Figure 2-19. This applies the Data Interpreter to the dataset and causes the Preview area to reformat the data, as you see in Figure 2-20.

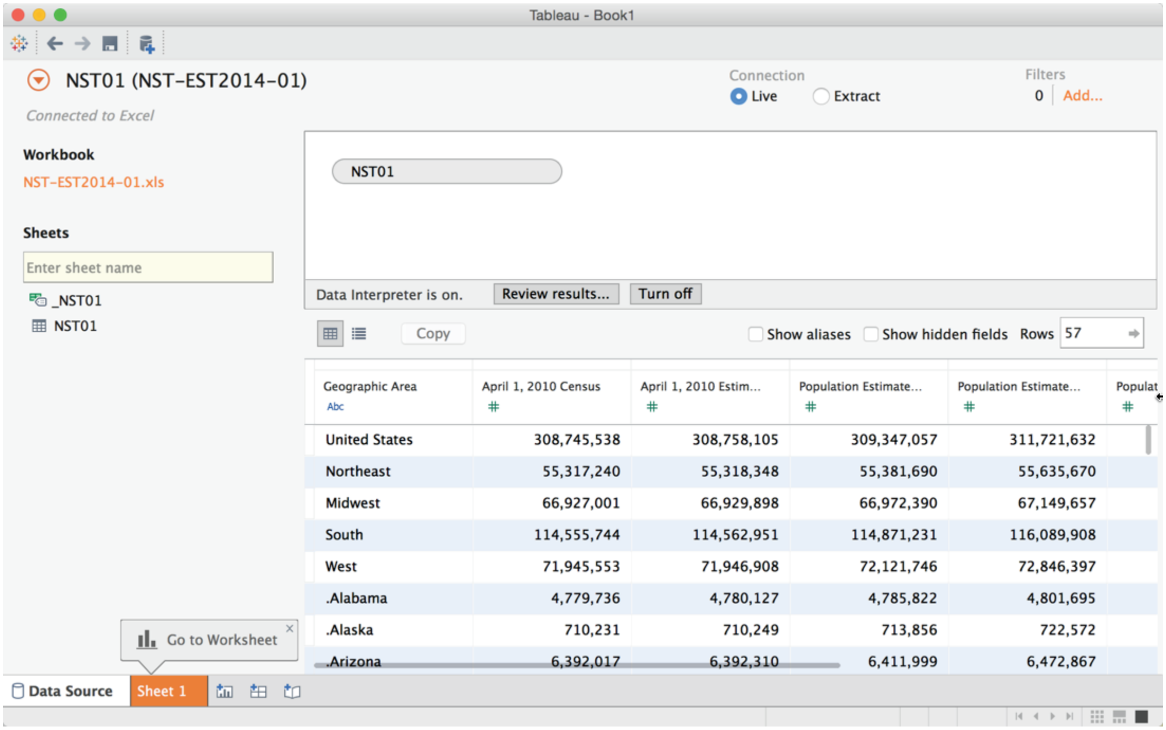
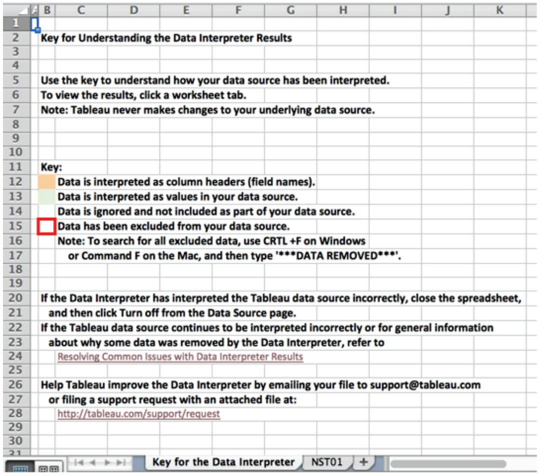


FIGURE 2-20 *Data Interpreter initiated*

You can see data looks more orderly. Clicking the Review results button at the top of the preview area generates a spreadsheet analysis of what Tableau did to reformat the data. Figure 2-21 shows a screenshot of the Review Results spreadsheet.

On the left side is a key that describes the color-encoding used in the view on the right side of Figure 2-21. Based on improved Preview area view and the additional confirmation from the review generated by Tableau, you can access the quality of the changes made by the Data Interpreter. If the data or formatting doesn't look right, you can turn off the interpreter and deal with the problems directly in the spreadsheet.



Key for Understanding the Data Interpreter Results

Use the key to understand how your data source has been interpreted.
To view the results, click a worksheet tab.
Note: Tableau never makes changes to your underlying data source.

Key:
Data is interpreted as column headers (field names).
Data is interpreted as values in your data source.
Data is ignored and not included as part of your data source.
Data has been excluded from your data source.
Note: To search for all excluded data, use CRTL +F on Windows
or Command F on the Mac, and then type '***DATA REMOVED***'.

If the Data Interpreter has interpreted the Tableau data source incorrectly, close the spreadsheet,
and then click Turn off from the Data Source page.
If the Tableau data source continues to be interpreted incorrectly or for general information
about why some data was removed by the Data Interpreter, refer to
Resolving Common Issues with Data Interpreter Results

Help Tableau improve the Data Interpreter by emailing your file to support@tableau.com
or filing a support request with an attached file at:
http://tableau.com/support/request

Key for the Data Interpreter | NST01

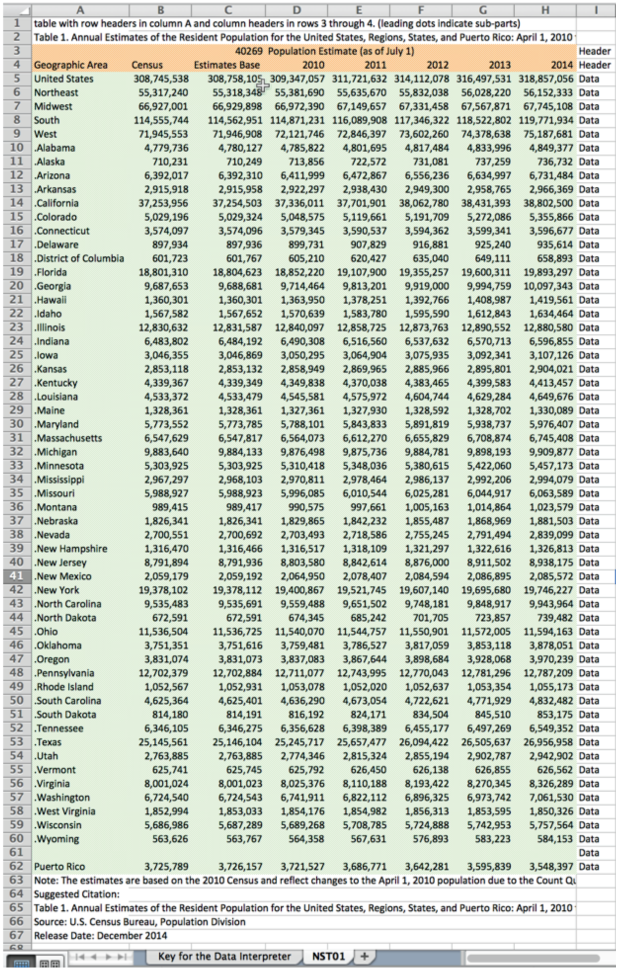


table with row headers in column A and column headers in rows 3 through 4. (leading dots indicate sub-parts)
Table 1. Annual Estimates of the Resident Population for the United States, Regions, States, and Puerto Rico: April 1, 2010

	A	B	C	D	E	F	G	H	I
3			40269	Population Estimate (as of July 1)					Header
4	Geographic Area	Census	Estimates Base	2010	2011	2012	2013	2014	Header
5	United States	308,745,538	308,758,105	309,347,057	311,721,632	314,112,078	316,497,531	318,857,056	Data
6	Northeast	55,317,240	55,318,348	55,381,690	55,635,670	55,832,038	56,028,220	56,152,333	Data
7	Midwest	66,927,001	66,929,898	66,972,390	67,149,657	67,331,458	67,567,871	67,745,108	Data
8	South	114,555,744	114,562,951	114,871,231	116,089,908	117,346,322	118,522,802	119,771,934	Data
9	West	71,945,553	71,946,908	72,121,746	72,846,397	73,602,260	74,378,638	75,187,681	Data
10	.Alabama	4,779,736	4,780,127	4,785,822	4,801,695	4,817,484	4,833,996	4,849,377	Data
11	.Alaska	710,231	710,249	713,856	722,572	731,081	737,259	736,732	Data
12	.Arizona	6,392,017	6,392,310	6,411,999	6,472,867	6,556,236	6,634,997	6,731,484	Data
13	.Arkansas	2,915,918	2,915,958	2,922,297	2,938,430	2,949,300	2,958,765	2,966,369	Data
14	.California	37,253,956	37,254,503	37,336,011	37,701,901	38,062,780	38,431,393	38,802,500	Data
15	.Colorado	5,029,196	5,029,324	5,048,575	5,119,661	5,191,709	5,272,086	5,355,866	Data
16	.Connecticut	3,574,097	3,574,096	3,579,345	3,590,537	3,594,362	3,599,341	3,596,677	Data
17	.Delaware	897,934	897,936	899,731	907,829	916,881	925,240	935,614	Data
18	.District of Columbia	601,723	601,767	605,210	620,427	635,040	649,111	658,893	Data
19	.Florida	18,801,310	18,804,623	18,852,220	19,107,900	19,355,257	19,600,311	19,893,297	Data
20	.Georgia	9,687,653	9,688,681	9,714,464	9,813,201	9,919,000	9,994,759	10,097,343	Data
21	.Hawaii	1,360,301	1,360,301	1,363,950	1,378,251	1,392,766	1,408,987	1,419,561	Data
22	.Idaho	1,567,582	1,567,652	1,570,639	1,583,780	1,595,590	1,612,843	1,634,464	Data
23	.Illinois	12,830,632	12,831,587	12,840,097	12,858,725	12,873,763	12,890,552	12,880,580	Data
24	.Indiana	6,483,802	6,484,192	6,490,308	6,516,560	6,537,632	6,570,713	6,596,855	Data
25	.Iowa	3,046,355	3,046,869	3,050,295	3,064,904	3,075,935	3,092,341	3,107,126	Data
26	.Kansas	2,853,118	2,853,132	2,858,949	2,869,965	2,885,966	2,895,801	2,904,021	Data
27	.Kentucky	4,339,367	4,339,349	4,349,838	4,370,038	4,383,465	4,399,583	4,413,457	Data
28	.Louisiana	4,533,372	4,533,479	4,545,581	4,575,972	4,604,744	4,629,284	4,649,676	Data
29	.Maine	1,328,361	1,328,361	1,327,361	1,327,930	1,328,592	1,328,702	1,330,089	Data
30	.Maryland	5,773,552	5,773,785	5,788,101	5,843,833	5,891,819	5,938,737	5,976,407	Data
31	.Massachusetts	6,547,629	6,547,817	6,564,073	6,612,270	6,655,829	6,708,874	6,745,408	Data
32	.Michigan	9,883,640	9,884,133	9,876,498	9,875,736	9,884,781	9,898,193	9,909,877	Data
33	.Minnesota	5,303,925	5,303,925	5,310,418	5,348,036	5,380,615	5,422,060	5,457,173	Data
34	.Mississippi	2,967,297	2,968,103	2,970,811	2,978,464	2,986,137	2,992,206	2,994,079	Data
35	.Missouri	5,988,927	5,988,923	5,996,085	6,010,544	6,025,281	6,044,917	6,063,589	Data
36	.Montana	989,415	989,417	990,575	997,661	1,005,163	1,014,864	1,023,579	Data
37	.Nebraska	1,826,341	1,826,341	1,829,865	1,842,232	1,855,487	1,868,969	1,881,503	Data
38	.Nevada	2,700,551	2,700,692	2,703,493	2,718,586	2,755,245	2,791,494	2,839,099	Data
39	.New Hampshire	1,316,470	1,316,466	1,316,517	1,318,109	1,321,297	1,322,616	1,326,813	Data
40	.New Jersey	8,791,894	8,791,936	8,803,580	8,842,614	8,876,000	8,911,502	8,938,175	Data
41	.New Mexico	2,059,179	2,059,192	2,064,950	2,078,407	2,084,594	2,086,895	2,085,572	Data
42	.New York	19,378,102	19,378,112	19,400,867	19,521,745	19,607,140	19,695,680	19,746,227	Data
43	.North Carolina	9,535,483	9,535,691	9,559,488	9,651,502	9,748,181	9,848,917	9,943,964	Data
44	.North Dakota	672,591	672,591	674,345	685,242	701,705	723,857	739,482	Data
45	.Ohio	11,536,504	11,536,725	11,540,070	11,544,757	11,550,901	11,572,005	11,594,163	Data
46	.Oklahoma	3,751,351	3,751,616	3,759,481	3,786,527	3,817,059	3,853,118	3,878,051	Data
47	.Oregon	3,831,074	3,831,073	3,837,083	3,867,644	3,898,684	3,928,068	3,970,239	Data
48	.Pennsylvania	12,702,379	12,702,884	12,711,077	12,743,995	12,770,043	12,781,296	12,787,209	Data
49	.Rhode Island	1,052,567	1,052,931	1,053,078	1,052,020	1,052,637	1,053,354	1,055,173	Data
50	.South Carolina	4,625,364	4,625,401	4,636,290	4,673,054	4,722,621	4,771,929	4,832,482	Data
51	.South Dakota	814,180	814,191	816,192	824,171	834,504	845,510	853,175	Data
52	.Tennessee	6,346,105	6,346,275	6,356,628	6,398,389	6,455,177	6,497,269	6,549,352	Data
53	.Texas	25,145,561	25,146,104	25,245,717	25,657,477	26,094,422	26,505,637	26,956,958	Data
54	.Utah	2,763,885	2,763,885	2,774,346	2,815,324	2,855,194	2,902,787	2,942,902	Data
55	.Vermont	625,741	625,745	625,792	626,450	626,138	626,855	626,562	Data
56	.Virginia	8,001,024	8,001,023	8,025,376	8,110,188	8,193,422	8,270,345	8,326,289	Data
57	.Washington	6,724,540	6,724,543	6,741,911	6,822,112	6,896,325	6,973,742	7,061,530	Data
58	.West Virginia	1,852,994	1,853,033	1,854,176	1,854,982	1,856,313	1,853,595	1,850,326	Data
59	.Wisconsin	5,686,986	5,687,289	5,689,268	5,708,785	5,724,888	5,742,953	5,757,564	Data
60	.Wyoming	563,626	563,767	564,358	567,631	576,893	583,223	584,153	Data
61									Data
62	Puerto Rico	3,725,789	3,726,157	3,721,527	3,686,771	3,642,281	3,595,839	3,548,397	Data

Note: The estimates are based on the 2010 Census and reflect changes to the April 1, 2010 population due to the Count Qu
Suggested Citation:
Table 1. Annual Estimates of the Resident Population for the United States, Regions, States, and Puerto Rico: April 1, 2010
Source: U.S. Census Bureau, Population Division
Release Date: December 2014

Key for the Data Interpreter | NST01

FIGURE 2-21 *Excel review of interpreter changes*

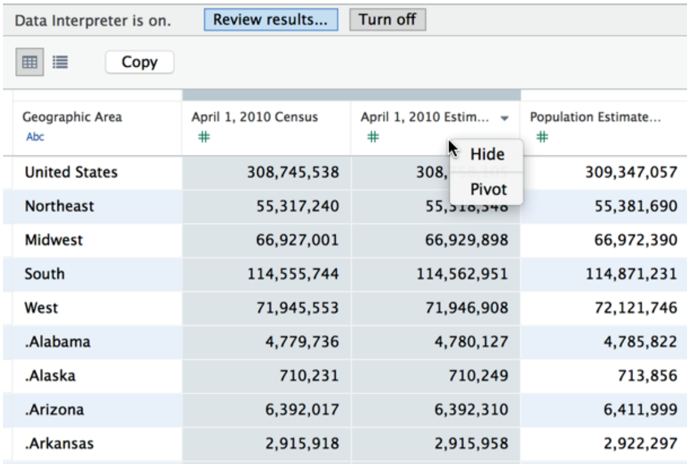
Next, you'll use the Preview area to hide unnecessary fields and rename columns that you want to use in your analysis.

Hiding and Renaming Columns

Hiding and renaming columns is easy to do in the Preview area. The spreadsheet includes the following fields:

- Geographic Area
- April 1, 2010 Census
- April 1, 2010 Estimate Base
- Population Estimate (as of July 1) 2010
- Population Estimate (as of July 1) 2011
- Population Estimate (as of July 1) 2012
- Population Estimate (as of July 1) 2013
- Population Estimate (as of July 1) 2014

Start by using the Data Interpreter to hide the Census and Estimate Base columns, as you see in Figure 2-22. Pointing at the drop-down arrow in a column exposes the Hide menu option. Clicking the drop-down arrow, which appears when you hover the pointer over a column name, exposes a menu that includes the Hide option.



Geographic Area (Abc)	April 1, 2010 Census (#)	April 1, 2010 Estim... (#)	Population Estimate... (#)
United States	308,745,538	308,[illegible]	309,347,057
Northeast	55,317,240	55,[illegible]	55,381,690
Midwest	66,927,001	66,929,898	66,972,390
South	114,555,744	114,562,951	114,871,231
West	71,945,553	71,946,908	72,121,746
.Alabama	4,779,736	4,780,127	4,785,822
.Alaska	710,231	710,249	713,856
.Arizona	6,392,017	6,392,310	6,411,999
.Arkansas	2,915,918	2,915,958	2,922,297

FIGURE 2-22 *Hiding unneeded columns*

Now that the unneeded columns are out of view, rename the population estimate columns as follows:

- Population Estimate (as of July 1) 2010: Rename as 2011
- Population Estimate (as of July 1) 2011: Rename as 2012
- Population Estimate (as of July 1) 2012: Rename as 2013
- Population Estimate (as of July 1) 2013: Rename as 2014
- Population Estimate (as of July 1) 2014: Rename as 2015

Figure 2-23 shows you how to rename the fields. The years are off by one because my data goes through the year 2015. I'm the person blending, and this is the use case that I require. Since the July 2014 data is the most recent, I'm matching it with the latest year in my data. That's one of the nice things about blending. You define the rules.

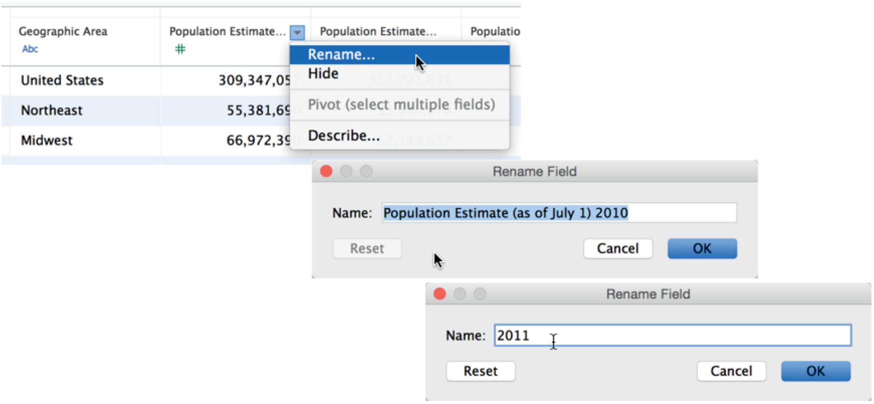
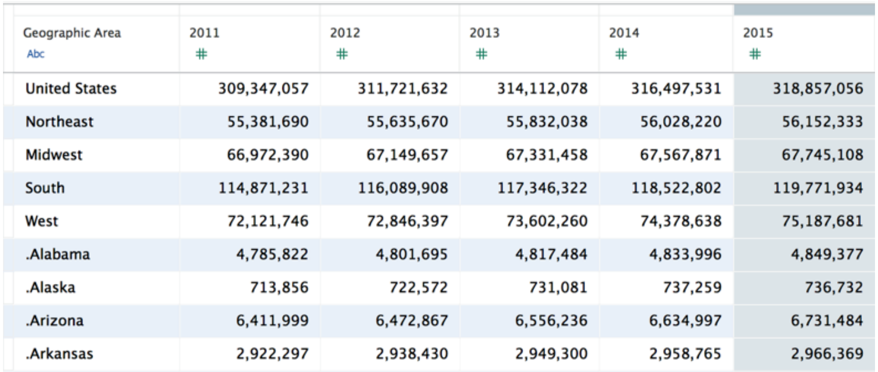


FIGURE 2-23 *Renaming columns*

Clicking the drop-down arrow in a column exposes the Rename Field dialog box, with the original field name as shown in the middle of Figure 2-23. Type in the year number desired and click OK. Repeat this step for each of the population estimate columns.

Split the Geographic Area Column

After hiding and renaming the columns, your Preview area should look like Figure 2-24.



Geographic Area	2011	2012	2013	2014	2015
United States	309,347,057	311,721,632	314,112,078	316,497,531	318,857,056
Northeast	55,381,690	55,635,670	55,832,038	56,028,220	56,152,333
Midwest	66,972,390	67,149,657	67,331,458	67,567,871	67,745,108
South	114,871,231	116,089,908	117,346,322	118,522,802	119,771,934
West	72,121,746	72,846,397	73,602,260	74,378,638	75,187,681
.Alabama	4,785,822	4,801,695	4,817,484	4,833,996	4,849,377
.Alaska	713,856	722,572	731,081	737,259	736,732
.Arizona	6,411,999	6,472,867	6,556,236	6,634,997	6,731,484
.Arkansas	2,922,297	2,938,430	2,949,300	2,958,765	2,966,369

FIGURE 2-24 *Renamed columns*

Focus on the row details below the Geographic Area column. There are rows containing geographic areas that are not states, and the state rows have a period in front of the state names. This is undesirable. Use the Split menu option to separate the period and the state names into separate columns. Figure 2-25 shows you how to do this.

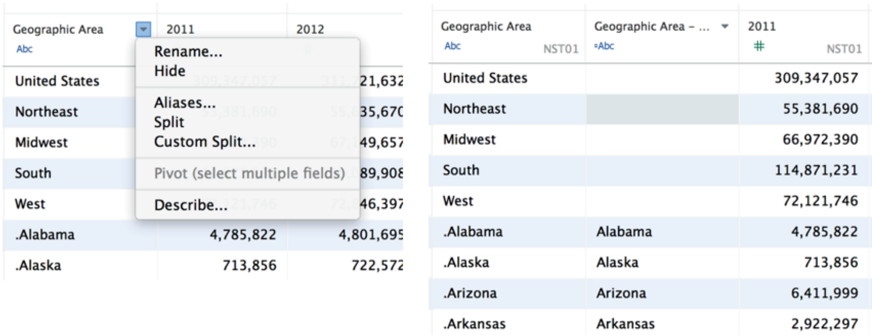


FIGURE 2-25 *Field split menu*

The left side of Figure 2-25 shows the menu accessed by clicking the drop-down arrow in the column. Choosing the Split option will result in the view you see on the right. States are now in their own column. The country and region information is split with blanks. Puerto Rico at the bottom has a blank column as well. This is not an issue because we are interested only in the lower 48 states, Alaska, Hawaii, and the District of Columbia.

Change the Datatype of the State Column

After hiding and renaming the columns, your Preview area should look like the top left of Figure 2-26. The Data Interpreter named the new column created by the split Geographic Area—Split 1.

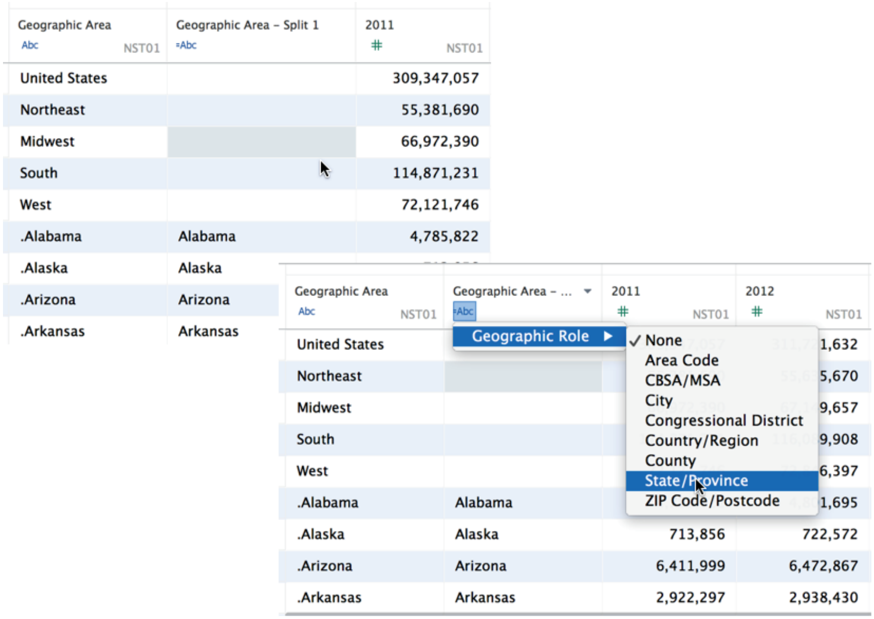


FIGURE 2-26 *Changing a datatype*

Rename that column and call it State. Then, change the datatype of the column giving it the geographic role of State/Province, as you see in the lower right of Figure 2-26. The column names and formatting are complete. The remaining step is to reshape the data into a format that is more like a database would store it.

Pivot the Year Columns

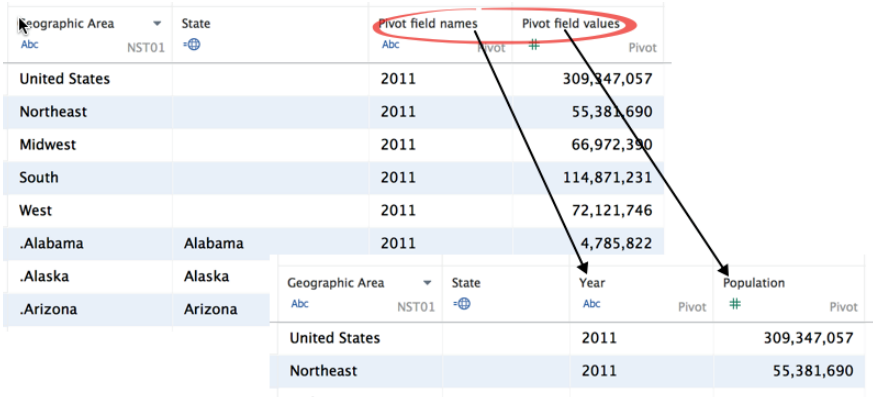
You could stop the reshaping at this point, but there could be a disadvantage. If you try to create a time series analysis of this data, it would still be possible, but it would require that you utilize the measure names and measure values to place different measure on the same axis. Why? Because Tableau will interpret each year column as a separate measure. The data in each of the renamed year columns is really the same measure (population estimate) for different years. In a properly structured database the table layout would consist of three columns: one for the geographic area, another for the year, and a third for the population count.

The Data Interpreter allows you to pivot the data quickly in the Preview area. First multi-select all of the population columns, as you see in Figure 2-27. If you do this correctly, the year columns should be highlighted. Choose the Pivot menu option from the 2010 column.

Geographic Area	State	2011	2012	2013	2014	2015
United States		309,347,057	311,721,632	314,112,078	316,497,531	318,857,056
Northeast		55,381,690	55,635,670	55,832,038	56,028,220	56,152,333
Midwest		66,972,390	67,149,657	67,331,458	67,567,871	67,745,108
South		114,871,231	116,089,908	117,346,322	118,522,802	119,771,934
West		72,121,746	72,846,397	73,602,260	74,378,638	75,187,681
.Alabama	Alabama	4,785,822	4,801,695	4,817,484	4,833,996	4,849,377
.Alaska	Alaska	713,856	722,572	731,081	737,259	736,732
.Arizona	Arizona	6,411,999	6,472,867	6,556,236	6,634,997	6,731,484
.Arkansas	Arkansas	2,922,297	2,938,430	2,949,300	2,958,765	2,966,369

FIGURE 2-27 *Pivot the year columns.*

When you click the Pivot menu option, the Data Interpreter will create a new three-column view of the data. It should look like the top left of Figure 2-28.

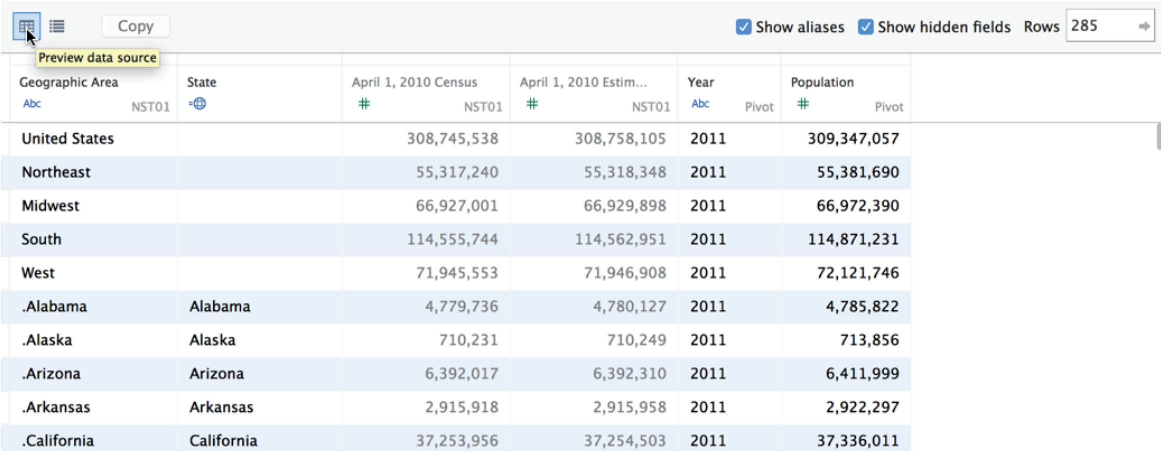


Geographic Area	State	Pivot field names	Pivot field values
United States		2011	309,347,057
Northeast		2011	55,381,690
Midwest		2011	66,972,390
South		2011	114,871,231
West		2011	72,121,746
.Alabama	Alabama	2011	4,785,822
.Alaska	Alaska		
.Arizona	Arizona		

Geographic Area	State	Year	Population
United States		2011	309,347,057
Northeast		2011	55,381,690

FIGURE 2-28 *Reshaped data*

Rename the second column as Year and the third column as Population, as you see in the lower right of Figure 2-28. At this point, your dataset is restructured, and the column headings are updated. Look at the reshaped data in the Preview area (see Figure 2-29).

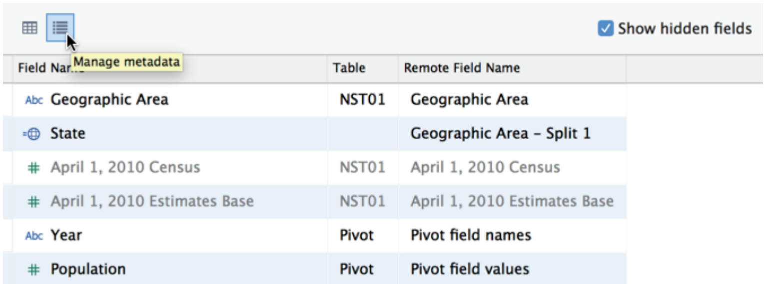


Geographic Area (Abc, NST01)	State	April 1, 2010 Census (#, NST01)	April 1, 2010 Estim... (#, NST01)	Year (Abc, Pivot)	Population (#, Pivot)
United States		308,745,538	308,758,105	2011	309,347,057
Northeast		55,317,240	55,318,348	2011	55,381,690
Midwest		66,927,001	66,929,898	2011	66,972,390
South		114,555,744	114,562,951	2011	114,871,231
West		71,945,553	71,946,908	2011	72,121,746
.Alabama	Alabama	4,779,736	4,780,127	2011	4,785,822
.Alaska	Alaska	710,231	710,249	2011	713,856
.Arizona	Arizona	6,392,017	6,392,310	2011	6,411,999
.Arkansas	Arkansas	2,915,918	2,915,958	2011	2,922,297
.California	California	37,253,956	37,254,503	2011	37,336,011

FIGURE 2-29 *Preview data source view*

Notice the check boxes for showing name aliases and hidden fields. Selecting the check box displays the two columns that were hidden. No row members (state names) were given name aliases in this example. If name aliases were used, checking/unchecking the Show Alias box allows you to toggle between the original field name or the name alias.

The two small icons at the top left of the Preview area allow you to select different views of the data. Figure 2-29 is currently showing the default preview data view. Figure 2-30 shows the Manage metadata view.



Field Name	Table	Remote Field Name
Abc Geographic Area	NST01	Geographic Area
State		Geographic Area – Split 1
# April 1, 2010 Census	NST01	April 1, 2010 Census
# April 1, 2010 Estimates Base	NST01	April 1, 2010 Estimates Base
Abc Year	Pivot	Pivot field names
# Population	Pivot	Pivot field values

FIGURE 2-30 *Manage metadata view*

The Manage metadata view is displaying hidden fields. It shows the source of the particular column (the original table or the pivot from the Data Interpreter) and the field names in the source dataset and the renamed fields.

All of this reformatting could be done in Excel, but it is faster and easier to prepare the data using the Data Interpreter. Test your work by building a map using the reshaped data.

Using the Reshaped Data in a Map

In order to confirm that all the data cleaning done has worked properly, build a map view of the data, as shown in Figure 2-31.

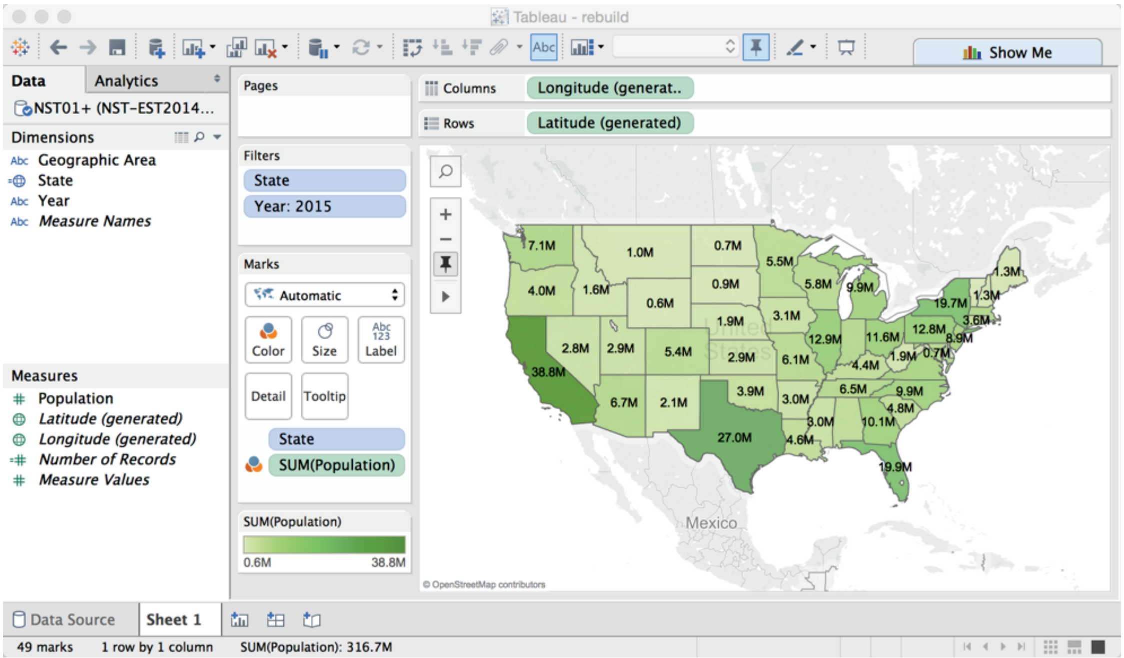


FIGURE 2-31 *Map showing population*

If you are not able to create the map, don't worry. In Chapters 3 and 5, you learn all of the details necessary to create maps and more.

Tableau's Data Interpreter helps you analyze spreadsheet data sources faster and more easily. You must have Tableau Desktop V9.0 or later to use this facility.

Joining Database Tables with Tableau

Seldom will your data include every bit of information you need in a single table. Even if you normally connect to Excel, it may be advantageous to use related data from more than one tab. As long as the data resides in a single spreadsheet or database and each table includes unique identifiers (key records) that tie the tables or tabs together, you can join these tables within Tableau.

Database joins can be complex. The basic principle is to bring together related information in your view. If you need to connect to an enterprise-class database, many more tables will be available. You should consult with a knowledgeable expert in your organization for assistance to ensure that your joins result in the correct definition.

Beginning with the release of V8.2, the join interface became more visual. Venn diagrams are now being used for defining the joins. Most Tableau users aren't database experts, and this more visual method for defining joins makes the task easier.

The Default Inner Join

You can define joins when you make your initial data connection or add them later. This example will use the Orders and Return tabs (tables) from the Superstore for TYD2 dataset. Figure 2-32 shows portions of the orders and returns tables. The Orders table includes all billing information. The Returns tab includes a smaller returned order table.

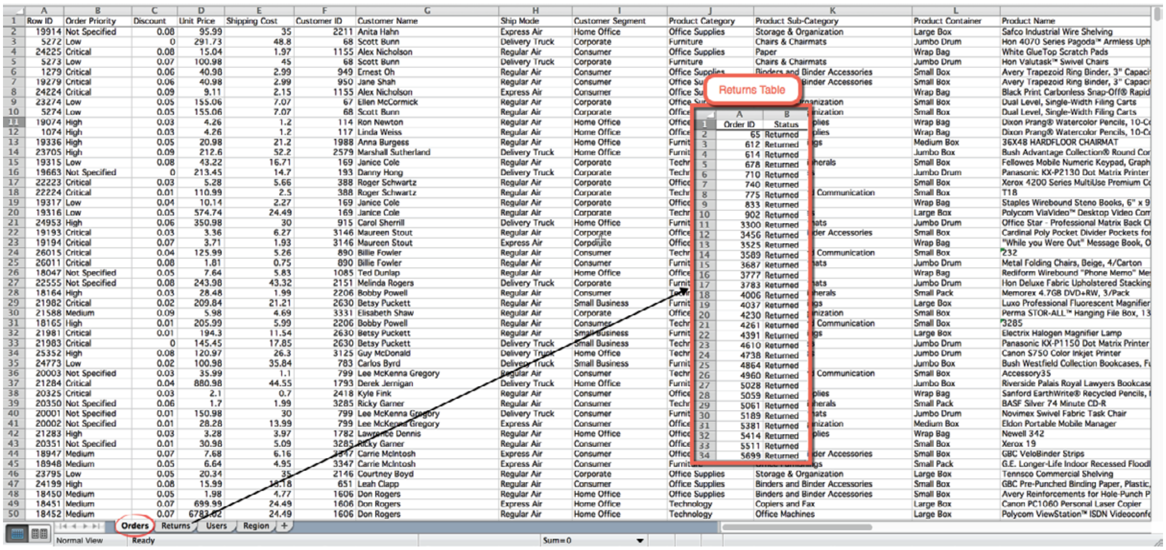


FIGURE 2-32 *Orders and Returns tables*

Start by connecting to the Orders sheet and then drag the Returns table into the join area, as you see in Figure 2-33.

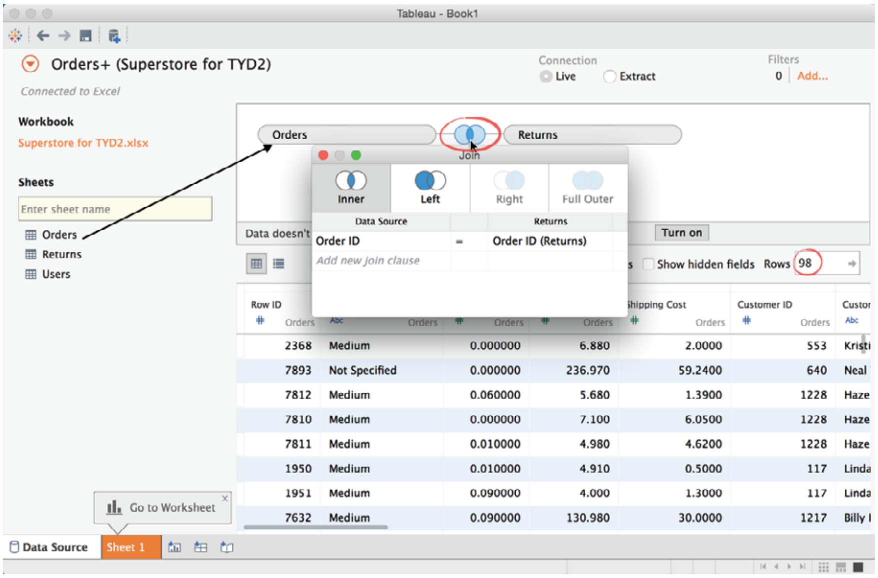


FIGURE 2-33 *Defining the join*

Clicking the Venn diagram in the join area causes the join window to be displayed. Join options that are dimmed are not available for the data source. Because Excel is the data source, the right and full outer join options are not available. These options are dependent on the data source. Most databases support these types of joins.

Notice that the inner join is selected and the resulting row count is 98. This means that there are 98 matching records between the Orders and Returns tables. Next let's look at what the left join option does.

The Left Join

Change to a left join by clicking the Venn diagram to expose the join options. A left join will cause all of the records from the Orders table (9,426), along with any matching records from the Returns table (98), to become available. Figure 2-34 displays the result in the Preview area.

The fields that don't have matching records in the Returns table will be displayed as NULL values in the combined table. If you are following along, try selecting the left join type and notice that the number of rows displayed in the Preview area will change.

You can add as many tables as necessary in the join area. Try joining the Users table to the view on your own and adjust the join type.

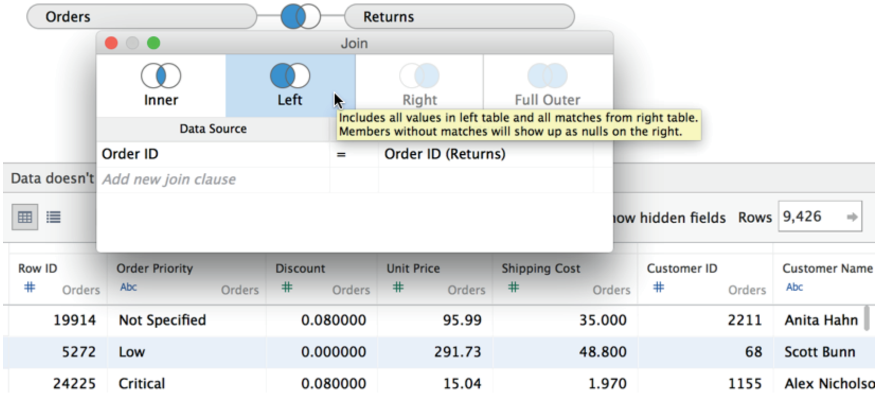


FIGURE 2-34 *The left join*

Customizing Tableau's Join Script

Prior to V9.0, it was possible to edit the connection script when you were working with Excel, Access, or text files. This facility is no longer available. This is the first time I'm aware of that Tableau has deprecated a feature that existed in a prior release. The Data Interpreter offers significant improvements for the majority of users. If the cost of the gained convenience of the Data Interpreter is the loss of full outer joins for Excel, Access, and text files, I think the value gained is worth the price. My expectation is that Tableau will find a way to bring this facility back in a future release.

In the meantime, if you are have more than one million records that you need to analyze, you should consider installing a real database. There are many low-cost commercial and open source databases with which Tableau provides all of the join options.

In the next section, you learn how to bring together data from different data sources into a single visualization using data blending.

Blending Different Data Sources

Wouldn't it be wonderful if all the data you needed to create your analysis always resided in a single database? Often, this isn't the situation. If you need to use data from more than one data source, Tableau provides a solution that does not require building a middle-layer data repository. If the disparate data sources have at least one common field, Tableau can combine them using data blending.

When to Use Blending versus Joins

If your data does reside in a single data source, it is always more desirable to use a join versus a data blend. In the last section, you saw that Tableau provides plenty of flexibility for creating joins to your data source. Joins are normally the best option because they are more robust, persist everywhere in your workbook, and offer more flexibility for defining the join clauses than blending. However, if your data isn't in one place, blending provides a viable way to quickly create a left join–like connection between your primary and secondary data sources.

Blends persist only on the worksheet page where they were created. This makes blending ideal for ad hoc analysis. Whatever data source is used first to build a view becomes the primary data source for that worksheet. Data sources used later become the secondary data source. This means that any data source can be the primary data source in a view.

In the next section, you learn how to blend population data from the U.S. Census Bureau with sales data from the Superstore dataset.

How to Create a Data Blend

Creating data blends requires some planning. If you are going to bring data that doesn't reside in your primary data source, you have to think about the fields you may need in order to achieve the desired result. Blending is easier if you have completely consistent field names in the data sources being used for the blend. The field names and the field members need to exactly match. Blending based on Cities will not match Saint Cloud from one source with St. Cloud in the other. If you can't rename fields to make them match, you can still blend the files but it requires slightly more effort.

If the blending fields have identical names, Tableau recognizes the fields in common and will automatically create the link for the blend. If the fields do not have identical names (City and Cities), you have to define the blend manually using the Edit Relationships menu.

Automatically Defined Relationship

The Superstore data source contains geographic sales data. What if you wanted to know what the per-capita sales for each state were for the year 2014? The Superstore dataset doesn't include population data but the United States Census Bureau website does.

The data in Figure 2-35 was downloaded from the web and adapted for this example. The spreadsheets used in this example are included in the Chapter 2 download on the companion website. This example uses the Superstore for TYD2 and the Population 2014 spreadsheet files.

1	State	Population
2	Alabama	4,849,377
3	Alaska	736,732
4	Arizona	6,731,484
5	Arkansas	2,966,369
6	California	38,802,500
7	Colorado	5,355,866
8	Connecticut	3,596,677
9	Delaware	935,614
10	District of Columbia	658,893
11	Florida	19,893,297
12	Georgia	10,097,343
13	Hawaii	1,419,561
14	Idaho	1,634,464
15	Illinois	12,880,580
16	Indiana	6,596,855
17	Iowa	3,107,126
18	Kansas	2,904,021
19	Kentucky	4,413,457
20	Louisiana	4,649,676
21	Maine	1,330,089
22	Maryland	5,976,407
23	Massachusetts	6,745,408
24	Michigan	9,909,877
25	Minnesota	5,457,173
26	Mississippi	2,994,079
27	Missouri	6,063,589
28	Montana	1,023,579
29	Nebraska	1,881,503
30	Nevada	2,839,099
31	New Hampshire	1,326,813
32	New Jersey	8,938,175
33	New Mexico	2,085,572
34	New York	19,746,227
35	North Carolina	9,943,964
36	North Dakota	739,482
37	Ohio	11,594,163
38	Oklahoma	3,878,051
39	Oregon	3,970,239
40	Pennsylvania	12,787,209
41	Rhode Island	1,055,173
42	South Carolina	4,832,482
43	South Dakota	853,175
44	Tennessee	6,549,352
45	Texas	26,956,958
46	Utah	2,942,902
47	Vermont	626,562
48	Virginia	8,326,289
49	Washington	7,061,530
50	West Virginia	1,850,326
51	Wisconsin	5,757,564
52	Wyoming	584,153

FIGURE 2-35 *Population data 2014*

Just two columns of data are included in the table. It is important to note the field descriptions for the states. For automatic blending to work, the field name for the blend must be the same in the Superstore for TYD2 dataset and the Population spreadsheets. If the fields are not the same, you will need to edit the name in the spreadsheet or rename the fields in Tableau so that they match.

To start an automatic blend, you must define the primary data source by placing one of its fields on the worksheet. The first data source used in the view becomes the primary data source for that particular view. Figure 2-36 shows a bar chart for sales by state.

Superstore for TYD2 is now the primary data source on this worksheet; rename the sheet Blending Example. The Population 2014 spreadsheet that will be blended into this view also includes state names, but the Superstore for TYD2 field name for state is State/Province. Before attempting the blend, rename the field to State. Now both datasets use the same name for the field.

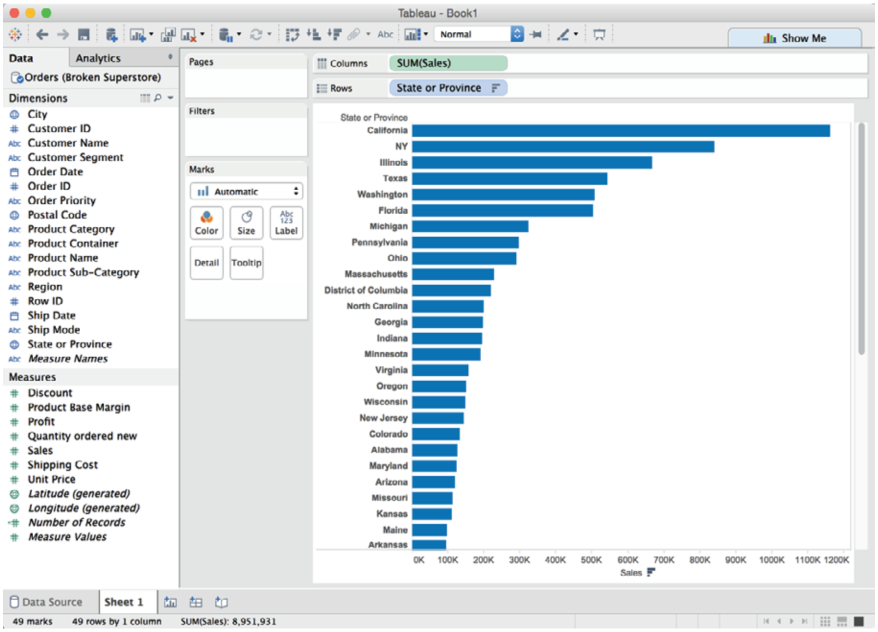


FIGURE 2-36 *Sales by state*

Next drag the Population measure from the Population 2014 data and place it on the Columns shelf. Once that is done, the data from the population spreadsheet can be used in the workbook. It becomes a secondary data source in the view. Dragging the Population field from the secondary data source onto the Columns shelf locks-in the blend. The orange check mark on the Population field and in the data window confirms the blend. Figure 2-37 now shows the sales data from Superstore next to the population for each state.

The orange border on the left side of the dimension and measures areas provides additional confirmation that the fields in the Population 2014 data are the secondary data source in this worksheet view.

A warning—when you perform data blending, you must ensure that all of the records you expected to blend actually came into the view. In Figure 2-37 this is not the case. New York population didn't come over in the blend because the state names used in the primary and secondary data sources do not match. You can correct this by right-clicking the abbreviated state name (NY), selecting the Edit Alias menu option, and renaming it as New York. This will reconcile the name and eliminate the null value. Figure 2-38 shows the blended view with labels added that include additional information in the data in the mark labels.

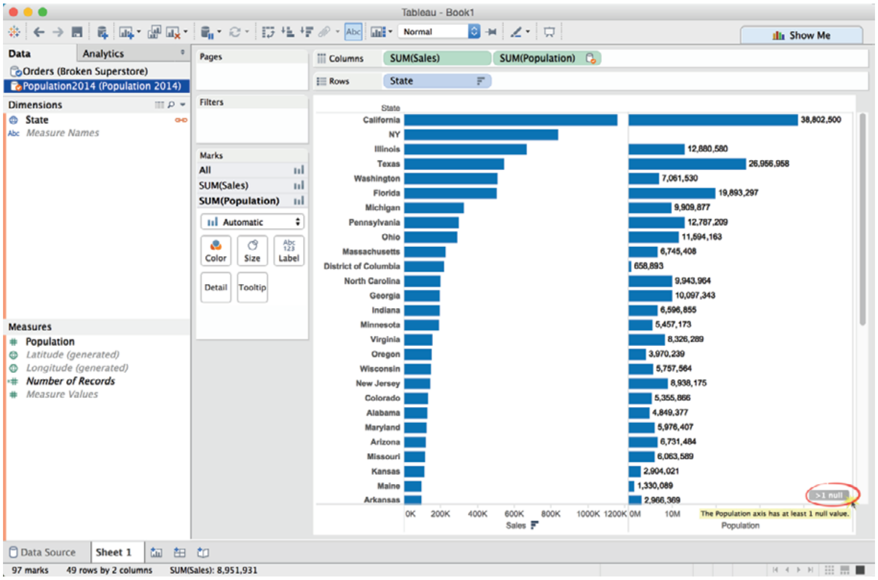


FIGURE 2-37 *Blended sales and population data*

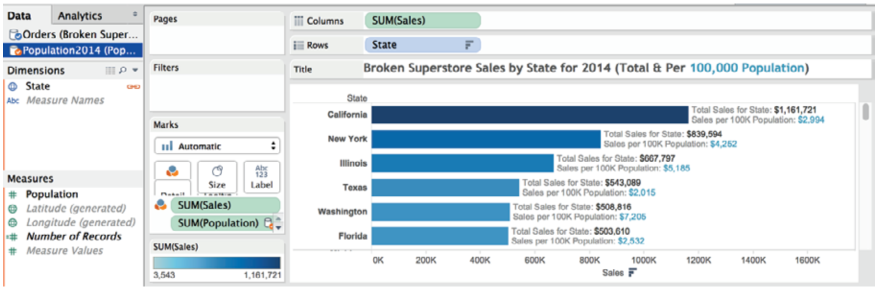


FIGURE 2-38 *Bar chart using blended data*

To save space, Figure 2-38 shows only the top six states by per-capita sales. The labels to the right of each bar show the total sales and the sales per hundred thousand people. The color of each bar encodes the total sales of each state. See if you can figure out how to create the formula to express that value. Look at the completed sample file included in the companion website materials. You will learn how to create calculations in Chapter 4.

Keep in mind that even if the field names denoting state are not the same, you can still blend the data sources using the Data/Edit Relationship menu option. The next example shows you how.

Manual Blending

What if your needs are more complicated? A scenario that requires the use of two or more dimensions is the comparison of budget data from a spreadsheet with actual data from your billing system. The field names in the data sources may be different even if they refer to the same thing. In this example, the Superstore for TYD2 file will be blended with the Budget Sales 2015 file. The budget file is included in the Chapter 2 companion website download. It contains monthly sales budgets by Product Category and Product Sub-Category. These are the steps used to build the bullet graph in Figure 2-39:

1. Connect to Superstore for TYD2.
2. Build a bar chart of actual sales for each product category by month filtered for the year 2015.
3. Connect to the Budget Sales 2015 file.
4. Pivot the 12 date fields and rename Pivot field names as Month and Pivot field values as Budget Sales.
5. Use the Data/Edit Relationship menu to define the blend on product category and month.
6. Drag the Budget Sales 2015 field on to the Detail button on the Marks card and then create a reference line using that field.

Encoding the bars with color was accomplished by adding the calculated value seen in the caption at the bottom of Figure 2-39 to the Color button on the Marks card.

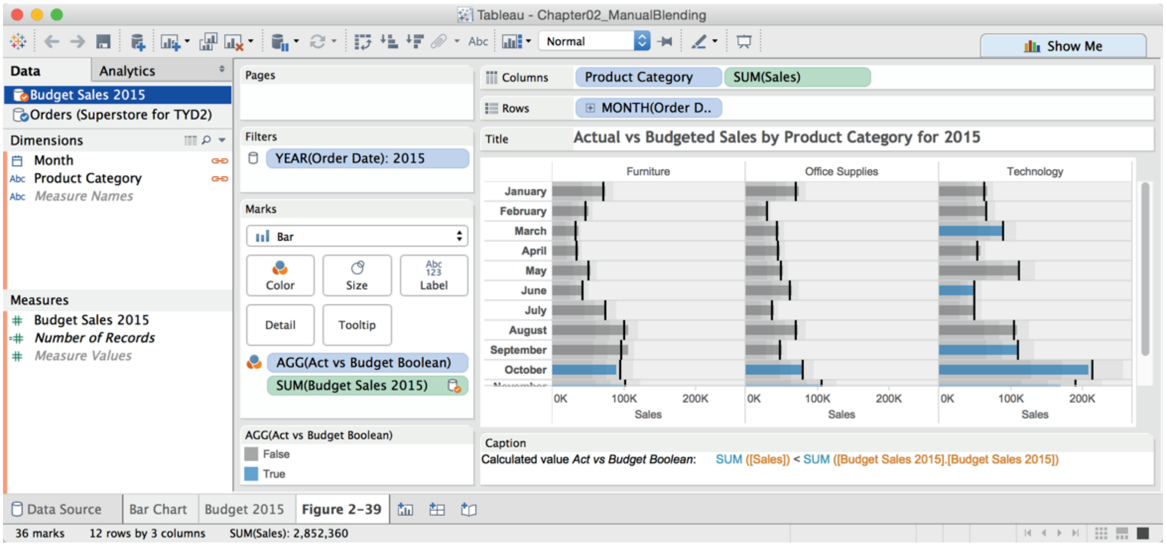


FIGURE 2-39 *Bullet graph with blended data*

Actual sales data from the primary data source, Orders (Superstore for TYD2), is displayed using bars. Budgeted data from the secondary data source (Budget Sales 2015) is plotted with vertical black reference lines for each cell. In Figure 2-39 the blend on both fields is denoted by the two orange links in the Dimensions shelf. This multi-field blend is created via the Data/Edit Relationships menu. Figure 2-40 shows the menu and the related dialog boxes.

From the Data menu option, select Edit Relationships, as shown in Figure 2-40. This exposes the relationships window. Product category will appear automatically because that field name exists in both data sources, and Tableau should automatically recognize the blend.

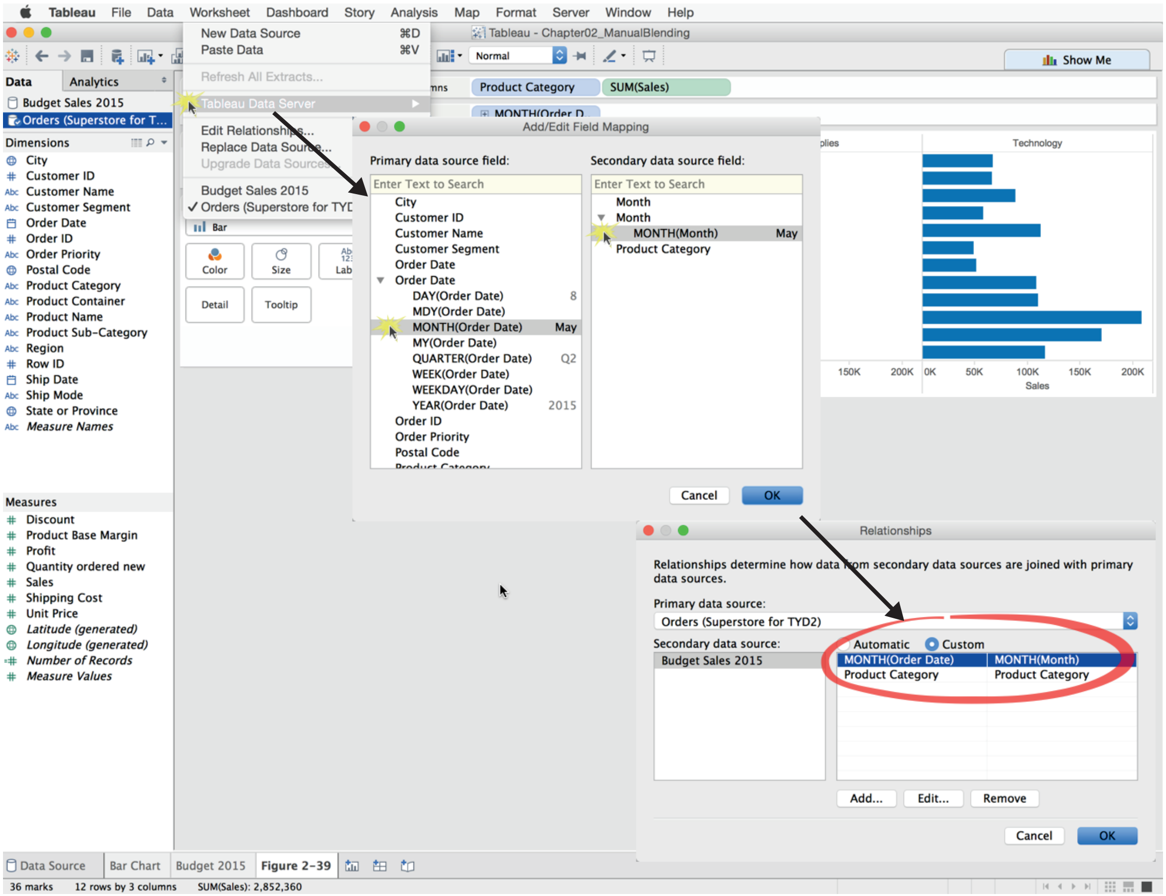


FIGURE 2-40 *The relationship menu*

Clicking the Add button exposes the Add/Edit field mapping window, where the specific date aggregation can be selected from each data source. After picking the appropriate date field, select OK to lock in the date. Click OK in the Relationship dialog box to finalize the links for both fields.

Review the pill placements in Figure 2-39 to see where fields were placed to create the chart. Be aware that you can display the fields for each data source in the data window by clicking the desired data source in the data window. The blue highlight on the Budget Sales 2015 data source you see in Figure 2-39 indicates that its fields are being displayed within the Dimensions and Measures shelves.

The black reference lines in each cell were created using the blended Budget Sales 2015 field from the secondary data source (indicated by the orange check mark in the data window and by the orange border on the left side of the Dimensions and Measures shelves.

The calculated field used to create the bar colors is displayed in the caption below the graph and is stored in the primary data source. Gray bars denote items above plan. The gray color gradient behind the sale bars comes from a reference distribution that uses color hue to show sales at 60 percent, 80 percent, 100 percent, and 120 percent of planned sales. You'll learn how to create bullet graphs with color-encoding like this in Chapter 7.

Next let's consider factors that can affect the load speed of your data connections.

Hardware Factors That Affect Performance

Four areas affect Tableau's speed:

- The server hardware, which hosts the database
- The database, which hosts the data
- The network, over which the data is sent
- Your own computer's hardware, which has Tableau Desktop

Like any chain, the weakest link dictates overall performance.

Your Personal Computer

Tableau doesn't require high-end equipment to run. But, you will find more internal memory, a new microprocessor, and a faster hard disk will all contribute to better performance—especially if you are accessing very large datasets. The video card and monitor resolution can contribute to the quality of how Tableau presents the visuals.

Random Access Memory (RAM)

Tableau 9 runs as a 32-bit or 64-bit application on Windows and on the Mac, which means the maximum memory that it can access is 4GB (32-bit) or

8GB (64-bit). Expanding RAM to the maximum amount the operating system supports provides the best performance.

Processor

A faster processor will help Tableau's performance, but you only really get a chance to change the processor when you buy your computer. Buy the best you can justify, and you should be fine.

Disk Access Times

Tableau is not normally a disk-intensive program, but having a faster hard disk drive or a solid-state drive (SSD) will help Tableau load faster. If you work with very wide and deep datasets that exceed your machine's internal memory capacity, it will slow down and will result in page swapping to the hard disk drive. In this circumstance a fast hard drive will help performance.

Screen Size

The resolution of your screen affects the level of detail that you're able to discern. A visualization on a large, high-resolution screen may provide better insight than a rendering on a lower-resolution monitor into. If you have a very good monitor, you must consider that other people may be consuming your analysis with equipment that isn't as good. If they have a lower-resolution video card, your visualization will not be the same on their computer. Chapter 8 includes tips and tricks for designing dashboards that allow you to manage your dashboard designs so that they capitalize on high-resolution hardware without penalizing information consumers with older, less-capable equipment.

Finally, consider the amount of work you are asking Tableau to do. While it is possible to plot millions of marks in a chart, ask yourself if all those marks add to understanding the data. If you run into performance issues, review the level of detail you're plotting. Using fewer marks in the view may improve the content's value and improve the rendering speed.

Your Server Hardware

The key consideration with regard to the specification of your server hardware is the volume and activity level you anticipate. Is your database currently deployed on the three-year-old production server with thousands of concurrent users? Does your server have other demanding applications running that may cause resource contention?

Tableau can run in the cloud and on servers that have other applications running, but as your deployment expands, it is best to dedicate a server to Tableau. For massive deployments, Tableau core licenses can be divided across multiple servers.

Specifying server hardware is not a one-size-fits-all proposition. Tableau provides guidelines on their website, but each situation is unique and requires some detailed planning. In general, oversizing the hardware a little isn't a bad idea. Tableau normally becomes very popular when it is deployed, so consider the potential for increasing demand and get professional assistance if you are unsure about the server hardware you should purchase. Chapter 11 covers this topic in more detail.

The Network

Like any other form of infrastructure (transport, power, water), data networking is a mundane but vital component for the efficient performance of any system. Networking is therefore the responsibility of specialists within your organization, and they can help you identify if there are choke points in your network that slow the performance of Tableau. For all but the very largest organizations, network capacity is seldom a bottleneck.

The Database

If you are using live connections to your data—as opposed to data extracts—the performance of your database is one of the most significant determining factors of speed.

As more people in your organization use Tableau, it is important to monitor resource load on the server, the network, and the database. Tuning your database is the responsibility of the database administrator. It is normally helpful if someone from the IT team is directly involved in the early phases of enterprise roll-outs, especially if it is expected that Tableau may create larger or different demands on the database.

If the database administrator understands the type, amount, and timing of the query loads that Tableau may generate, proper planning can ensure that system performance will not be degraded due to inadequately indexed database tables or an overloaded database server.

NOTES

1. Gordon Mackenzie, *Orbiting the Giant Hairball: A Corporate Fool's Guide to Surviving with Grace* (New York: Viking, 1998), p. 88.
2. United States Census Bureau, "Annual Estimates of the Resident Population for the United States, Regions, States, and Puerto Rico: April 1, 2010 to July 1, 2014 (NST-EST2014-01," downloaded April 4, 2015, `http://www.census.gov/popest/data/state/totals/2014/index.html`

CHAPTER 3

Building Your First Visualization

"If we have made this our task, then there is no more rational procedure than the method of trial and error—of conjecture and refutation: of boldly proposing theories; of trying our best to show that these are erroneous; and of accepting them tentatively if our critical efforts are unsuccessful."

Karl Popper[1]

Now that you've learned how to connect Tableau to a variety of data sources, you can start building visualizations. In this chapter, you learn about all of the chart types provided by the Show Me button. You will discover how to add trend lines and reference lines and how to control the way your data is sorted and filtered. You'll see how creating ad hoc groups, sets, and hierarchies can produce information not available in the data source. The chapter covers Tableau's discrete and continuous data hierarchies, along with how you can alter Tableau's default date hierarchies by creating your own custom dates.

FAST AND EASY ANALYSIS VIA SHOW ME

Tableau's mission statement is to help you see and understand your data by enabling self-service visual analytics. The software is designed to facilitate analysis for non-technical information consumers. This is the concept behind Tableau's Show Me button. Consider Show Me to be your expert helper. Show Me tells you what chart to use and why. It will also help you create complicated visualizations faster and with less effort. For example, advanced map visualizations are best started via Show Me because Tableau will properly place multiple Dimensions and Measures pills on the appropriate shelves with a single click. If you know what you want to see, Show Me will get you to your desired destination quickly.

NEW FEATURES

Since the first edition of this book was published, two additional chart types have been added to Show Me: the box plot and the dual axis combination chart. While it has been possible to create these chart types in Tableau for quite some time, their addition to Show Me makes them accessible to novice users. A new Analytics tab has also been added to the Dimensions shelf, providing a more intuitive way for new users to add trend lines, reference lines, or forecasts to visualizations.

HOW SHOW ME WORKS

Show Me looks at the combination of measures and dimensions you've selected and interprets what chart types display the data most effectively. Most of the examples in this chapter use a version of the Superstore sales dataset that will not be the same as the version that ships with Tableau. You can download a copy of the workbook used to create all of the figures in this chapter at the book's companion website (see Appendix F for instructions). Or, you can use the Superstore dataset that shipped with Tableau desktop. Just realize that your results may look different than the figures in the text.

Regardless of which version you decide to use, selecting the Order Date field from the Dimensions shelf and Sales from the Measures shelf and then clicking Show Me will expose the options available for that combination that you see in Figure 3-1.

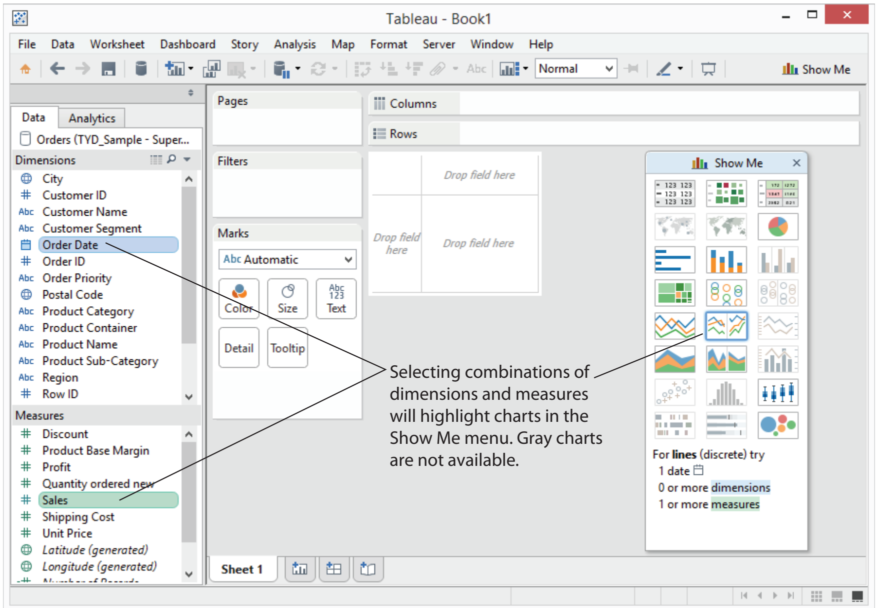


FIGURE 3-1 *Show Me chart options*

Tableau recommends a line (discrete) time series chart in Show Me—denoted with a blue outline. At the bottom of the Show Me area, you also see additional details regarding requirements needed for building any available chart. The time series chart requires one date, one measure, and zero or more dimensions. Selecting the highlighted chart and rotating the axis causes the time series chart in Figure 3-2 to be displayed.

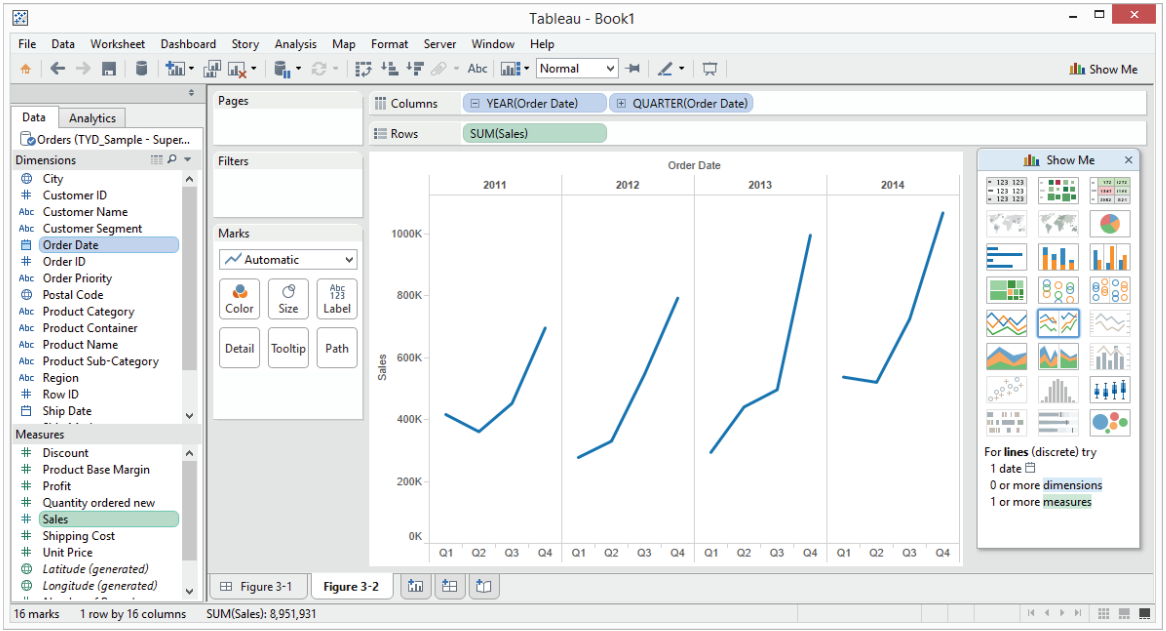


FIGURE 3-2 *Discrete date time series chart*

Pointing at other chart options in the Show Me menu changes the text at the bottom of the menu. This text provides guidance on the combination of data elements required for the chart being considered. Clicking on any of the highlighted Show Me icons alters the visualization in the worksheet.

Chart Types Provided by the Show Me Button

Show Me now contains 24 chart types. The most recent additions are the dual axis combination time series chart and the box-and-whisker plot. In Figure 3-3 you can see a blue outline around the symbol map.

At the bottom of the Show Me menu, descriptions of the data elements required for the chart are provided. As you hover over other chart types, these descriptions change. Next you'll learn about all of the chart types facilitated by Show Me in more detail.

FIGURE 3-3 *The Show Me menu*

In the following sections, you will see examples of every chart type that Show Me supports in the order that they appear in the Show Me menu.

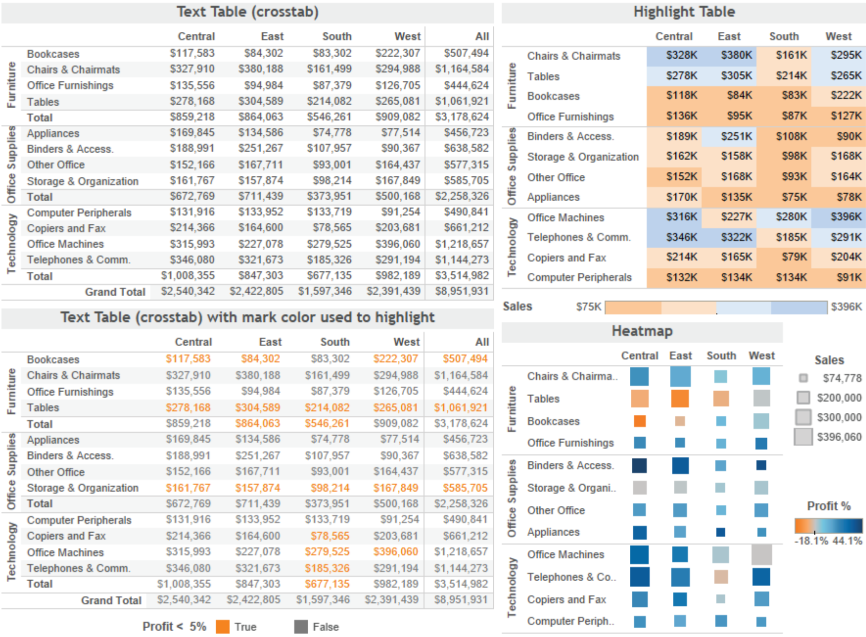
Text Tables, Heat Map, and Highlight Table

Text Tables look like grids of numbers in a spreadsheet. These are also referred to as text tables and are useful for looking up values. Figure 3-4 shows a standard text table in the upper left.

The text table in the lower left of Figure 3-4 has been enhanced by adding a Boolean calculation to highlight items with less than 5 percent profit ratio. Individual marks with less than 5 percent profit ratio are orange. You learn how to create calculated values in Chapter 4.

In the upper right of Figure 3-4 is a highlight table. These are similar to text tables but use a color background to help you see the range of values included. The larger values have blue backgrounds with the lower values being displayed with an orange background. The color legend below the highlight table indicates that the full range of values includes a low value of $75,000 and a high value of $396,000. In this way, highlight tables help you see the outliers more clearly than a text table.

The chart in the lower right of Figure 3-4 is a heat map. Heat maps can use size and color to display multiple measures. In the example, sales are represented by the size of the mark and profit ratio is conveyed using color. While the actual numbers aren't displayed as they are in text tables or highlight tables, heat maps can display a lot of data in a small amount of space.



Text Table (crosstab)

		Central	East	South	West	All
Furniture	Bookcases	$117,583	$84,302	$83,302	$222,307	$507,494
	Chairs & Chairmats	$327,910	$380,188	$161,499	$294,988	$1,164,584
	Office Furnishings	$135,556	$94,984	$87,379	$126,705	$444,624
	Tables	$278,168	$304,589	$214,082	$265,081	$1,061,921
	Total	$859,218	$864,063	$546,261	$909,082	$3,178,624
Office Supplies	Appliances	$169,845	$134,586	$74,778	$77,514	$456,723
	Binders & Access.	$188,991	$251,267	$107,957	$90,367	$638,582
	Other Office	$152,166	$167,711	$93,001	$164,437	$577,315
	Storage & Organization	$161,767	$157,874	$98,214	$167,849	$585,705
	Total	$672,769	$711,439	$373,951	$500,168	$2,258,326
Technology	Computer Peripherals	$131,916	$133,952	$133,719	$91,254	$490,841
	Copiers and Fax	$214,366	$164,600	$78,565	$203,681	$661,212
	Office Machines	$315,993	$227,078	$279,525	$396,060	$1,218,657
	Telephones & Comm.	$346,080	$321,673	$185,326	$291,194	$1,144,273
	Total	$1,008,355	$847,303	$677,135	$982,189	$3,514,982
	Grand Total	$2,540,342	$2,422,805	$1,597,346	$2,391,439	$8,951,931

Text Table (crosstab) with mark color used to highlight

		Central	East	South	West	All
Furniture	Bookcases	$117,583	$84,302	$83,302	$222,307	$507,494
	Chairs & Chairmats	$327,910	$380,188	$161,499	$294,988	$1,164,584
	Office Furnishings	$135,556	$94,984	$87,379	$126,705	$444,624
	Tables	$278,168	$304,589	$214,082	$265,081	$1,061,921
	Total	$859,218	$864,063	$546,261	$909,082	$3,178,624
Office Supplies	Appliances	$169,845	$134,586	$74,778	$77,514	$456,723
	Binders & Access.	$188,991	$251,267	$107,957	$90,367	$638,582
	Other Office	$152,166	$167,711	$93,001	$164,437	$577,315
	Storage & Organization	$161,767	$157,874	$98,214	$167,849	$585,705
	Total	$672,769	$711,439	$373,951	$500,168	$2,258,326
Technology	Computer Peripherals	$131,916	$133,952	$133,719	$91,254	$490,841
	Copiers and Fax	$214,366	$164,600	$78,565	$203,681	$661,212
	Office Machines	$315,993	$227,078	$279,525	$396,060	$1,218,657
	Telephones & Comm.	$346,080	$321,673	$185,326	$291,194	$1,144,273
	Total	$1,008,355	$847,303	$677,135	$982,189	$3,514,982
	Grand Total	$2,540,342	$2,422,805	$1,597,346	$2,391,439	$8,951,931

Profit < 5% True False

Highlight Table

		Central	East	South	West
Furniture	Chairs & Chairmats	$328K	$380K	$161K	$295K
	Tables	$278K	$305K	$214K	$265K
	Bookcases	$118K	$84K	$83K	$222K
	Office Furnishings	$136K	$95K	$87K	$127K
Office Supplies	Binders & Access.	$189K	$251K	$108K	$90K
	Storage & Organization	$162K	$158K	$98K	$168K
	Other Office	$152K	$168K	$93K	$164K
	Appliances	$170K	$135K	$75K	$78K
Technology	Office Machines	$316K	$227K	$280K	$396K
	Telephones & Comm.	$346K	$322K	$185K	$291K
	Copiers and Fax	$214K	$165K	$79K	$204K
	Computer Peripherals	$132K	$134K	$134K	$91K

Sales $75K $396K

FIGURE 3-4 *Text tables, highlight table, and heat map*

Symbol Map, Fill Map, and Pie Chart

Selecting a field with a small globe icon makes maps available in Show Me. Figure 3-5 shows examples of the two kinds of maps Show Me provides.

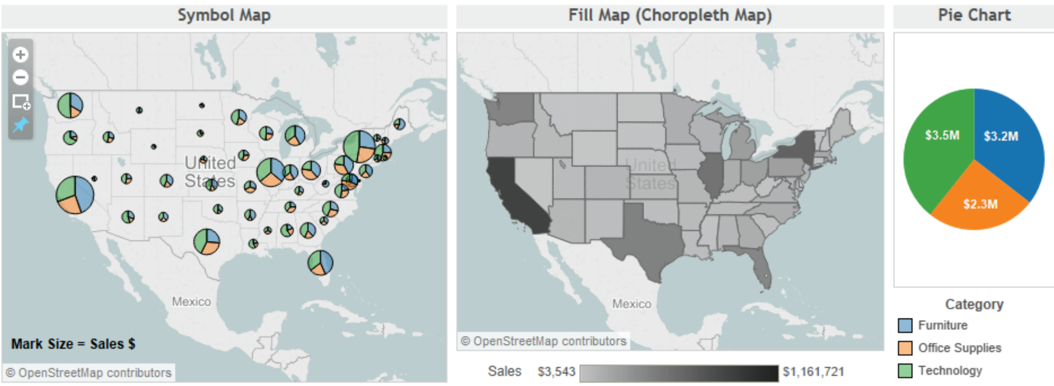


FIGURE 3-5 *Symbol map, fill map, pie chart*

Symbol maps are most effective for displaying very granular details or if you need to show multiple members of a small dimension set. The symbol map on the left of Figure 3-5 uses small pie charts to display product category. In symbol maps, it is a good idea to make the marks more transparent and add dark borders because marks tend to cluster around highly populated areas. Using the Color button on the Marks card to do this makes the individual marks easier to see.

Filled maps (also known as choropleth maps—see the glossary for a detailed definition) display a single measure using color within a geographic shape. If you restrict filled maps to smaller geographic areas (state, province), they effectively display more granular areas such as county or postal code.

The pie chart on the right side in Figure 3-5 shows the relative sales values for the three product categories included in the dataset. Pie charts are commonly used to present relative values, but they are not the best choice if precise visual comparisons are desired. A better way to present this data, if more precise comparisons need to be made, is the bar chart covered in the next section.

Bar Chart, Stacked Bar, Side-by-Side Bars

Bar charts facilitate comparisons between different values. Figure 3-6 includes three different examples.

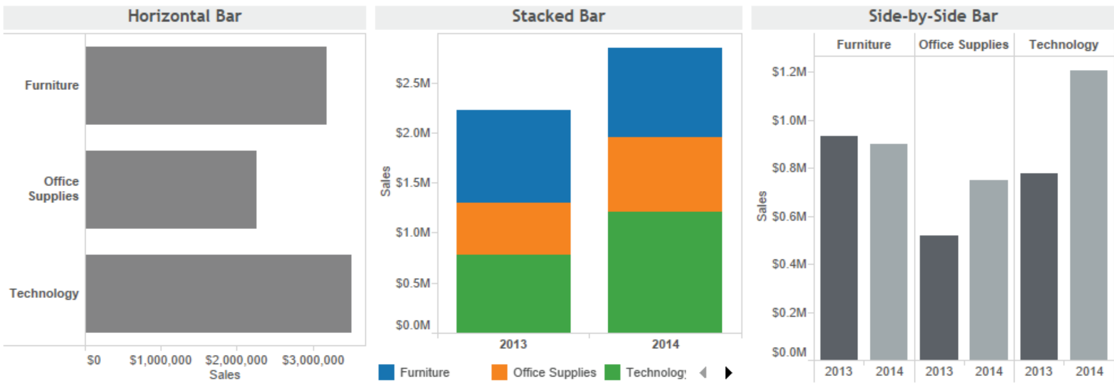


FIGURE 3-6 *Bar chart, stacked bar chart, and side-by-side bar chart*

Bar charts are the most effective way to compare values across dimensions, their linear nature making precise comparisons easy. The horizontal bar chart on the left of Figure 3-6 makes precise comparison of the three values displayed easy. You can see that Technology has the most sales while Office Supplies is the lowest. And, even though Furniture sales are nearly the same as Technology, you can see that they are not as big.

Stacked bar charts in the middle of Figure 3-6 should not be used when there are many different dimensions because they can be overwhelming if too many colors are plotted in each bar. On the right of Figure 3-6 is the side-by-side bar chart. This chart type provides a more precise way to compare measures. You can see that it is easier to see the relative values in the side-by-side bar chart than in the stacked bar chart.

Treemap, Circle View, Side-by-Side Circles

Comparing more granular combinations of dimensions and measures in a relatively small space can be done effectively with each of these charts. Heat maps use color and size to compare up to two measures. On the left side of Figure 3-7 is a treemap.

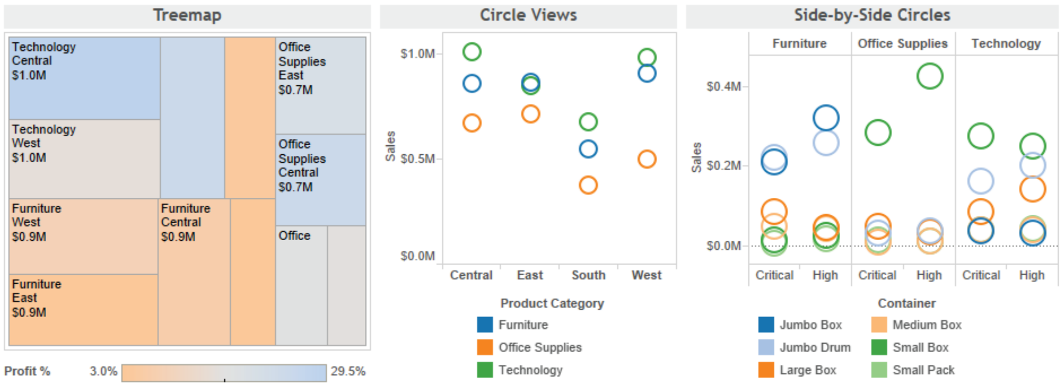


FIGURE 3-7 *Treemap, circle views, side-by-side circles*

Treemaps effectively display larger dimension sets in a relatively small space using color and size. The example shows product category, sales value using size, and profit % using color. Labels indicate the product category. When the size of the block is too small, no labels are displayed. The measures and dimensions that make up each block can be displayed in tooltips that pop out when the information consumer hovers over the mark. This is a default behavior in Tableau for any chart type.

Circle views can display data in a small space. The example in the middle of Figure 3-7 shows sales by product category and region. Side-by-side circles can add an additional dimension for comparison. While these examples show a relatively small number of marks (to save space), it is possible to use these chart types to display very granular values.

In the next two sections, you'll see how to use Show Me to display time series data.

Line Charts for Time Series Analysis

Line charts are the most effective way to display time series data. One variable to consider when presenting time series is the treatment of time as a discrete (bucketed) entity or as a continuous (unbroken) series progression. Discrete line charts place breaks between time units (year, quarter, and month).

Most people are familiar with time series charts that are presented in unbroken lines. The Lines Continuous chart in Figure 3-8 presents a single measure (sales) by year and month using continuous time.

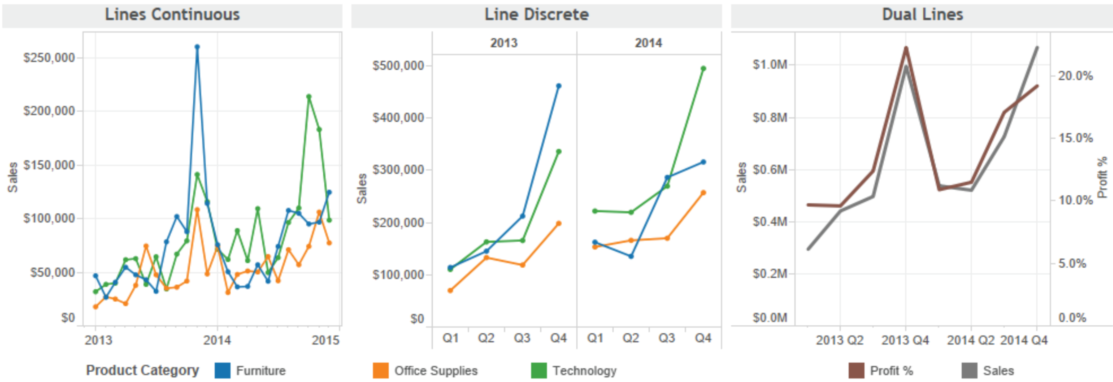


FIGURE 3-8 *Time series presented using continuous time*

The chart includes 24 months of history beginning January 2013 and ending December 2014. The Line Discrete chart in the middle of Figure 3-8 shows the same data using a discrete notion of time. Note the thin gray line dividing each year and the broken line. Later in this chapter you learn how to add trend lines and reference lines that take advantage of discrete "panes" of time created in time series using discrete time.

The dual line chart on the right of Figure 3-8 presents two measures (sales and profit) using asynchronous axis ranges. Show Me assumes dual axis charts will be used to present values that are dissimilar and plots the marks using different axis ranges. The left axis displays sales dollars, while the right axis shows profit %. In the next section, you see three additional options for displaying time series data.

Area Charts and Dual Combination Chart for Time Series Analysis

Area charts can provide a visually appealing way to view time series data; however, care must be taken to explain what is actually being conveyed. The area chart (Continuous) on the left side of Figure 3-9 shows sales of three product categories over time.

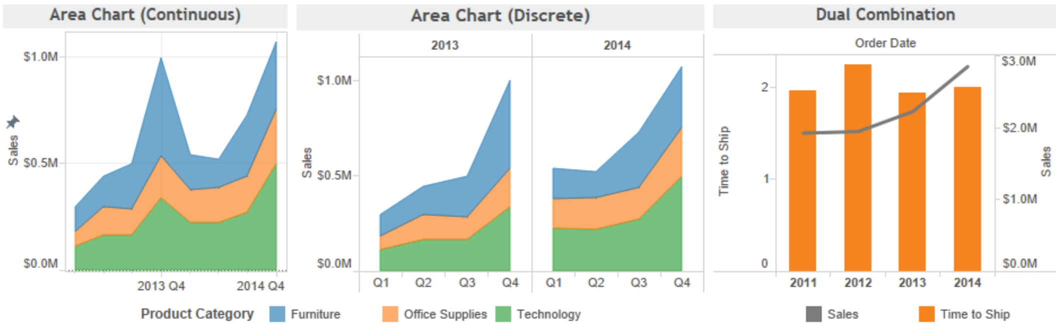


FIGURE 3-9 *Area charts and dual combination chart*

Area charts do this by stacking each dimension member (furniture, office supplies, technology) on top of each other. Area charts use the color-filled area to denote the relative values of the measure. For this reason care should be taken to explain that the height of the colored area represents all three product categories sales and not just the blue area denoting furniture sales. To precisely compare the relative sales value of each product category, the line charts in Figure 3-8 are a better choice.

The area chart (discrete) in the middle of Figure 3-9 shows the same data plotted using discrete time. Area fill charts make very effective sparklines (see glossary). Building sparklines is covered in Chapter 7.

The dual combination chart on the right side of Figure 3-9 is similar to the dual line chart presented in Figure 3-8 but uses a bar to convey one measure and a line to convey another. Using different mark types in this way highlights that the data being plotted are different measures.

Scatter Plot, Circle View, and Side-by-Side Circle Plots

Enabling analysis of fine-grained data across multiple dimensions, the scatter plot is a great tool for exploring datasets. Figure 3-10 shows an example of a scatter plot.

The scatter plot uses two axes for comparing profit and shipping cost. Color and shape provide insight into two dimensions. Size isn't being used in the example but could be used for a third measure. Each mark represents a customer, and nearly 5,000 marks are plotted in the example. In a relatively small space, you can plot two measures and up to three dimension using color, shape, and size. This is why scatter plots are great for finding outliers in your data.

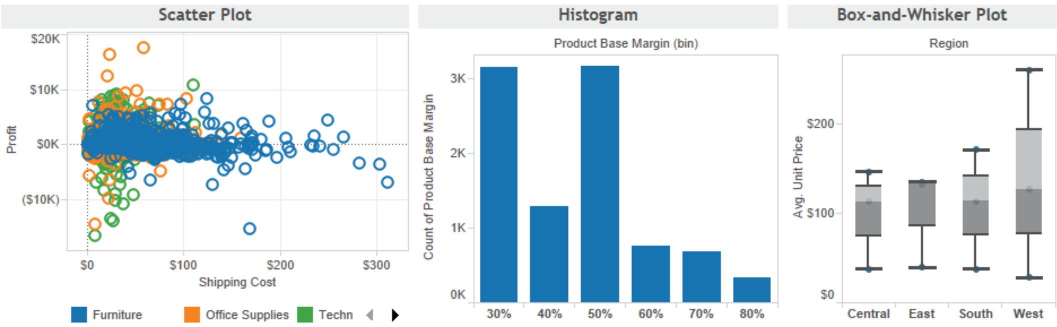


FIGURE 3-10 *Scatter plot, histogram, and box-and-whisker plot*

Histograms are used to display distributions of a single measure over binned ranges of values. The example in the middle of Figure 3-10 shows Product Base Margin in buckets of 10-point ranges. For example, the 50–60% range contains the largest count of transactions (over 3,000). Histograms provide insight in the number of values contained within a specific range of values. Tableau also makes it easy to parameterize these value ranges so that you can alter the size of the bins. You learn how to create parameters in Chapter 4.

Box-and-whisker plots are a new addition to Show Me. The example on the right of Figure 3-10 shows the range of average selling prices by region. The upper and lower values are shown via the reference lines that plot the maximum and minimum values. The gray-shaded areas show the quartile ranges, and the median value is shown via a small blue dot. Box-and-whisker plots are helpful in comparing the dispersion of values within a set of numbers. In the example presented, you can see that the West Region has a much larger range of average selling price than the other regions and that there were no outliers.

Bullet Graph, Packed Bubble, and Gantt Charts

The last three chart types provided by Show Me are completely different tools. Figure 3-11 shows them together, but their uses are very different.

You may have seen Gantt charts being used in project planning. These are particularly useful when you want to visualize the timing and duration of events. In the example on the left of Figure 3-11, the length of the bar color is the duration of time required to complete a shipment. The starting position of the bar is the date the order was received.

Bullet graphs are bar charts that include a reference line and reference distribution for each cell in the plot. In the example, current year sales (green bars) are compared to prior year sales (black reference lines). The color band behind the bar represents 60 and 80 percent of the prior year sales. Bullet graphs pack a lot of information into a small space, which makes them very good candidates for use in dashboards.

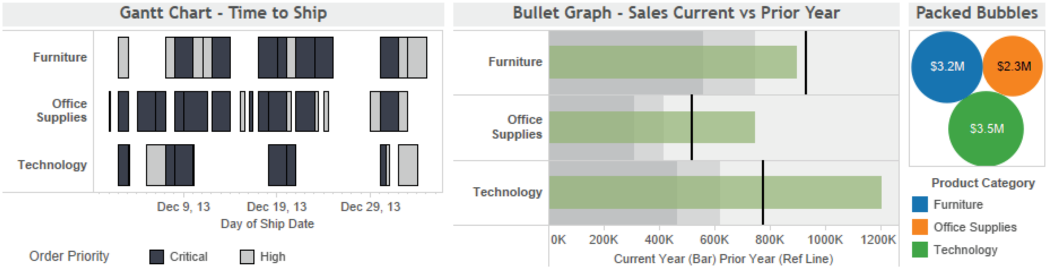


FIGURE 3-11 *Gantt chart, bullet graph, and packed bubbles*

Bubble charts offer another way to present relative values by using size and color. They can be interesting to look at but do not allow for very precise comparisons between the different bubbles. For this reason, limit their situations that don't require precise visual ranking of the bubbles.

Show Me is a real time-saver. If you are new to Tableau and don't completely understand what the shelves do, Show Me will help you build effective charts even if you don't completely understand how to create the view. In this way, Show Me provides insight into what happens, and the tool will speed your learning. Also, you can leave the Show Me menu open and move it around the screen. This allows you to quickly select different chart types and see the results immediately.

Once you have a chart in view, you can use that chart structure to add additional information. Two common ways to do this is by adding trend lines or reference lines to your chart. The numbers used to derive trend lines and reference lines can come from the view in Tableau itself and don't necessarily require that the data exist in your data source. You'll learn about the new Analytics pane along with trend lines and references lines next.

THE ANALYTICS PANE

A new feature added in Tableau V9.0 makes it easier to create trend lines, reference lines, and forecasts. The Analytics pane shown in Figure 3-12 eliminates the need to go to a menu or to know the exact place to right-click (Control-click on the Mac) to add trend lines, reference lines, or forecasts.

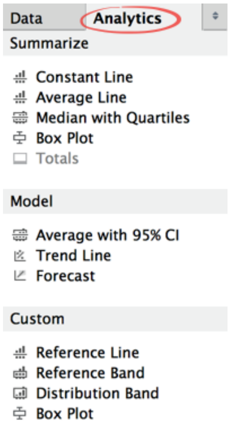


FIGURE 3-12 *The Analytics pane*

This exposes options for summarizing, modeling, and forecasting the data in the view. A linear regression line with confidence lines can be added to your view by double-clicking the trend line option in the analytics. If you want more control over the type of trend line added to the view, drag the Trend Line option from the Analytics pane onto the polynomial option, as you see in Figure 3-13.

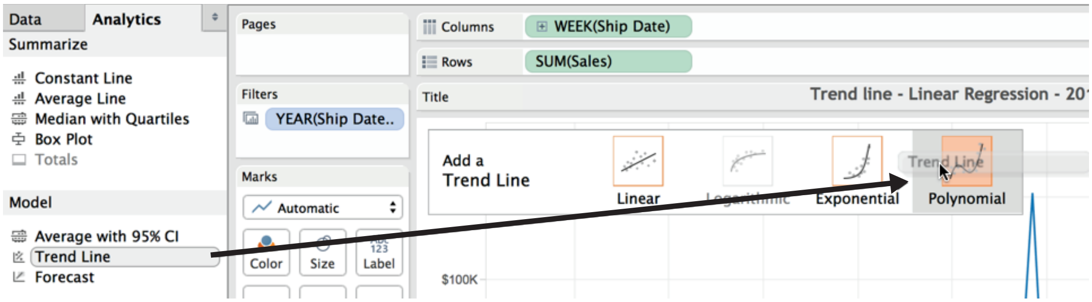


FIGURE 3-13 *Adding a polynomial trend line*

Dragging the Trend Line option into the view exposes more options. Figure 3-14 displays the resulting polynomial trend line.

A quadratic trend line (degree 2) is fitted to the data in Figure 3-14. Higher-order polynomials can be selected on the Trend Lines Options menu, as shown in Figure 3-15.

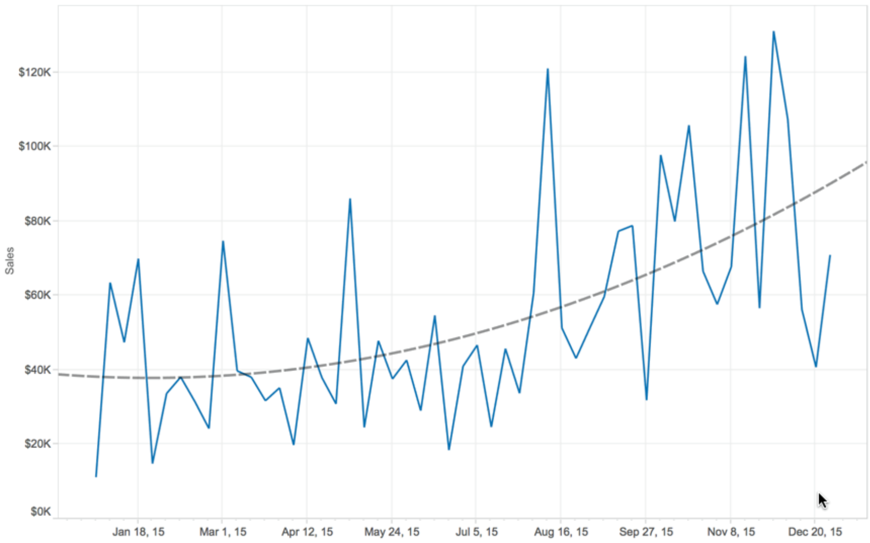


FIGURE 3-14 *Polynomial trend line*

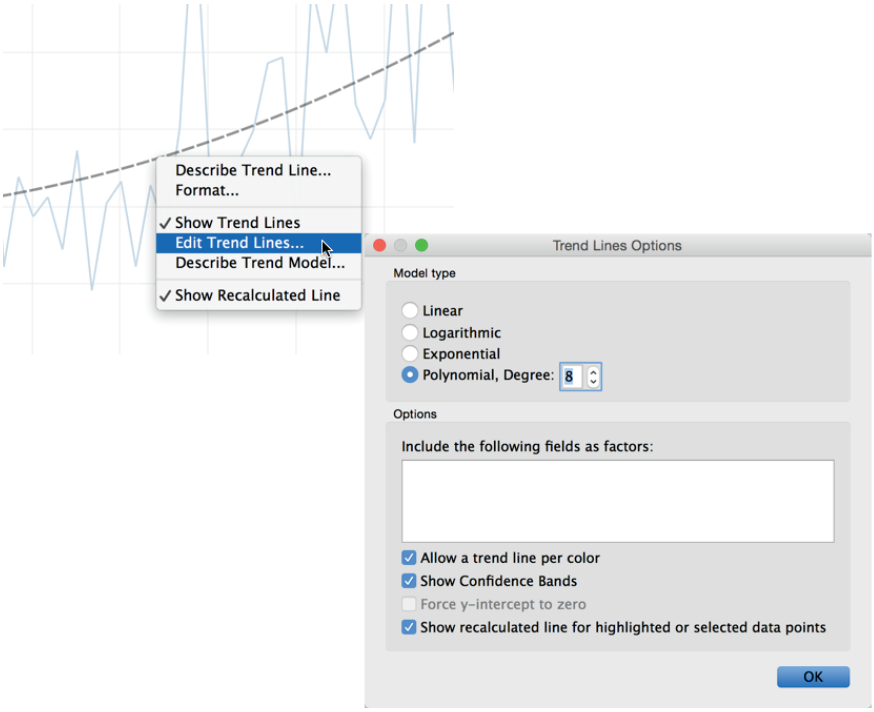


FIGURE 3-15 *Editing the trend line*

The resulting trend line in Figure 3-16 uses 4 degrees of freedom.

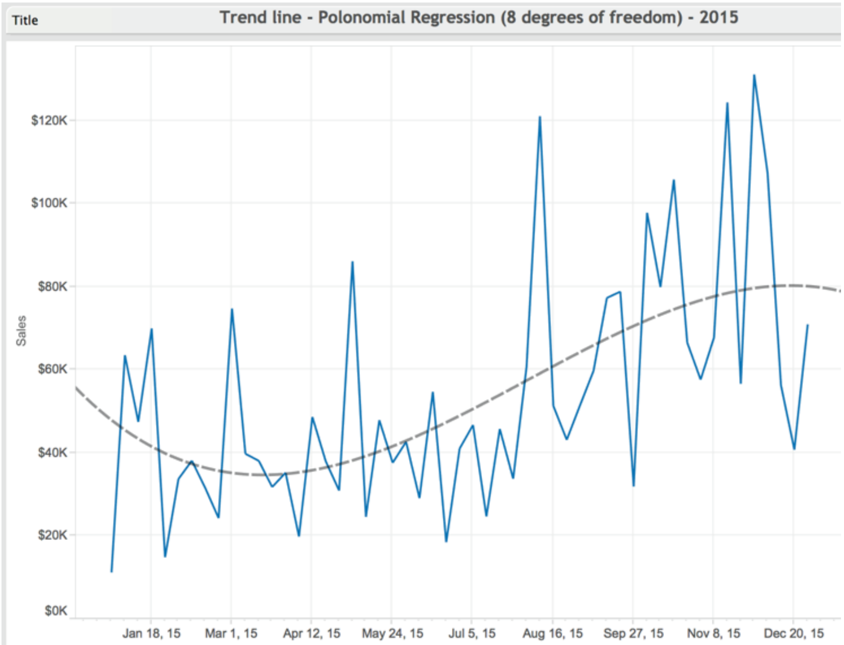


FIGURE 3-16 *Polynomial (degree 4) trend line*

If you want to add a reference line to a view containing discrete time buckets, dragging the average line from the summarize area of the Analytics pane enables you to define scope, as you see in Figure 3-17.

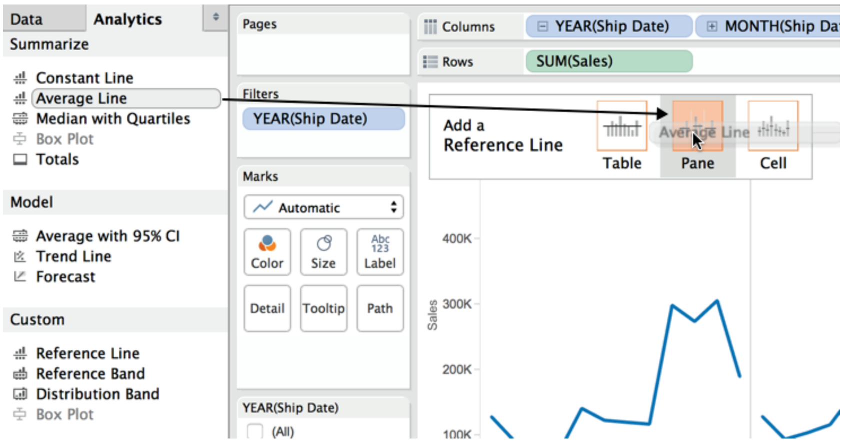


FIGURE 3-17 *Selecting reference line scope*

The resulting average line plot shows the average monthly values for each pane (year), as you see in Figure 3-18.

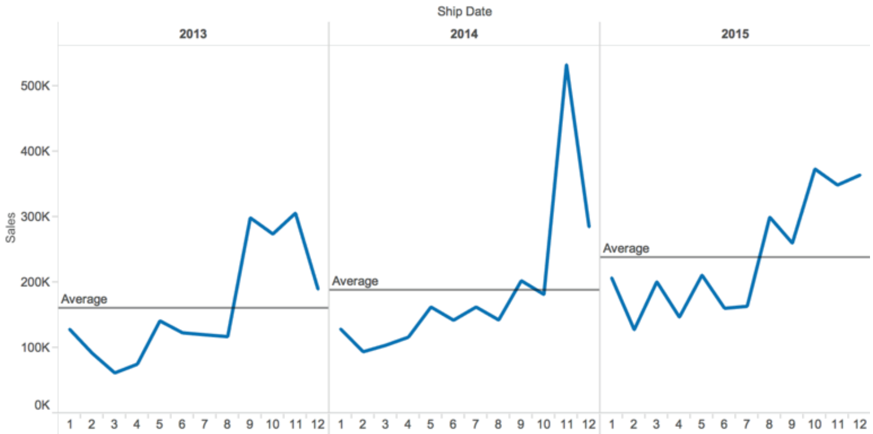


FIGURE 3-18 *Average by year reference line*

How would you add another reference line that provides a different scope and calculation to the data in the plot? Dragging the reference line from the custom area of the analytics pane exposes the Add a Reference Line dialog box. Drag the Reference Line, Band or Distribution icons from the Analytics tab Custom pane will display more options as you see in Figure 3-19.

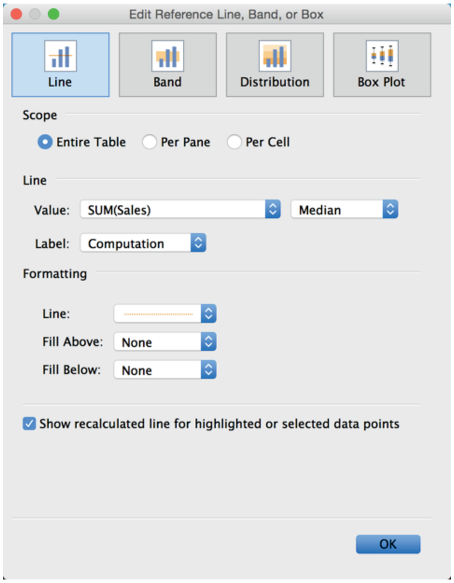


FIGURE 3-19 *Defining a second reference line*

You can see that the scope selected is Entire Table, the value calculation on sales is Median, and the formatting of the reference line has been altered to an orange dotted line to differentiate the median line from the averages already in the view. Figure 3-20 show the median line added to the view.

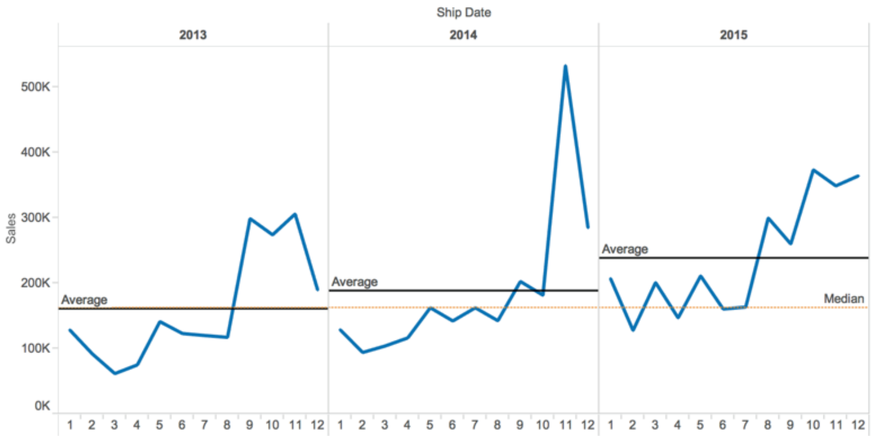


FIGURE 3-20 *Median reference line added*

The resulting view now contains two different reference lines. Notice the median line scope (table) is different than the average line scope (pane). The line color and style are different, and the positioning of the median label is on the far right. You can change the position of that label by pointing at the reference line, right-clicking (Control-click on the Mac), and selecting the Format menu to edit the alignment of the label text.

Add a forecast to the view using the Analytics pane by dragging the forecast option from the model section of the Analytics pane. Figure 3-21 shows the view with a 2016 forecast added.

You will have to edit the Forecast ➢ Forecast Options menu to get the forecast plot to look exactly like Figure 3-21. To edit the forecast options, point at the forecast line plot, right-click to expose the Forecast menu, and then select the Forecast Options menu, as you see in Figure 3-22.

Tableau assumes that the last month before the forecasted time period contains incomplete information. The default forecast applied ignored the month of December 2015 in the forecast calculation. The dataset used for this view has complete data for that month. Edit the forecast options to ignore the last “0” months, as you see in Figure 3-22, and your view will look like Figure 3-21.

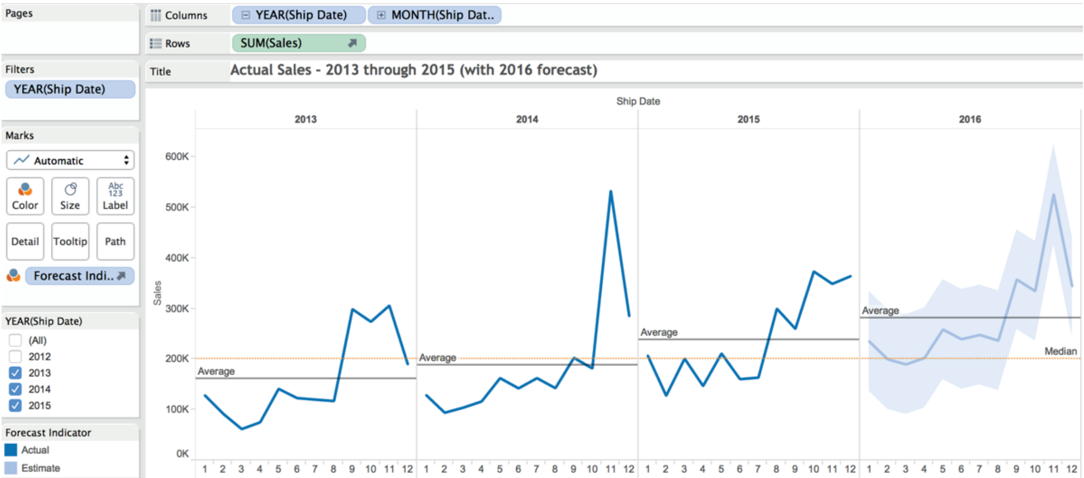


FIGURE 3-21 *Forecast for 2016*

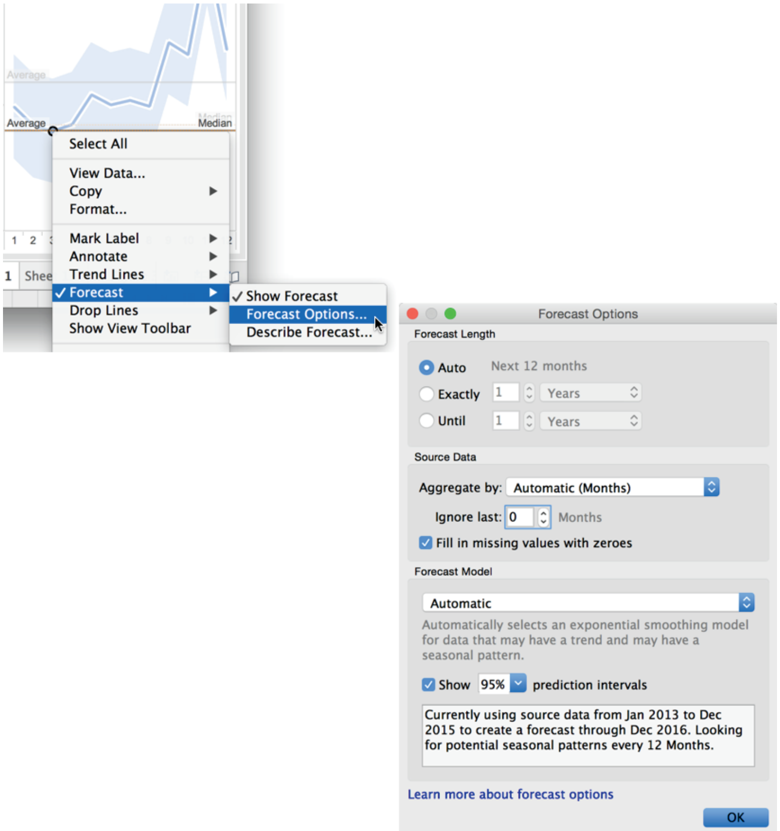


FIGURE 3-22 *Editing forecast options*

More complex views containing multiple measures result in more options being exposed when you drag more dimensions into the view. Refer to Tableau's online manual from the Help menu to see more examples. Chapter 6 provides more details on forecasting. Now let's take a closer look at how trend lines and reference lines help you understand your data.

Trend Lines and Reference Lines

Visualizing granular data sometimes results in random-looking plots. Trend lines help you interpret the data by fitting a straight or curved line that best represents the pattern contained within detailed data plots. Reference lines provide visual comparisons to benchmark figures, constants, or calculated values that provide insight into marks that don't conform to expected or desired values. Trend lines help you see patterns in data that are not apparent when looking at your chart of the source data by drawing a line that best fits the values in view.

Reference lines allow you to compare the actual plot against targets or to create statistical analysis of the deviation contained in the plot, or the range of values based on fixed or calculated numbers. Trend lines help you see patterns in data that are not apparent when looking at your chart of the source data by drawing a line that best fits the values in view. Figure 3-23 provides examples of a trend line and a reference line.

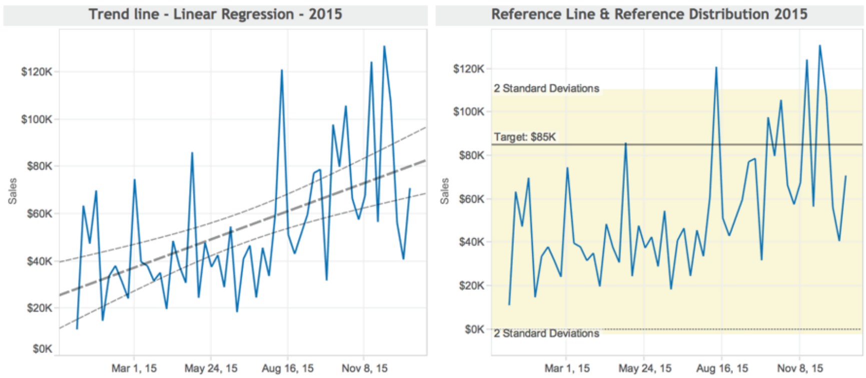
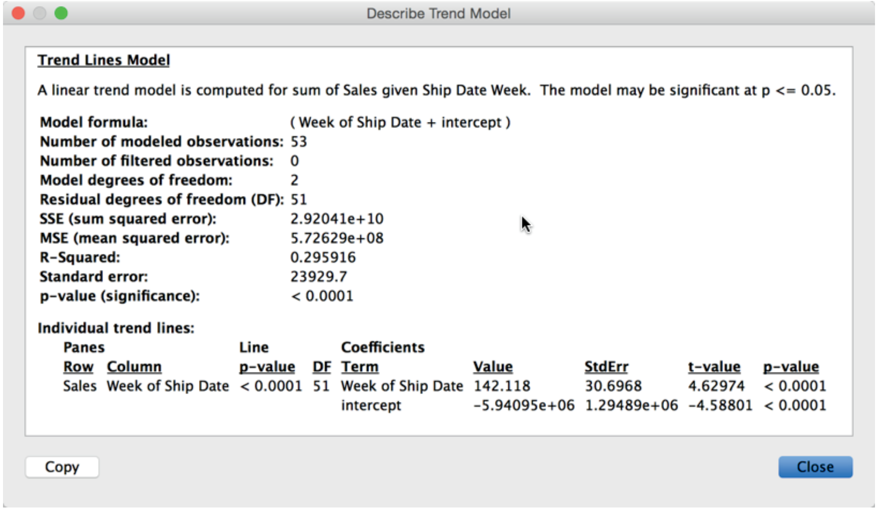


FIGURE 3-23 *Trend line and reference line*

The chart on the left uses a linear regression line to plot the trend for 2015 weekly sales figures. The pattern of sales is volatile—making it more difficult to see the overall pattern. A linear regression trend line has been added to the plot highlighting that the overall trend in 2015 is up. How reliable is the trend line plot? That question can be answered by pointing at the trend line and reviewing the statistical values displayed or by pointing at the trend line, right-clicking, and selecting Describe Trend Model. Figure 3-24 shows the more detailed description of the trend model statistics.


Describe Trend Model

Trend Lines Model

A linear trend model is computed for sum of Sales given Ship Date Week. The model may be significant at p <= 0.05.

Model formula:	(Week of Ship Date + intercept)
Number of modeled observations:	53
Number of filtered observations:	0
Model degrees of freedom:	2
Residual degrees of freedom (DF):	51
SSE (sum squared error):	2.92041e+10
MSE (mean squared error):	5.72629e+08
R-Squared:	0.295916
Standard error:	23929.7
p-value (significance):	< 0.0001

Individual trend lines:

Panes		Line		Coefficients				
Row	Column	p-value	DF	Term	Value	StdErr	t-value	p-value
Sales	Week of Ship Date	< 0.0001	51	Week of Ship Date	142.118	30.6968	4.62974	< 0.0001
				intercept	-5.94095e+06	1.29489e+06	-4.58801	< 0.0001

Copy Close

FIGURE 3-24 *Description of the trend model*

Describing the trend model exposes the statistical values that describe the trend line plot. If you are a statistician, all the figures will mean something to you. If you aren't a statistics expert, focus on the p-value and R-Squared figures. They help you evaluate the reliability and predictive value of the trend line plot. If the p-value is greater than .05, then the trend line doesn't provide much predictive value. R-Squared provides an indicator of how well the line fits the individual marks. The linear regression trend line displayed on the left side of Figure 3-21 is almost certainly not due to random effects (p-value is <0.0001), which implies a confident interval of around 99 percent. The line doesn't fit the marks particularly well because there is so much volatility in the weekly sales figures. The R-Square value (.295916) is low, indicating that the variation about the regression line is only 30 percent smaller than the variation about the mean.

The chart on the right in Figure 3-23 uses the same data as the chart on the left, but this time a reference line has been applied to show the target value of $85,000. A reference distribution has also been calculated to show two standard deviations from the mean value of the plot. Assuming the data is normally distributed, marks outside of that range indicate abnormal variation that would warrant further investigation to determine the cause of the variance. The points above the two standard deviation band are all in recent weeks. This is expected based on the previous regression analysis, which had a pronounced positive slope.

You don't need to become a statistics expert to use trend lines and reference lines. But understanding the basics will certainly help you interpret the plots. If you want to learn more about the statistics involved, a web search will provide more details regarding the mathematics.

Trend Lines

You can also add a trend line to your visualization by right-clicking the white space in the view and selecting the menu option Trend Lines ➢ Show Trend Lines. This adds a linear regression line to the chart. More trend line options are available if you point at the trend line, right-click, and select Edit Trend Lines. This exposes the trend line menu you see in the left side of Figure 3-25.

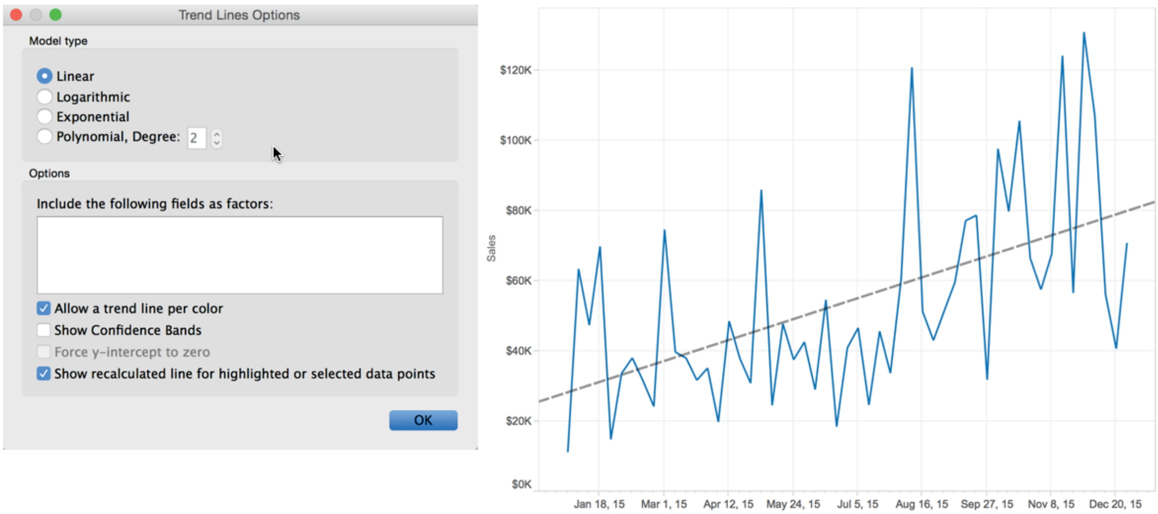


FIGURE 3-25 *Trend line options*

The Trend Lines menu provides options for changing the trend line type. If your chart uses color to express a dimension, you can choose to create separate trend lines for each colored line in the view—or not. Selecting Show Confidence Bands adds upper and lower bounding lines based on the variation of the data. Tableau adds the confidence bands by default. Notice that the confidence bands option is not checked, and the chart on the right side of Figure 3-25 does not include them.

Reference Lines

There are many different options for reference lines, and you have already learned that you can apply more than one reference line to an axis.

Without using the analytics pane, you can add a reference line by right-clicking on the axis from which you want to apply the reference line. Be careful to point at the white space and not at a title or axis label. Figure 3-26 shows the reference line menu selections used to add the standard deviation reference distribution displayed in the time series plot on the right side of Figure 3-23.

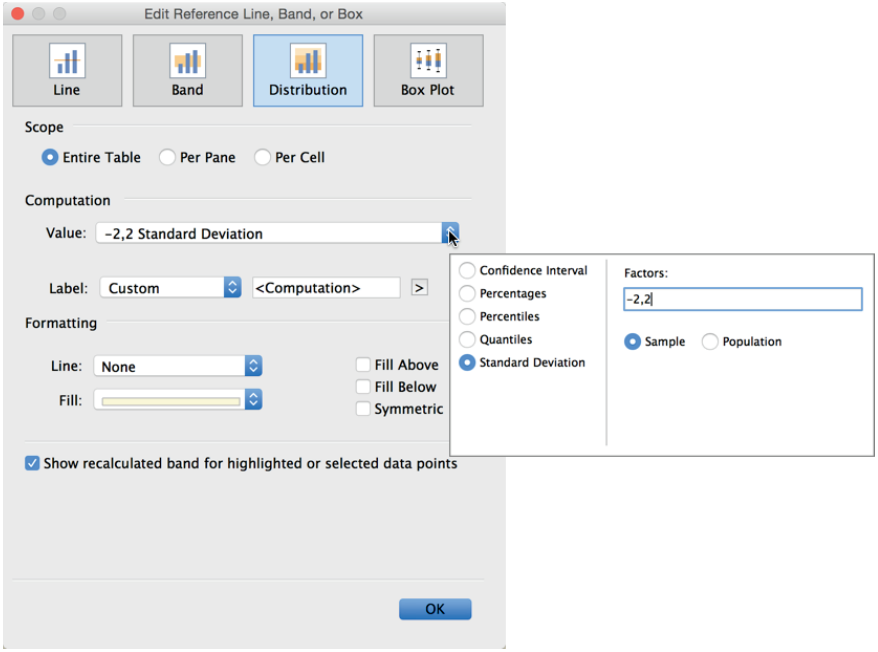


FIGURE 3-26 *Reference line menu settings for two standard deviations*

The same chart in Figure 3-23 includes a second reference line that displays a constant target value. This was added by selecting the reference "line" type to display a manually entered constant value of $85,000.

Two more reference line examples along with the related reference line menu selections can be seen in Figure 3-27.

The example on the left in Figure 3-27 combines a reference line displaying the median with reference bands for maximum and minimum values. The chart on the right side of Figure 3-27 uses a reference distribution to plot quintile ranges. Note the use of the Symmetric Color option. Selecting this causes the color bands outside of the widest quintile lines to use the same color hue. If Symmetric Color wasn't selected, the band color would get darker from top to bottom. Alternatively, if the symmetric options were left unchecked and the reverse was selected, the color bands would get lighter from top to bottom.

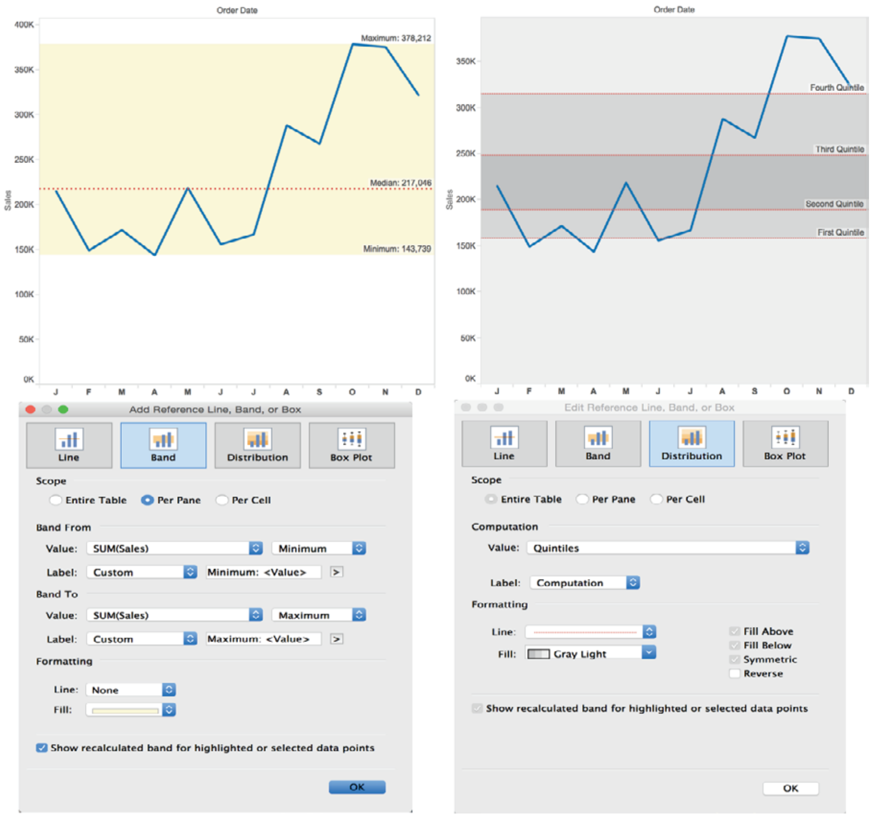


FIGURE 3-27 *Reference bands and reference distributions*

Applying color fill above or below reference lines calls attention to specific areas of your visualization. Use trend lines and reference lines in moderation. They add insight to your visualizations, but too many reference lines can lead to chart clutter and make it more difficult to understand.

Why the Concept of Scope Is Important

Understanding how the scope in trend line and reference line calculations determines the resulting appearance of the line is important not only for the deriving trend and reference lines but for understanding how calculated values and table calculations work in Tableau. We'll cover these topics in detail in Chapter 4, but the concept of scope (Cell, Pane, Table) that you learn here will help you when you do more advanced calculations later.

Figure 3-28 includes a time series chart on the left that contains two different reference lines, and the bullet graph on the right contains a single reference line for each bar (cell) in the view.

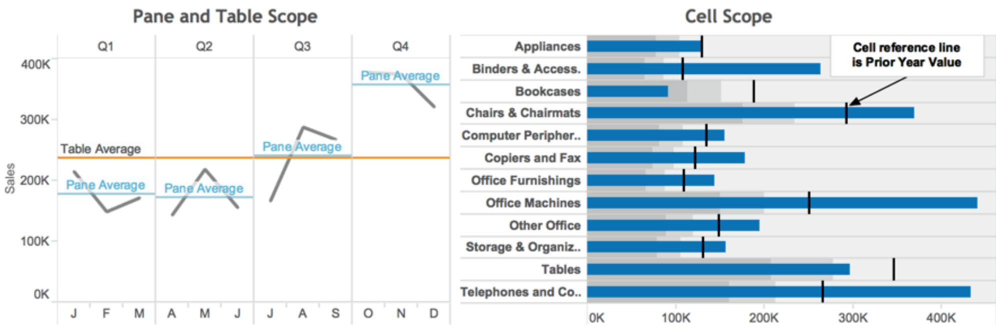


FIGURE 3-28 *Reference lines using Table, Pane, and Cell*

The time series chart on the left uses discrete dates to create panes by quarter. Tableau outlines the panes using gray lines. The scope of the calculation Tableau uses to create the orange reference line is Table. It shows the average value for the entire table. The scope of the blue line labeled Pane Average is Pane. By coincidence, the Table Average and Pane Average lines overlap in the third quarter. In all other quarters in the view, the pane average differs from the average for the entire year (table scope). The bullet graph on the right compares current-year values (blue bars) with prior-year values plotted using thick black reference lines. Those reference lines are applied using cell scope. The gray background is a reference distribution using Cell scope and showing 60 and 80 percent of prior year values.

Changing the Scope of Trend Lines

Scope can also be used to change the appearance of trend lines, but this is done in a different way than reference lines. Figure 3-29 shows examples of different trend line scopes applied to the same chart. Each chart is plotting the same time period and detail, but the trend lines use different details for calculating the lines.

Below each of the charts are the related settings for the trend line options. Notice that the scope of the left chart includes quarter, while the right side doesn't. This results in the right-side chart plotting a more general trend for the entire year, while the left chart is plotting the trend within each quarter. Also note that the default confidence bands have been turned off in both charts. Access the Trend Lines Options dialog box by right-clicking in the view while pointing at white space or at the trend line.

FIGURE 3-29 *Trend lines options*

Tableau provides four different kinds of trend lines (linear, logarithmic, exponential, and polynomial). Most people are accustomed to seeing linear (straight) regression lines in time series data. Polynomial regression provides a more curved line. Increasing the Polynomial degree (as you saw in the section "The Analytics Pane") will make the trend line follow the plot of the individual marks more closely.

Exponential and Logarithmic Regression Lines

One reason for using trend lines is predictive analysis—to help you see a possible future condition. The choice of method for calculating trend lines requires some professional judgment and is dependent on the data. People associate the word "exponential" with rapid growth. A real-world example of this is provided by the rapid advance of computing power over the past 40 years. Plotting numbers that change drastically and making those figures easy to interpret can be challenging. Figure 3-30 shows three different ways to plot a rapidly changing dataset.

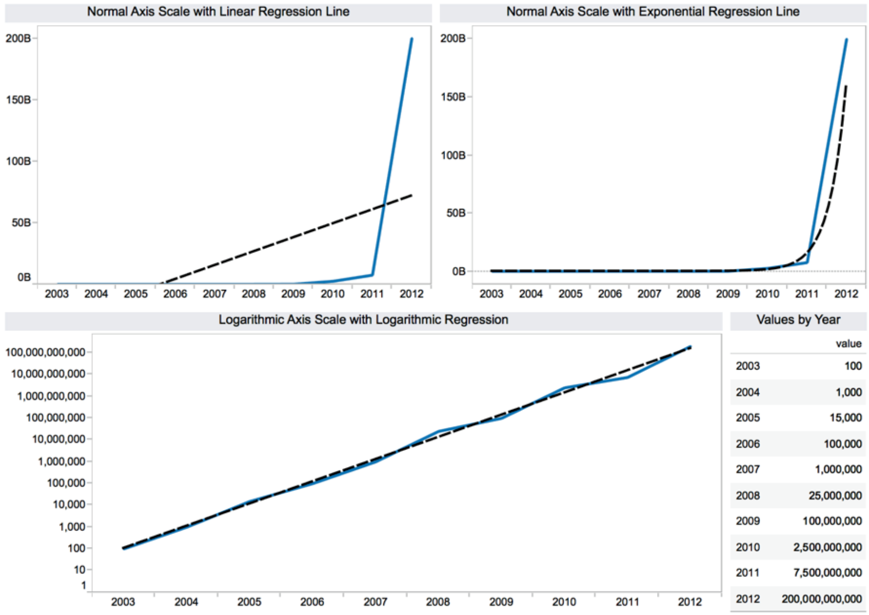


FIGURE 3-30 *Rapidly increasing time series*

You can tell by looking at the top two time series plots that the values plotted are increasing very rapidly over a ten-year period. These charts use a linear axis scale. In the top-left chart, a linear trend line is also used to smooth the data. The top-right chart uses an exponential regression line. It's obvious that the exponential trend line fits the data better. The bottom chart utilizes a logarithmic axis scale, which was altered by right-clicking in the white space of the axis and picking the logarithmic scale option. The trend line is also computed using logarithmic regression.

Tableau's logarithmic axis scale makes it easier to compare very different values in the same chart. The logarithmic regression line also makes it easier to see what next year's value might be. If you feel that logarithmic or exponential trend lines might benefit your analysis, you should arm yourself with the technical expertise to explain what the lines mean. As with all statistics, judgment should be applied. History may not repeat.

If you know a friendly statistician, ask them to explain the underlying theory and math. Alternatively, go to Khan Academy's website at `https://www.khanacademy.org/math/probability/regression` and watch the videos related to regression, statistics, and probability. Unless you understand the statistics supporting exponential and logarithmic smoothing, you should stick to what you feel comfortable explaining to your audience.

SORTING DATA IN TABLEAU

Tableau provides basic and advanced sorting methods that are easily accessed through icons or menus. Sorting isn't limited to fields that are visible in the chart—any field in the data source can be used for sorting. You can sort in ways that are temporary, or you can define custom default sorting order for your views.

Manual Sorting via Icons

The most basic way to sort is via the icons that appear in the toolbar menu. Sorting via the toolbar icons causes Tableau to sort based on the measure presented in the view. Sorting defined using the toolbar icons or the icons contained within views are temporary sorts. This way of sorting doesn't override Tableau's default sorting method. The toolbar menu sort icons provide ascending and descending sorts. Figure 3-31 shows a bar chart in which a manual sort was applied from the toolbar icon.

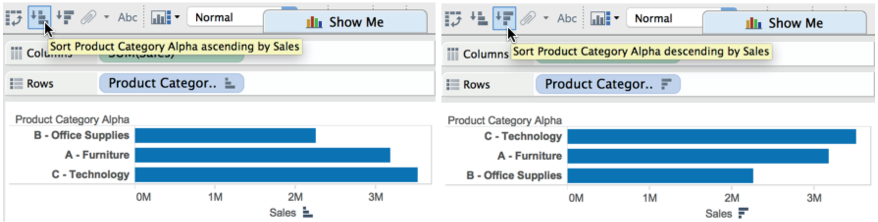


FIGURE 3-31 *Manual sorting applied from the toolbar icon*

Sorting using the toolbar icons provides ascending and descending sorts based on the measure in the view. Tableau also provides sorting icons near the dimension headings in the view. Figure 3-32 shows the three different ways you can apply temporary manual sorts using the icons over the dimension heading.

In the Figure 3-32 example, I duplicated the product category field by right-clicking the product category field on the Dimensions shelf and selecting the Duplicate option. Then I aliased the names of the three members in the product category set, prefixing A, B, and C to make it easier to follow how the sorting is applied. Sorting from the dimension heading column provides an additional sorting option.

The three manual temporary sorts provided by this option are all based on the field. In this example, that is Product Category Alpha. The first sort in Figure 3-32 is the data source order. The second is based on an ascending alphabetical sort of the field members (the same as data source order in this case), and the third is descending alphabetical sort of the field members. It doesn't matter how many levels of hierarchy are added to the view; you can sort on each level. Experiment by clicking the sort icons above each dimension and note the different outcomes. Note the subtle information contained in the right side of the field pill on the Rows shelf. When data source order is used, the pill doesn't display any bar chart icon, but when ascending or descending sorts are applied, a small bar chart icon appears.

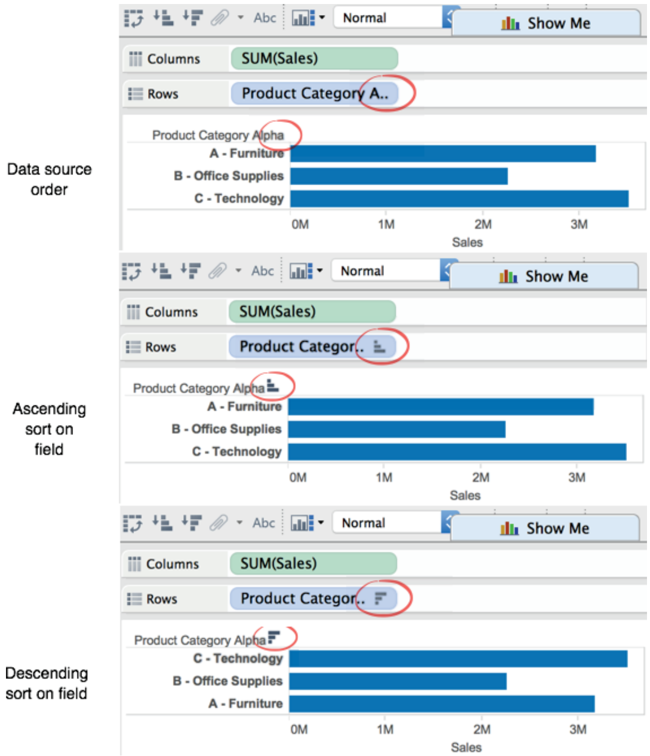


FIGURE 3-32 *Dimension heading manual sort*

Manual temporary sorts can also be applied when you hover the pointer above or below the measure marks in the view. Figure 3-33 displays an example.

Sorting via the measure icon cycles through three sorts: ascending or descending value in the view (in this example, sales) or the data source order of the dimension (in this example, the Product Sub-Category field).

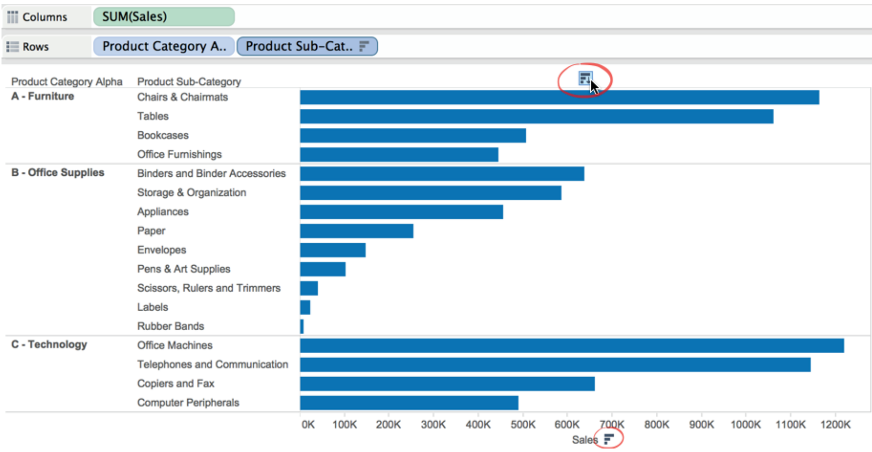


FIGURE 3-33 *Measure icon manual sort*

Calculated Sorts Using the Sort Menu

To replace Tableau's default sort with a custom sort point at a dimension pill, right-click (or select the drop-down arrow in the pill) and select Sort. This exposes the Sort dialog box you see on the right side of Figure 3-34.

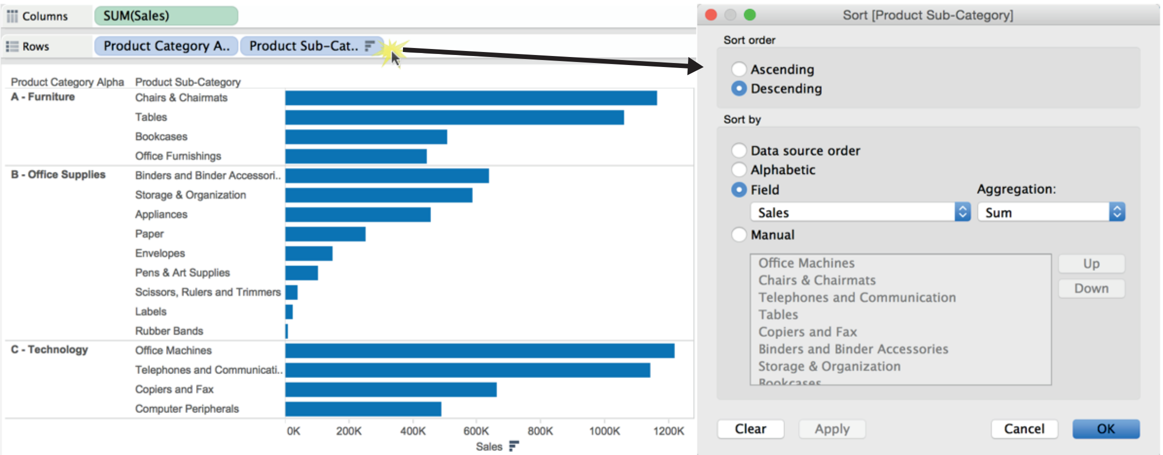


FIGURE 3-34 *Defining a custom default sort*

The dialog box allows you to redefine the default sort order within the view. In the example, you can see that the view is being sorted based on descending sales. You aren't limited to sorting based on visible fields in the view. For example, you could also apply Ascending Sort by Average Profit.

Try leaving the Sort menu open and select different fields to base the sort upon. Use the Apply button at the bottom right of the menu to see the result. To save the sort, click OK.

Sorting via Legends

Another useful sort feature is enabled within legends. Figure 3-35 shows two versions of the same bar chart. The right view has the blue delivery truck dimension on the bottom. The chart on the left shows green regular air at the bottom. Reordering the position of the colors displayed within the color legend causes the order of the colors appearing in the bars to change. Reposition the colors within the color legend by pointing at a color, holding down the left mouse button, and dragging the color to the desired position.

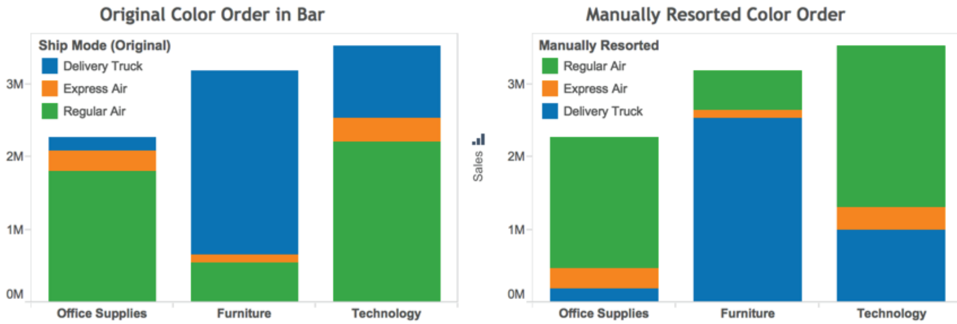


FIGURE 3-35 *Reordering the color in charts*

The ability to reorder colors in a stacked bar chart is important because precise comparisons are most easily made for the color that starts at the zero point on the axis. All of the other colors are not as easily compared because they don't start at the same value.

ENHANCING VIEW WITH FILTERS, SETS, GROUPS, AND HIERARCHIES

Sorting isn't the only way to arrange data. Creating drill-down hierarchies is easy in Tableau. Perhaps your data includes a dimension set with too many members for convenient viewing. Grouping dimensions within a particular field is available. Interacting with your data may uncover measurement outliers that you would like to save and reuse in other visualizations. That capability is enabled via sets. Even groups of sets can be created on the fly.

Making Hierarchies to Provide Drill-Down Capability

Hierarchies provide a way to start with a high-level overview of your data and then drill down to lower levels of detail on demand. In Figure 3-33, you see a two-level view of the data that includes product category and then subcategory. That presentation may include more detail than you prefer to see. A hierarchy that combines category and subcategory can address both needs. Figure 3-36 uses a hierarchy to show category first and Subcategory on demand.

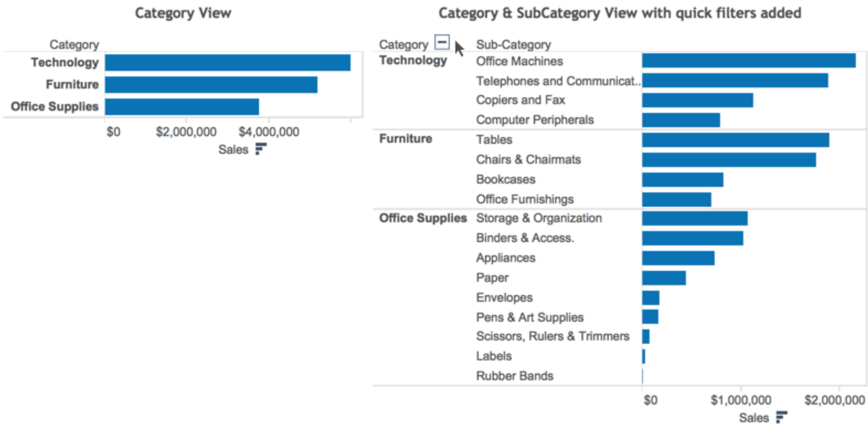


FIGURE 3-36 *Hierarchy category and subcategory*

The bar chart on the left displays the summary product category. Hierarchy pills have a plus/minus control box preceding the field name. By pointing at the category heading, a small plus sign will appear. Clicking that causes the subcategory level of detail to be exposed. To collapse the hierarchy, point at the category heading again and click the minus sign. You can create as many levels in your hierarchy as you desire.

Hierarchies are created by pointing at a dimension field and dragging it on top of another field. The order of appearance is defined by dragging the field names contained within the hierarchy icon to the desired position. Figure 3-37 shows the hierarchy icon with category and subcategory. You can change the hierarchy name by pointing at the text to the right of the hierarchy icon and typing **Product Hierarchy**.

Other fields can be added to the hierarchy by dragging them to the desired position inside the hierarchy grouping on the Dimensions shelf.

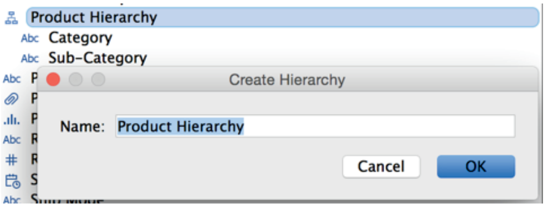


FIGURE 3-37 *Making a custom hierarchy*

Creating and Using Filters

There are a few ways to add filtering to your visualization. Dragging any dimension or measure on to the Filters shelf provides filtering that is accessible to the designer. Make that filter accessible to more people by turning it into a quick filter. This places it on the desktop where it is accessible to anyone—even those reading your report via Tableau Reader or Tableau Server. You can also create conditional filters that operate according to rules you define.

Creating a Filter with the Filters Shelf

In Figure 3-36, the Category and Subcategory view contains seventeen rows of data. Suppose you want to hide five of those rows from view. Dragging the subcategory field from the Dimensions shelf and placing it in the Filters shelf exposes the Filter dialog. Figure 3-38 shows the filtered data with the general tab of the Filter dialog. The subcategories that do not have check marks have been filtered out of view.

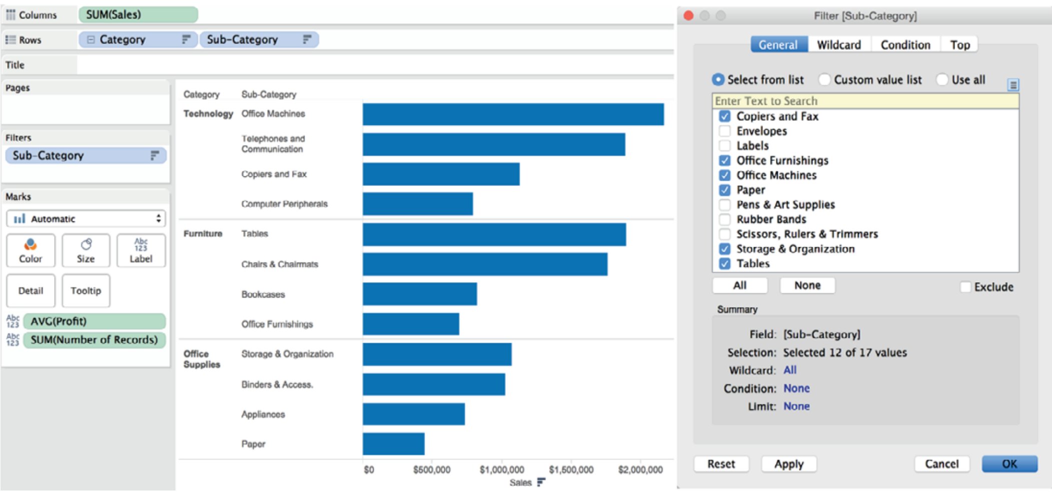


FIGURE 3-38 *Applying a filter via the Filters shelf*

Notice that there are three other tabs on the Filter dialog. The Wildcard tab is typically used to search for text strings to filter. If you want to filter using another field that isn't in your view, you can use the Condition tab to select any field in your data source and filter using that field. The Top tab facilitates building top and bottom filtering or filtering requiring other formula conditions. If you use more than one of the filtering options tabs to define your filter, Tableau applies the conditions defined in each tab in the order the tabs appear from left to right. General conditions will be applied first, then wildcard, then condition, and the top tab conditions last.

Below the general field list to the right of the None button is a check box for the Exclude option. If Exclude is selected, the items that include check marks are filtered out of view. Exclude filters can take a little longer to execute than Include filters, especially if your dataset is very large.

Quick Filters

If you want to make the filter available for people who are viewing the report via Tableau Reader or Server, you need to expose the filter control on the desktop. To create a quick filter, point at and right-click any pill used on any shelf in your worksheet; then select the Show Quick Filter option. Figure 3-39 includes quick filters using the category and sales fields.

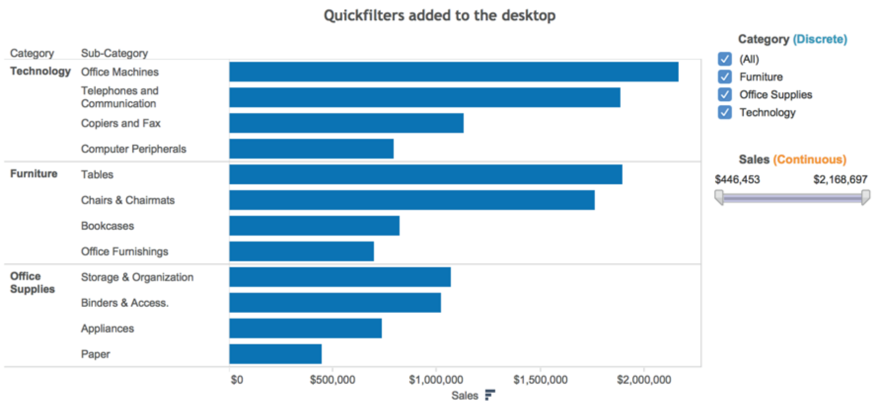


FIGURE 3-39 *Category and Sales quick filters*

The default quick filter styles are dependent on the type of field you apply within the Quick Filter control. In Figure 3-39, the discrete category field results in discrete filter options (furniture, office supplies, and technology). Discrete filters are expressed using radio buttons or multi-select boxes. The second quick filter for sales (a continuous range of values) is expressed using slider-type filters.

You can edit the quick filter type from inside the control itself. Click the title bar of the filter to expose the available options. Figure 3-40 shows examples of the menus that can be activated from the category and sales quick filter title bars.

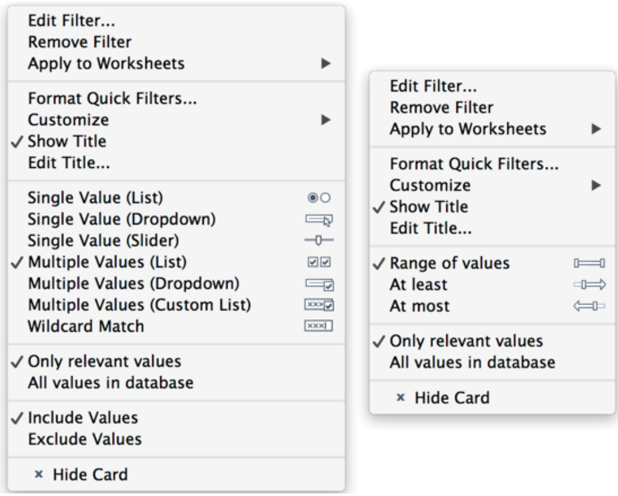


FIGURE 3-40 *Editing quick filter types*

The menu on the left side of Figure 3-40 relates to discrete category filters. The right menu is for the continuous filters. In addition to controlling the filter style, you can adjust other attributes. The quick filters were modified to include the words "discrete" and "continuous." Custom colors were applied to each word, and each title was centered in the title block of the Quick Filter. The results of the changes are displayed in Figure 3-39.

These are the Quick Filter menu options (both continuous and discrete):

- **Edit filter:** Exposes the main filter menu
- **Remove filter:** Removes the quick filter
- **Apply to worksheets:** Applies the filter to all or selected worksheets
- **Customize:** Turns on or off different filter controls
- **Show title:** Turns off or on the quick filter title
- **Edit title:** Modifies the text in the quick filter title
- **Only relevant values:** Reduces the set members displayed in the filter so that only values included in the filtered set are displayed
- **Include values:** Causes selected items in the filter to be included in the view

- **Exclude values:** Causes selected items in the filter to be excluded from view
- **Hide card:** Removes the quick filter from view but leaves it on the filter shelf

These are the Quick Filter menu items that appear only if the quick filter is on a dashboard:

- **Floating:** If activated, allows the filter to float on top of other worksheet objects
- **Select layout container:** Activates the layout container in the dashboard
- **Deselect:** Removes the layout container selection in the dashboard
- **Remove from dashboard:** Removes the quick filter from the dashboard

The remaining sections of each filter type control the style of quick filter. There are seven styles of discrete and three styles of continuous quick filter types available. One other feature available directly from the quick filter is the ability to control the relevant values displayed directly from the desktop.

Context Filters

One type of filter that many experienced Tableau users are unaware of is the context filter. Context filters do not only filter the data; they also cause Tableau to create a temporary table that contains only the filtered data. For this reason, they execute more slowly than a normal filter. Context filters are denoted by a gray-colored pill. They can be useful if you want to work with a subset to achieve a particular result. Don't use a context filter if you plan to alter the filter frequently.

Tableau provides robust filtering. In Chapter 8, you learn how to save space on dashboards by making the data panes act as filters.

Grouping Dimensions

When you have a dimension that contains many members and your source data doesn't include a hierarchy structure, grouping can provide summarized views of the data. You can manually group items from headers or multi-select marks in a chart. Tableau also provides a menu option with fuzzy search that will help you group by searching strings in large lists of values. You can even group by selecting marks in a view. If you need to work with data that isn't structured the way you want it, grouping allows you to build that structure within Tableau.

Creating Groups Using Headers

Figure 3-41 includes a bar chart that compares product subcategories within each product category. The office supplies dimension has too many small members with very low sales values. Grouping the six smallest categories in office supplies into a single (ad hoc) category creates a grouping that is more comparable to the other subcategories.

There are three ways to group headings. The easiest way is to click the paper-clip icon in the tooltip that appears when you multi-select the headers, as you see in Figure 3-41.

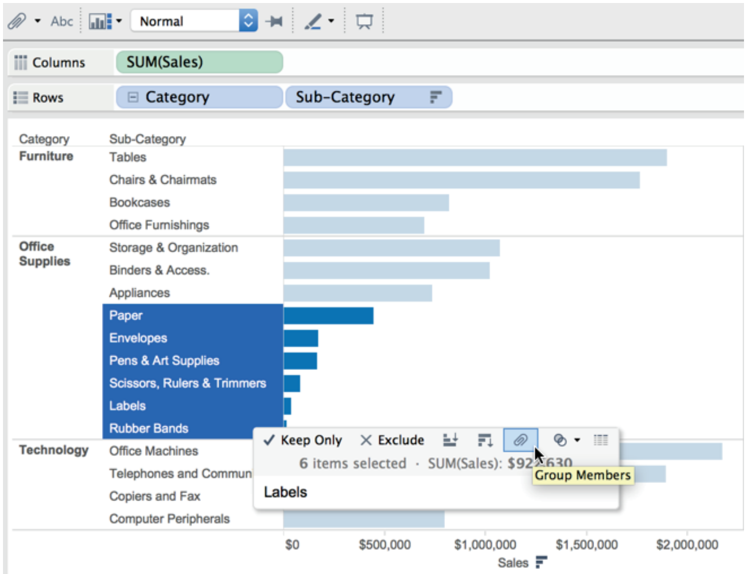


FIGURE 3-41 *Grouping from headers*

After creating the group, all six members will be combined into a single bar. The name that appears in the heading will be a concatenated list of the individual headings. To rename the combined list heading, right-click while pointing at the new group, choose Edit Alias, and type in a shorter name. The example group will be called "Other office." The combined members are then displayed, as you see in Figure 3-42.

You can also create groups by selecting marks in the worksheet. This method is a great way to highlight items of interest when you are performing ad hoc analysis. In Figure 3-43, you see a cluster of marks that have been selected.

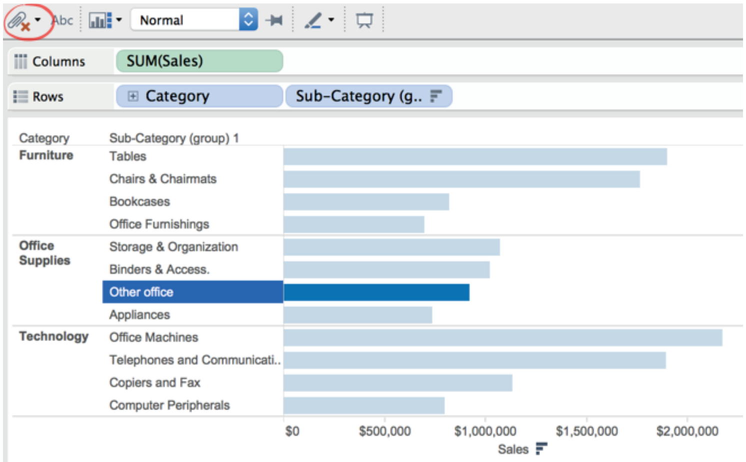


FIGURE 3-42 *Other office grouping*

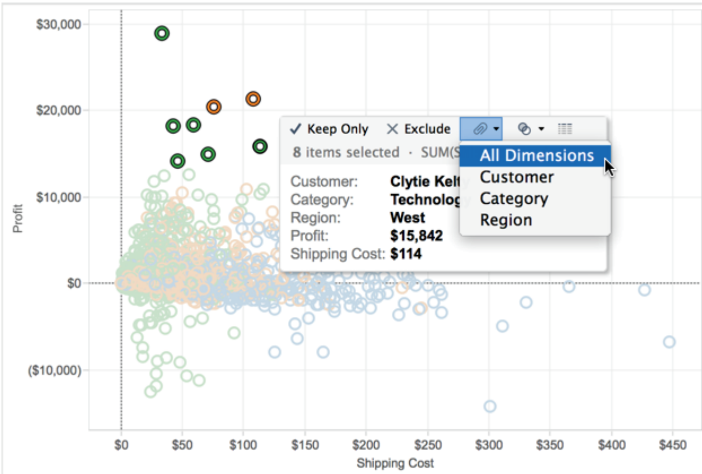


FIGURE 3-43 *Grouping marks using all dimensions*

These marks can be grouped using the paperclip icon inside the tooltip menu that appears when you point at any of the selected marks. Select All Dimensions to create the group. The result is shown in Figure 3-44.

Tableau's visual grouping causes the selected marks to be highlighted using a different color than the marks that are not included in the group. These methods work well if you have a small number of members to group, or you can easily select the marks that you want to highlight.

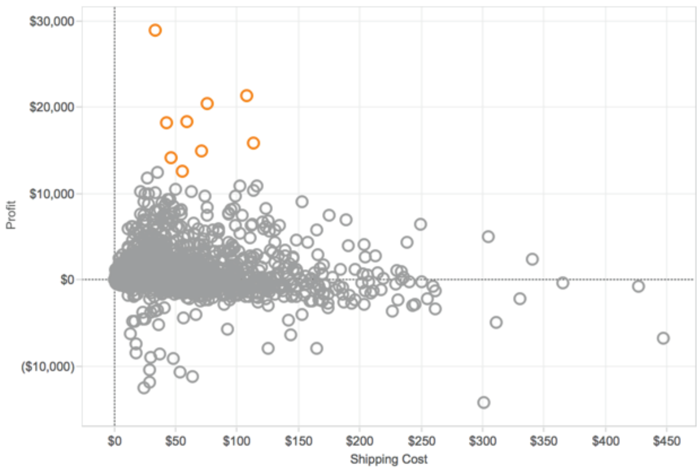


FIGURE 3-44 *Manually selecting a group*

If you have a very large set of dimensions that you need to group or the grouping must be created using portions of field names, these methods would be tedious. Tableau provides a more robust way to create groups using fuzzy search. Figure 3-45 shows another grouping menu that can be accessed by right-clicking a specific dimension field within the Dimensions shelf.

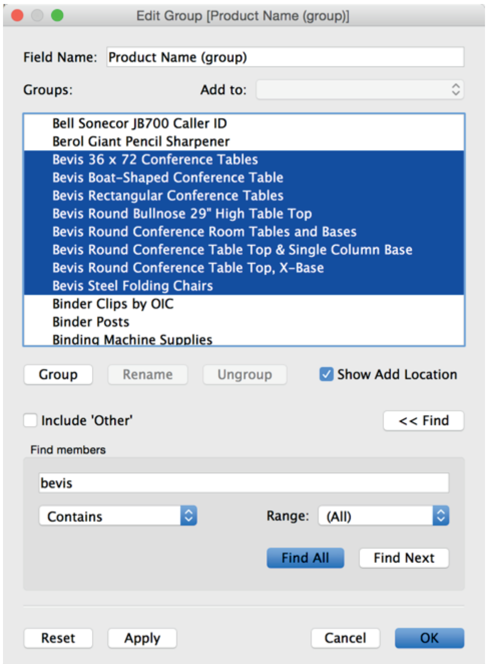


FIGURE 3-45 *Using string search to group*

In this case, the string search is initiated by right-clicking on the product name (group) field. In this dataset, the vendor name is included at the beginning of the product name string. You can use this as a way to group products by vendor. Figure 3-45 shows a search for all products provided by the vendor Bevis.

Clicking the Find button exposes the Find Members search. When “bevies” is entered along with the “contains” option, Tableau executes a string search in all the product names that include that string anywhere in the product name field. After checking to ensure the group contains the correct information, clicking the Group button will create a new grouping of the products. You can also alias the group name within the menu. After completing all the vendor groups you require, selecting the Include Other check box will generate a group that contains all the other items in the dimension that haven’t already been assigned to a vendor group.

Please note that any new group members that are added to your data source after your initial grouping using this method will not automatically appear in the group. You must add them manually the first time they appear in the data source.

Using Sets to Filter for Specific Criteria

Think of sets as special kinds of filters that enable you to share findings made in one worksheet across other worksheets in your workbook. Or, perhaps you want to create an exception report that only displays records that meet specific criteria. Sets can be created several different ways:

- Multi-selecting marks
- Right-clicking a field in the Dimensions shelf
- Combining sets on the set shelf

Constant sets are created by multi-selecting marks in a view and are static. Computed sets are made by defining criteria for inclusion in the set. Because they are defined by formula, they are dynamic, and results can change as your data changes. Keep in mind that computed sets can be defined using only a single dimension.

Creating a Constant Set to Analyze Outliers

A use case for constant sets occurs when you identify outliers in one view that you want to use as a filter in other views. Creating a set by selecting marks in a view is fast and easy. Figure 3-46 shows a scatter plot that is comparing profit and shipping cost. Low profit marks have been selected in the view by holding the left mouse button down and scrolling to highlight the marks to be saved in the set.

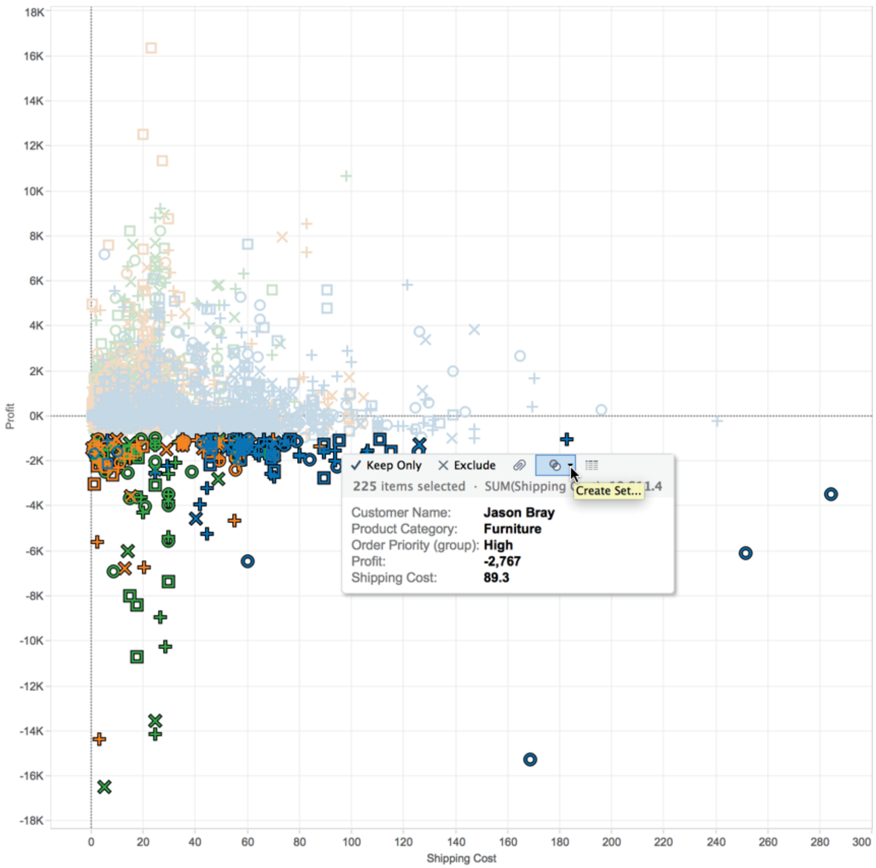


FIGURE 3-46 *Selecting marks to create a set*

Pointing at any one of the highlight marks causes the tooltip to be exposed that contains the Create Set icon. Selecting the Create Set menu option exposes the dialog box in Figure 3-47.

The scatter plot includes the customer name, order priority (group), and product category dimensions. If you want to exclude any row or column from the set, hovering the mouse over the row or column header exposes a red *x*. Selecting it removes the content of the item from being included in the set. For this example, all columns in Figure 3-47 will be included. Notice that the set has been named Low Profit Orders.

Clicking OK causes the Sets shelf to appear containing the Low Profit Orders set you see in Figure 3-48.

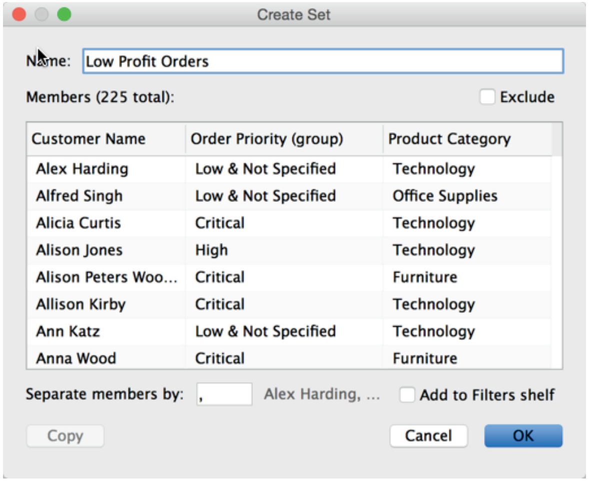


FIGURE 3-47 *Editing fields included in a set*

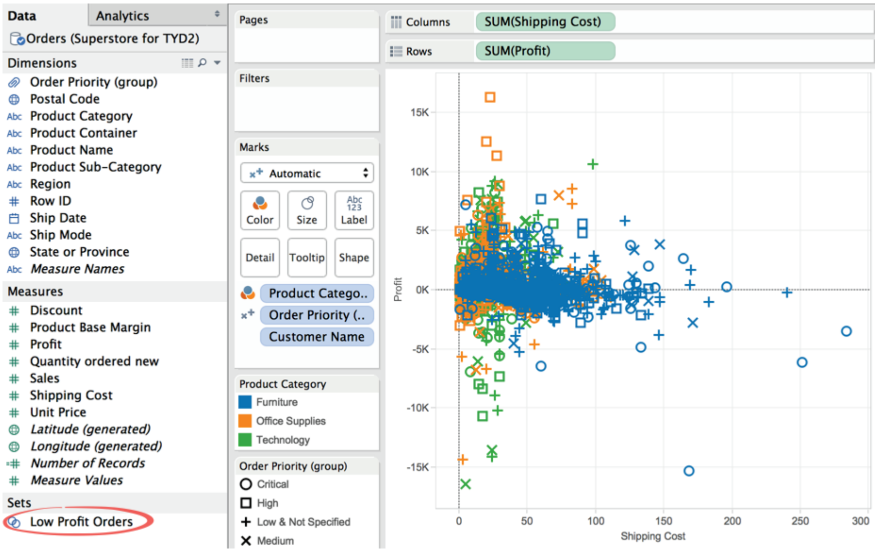


FIGURE 3-48 *Low Profit Orders set*

This set is now available for use in any other worksheet in the workbook. Figure 3-49 shows three different ways to use the Low Profit Orders set.

FIGURE 3-49 *Applying constant sets in views*

The first time series on the left of Figure 3-49 displays record count and profit dollars by month for the year 2015 without the Low Profit Orders set being used. Dragging the Low Profit set to the filter will change the view to display only the records included in the set. Notice the middle chart in Figure 3-49 has fewer records, and the Filters shelf for that chart includes the Low Profit Orders set. The profit trend line below the bar chart reflects the sum of only the low profit orders set.

Another way use the set is shown on the right side of Figure 3-49. This view was created by double-clicking the Low Profit Orders set on the Set shelf. This causes the set to be expressed using the Marks card Color option. Low Profit set items are encoded blue while items that are not part of the set are gray.

These constant (static) sets are very useful for preforming ad hoc analysis because you can quickly transfer findings in one view to many other views. But what if you want to create an exception report that requires a cross-section of two dynamic sets? You can accomplish this using computed sets.

Creating Computed Sets

Computed sets can provide a useful technique for creating exception reports that provide updated analysis every time the data source is refreshed. The filtering enabled by the sets can also be used to create individual charts that would be difficult or impossible to present otherwise. They are created using a single dimension. But, you can also join individual dimension sets together to create a combination set.

In the example that follows, you learn how to do this and the importance of validating your work. Computed sets are easy to build but can require more advanced skills to fully appreciate their power. The example steps you through creating, using, and validating the results for a computed set. You'll be using the World Indicators saved data source that ships with Tableau for this example.

Tableau updates these sample datasets regularly. I suggest you go the companion website at `http://tableauyourdata.com/downloads/` and download the Chapter 3 folder. You'll find the completed workbook along with a Tableau data extract (*.tde) called `Connect to the ComputedsetTYD2.tde` as the starting point for this example.

The following are the steps you'll follow to build this example:

1. Build some basic views to get an understanding of the dataset.
2. Create a field using the combination of the country and year fields.
3. Make the child mortality ≤ 1% Set.
4. Make the health expense per capita ≤ $1,000 Set.
5. Create the combination set using these two sets.
6. Test the result in a view.
7. Validate that the results are correct.

Unlike constant sets, computed sets require a little more effort to set up and validate. Some of the skills required are a bit more advanced, so don't get frustrated if you are new to Tableau and some of the concepts in this example aren't clear to you. The best way to learn Tableau is through examples. If you get stuck, the solution workbook will help. In addition, string calculations will be covered in detail in Chapter 4. Refer to Appendix E for additional examples and explanations.

Get to Know the Data

Once you've connected to the sample data, build a couple of views to get familiar with the information that you will be working with. Figure 3-50 shows a scatter plot comparing per-capita health expense with infant mortality rates for every country in the dataset.

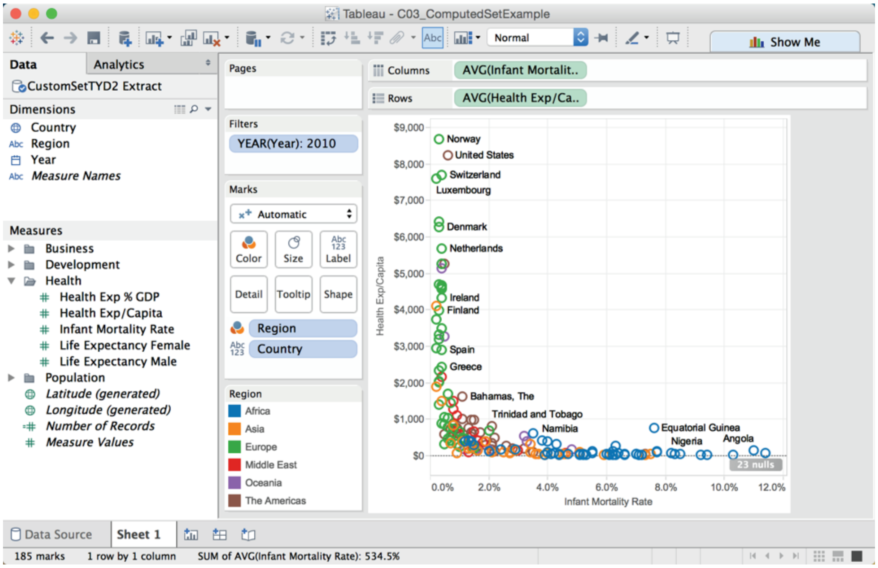


FIGURE 3-50 *Health cost versus infant mortality*

If you have trouble matching the image in the figure, copy the pill placements that you see on the shelves. Your result should look similar; you should have 185 marks in the view, and you should see the small gray pill in the bottom right of the view that indicates 23 nulls. This means 23 countries in the dataset do not have any data for the year 2010. This is okay for the purpose of this example. Click the pill and select the Filter Data option. This will filter the countries without records out of the view.

What is interesting about this view? I noticed that Norway and the United States have low infant mortality rates but very high per-capita health spending. It is interesting that there is a significant cluster of marks in the lower left of the scatter plot with very low per-capita health cost and low infant mortality.

Building a view to zero in on the cluster of marks that make up countries in the low cost and low mortality rate area of the scatter plot will expose more specific details on those countries and may provide additional insight. Figure 3-51 shows one possible view.

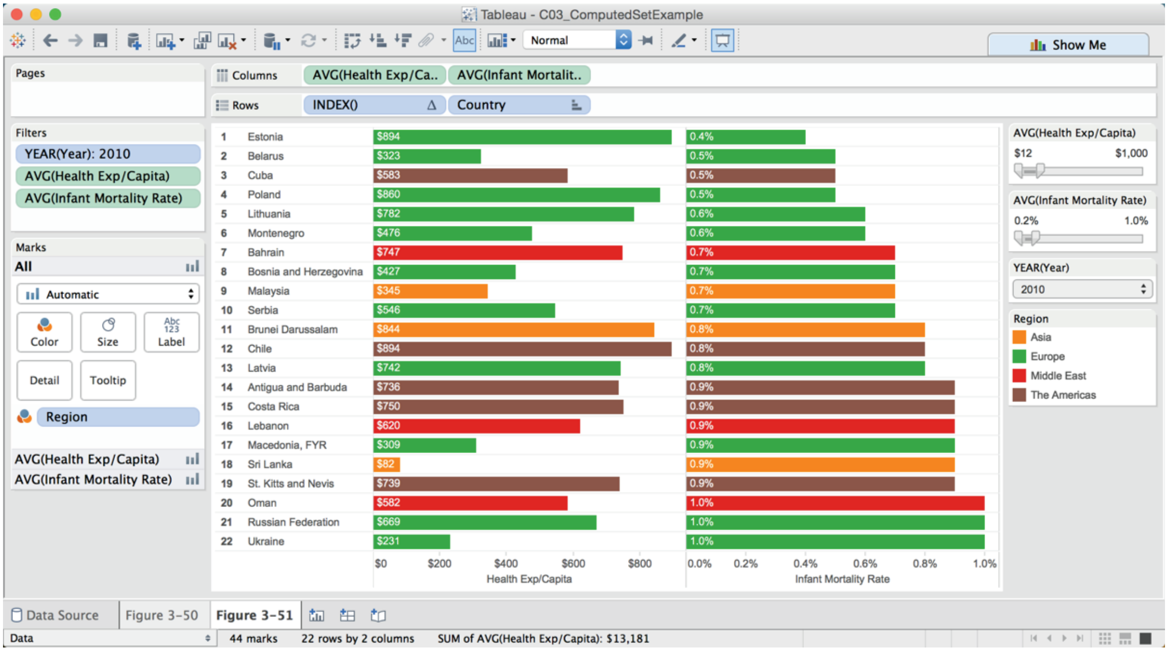


FIGURE 3-51 *Bar charts with filters*

The bar charts provide a much more detailed view of the countries with very low per-capita health costs and very low infant mortality. Notice the data is filtered for the year 2010, per-capital health costs of $1,000 or less, and infant mortality of 1.0% or less. Only 22 countries meet this criteria. Four of the six regions are represented. Comparing the scatter plot in Figure 3-50 with the bar chart in Figure 3-51, you can see that no countries in Africa or Oceania met the combination of criteria for inclusion in the bar chart.

Someone responsible for supporting global health initiatives might be interested in monitoring counties that meet these strict criteria to investigate exactly how they achieved these excellent results. Next, you'll use a computed set to monitor this combination of criteria.

Creating a Field Using Country and Year

Computed sets can be applied only to a single dimension. This data requires that you create a new field that combines the year and country because the source data includes different measures spanning many years.

Your goal is to create a computed set that combines both country and year for the measures of interest. In practice, knowing when and how to create a new field requires some experience along with some trial and error. That is why knowing your data is important. Creating computed sets normally means you realize something is important in your data and you are looking for a way to create specific reporting and analysis that will be used on a continuing basis.

You learn about calculations in Chapter 4. For this example, create a calculated value named Country-Year by copying what you see in Figure 3-52. Go to the Analysis menu and select the Create Calculated Field option to open the calculation window.

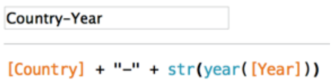


FIGURE 3-52 *County-Year calculation*

This field is the combination of the Country field and the Year field. You need both in the single field for the computed to provide the correct answer for any year in the dataset. The formula combines country with year. Tableau will not concatenate text with a number, so the string function STR converts the year to a string. The data in the dataset includes full dates, so the DATEPART function (year) converts the date into a year. This provides a single dimension that you'll utilize to build the computed sets.

Building the Computed Sets

Now that you have a dimension that includes the right combination of fields, you will start by building sets for child mortality and per capita health expense.

You create the sets by pointing at the Country-Year field on the Dimensions shelf and right-clicking; then select Create ➢ Set, as you see in Figure 3-53.

Once the Create Set dialog box is open, you have several options for defining the set. Pick the Condition option so that you can select and define the two sets needed using the appropriate fields and logical value definitions, as you see in Figure 3-54.

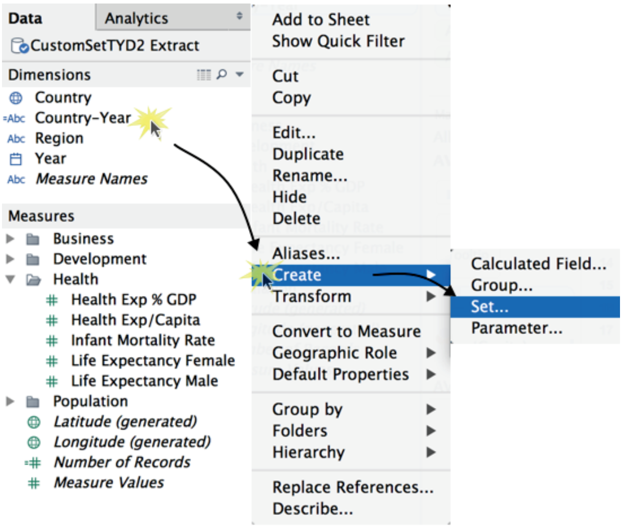


FIGURE 3-53 *Opening the Create Set dialog box*

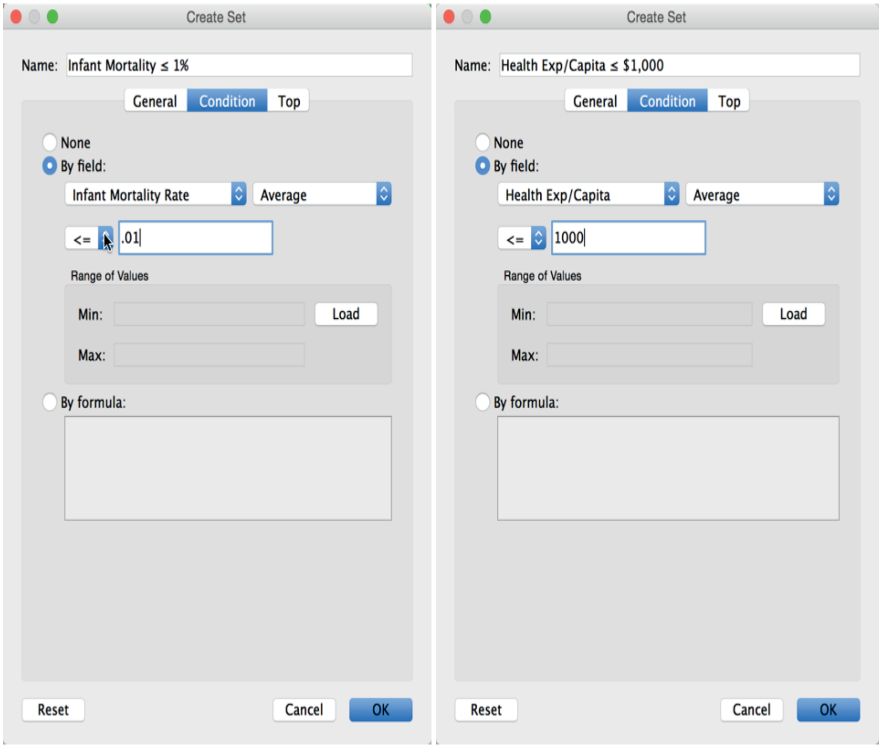


FIGURE 3-54 *Defining the sets*

You will create two different sets. On the left side of Figure 3-54 is the definition needed for infant mortality, and the right shows the per capita health set. Explore the Tableau online manual from the Help menu to see additional types of set definitions.

Your objective is to create a set that combines low infant mortality and low per-capital heath cost by country. Next you'll learn how to define the combination computed set.

Combining Multiple Sets to Create a Combination Set

Tableau has made the creation of combination sets much easier over the past two years. The dialog box used to define set combination logic is more visual and Tableau includes improved definitions for the options. Figure 3-55 shows the completed definition.

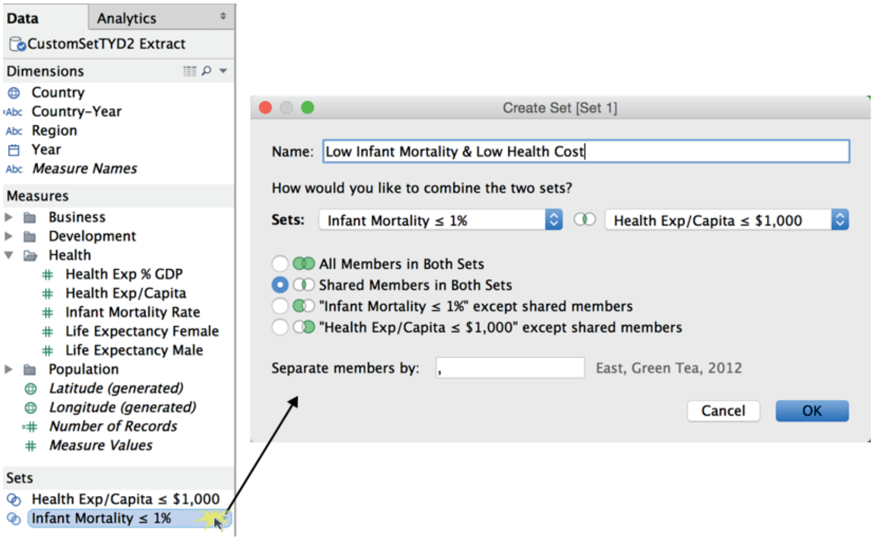


FIGURE 3-55 *Combining two sets*

To access the Create Set dialog box, right-click one of the computed sets in Data Pane Sets area. Doing that will expose the Create Set dialog box. Enter the name of the set and then use the drop-down selectors to pick the two sets that you want to combine.

Notice that the Shared Members option is selected. Click OK to create the computed set. It should now appear in the Sets area. You can now use the combined sets in views.

Creating Views Using the Combined Set

Creating views using the combined set is easy. Duplicate the views you previously created for Figures 3-50 and 3-51, but remove the sliding filters and replace them with the combination set. The scatter plot in Figure 3-56 shows that the data is filtered for the year 2010 and the combination set for the low infant mortality and low per-capita health cost has been applied to the view.

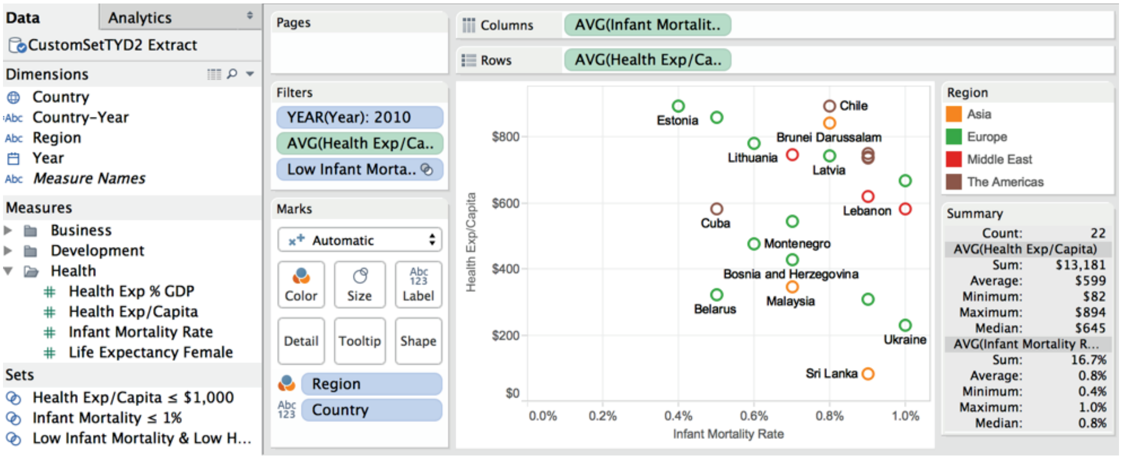


FIGURE 3-56 *Scatter plot using the combined set*

Try adding a quick filter for the year to this view by right-clicking the year Filters sin the filter shelf and following the menu options. You can edit that filter so that it looks like the one you see in Figure 3-57.

Notice that a drop-down style quick filter was added to the view in Figure 3-57 and that the date is filtered for year 2010. An index table calculation was added to the view as well to get a count of the countries in the filtered view. Don't worry about how that was created now. You'll learn about table calculation functions in Chapter 4.

If you managed to create these views, try to create the time series chart you see in Figure 3-58 using the combination set filter.

Using the set filter makes it easy to create the chart you see in Figure 3-58. The time series shows the annual figures for the number of countries that make up the low infant mortality and low per capita health cost combined set. The top section shows the number of countries that meet the defined criteria. The next two plots show the average values for annual per-capital health costs and infant mortality for the same countries.

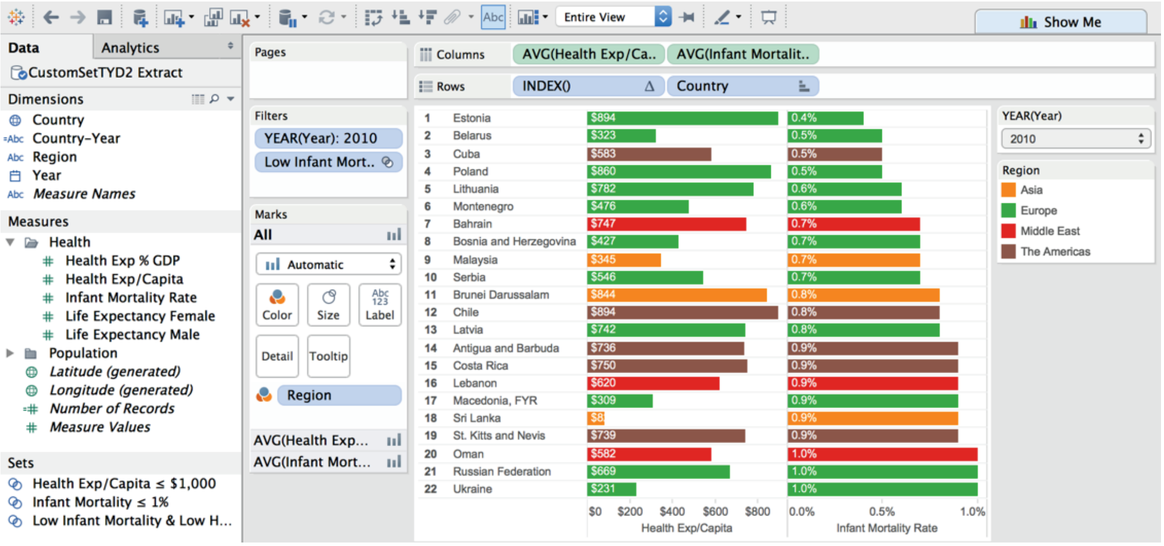


FIGURE 3-57 *Bar chart using combined set*

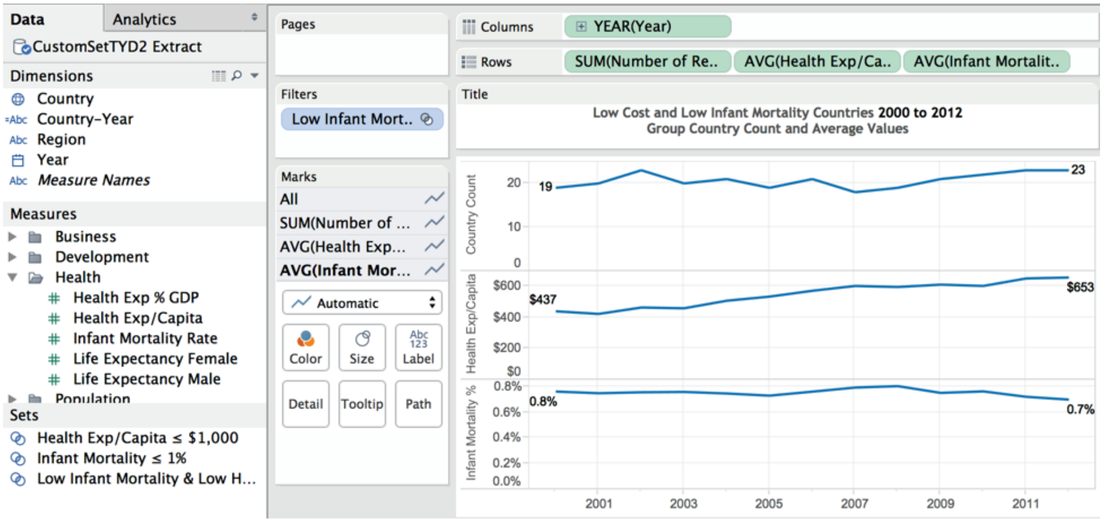


FIGURE 3-58 *Time series using combined set*

Best of all, you can combine these filtered views with unfiltered views in dashboards. Figure 3-59 shows one possible way you could show the combined set data with a few that shows all countries.

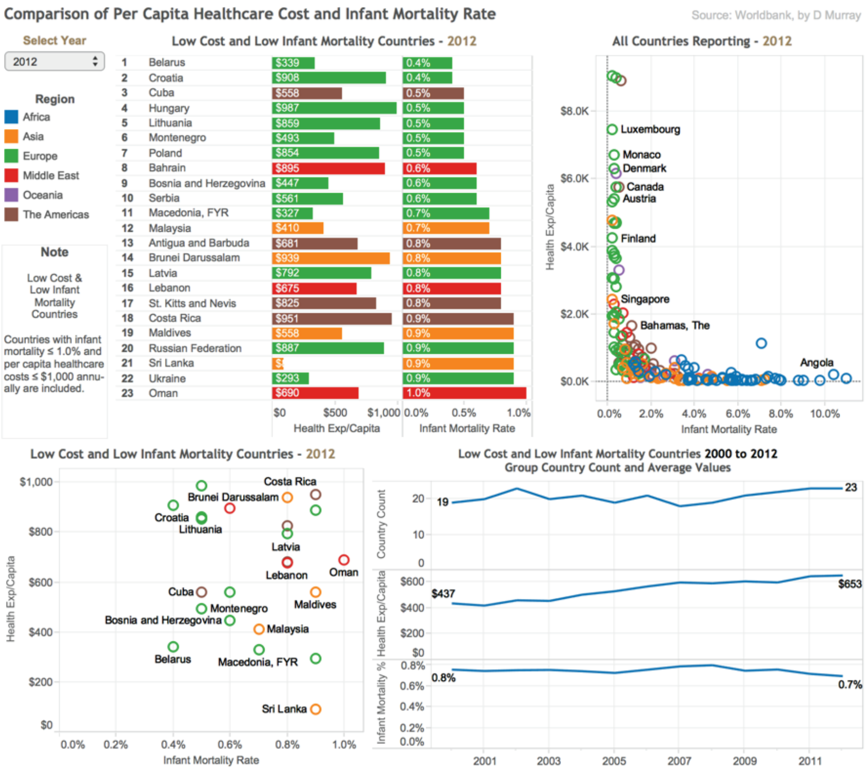


FIGURE 3-59 *Dashboard analysis*

In the upper left of Figure 3-59 of the chart, you can select the year to display in every chart except the time series view that shows all years. The scatter plot in the upper right shows all countries in a particular year and is not filtered using the combination set. The titles are dynamic and will change depending on the year selected. You can click the marks in the views, and the related regions will highlight in all of the charts except for the time series. You will learn how to build dashboards in Chapter 8. The sample data that you downloaded—`Computed Sets Example.twbx`—includes the dashboard if you want to play with it.

Combination Sets for Early Warning

There are many different ways you can use constant and computed sets for data discovery, reporting, and analysis. Now let's examine different ways you can use date dimensions with Tableau.

HOW TABLEAU USES DATE FIELDS

Tableau recognizes dates that are contained in your source data and allows you to change the level of detail displayed via an auto-generated hierarchy. It is also possible to rearrange date levels by changing the order of date pills on the Rows or Columns shelves.

Discrete and Continuous Time

You've probably noticed by now that some pills are green and others are blue. Similarly, icons can be in blue or green colors. Most beginners believe blue pills and icons denote dimensions, while green pills are used to display measures. While this is frequently the case, the truth is more subtle. Blue pills/icons denote "discrete" fields. Green pills/icons denote "continuous" fields. Dates can be both discrete and continuous. Figure 3-60 shows Tableau's default way of displaying time—as a discrete time hierarchy.

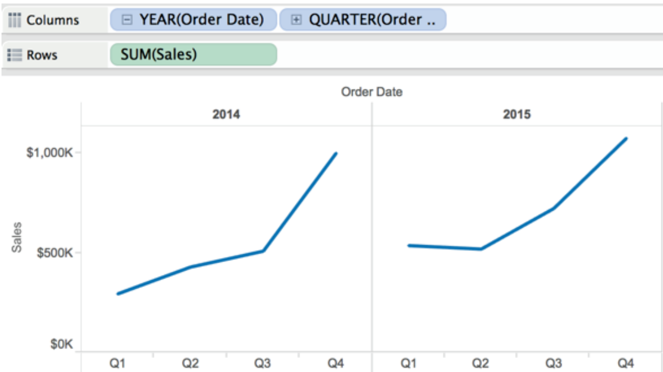


FIGURE 3-60 *Discrete time series*

You can see that time has been discretely segmented in the time series chart by year. Clicking the plus sign in the quarter pill would cause the date hierarchy to expand to include months and panes for each quarter. Continuous dates don't discretely bucket time but will cause a drill down to a lower level of detail. Figure 3-61 shows a similar time series chart that uses continuous time and the level of detail as month.

The green pill on the Columns shelf in Figure 3-61 indicates the level of detail being displayed. Notice that there are no panes in view. Time is continuously displayed as an unbroken line.

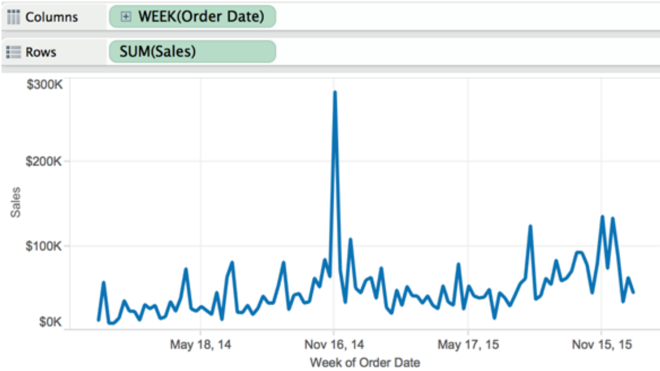


FIGURE 3-61 *Continuous time series*

Tableau's Date Hierarchy

Time can be expanded to more fine or coarse levels by clicking the plus sign within the date pill. Experiment with this and note that you can rearrange time buckets just by changing the order of the pills by repositioning them. It's also possible to change the level of detail displayed by right-clicking the date pill. If the discrete quarter pill in Figure 3-60 is right-clicked, this exposes the menu in Figure 3-62.

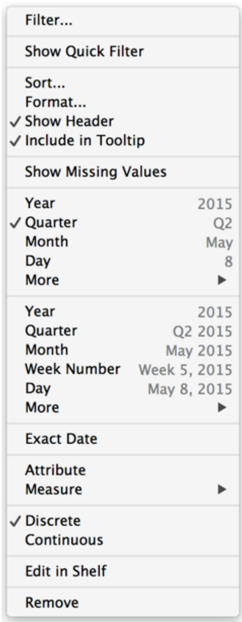


FIGURE 3-62 *Changing the date level of detail*

The menu includes two different date sections that start with the year. The first group provides discrete date parts. Notice that there is a check mark by quarter in the discrete date section of the menu. Also note the check mark near the bottom of the menu that confirms a Discrete date has been selected. The second group of date selection provides continuous date values. Figure 3-63 was created by changing the date displayed in Figure 3-60—altering the quarter pill to display the month.

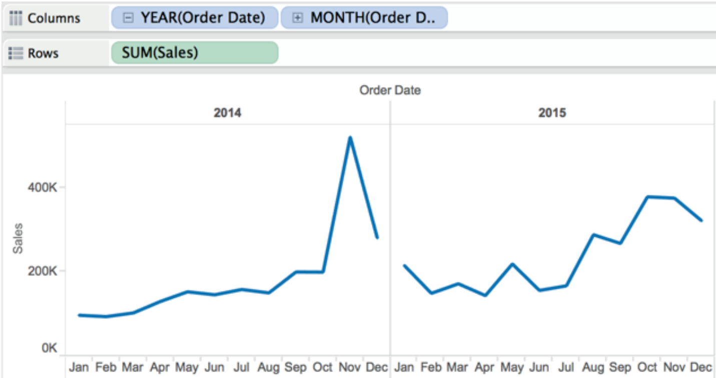


FIGURE 3-63 *Time series displaying discrete year-month*

In Figure 3-62, note that the menu option More appears twice. The first time it appears is within the discrete date section of the menu. The second time, it is in the continuous date section. Explore the menu option More in both the discrete and continuous time portions of the menu. The More menu options provide many options for controlling how date and time are presented in your view.

Rearranging Time with Tableau

There are many different date and time combinations that can be displayed. Figure 3-64 rearranges time to display weekday first, then year. Each day is a discrete time bucket. You can also add a reference line by pane that displays the average sales value for each weekday across all four years. This is one of the ways you can leverage discrete time to provide additional information.

If your data supports very fine-grained views of data, Tableau can display details down to the second. This is particularly useful if you need to analyze clickstream data on a website.

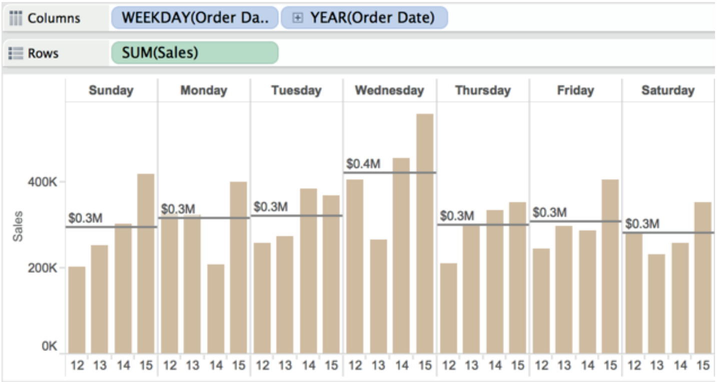


FIGURE 3-64 *Rearranging time and applying a reference line*

Creating Customized Date Fields

Tableau's date hierarchy is always available. Even people consuming reports via Tableau Reader or Server can expand time. When hovering your mouse pointer over an axis, you will see a small plus or minus sign appear. Clicking those signs expands or contracts the date hierarchy displayed.

Designers with Tableau Desktop can alter Tableau's default date hierarchy by creating custom date fields and then building unique date hierarchies. Making custom date hierarchies requires three steps:

1. Create a custom date.
2. Create the date hierarchy.
3. Use the custom date in your view.

To create a custom date, point at a date field on the Dimensions shelf and right-click. This will expose a dialog box that provides a means for defining a custom date or time element, as you see in Figure 3-65.

The two menu windows at the bottom of Figure 3-65 show the custom date definitions for the year and month used in Figure 3-66. Date parts are discrete. Date values are continuous.

Create a custom year date by naming the field Year and defining the date as a discrete year date part. You can also add another discrete date for month. By dragging the custom month on top of the custom year, you can add a new custom date hierarchy. Figure 3-66 shows the resulting date hierarchy.

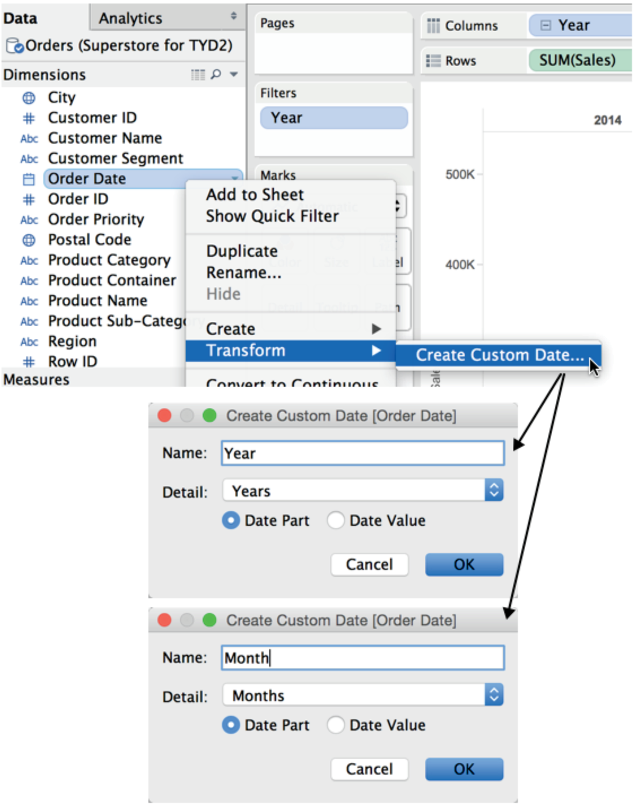


FIGURE 3-65 *Creating a custom date*

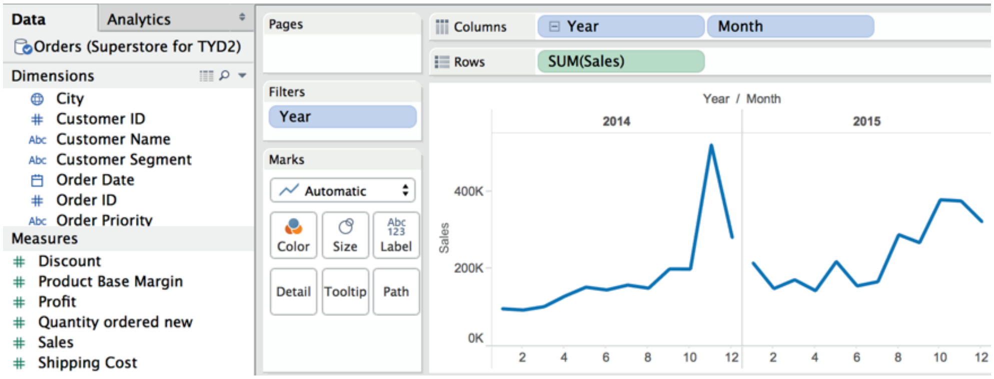


FIGURE 3-66 *Custom date hierarchy*

You can see the custom hierarchy in Figure 3-66 on the Dimensions shelf. The year and month custom dates are all that will be displayed in this time series chart. In this way, you can change how Tableau expands and contracts the dates used in your visualizations.

Tableau's date facility encourages explorations of data over different time slices because it is very easy to use and requires no special skill to master. Creating custom date hierarchies can help you control how much space is consumed within a view. This is particularly helpful for dashboards that have limited publishing space.

Taming Data with Measure Names and Values

Sometimes your data isn't clean and it may not be structured in a way that supports the analysis you need to perform. You might also be looking at a dataset for the first time and want to quickly get familiar with it. Tableau's Measure Names and Measure Values fields help you with all these tasks.

What Are Measure Names and Measure Values?

Measure names and measure values do not exist in your data source. These fields are generated by Tableau. Measure names hold the names of every measure in your dataset. Measure values hold all of the measures in your data set.

When to Use Measure Names and Measure Values

Any time you need to combine more than one measure on a single axis, measure values and measure names facilitate your analysis. Figure 3-67 displays a summary of all of the measures in the Superstore for TYD2 dataset.

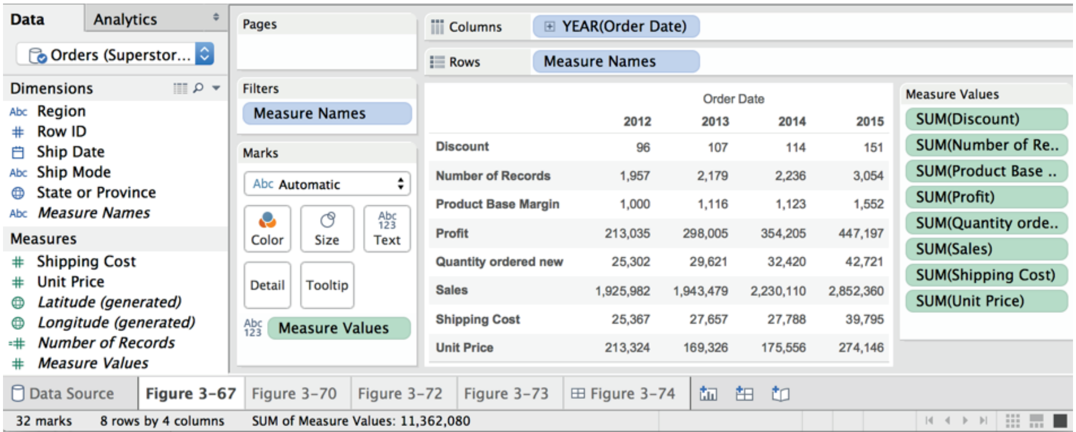


FIGURE 3-67 *Measure names and measure values*

The cross tab in Figure 3-67 was created by double-clicking on measure names on the Dimensions shelf, swapping the axis, and adding order date on the Columns shelf. This technique provides a very quick overview of the data. The status bar at the bottom left tells you that there are 32 data marks and 4 years of data. Notice the Measure Values shelf on the right provides details on the aggregation used to display the information. You can change that by right-clicking any of those pills and picking another option.

Another way to use measure names and measure values is to tame data that is poorly structured. Figure 3-68 includes a sales projection in a spreadsheet that isn't structured in the best way for analysis using Tableau.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Product Code	Product Name	Jan-15	Feb-15	Mar-15	Apr-15	May-15	Jun-15	Jul-15	Aug-15	Sep-15	Oct-15	Nov-15	Dec-15
2	001	Widget 1	100	110	110	105	155	160	150	160	170	160	155	145
3	002	Wangle 2	45	45	50	48	49	52	55	60	70	65	55	50
4	003	Widget 3	25	30	40	50	55	60	60	60	70	70	65	60
5	004	Wangle 1	100	100	105	100	110	110	100	105	115	110	100	90
6	005	Waxel 1	30	30	35	35	35	40	45	45	50	48	45	40

FIGURE 3-68 *Sales forecast in spreadsheet form*

Each product has unit sales projected in columns for each month in the year. Tableau will interpret each column as a separate measure. In a database, this information would be stored in a structure that looks like Figure 3-69.

	A	B	C	D
1	Product Code	Product Name	Month-Year	Unit Forecast
2	001	Widget 1	January-15	100
3	001	Widget 1	February-15	110
4	001	Widget 1	March-15	110
5	001	Widget 1	April-15	105
6	001	Widget 1	May-15	155
7	001	Widget 1	June-15	160
8	001	Widget 1	July-15	150
9	001	Widget 1	Aug-15	160
10	001	Widget 1	September-15	170
11	001	Widget 1	October-15	160
12	001	Widget 1	November-15	155
13	001	Widget 1	December-15	145
14	002	Wangle 2	January-15	45

FIGURE 3-69 *Sales forecast in a database format*

General-purpose databases normally store data in a row-oriented format. Tableau can connect to both kinds of data sources. If your data is more column-oriented, like Figure 3-68, measure names and measure values provide a means for creating views that wouldn't be supported otherwise. Figure 3-70 shows a time series chart that was built using the spreadsheet data source.

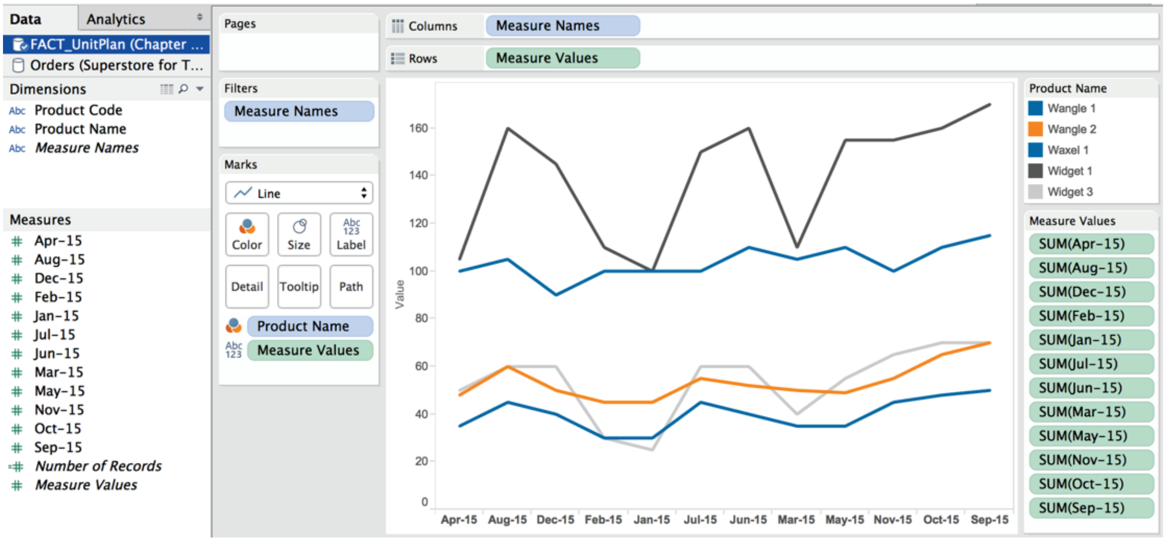


FIGURE 3-70 *Using the spreadsheet data source*

Tableau interprets each column in the spreadsheet as a separate measure. A time series chart is used to plot the change in a single measure over time. Look at the Measures shelf in Figure 3-70. Because each month is presented in a separate column in the spreadsheet in Figure 3-69, Tableau is interpreting each month as a separate measure. The columns in the spreadsheet are displaying the same measure for 12 months. Measure names and Measure values allow you to place multiple measures on a single axis. Look at the Rows shelf in Figure 3-70 and the related Measure Values shelf that appears below the Product Name color legend. Tableau uses Measure Names to treat all of the different month Measures like a single measure. Even though the data format in the spreadsheet is not stored like it would be in a database, using Measure Names and Measure Values allows you to achieve the desired presentation quickly.

If your data source is formatted as you see in Figure 3-69 (like a database would store the information) Measure Names and Measure Values would not be required to make the chart you see in Figure 3-70.

The column and row structure you see in Figure 3-69 directly supports the creation of the view without having to use measure names and measure values because the Month-Year is contained in a single dimension and the Unit Forecast is contained in a single measure. Placing the Product name field on the color button within the Marks card would express each product with its own color. Placing the Month-Year field on the Columns shelf puts each month in its own column, and the Unit Forecast can be placed on the Columns shelf.

This would replicate the chart you see in Figure 3-70 without using Measure Names and Measure Values. Unfortunately, you may have to deal with spreadsheet structures and build views. Measure names and measure values help you deal with an untidy data structure.

Advanced Uses for Measure Names and Measure Values

Measure names and measure values facilitate more advanced chart types as well. Your spreadsheet includes another tab containing unit price and cost information that looks like Figure 3-71.

	A	B	C
1	Product Code	Price	Cost
2	001	$99.00	$39.62
3	002	$50.00	$22.31
4	003	$60.00	$26.29
5	004	$25.00	$9.85
6	005	$125.00	$52.63

FIGURE 3-71 *Unit price and cost data*

You can join that tab to the forecasted units tab in Figure 3-69 via the product code key record. This will allow you to create calculated values extending sales, cost, and gross margin dollars. Then you can create a time series view of the three measures, as you see in Figure 3-72.

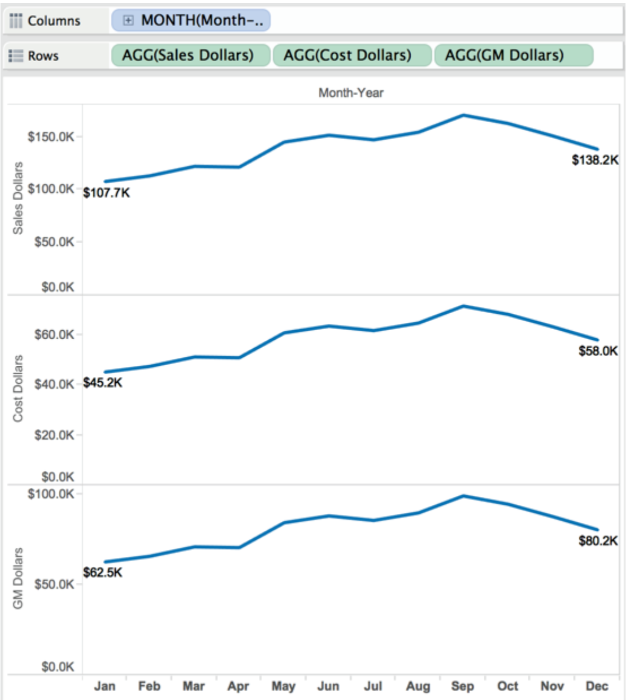


FIGURE 3-72 *Tableau's default presentation*

Tableau's default presentation of this information uses a separate axis to display each measure. Using measure names and measure values, you can combine all three measures on a single axis and use color to differentiate the values. Figure 3-73 shows the combined axis time series chart.

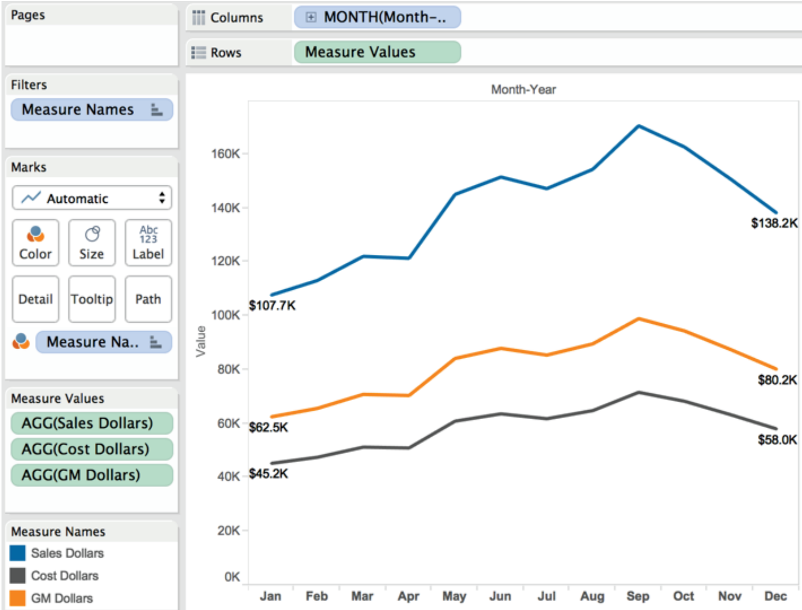


FIGURE 3-73 *Multiple measures on one axis*

The combined axis chart in Figure 3-73 can be created a few different ways. Using the view in Figure 3-72 as a starting point, the most intuitive way to combine the measures on one axis is to drag each measure to a single axis. Pointing at the cost axis in the upper left makes a green fold mark appear. Pointing at the green fold makes a cross arrow appear. Drag the green fold from the cost axis on top of the sales axis as you see in Figure 3-74.

Repeating the same method, you can relocate the gross margin axis so that all three measures are displayed together on a single axis. Note in Figure 3-73 that the Rows shelf now holds the Measure Values pill. Measure names are on the Marks card for color. The measure values shelf also appears, displaying all three measures. Tableau automatically creates the measure name and measure value pills when a second measure is moved to the same axis. If you are wondering how the labels were made to appear at the beginning and end of each line, explore the Label button on the Marks card.

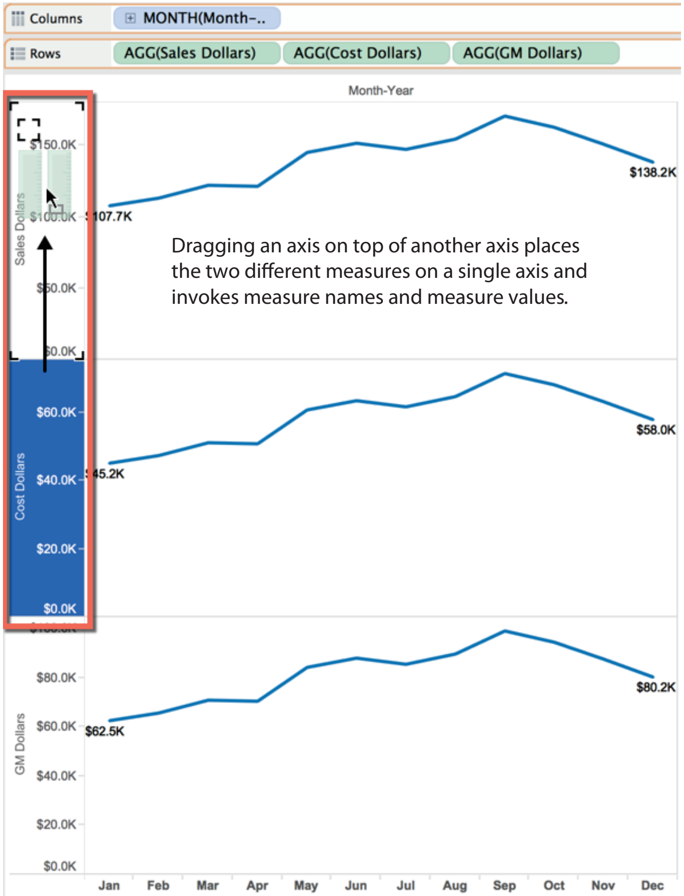


FIGURE 3-74 *Dragging to combine axes*

Measure names and measure values help you contend with poorly formatted data and facilitate creating more advanced chart types. Many beginning and intermediate Tableau users avoid these specific fields because they are unfamiliar. Spend a little time working with them. The added flexibility you gain will be worth the effort.

You've seen all the ways that Tableau's built-in chart types, trend lines, references lines, groups, sets, sorting, and filtering help you add meaning to your data within Tableau. In the next chapter, you learn how calculations provide even more flexible ways to add information to your charts and dashboards.

NOTE

1. Karl R. Popper, *Conjectures and Refutations: The Growth of Scientific Knowledge* (London: Routledge, 1989), p. 51.

CHAPTER 4

Creating Calculations to Enhance Data

Each new paradigm gives us the opportunity to "see" phenomena that were before as invisible to us as the colors of the sunset to the fog.

—BENJAMIN AND ROSAMUND ZANDER[1]

Tableau provides different methods for making calculations to enhance your data through the creation of new fields that don't exist in your data source. You can also turn single-purpose dashboards and views into multi-purpose analysis environments though the use of parameter controls. Parameters are formula variables that can be used to provide filter-like controls that allow users to change the measures and dimensions used in a dashboard or worksheet. Starting with Tableau V9.0, an entirely new class of functions was added that eases control over the granularity of the data expressed in your views. Level of detail (LOD) expressions lower the technical bar for achieving effects that formerly required advanced SQL-scripting skills.

In this chapter, you will learn how to use Calculated Fields and Table Calculations to derive facts and dimensions that don't exist in your source data. Tableau's formula editing window will be explained as well as the Quick Table Calculation menu and how to modify the calculation defaults to address your specific needs.

You will learn about parameter controls—basic and advanced—so that you can make views that address different needs using the same basic visual design. Level of detail (LOD) expressions are covered at the end of the chapter.

Tableau makes formula creation as easy as it can possibly be, but it helps to understand the concept of aggregation and the functions and operators that are available to use before you start making formulas. For those of you who want to dive deeply into Tableau's functions, Appendix E provides in-depth coverage of every Tableau function and level of detail expressions, with basic, intermediate, and advanced examples.

WHAT IS AGGREGATION?

Aggregation defines how values are expressed. Most Tableau functions are calculated at the database server with only the results being sent to Tableau. If you are familiar with SQL, you will find many of the functions in Tableau are an extension of SQL. Tableau uses the Sum aggregation by default. If the default aggregation isn't what you want, you can change it per view or change the default aggregation for the field. To change it in a view, point at the pill of the measure you've placed into the view—right-click, and select a more appropriate aggregation. To change the default settings, right-click the field in the Measures pane and select Defaults Properties ➢ Aggregation ➢ {select a new default format}. Supported aggregation types include:

- Sum
- Average
- Median
- Count
- Count Distinct
- Minimum
- Maximum
- Percentiles
- Standard Deviation
- Standard Deviation of a Population
- Variance
- Variance of a Population

These are defined in Tableau's online manual. Search the help menu to read more about each of them. Note that in older versions of Tableau, Desktop Count Distinct and Median aggregation types were not supported when using a direct connection to Excel, Access, or text files. The data had to be extracted into Tableau's data engine. Beginning in Tableau V9.0, extracts are no longer required.

Figure 4-1 shows a text tables in the Sample – Superstore Sales dataset, displaying all of the different aggregations available for the sales field in the dataset.

		Measure Values
Avg. Sales	949.71	AVG(Sales)
Median Sales	203.46	MEDIAN(Sales)
Count of Sales	9,426.00	CNT(Sales)
Distinct count of Sales	8,674.00	CNTD(Sales)
Min. Sales	1.32	MIN(Sales)
Max. Sales	100,119.16	MAX(Sales)
Percentile (5) of Sales	14.52	PCT5(Sales)
Percentile (10) of Sales	24.41	PCT10(Sales)
Percentile (25) of Sales	61.28	PCT25(Sales)
Percentile (50) of Sales	203.46	PCT50(Sales)
Percentile (75) of Sales	776.40	PCT75(Sales)
Percentile (90) of Sales	2,323.65	PCT90(Sales)
Percentile (95) of Sales	4,209.38	PCT95(Sales)
Std. dev. of Sales	2,598.02	STDEV(Sales)
Population std. dev. of Sales	2,597.88	STDEVP(Sales)
Variance of Sales	6,749,706.98	VAR(Sales)
Population variance of Sales	6,748,990.90	VARP(Sales)
Distinct count of Order ID	6,455	CNTD(Order ID)
Distinct count of City	1,424	CNTD(City)
Distinct count of State or Province	49	CNTD(State or Pro..
Distinct count of Region	4	CNTD(Region)

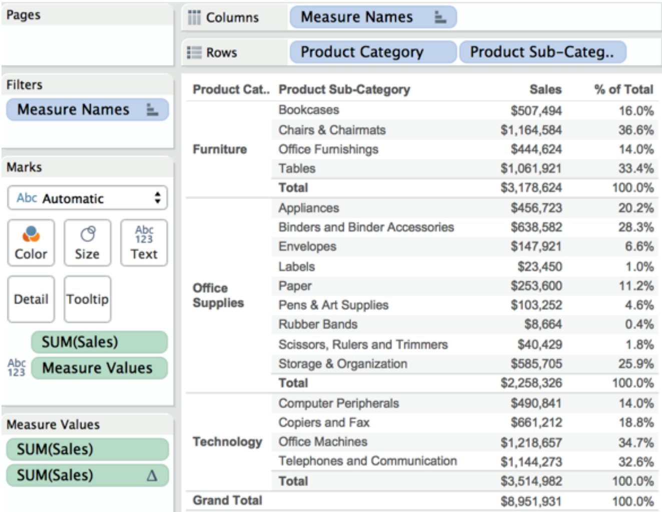
FIGURE 4-1 *Different aggregation of sales*

Notice that the bottom four rows are expressing Count Distinct values for different dimensions. By dragging each of those dimension fields into the text table using the right mouse button (Option+drag a Mac), the Count Distinct aggregation can be expressed for each dimension. As you can see, the dataset includes 6,455 different orders, 1,424 cities, 49 states, and 4 regions. This view was created using a direct connection to an Excel spreadsheet.

DIMENSION VERSUS ATTRIBUTE

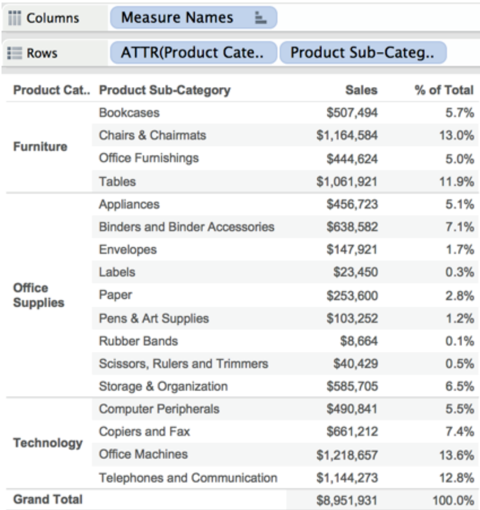
Aggregation behavior can be changed by altering the default method by which Tableau expresses dimensions. Figure 4-2 shows a text table containing sales by product category and subcategory. A table calculation is being used to display the percent of total sales that each row represents within each product category pane.

By default, Tableau partitions the result by the category dimension. Subtotals have been added by using the main menu option Analysis ➢ Totals and then showing subtotal and column totals. The amount of sales and percent of sales are totaled within each category pane. But, if the category dimension is changed to an attribute, the category dimension will become a label only and no longer cause the data to be partitioned. Figure 4-3 shows the same dataset but with the category field changed to an attribute.



Product Cat..	Product Sub-Category	Sales	% of Total
Furniture	Bookcases	$507,494	16.0%
	Chairs & Chairmats	$1,164,584	36.6%
	Office Furnishings	$444,624	14.0%
	Tables	$1,061,921	33.4%
	Total	$3,178,624	100.0%
Office Supplies	Appliances	$456,723	20.2%
	Binders and Binder Accessories	$638,582	28.3%
	Envelopes	$147,921	6.6%
	Labels	$23,450	1.0%
	Paper	$253,600	11.2%
	Pens & Art Supplies	$103,252	4.6%
	Rubber Bands	$8,664	0.4%
	Scissors, Rulers and Trimmers	$40,429	1.8%
	Storage & Organization	$585,705	25.9%
	Total	$2,258,326	100.0%
Technology	Computer Peripherals	$490,841	14.0%
	Copiers and Fax	$661,212	18.8%
	Office Machines	$1,218,657	34.7%
	Telephones and Communication	$1,144,273	32.6%
	Total	$3,514,982	100.0%
Grand Total		$8,951,931	100.0%

FIGURE 4-2 *Product category as a dimension*



Product Cat..	Product Sub-Category	Sales	% of Total
Furniture	Bookcases	$507,494	5.7%
	Chairs & Chairmats	$1,164,584	13.0%
	Office Furnishings	$444,624	5.0%
	Tables	$1,061,921	11.9%
Office Supplies	Appliances	$456,723	5.1%
	Binders and Binder Accessories	$638,582	7.1%
	Envelopes	$147,921	1.7%
	Labels	$23,450	0.3%
	Paper	$253,600	2.8%
	Pens & Art Supplies	$103,252	1.2%
	Rubber Bands	$8,664	0.1%
	Scissors, Rulers and Trimmers	$40,429	0.5%
	Storage & Organization	$585,705	6.5%
Technology	Computer Peripherals	$490,841	5.5%
	Copiers and Fax	$661,212	7.4%
	Office Machines	$1,218,657	13.6%
	Telephones and Communication	$1,144,273	12.8%
Grand Total		$8,951,931	100.0%

FIGURE 4-3 *Product category as an attribute*

The view still shows the light gray boundary lines between each category, but because the category dimension has been changed to an attribute, it no longer partitions the view. The sales total reflects the total for the entire text table and the percent of total sales is now expressing the percentage of total sales, not the sales within each category. This may appear to be trivial, but as your skills advance and you begin to employ more advanced table calculations, you will need to understand how attributes change Tableau's behavior.

WHAT ARE CALCULATED FIELDS AND TABLE CALCULATIONS?

Calculated Fields and Table Calculations allow you to add new data to your Tableau workbook, but the way you add the data, and where the calculations occur, is different for each method.

Calculated Fields are defined by entering a formula into Tableau's formula editing dialog box. For example, if you have gross margin dollars and sales dollars in your source data, you may want to add a new field called Gross Margin Percent by creating a calculated value. The formula to create the gross margin percent is `sum([gross margin dollars])/sum([sales dollars])`.

The Sum aggregation function in front of each field name tells the source database what to return to Tableau. Table calculations are created in a different way—using your data visualization as the source for the formula.

Predefined Quick Table Calculations remove the need for you to create the formula manually, but these are always processed locally because they rely on the data presented in your view to derive the result.

Calculated Fields can also include Table Calculation functions. These are functions you use in Calculated Fields that are processed locally just like Quick Table calculations.

HOW DO CALCULATED FIELDS WORK?

Calculated Fields are normally (but not always) executed at the database level, where the heavy-lifting happens is dependent on the type of functions utilized in the formula. Calculated Fields can be used to generate numbers, dates, date-times, strings, or Boolean (true/false) conditions. Formulas need these elements:

- **Functions:** Including aggregate, number, string, date, type conversion, logical, user, and table calculation types
- **Fields:** Selected from the data source
- **Operators:** For math and comparison of values, dates, and text
- **Optional elements:** Can be added within the formula dialog box including:
    - **Parameters:** For creating formula variables that are accessible to information consumers
    - **Comments:** For documenting formula syntax and notes within the formula dialog box

There are two ways to create a Calculated Field: by initiating a formula dialog or via ad hoc calculations. People experienced at writing SQL script or creating spreadsheet formulas normally have very little difficultly learning how to write formulas in Tableau. Those with very little experience writing formulas may need more help. Tableau provides assistance via a real-time Calculation Editor and a help window in the formula editing window, as well as an online manual that is accessible from the editing window.

CREATING CALCULATED FIELDS WITH THE CALCULATION EDITOR

Start the formula dialog box via the main menu using the Analysis ➢ Created Calculated Field or by right-clicking on a field. In the formula dialog box, you enter the functions, operators, and parameters to create the logic for your formula. Figure 4-4 shows the required menu options.

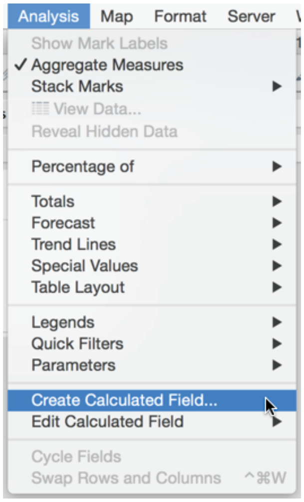


FIGURE 4-4 *Menu for creating Calculated Fields*

Alternatively, right-clicking a field in the Dimensions or Measures shelves opens the formula dialog box you see in Figure 4-5.

This alternative saves a little time because Tableau will automatically insert the field into the window.

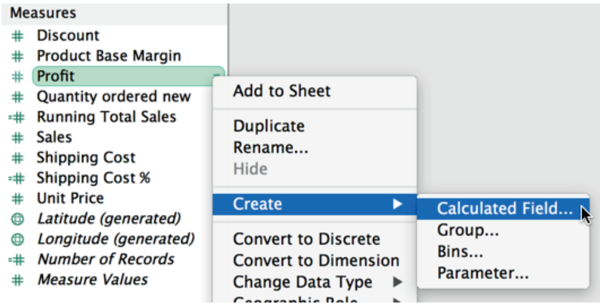


FIGURE 4-5 *Right-click to create a Calculated Field.*

PERFORMING AD HOC CALCULATIONS

Ad hoc calculations are started by double-clicking within a shelf and typing as you see in Figure 4-6.

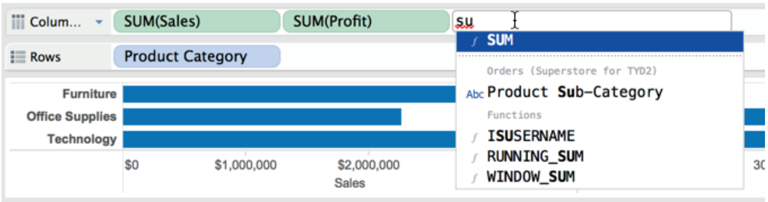


FIGURE 4-6 *Ad hoc calculation*

Initiating the process of writing a formula this way is similar to writing a formula in a spreadsheet. This new Tableau V9.0 facility keeps you in the flow of your analysis.

HOW DO TABLE CALCULATIONS WORK?

Table calculations are derived from the structure of the data included in your visualization, so table calculations are dependent on the source worksheet view contained in your workbook. That means these calculations are always derived locally using your personal computer's processor to return the result.

Understanding exactly how table calculations work takes a little time because table calculations can change as your visualization is altered. As with any new concept, after you create some table calculations, you'll get comfortable with how they behave in different situations. Tableau's online manual has a large number of examples that you can view that provide a good basic introduction.

Creating a Table Calculation requires that you have a worksheet with a visualization. A good way to create them is to right-click a measure pill used in the view to expose the Quick Table Calculation menu. Quick Table Calculations are provided for:

- Running total
- Difference
- Percent difference
- Percent of total
- Rank
- Percentile
- Moving average
- YTD total
- Compound growth rate
- Year over year growth
- YTD growth

Depending on the view of the data included in your worksheet, some of these may be unavailable because your worksheet view doesn't support the calculation. Unavailable calculations will be visible in the menu but will appear grayed out.

A WORD ON CALCULATIONS AND CUBES

Tableau connects to relational databases, spreadsheets, columnar-analytic databases, data services, and data cubes (multi-dimensional data sources). Data cubes are different from regular database files because they pre-aggregate data and define hierarchies of dimensions in specific ways.

If you need to access pre-aggregated data that is stored in a multi-dimensional data source, you can still perform calculations using Tableau formulas or create formulas using the standard query language of multi-dimensional databases, Multidimensional Expressions (MDX). The syntax is a bit more complex, but MDX also provides the ability to create more complex formulas. If you want to learn more about options for creating calculations when accessing data cubes, refer to Tableau Software's quick-start guide "Creating Calculated Fields-Cubes." Tableau's behavior when you connect it do a data cube is different because the cube controls aggregation. For example, date fields behave differently because the cube controls date aggregation in specific ways.

USING THE CALCULATION EDITOR TO BUILD CALCULATED FIELDS

Calculated Fields require that you enter fields, functions, and operators. Tableau strives to make formula creation fast and easy, so it is possible to write formulas with minimal typing. Once you've connected to a data source, you can create a Calculated Field from the main menu by selecting Analysis/Create Calculated Field. This example uses the Superstore for TYD2 spreadsheet. Figure 4-7 shows the Calculation Editor.

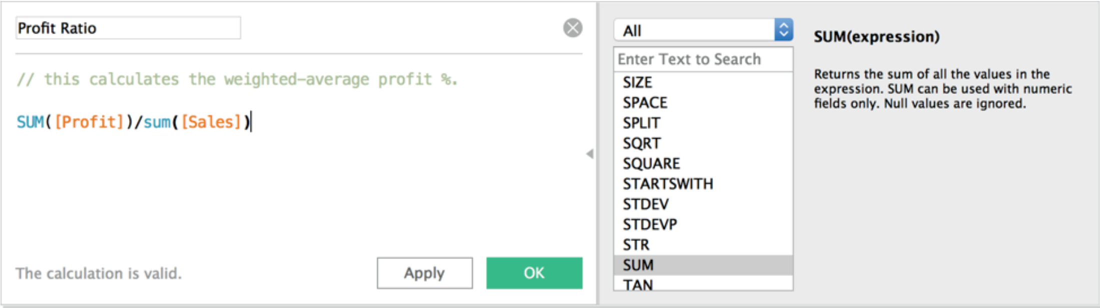


FIGURE 4-7 *Calculation Editor*

The figure shows a calculation for Profit Ratio that uses two fields from the Superstore file to derive the result. The Name field at the top of Figure 4-7 is where you type the name of your Calculated Field as you want it to appear in the date pane of the worksheet. The Formula box is used to write the script for the formula. After creating a Calculated Field, right-click it and set the default properties. In this example, you want the calculated ratio to be treated as a percentage, so select Number Format ➢ Percentage ➢ OK.

This Calculation Editor marks a significant departure from earlier versions of Tableau that included separate dialog boxes for fields, parameters, and functions. Tableau's new autofill feature eliminates the need for the dialog boxes. You simply start to type the name of the desired element and Tableau's autofill presents the available options. Or, drag fields from the data pane or a shelf into the view Calculation Editor. This method for building Calculated Fields more closely resembles how you build visualizations in Tableau Desktop by keeping you in the flow of your build.

Observe that Tableau color-encodes different elements of formulas so that they are easy to separate visually. Fields are orange, parameters are purple, and functions are blue. Notice the example in Figure 4-7 includes comments at the top, color-encoded in green. Comments are useful for documenting sections of complex formulas or for adding basic descriptive information to other analysts that may use your formula in their work. You can add comments anywhere in the Calculation Editor by typing two forward slashes (//) in front of the text.

If your formula syntax is correct, you will see gray text at the bottom of the window that confirms that the calculation is valid, as you see in Figure 4-8. If you make a mistake, the Calculation Editor will help you find the cause.

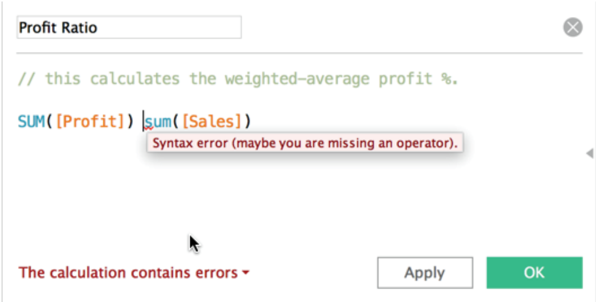


FIGURE 4-8 *Detecting formula errors*

Hover over the squiggly red line in the formula or click the drop-down box at the bottom to see the details. Forgetting to add an operator is a common mistake. Figure 4-8 shows how Tableau helps you identify the missing syntax from your formula.

AD HOC CALCULATED FIELDS

Creating formulas using Tableau's ad hoc formula editor is similar to building formulas in a spreadsheet. To build a formula this way, double-click the Rows shelf, Columns shelf, or Marks card. Figure 4-9 shows an ad hoc formula for profit ratio.

As you become more familiar with Tableau functions, you will probably gravitate toward ad hoc calculations because it's the fastest way to create a formula. If you decide that you want to reuse an ad hoc calculation, drag its pill to the data pane and name it, as you see in Figure 4-10.

Dropping the pill on the data pane allows you to save the new field permanently, name the field specifically, and edit the default format or data type for the field. Figure 4-10 shows the Rename Field dialog box. Edit the datatype or default number format as you would any field in the data pane.

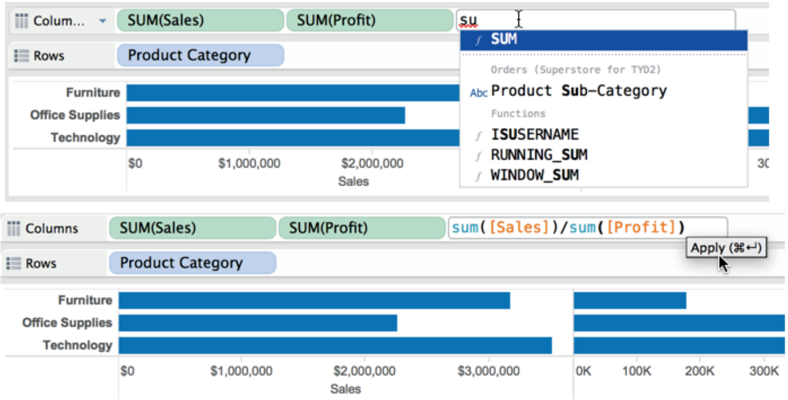


FIGURE 4-9 *Ad hoc Calculation Editor*

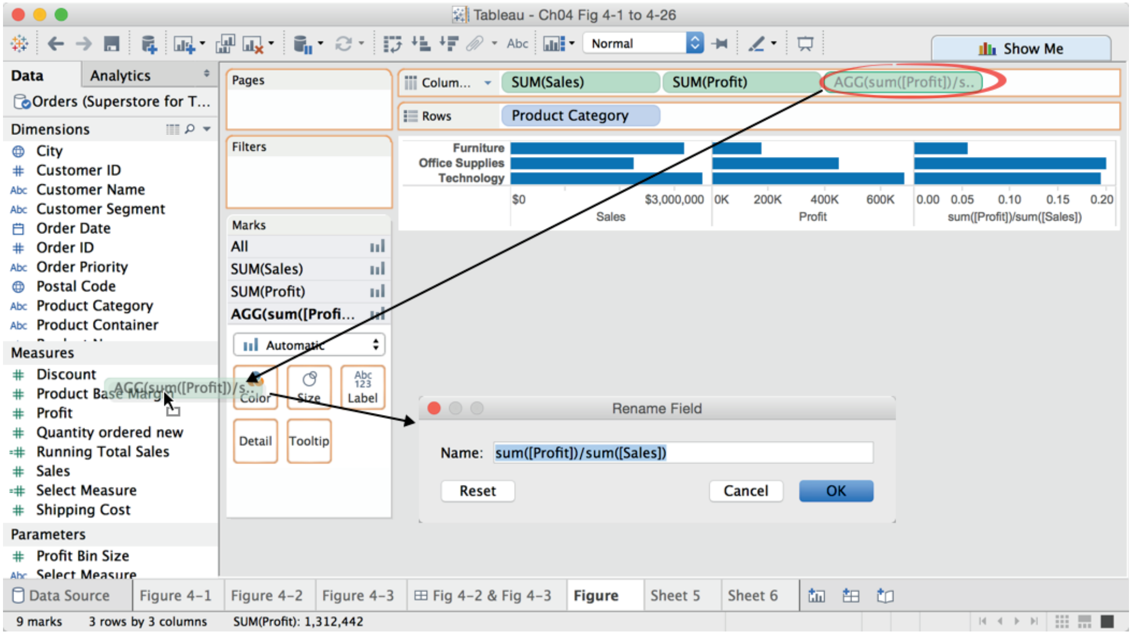


FIGURE 4-10 *Naming an ad-hoc calculation*

As you are adding new Calculated Fields, using the ad hoc calculations, you can see the result of your work before saving it by hovering your cursor over the pill. This will make the small Apply window you see in Figure 4-11 appear.

Selecting the Apply option allows you to preview the result of the Calculated Field in the view as you see in the view area of Figure 4-11. Ad hoc calculations accelerate your ability to create and test new formulas.

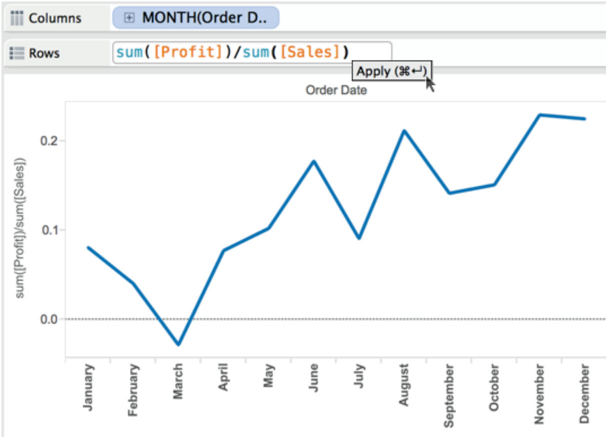


FIGURE 4-11 *Ad hoc calculation preview*

BUILDING FORMULAS USING TABLE CALCULATIONS

In contrast to Calculated Fields, Table Calculations use the data in your visualization to create a formula. Before you can use Table Calculations, you must first create a view. Using Superstore for TYD2, Figure 4-12 displays a time series of monthly sales on top. The bottom half employs a quick table calculation to derive the running total of sales for each year. Notice the second SUM(Sales) pill contains a small triangle. This icon denotes the pill is a table calculation.

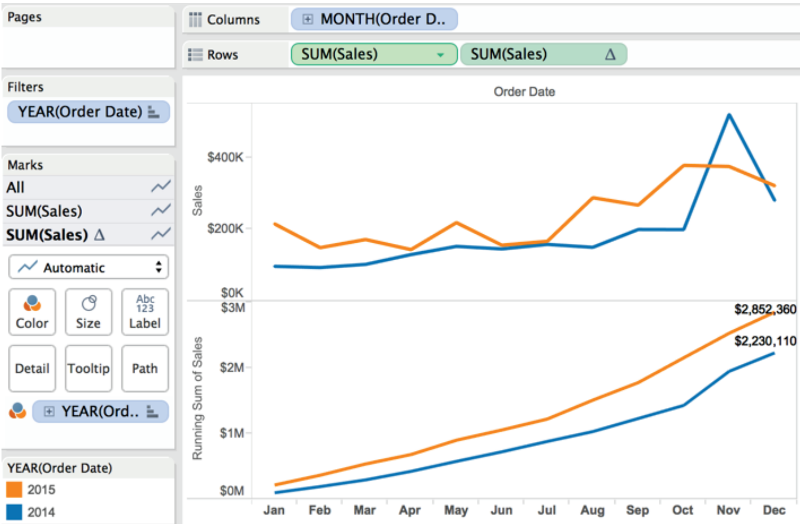


FIGURE 4-12 *Time series using a running sum*

The following are the steps required to build the charts in Figure 4-12:

1. Add Order Date (discrete month) to the Columns shelf.
2. Add sales to the Rows shelf.
3. Filter the order date for the year(s) 2014 and 2015.
4. Add the order date to the Color Marks button.
5. Format the lower axis selecting Dates: Abbreviation.

The data from the time-series chart will serve as the data source for the table calculation that will be used to create the chart in the bottom half of Figure 4-12. That chart displays the running sum of sales for each month within the displayed years. The following are the steps required to add that portion of the view:

1. Duplicate the sales pill on the Rows shelf to create a duplicate chart (Ctrl+drag and drop).
2. Right-click (or pick the drop-down arrow on the right side of the second sales pill).
3. Select Quick Table Calculation—Running Total.
4. On the Marks card, click the bottom Sum(Sales) to modify that specific chart.
5. Turn on field labels for the Line Ends and un-check label start of line.

Figure 4-13 shows how right-clicking the duplicate sales pill exposes the Quick Table Calculation menu.

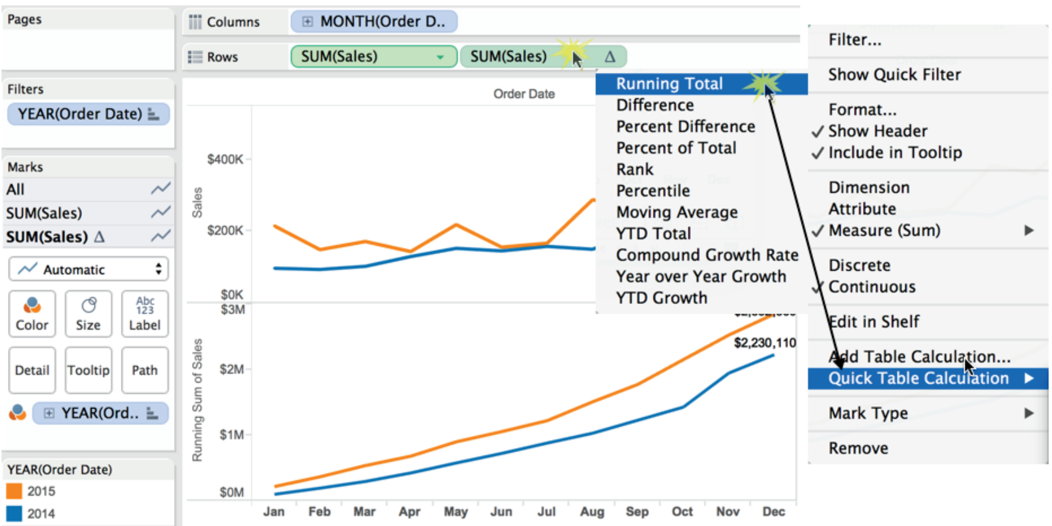


FIGURE 4-13 *Creating the Quick Table Calculation*

Selecting Running Total generates the table calculation that is plotted in the lower half of the view in Figure 4-13. The number format for the mark label was also changed to display the results in thousands in the top chart and millions in the lower chart with the decimal places set to zero for both. To format the numbers, point at the number, right-click, and select the desired number format from the formatting menu that appears. Only 30 seconds were required to build this chart.

Editing Table Calculations to Suit Your Purpose

You can also see in Figure 4-13 that there are many other Quick Table Calculation options available. There is also a menu option called Edit Table Calculation. You can also customize table calculations by selecting the Edit Table Calculation menu as well.

Understanding how Table Calculations work takes a little time—playing with the options and looking at the results. Take a close look at the Table Calculation dialog box displayed in Figure 4-14.

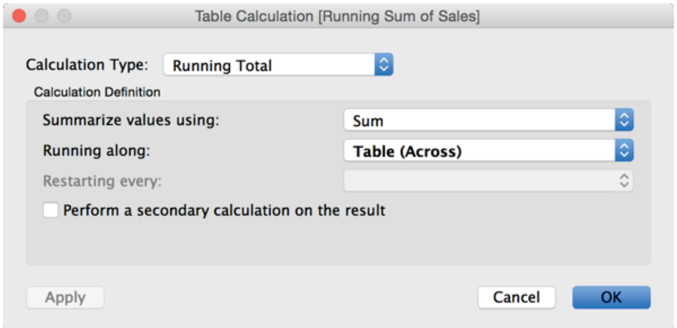


FIGURE 4-14 *The Table Calculation dialog*

Table Calculations require selections of following options:

- **Calculation Type:** As shown in Figure 4-13
- **Summarize values using:** Sum, average, median (these will change depending on the content of your source)
- **Running along:** Defines the direction and the scope of the table calculation (Table Across, Table Down, and so on)

Modifying the Month (Order Date) field to show time as discrete quarters and months creates quarterly partitions, as shown in Figure 4-15.

FIGURE 4-15 *Using discrete quarter and month*

The bottom time series showing the Running Sum of Sales still uses Table Across to calculate the total. Notice the labels at the end of each quarter reflect the running total sales for the entire table. Right-clicking on the table calculation pill (identified by a small triangle on the right side of the pill) and selecting the Edit Table Calculation menu exposes the Table Calculation control. Figure 4-16 shows the Table Calculation editing menu for Compute Using Pane (Across) and includes more options. Adding the partition for quarter creates quarterly panes that revise the scope of the table calculation used in the bottom half of the view.

Changing the scope of the calculation to Pane Across causes the running sum calculation to reset every quarter (pane). The bottom half of Figure 4-16 reflects the revised scope. As you see, the running totals restart at the beginning of each quarter.

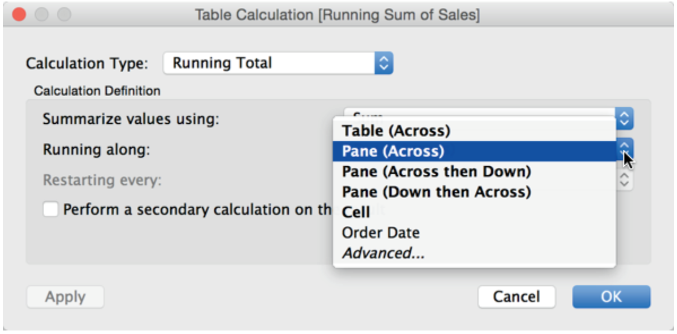


FIGURE 4-16 *Changing table calculation scope*

Understanding Table Calculation Behavior

Learning exactly how Table Calculations behave in different visualizations takes a little time. The best way to learn is to build a text table report and then start playing with different tableau calculation options to see the results. Tableau's online manual provides many different examples. Figure 4-17 shows the percent of total table calculations using all of the different direction and scope options.

Notice that the example for the Table scope returns the same result as the Table Down Then Across scope. Also, the Cell scope is calculating the mark value of itself, resulting in 100 percent in every cell. Depending on the structure of your view, it is not uncommon for different scope options to return the same values. In general, adding more dimensions to your view will increase the number of available options provided by table calculations. Experiment with different visualizations and test your results. With practice, you'll be able to anticipate how they behave.

Reusing and Customizing Table Calculations

Quick table calculations don't automatically result in any new fields appearing in the data pane. What if you want to use the result returned by a Quick Table Calculation on another worksheet? Is this possible? It is. The method for doing this has changed. Prior to V9.0, the table calculation editing window contained a Customize button. Clicking that created a new field and generated script containing table calculation functions.

Starting in V9.0, the Calculation Editor no longer includes this button because the new ad hoc calculation facility allows you to drag and drop table calculations from your view to the data pane. When you do that, Tableau will create a new Calculated Field that you can rename.

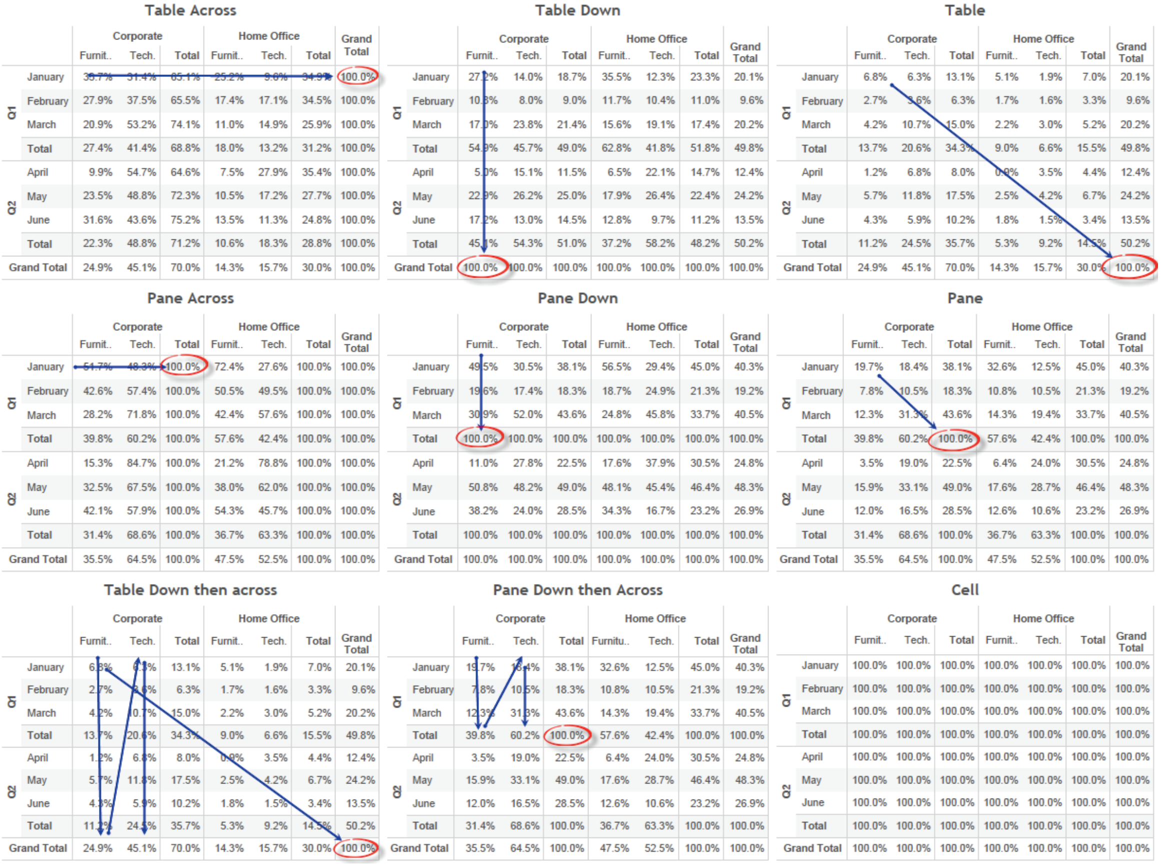


Table Across

		Corporate			Home Office			Grand Total
		Furnit..	Tech.	Total	Furnit..	Tech.	Total	
Q1	January	33.7%	31.4%	65.1%	25.2%	9.6%	34.9%	100.0%
	February	27.9%	37.5%	65.5%	17.4%	17.1%	34.5%	100.0%
	March	20.9%	53.2%	74.1%	11.0%	14.9%	25.9%	100.0%
	Total	27.4%	41.4%	68.8%	18.0%	13.2%	31.2%	100.0%
Q2	April	9.9%	54.7%	64.6%	7.5%	27.9%	35.4%	100.0%
	May	23.5%	48.8%	72.3%	10.5%	17.2%	27.7%	100.0%
	June	31.6%	43.6%	75.2%	13.5%	11.3%	24.8%	100.0%
	Total	22.3%	48.8%	71.2%	10.6%	18.3%	28.8%	100.0%
Grand Total		24.9%	45.1%	70.0%	14.3%	15.7%	30.0%	100.0%

Table Down

		Corporate			Home Office			Grand Total
		Furnit..	Tech.	Total	Furnit..	Tech.	Total	
Q1	January	27.2%	14.0%	18.7%	35.5%	12.3%	23.3%	20.1%
	February	10.8%	8.0%	9.0%	11.7%	10.4%	11.0%	9.6%
	March	17.0%	23.8%	21.4%	15.6%	19.1%	17.4%	20.2%
	Total	54.9%	45.7%	49.0%	62.8%	41.8%	51.8%	49.8%
Q2	April	5.0%	15.1%	11.5%	6.5%	22.1%	14.7%	12.4%
	May	22.9%	26.2%	25.0%	17.9%	26.4%	22.4%	24.2%
	June	17.2%	13.0%	14.5%	12.8%	9.7%	11.2%	13.5%
	Total	45.1%	54.3%	51.0%	37.2%	58.2%	48.2%	50.2%
Grand Total		100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%

Table

		Corporate			Home Office			Grand Total
		Furnit..	Tech.	Total	Furnit..	Tech.	Total	
Q1	January	6.8%	6.3%	13.1%	5.1%	1.9%	7.0%	20.1%
	February	2.7%	3.6%	6.3%	1.7%	1.6%	3.3%	9.6%
	March	4.2%	10.7%	15.0%	2.2%	3.0%	5.2%	20.2%
	Total	13.7%	20.6%	34.3%	9.0%	6.6%	15.5%	49.8%
Q2	April	1.2%	6.8%	8.0%	0.9%	3.5%	4.4%	12.4%
	May	5.7%	11.8%	17.5%	2.5%	4.2%	6.7%	24.2%
	June	4.3%	5.9%	10.2%	1.8%	1.5%	3.4%	13.5%
	Total	11.2%	24.5%	35.7%	5.3%	9.2%	14.5%	50.2%
Grand Total		24.9%	45.1%	70.0%	14.3%	15.7%	30.0%	100.0%

Pane Across

		Corporate			Home Office			Grand Total
		Furnit..	Tech.	Total	Furnit..	Tech.	Total	
Q1	January	51.7%	48.3%	100.0%	72.4%	27.6%	100.0%	100.0%
	February	42.6%	57.4%	100.0%	50.5%	49.5%	100.0%	100.0%
	March	28.2%	71.8%	100.0%	42.4%	57.6%	100.0%	100.0%
	Total	39.8%	60.2%	100.0%	57.6%	42.4%	100.0%	100.0%
Q2	April	15.3%	84.7%	100.0%	21.2%	78.8%	100.0%	100.0%
	May	32.5%	67.5%	100.0%	38.0%	62.0%	100.0%	100.0%
	June	42.1%	57.9%	100.0%	54.3%	45.7%	100.0%	100.0%
	Total	31.4%	68.6%	100.0%	36.7%	63.3%	100.0%	100.0%
Grand Total		35.5%	64.5%	100.0%	47.5%	52.5%	100.0%	100.0%

Pane Down

		Corporate			Home Office			Grand Total
		Furnit..	Tech.	Total	Furnit..	Tech.	Total	
Q1	January	49.5%	30.5%	38.1%	56.5%	29.4%	45.0%	40.3%
	February	19.6%	17.4%	18.3%	18.7%	24.9%	21.3%	19.2%
	March	30.9%	52.0%	43.6%	24.8%	45.8%	33.7%	40.5%
	Total	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
Q2	April	11.0%	27.8%	22.5%	17.6%	37.9%	30.5%	24.8%
	May	50.8%	48.2%	49.0%	48.1%	45.4%	46.4%	48.3%
	June	38.2%	24.0%	28.5%	34.3%	16.7%	23.2%	26.9%
	Total	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
Grand Total		100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%

Pane

		Corporate			Home Office			Grand Total
		Furnit..	Tech.	Total	Furnit..	Tech.	Total	
Q1	January	19.7%	18.4%	38.1%	32.6%	12.5%	45.0%	40.3%
	February	7.8%	10.5%	18.3%	10.8%	10.5%	21.3%	19.2%
	March	12.3%	31.3%	43.6%	14.3%	19.4%	33.7%	40.5%
	Total	39.8%	60.2%	100.0%	57.6%	42.4%	100.0%	100.0%
Q2	April	3.5%	19.0%	22.5%	6.4%	24.0%	30.5%	24.8%
	May	15.9%	33.1%	49.0%	17.6%	28.7%	46.4%	48.3%
	June	12.0%	16.5%	28.5%	12.6%	10.6%	23.2%	26.9%
	Total	31.4%	68.6%	100.0%	36.7%	63.3%	100.0%	100.0%
Grand Total		35.5%	64.5%	100.0%	47.5%	52.5%	100.0%	100.0%

Table Down then across

		Corporate			Home Office			Grand Total
		Furnit..	Tech.	Total	Furnit..	Tech.	Total	
Q1	January	6.8%	6.3%	13.1%	5.1%	1.9%	7.0%	20.1%
	February	2.7%	3.6%	6.3%	1.7%	1.6%	3.3%	9.6%
	March	4.2%	10.7%	15.0%	2.2%	3.0%	5.2%	20.2%
	Total	13.7%	20.6%	34.3%	9.0%	6.6%	15.5%	49.8%
Q2	April	1.2%	6.8%	8.0%	0.9%	3.5%	4.4%	12.4%
	May	5.7%	11.8%	17.5%	2.5%	4.2%	6.7%	24.2%
	June	4.3%	5.9%	10.2%	1.8%	1.5%	3.4%	13.5%
	Total	11.2%	24.5%	35.7%	5.3%	9.2%	14.5%	50.2%
Grand Total		24.9%	45.1%	70.0%	14.3%	15.7%	30.0%	100.0%

Pane Down then Across

		Corporate			Home Office			Grand Total
		Furnit..	Tech.	Total	Furnitu..	Tech.	Total	
Q1	January	19.7%	18.4%	38.1%	32.6%	12.5%	45.0%	40.3%
	February	7.8%	10.5%	18.3%	10.8%	10.5%	21.3%	19.2%
	March	12.3%	31.3%	43.6%	14.3%	19.4%	33.7%	40.5%
	Total	39.8%	60.2%	100.0%	57.6%	42.4%	100.0%	100.0%
Q2	April	3.5%	19.0%	22.5%	6.4%	24.0%	30.5%	24.8%
	May	15.9%	33.1%	49.0%	17.6%	28.7%	46.4%	48.3%
	June	12.0%	16.5%	28.5%	12.6%	10.6%	23.2%	26.9%
	Total	31.4%	68.6%	100.0%	36.7%	63.3%	100.0%	100.0%
Grand Total		35.5%	64.5%	100.0%	47.5%	52.5%	100.0%	100.0%

Cell

		Corporate			Home Office			Grand Total
		Furnit..	Tech.	Total	Furnit..	Tech.	Total	
Q1	January	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
	February	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
	March	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
	Total	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
Q2	April	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
	May	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
	June	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
	Total	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
Grand Total		100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%

FIGURE 4-17 *Comparison of different scope options*

Refer to Figure 4-12. In that view a quick table calculation was used to make a Running Sum of Sales chart at the bottom half of the view. Dragging that Table Calculation from the row shelf to the data pane, as you see in Figure 4-18, allows you to add the field to the Measures pane. Notice the field is now named Running Sum Table Calc.

Once you've created the new field (Running Sum of Sales), you can use it in another worksheet. Figure 4-19 shows one way that it could be deployed. This new view contains year and month date granularity. Note that the table calculation in the view resets at the beginning of each year. To make this view requires that the Calculated Field—Running Sum of Sales—be revised to compute the value using Pane Across versus Table Across. Figure 4-19 shows you how.

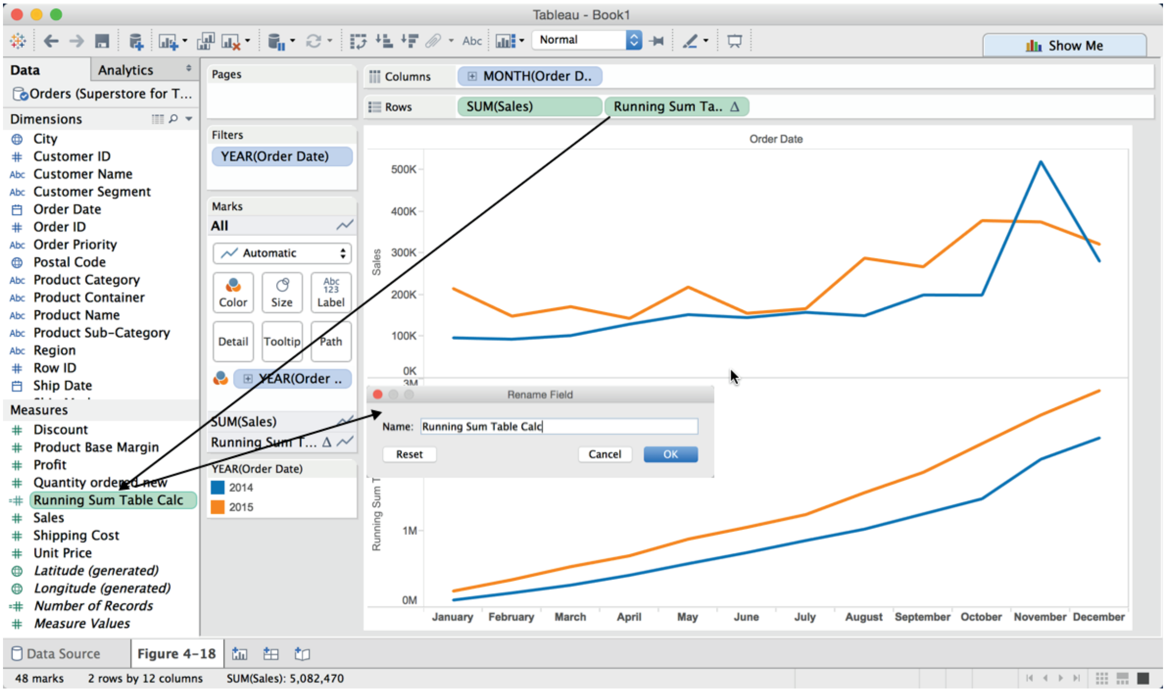


FIGURE 4-18 *Creating a Calculated Field from a table calculation*

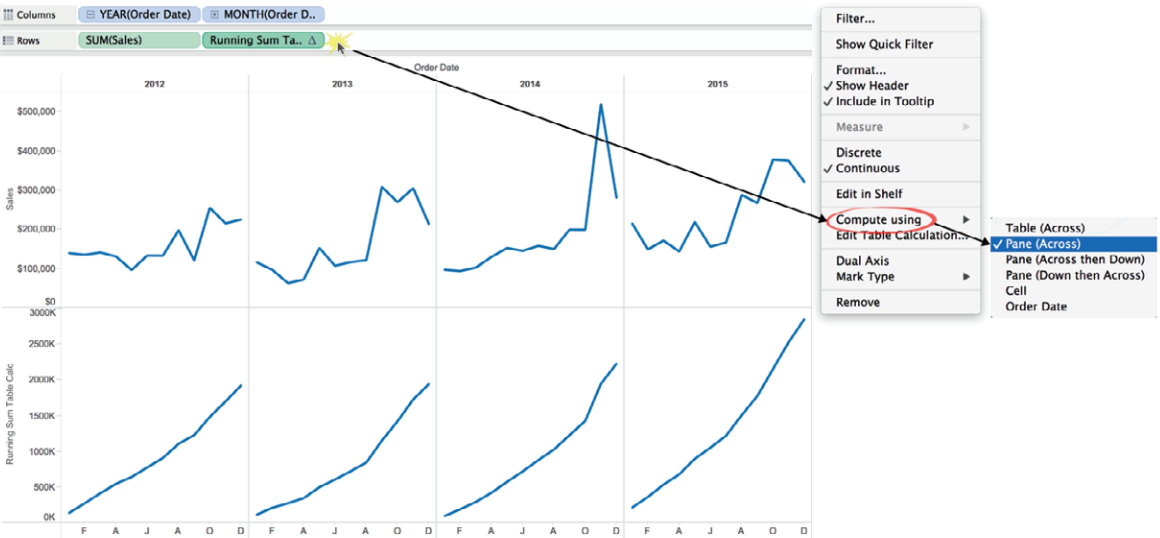


FIGURE 4-19 *Using the custom table calculation in another worksheet*

To build this view, place the order date on the Columns shelf, and then click the plus sign in the Year (order date) pill. After that, modify the Quarter (order date) pill to display the month by selecting the drop-down and picking the Discrete month option. Next, place Sum (Sales) and the running sum Calculated Field on the Rows shelf. Customize the running sum to compute using the Pane Across option by selecting the drop-down arrow in the pill (or right-clicking) and picking the desired option indicated in Figure 4-19. Format the bottom axis by right-clicking it and selecting Format ➢ Header ➢ Dates: ➢ First letter.

Secondary Table Calculations

Secondary table calculations allow you to pass the result of an initial table calculation to a second table calculation to derive a result. In the next example, you use the World Indicators dataset that ships with Tableau desktop as your data source. That file includes country Gross Domestic Product (GDP) data. You'll see how a secondary table calculation can be used to enhance analysis of GDP by country.

An initial view of the data shows the GDP information in a bar chart, sorted by descending GDP. You can see the result in Figure 4-20.

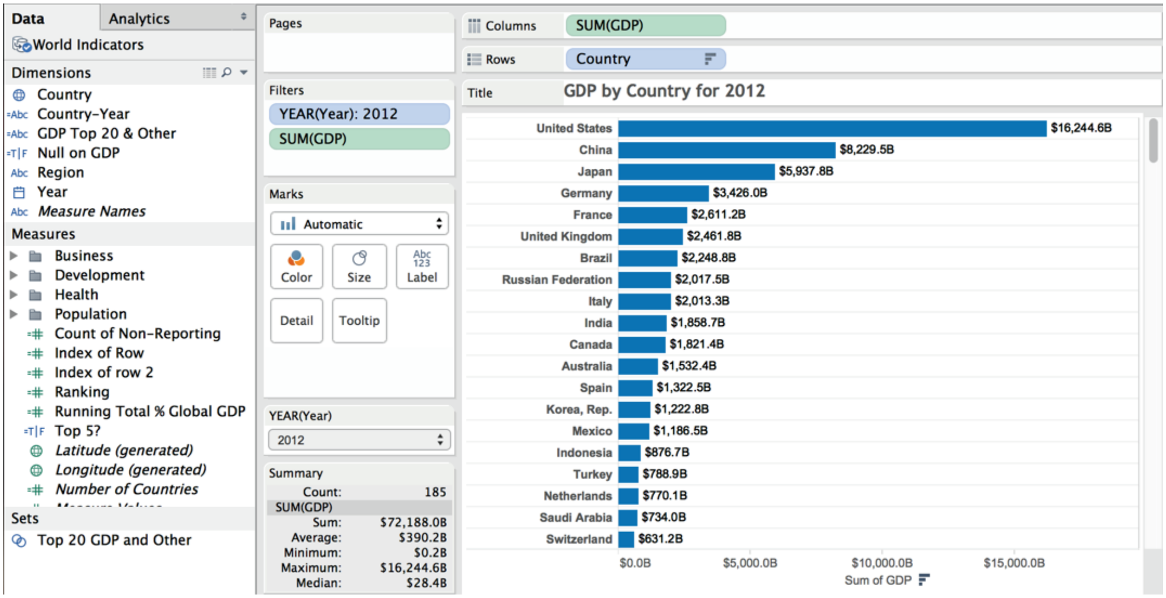


FIGURE 4-20 *GDP by country*

Looking at Figure 4-20, you get a quick sense of the top countries in terms of GDP. Notice that the view is filtered for the year 2012 and that the dataset includes 185 records. The total reported GDP for all reporting countries is $72,188.08B ($72 trillion). How could this be enhanced using table calculations?

- Create a table calculation that computes the running total GDP by country and another that computes the running cumulative percent (%) of global GDP.
- Add an INDEX table calculation function to show a ranking of the row.
- Make a quick filter that utilizes the INDEX function to filter the view.
- Apply these three items provided to the details for the view you see in Figure 4-21.

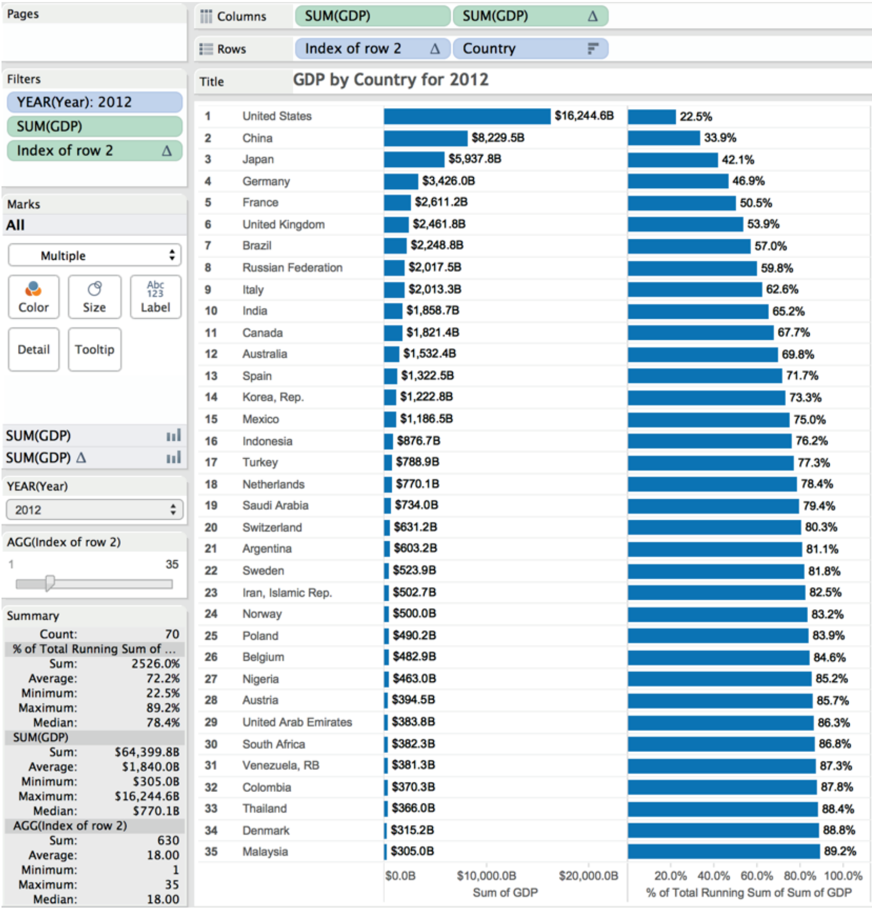


FIGURE 4-21 *GDP by Country enhanced*

Note the cumulative percentage values on the right side of Figure 4-21. That bar chart was created using as secondary table calculation. The ranking number on the left side of the few was provided by an Index Table Calculation function, and the quick filter also utilizes the index function.

To create the Cumulative Percentage bar chart, start by dragging GDP from the Data Shelf Measures area to the Columns shelf, and then edit the aggregation to be SUM by selecting the drop-down arrow in the pill. Then edit the pill to be a Quick Table Calculation to show the running total GDP. Next, edit the Table Calculation by clicking the drop-down again and selecting the Edit Table Calculation menu option you see in Figure 4-22.

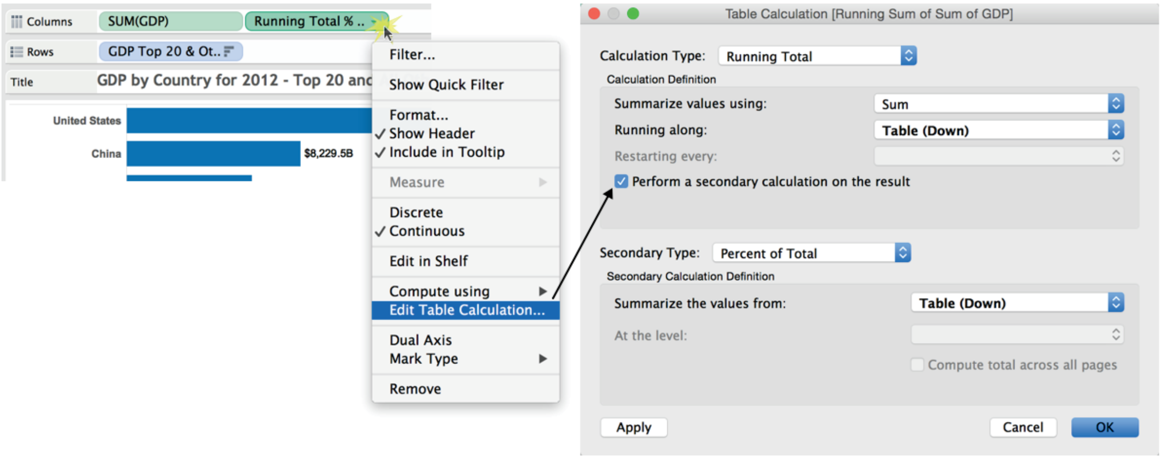


FIGURE 4-22 *Creating the secondary table calculation*

Clicking the check box to perform a secondary calculation exposes the lower section of the table calculation window. Define the secondary table calculation for the percent of table there. Notice both table calculations define a table (down) summary. Try other summarize options to see what is available. Computer Using ➢ Country would also work in this example.

If you create more advanced views that involve multiple dimensions, you can use the advanced options to define not only the type of calculation but how it should restart. See Chapter 7 for an example of this being used in a Pareto chart.

After defining the secondary table calculation, your chart should look like Figure 4-21 but will not include the ranking numbers on the left. A table calculation function can be used to add that data.

Using Table Calculation Functions

The index function used in Figure 4-21 is a table calculation function that counts the position of a row or column in a view. To add this measure to the view, you can type the formula on the Rows shelf or you can go to the Analysis menu to build the formula. If you are familiar with Tableau formula-writing, the ad hoc method is faster. The menu option provides more help. Initiate the calculation by going to the Analysis ➢ Create Calculated Field menu to expose the calculation window you see in Figure 4-23.

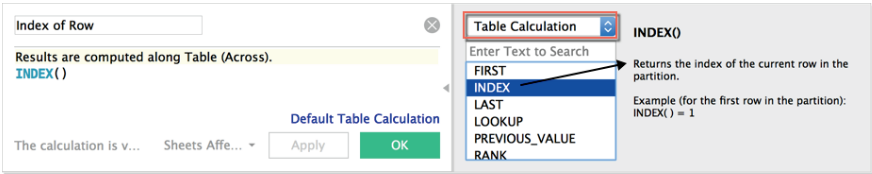


FIGURE 4-23 *Using the index table calculation function*

Name the field Index of Row. You can see that the formula syntax is very simple. Using the Enter Text to Search dialog box at the right of Figure 4-23, you can search for a specific function or filter the list for a specific class of functions. The list has been filtered for table calculation functions. Selecting the index function displays a brief explanation of the function. If you want to get more detailed explanations, refer to the Tableau online manual. In addition, Appendix E provides details on the function syntax along with code samples. Double-clicking on the function will insert it into the formula dialog box. The index function requires no additional arguments. Clicking OK creates a new measure to appear in the Data pane Measures shelf.

Add the measure to the shelf. When you do that, your view will change. Don't panic. Point at the Index of Row pill, select the drop-down, and change the measure type from Continuous to Discrete, as you see in Figure 4-24. Reposition the pill to the left of Country on the Rows shelf.

Now your view should display the country ranking. Figure 4-21 also includes a quick filter that uses the same calculated value. To add that quick filter to the view, drag the Index of Row pill while holding the Control button (Command+Index of pill on a Mac). This will copy the field into the Filters shelf. Once the pill is in the Filters shelf, change it back to a continuous measure. The pill should be green, as you see in Figure 4-25.

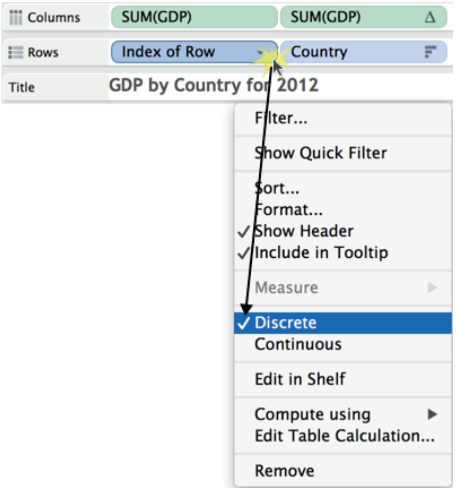


FIGURE 4-24 *Convert the Index of Row to Discrete.*

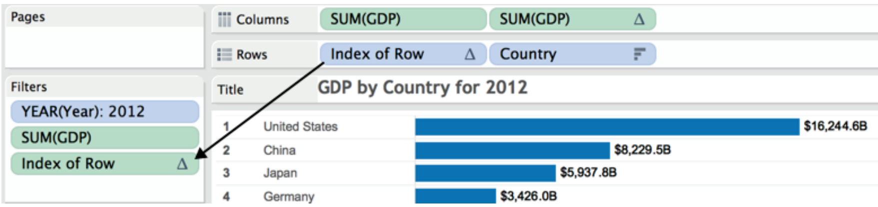


FIGURE 4-25 *Copy the Index of Row to the Filters shelf*

You can see that the Index of Row calculated value has been converted to a continuous measure in Figure 4-25. To add it to the view as a quick filter, click the pill's drop-down menu and select the Show Quick Filter option. Play with the filter to control the number of countries being displayed in the view. In Figure 4-21, the top 35 countries for 2012 are being displayed.

ADDING FLEXIBILITY TO CALCULATIONS WITH PARAMETERS

Parameters empower information consumers to change the content that appears in worksheets and dashboards. Designers have two different ways to enable parameter controls. Basic parameters are helpful when creating flexible top-down filters or adding dynamic reference lines, or in histograms to facilitate variable bin sizes. Advanced parameters provide the ability to address unique use cases and offer more flexibility. But, they do require more planning to create.

What Are Basic Parameters?

Basic parameters are variables that are provided in specific situations that reduce the number of steps required to create a parameter control. Basic parameters are available in histograms to specify the size of each bin. They are also available for creating flexible top or bottom filters, and they can be a way to make reference lines user-selectable. Figure 4-26 shows examples for these three use cases.

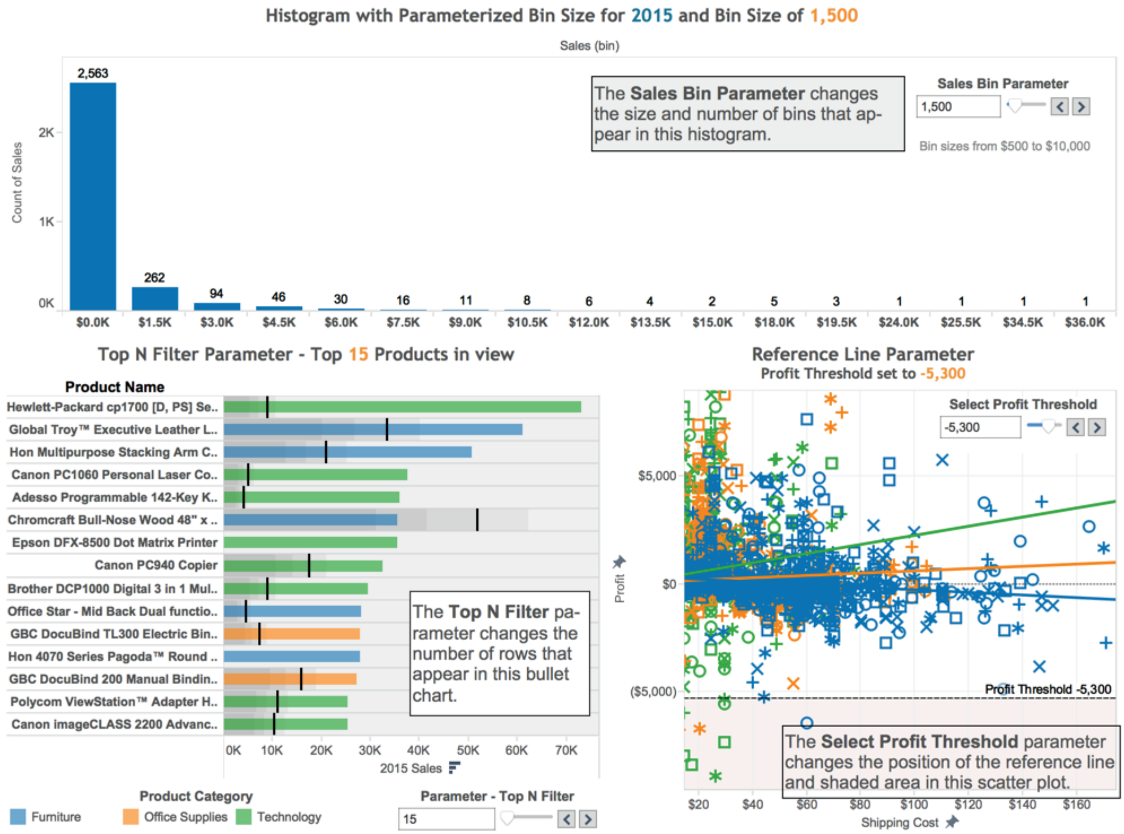


FIGURE 4-26 *Basic parameter controls*

The histogram on the top of Figure 4-26 displays order counts by the size of orders. The Sales Bin Parameter allows end users to change the size of each bin. The Parameter Size Range is from $500 to $10,000. The parameter is currently displaying bin sizes of 1,500. The bullet graph in the lower left of Figure 4-26 compares sales (bars) to prior year sales (black reference lines) for every product name. The dataset includes over 1,000 product names. The Parameter–Top N Filter enables the user to change the number of products displayed in the bar chart. The top 15 products are being displayed in Figure 4-26. The scatter plot in the lower right includes a reference line called Profit Threshold that allows users to change the threshold value affecting the position of the reference line and the corresponding color of the shading below the reference line. It also displays the profit threshold value currently being used to draw the reference line –5,300.

All of these are basic parameters that are selectable options from menus associated with dimension fields for the histogram and the bullet graph (sales bin and product name). The reference line parameter used in the scatter plot was defined when the reference line was added to the view.

To edit the existing histogram's bin size right-click the bin field name that appears in the Dimensions shelf or select the drop-down arrow in the Sales Bin Parameter and select Edit Parameter. To set up the flexible filter in the bullet graph, place Product Name on the Filters shelf, select the top tab, check the By field, expand the value drop-down, and select Create a New Parameter in the filter dialog box.

The Reference Line parameter is accessed when adding the reference line by clicking the Value drop-down selector and picking the Create a parameter option. Figure 4-27 shows the Edit menus for all three parameter controls used to create the controls displayed in Figure 4-26.

Define the parameter for the histogram by right-clicking the sales (bin) field on the Dimension shelf then choose Create > Parameter.

Define the parameter for the Top N filter in the bullet graph by right-clicking the Product Name field on the Dimension shelf then choose Create > Parameter.

Define the parameter for the Reference Line by right-clicking the axis you want to add the Reference Line to then select Create > Parameter.

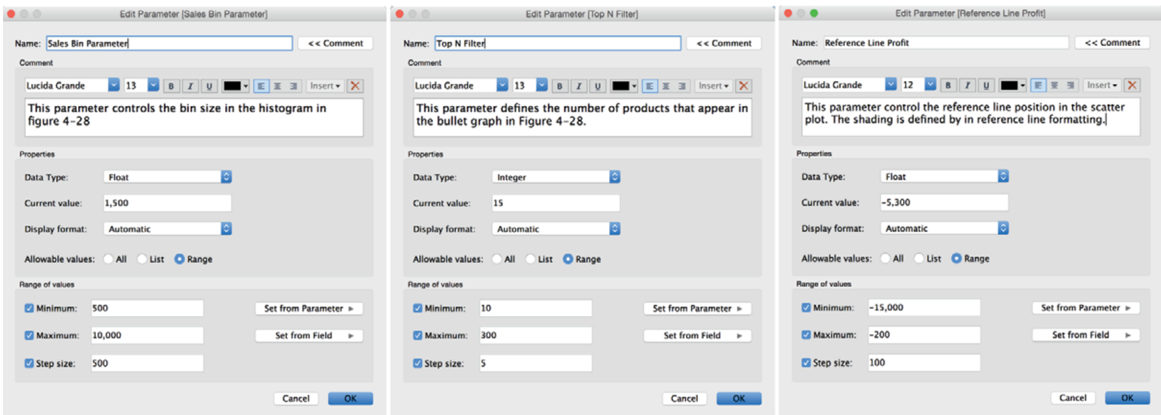


FIGURE 4-27 *Parameter definition windows*

Basic parameters are easy to create but they are also limited to the specific use cases you see in Figure 4-26: bin sizing, top or bottom filters, or flexible reference lines. If you want to create more advanced parameters, these require a little more effort.

What Are Advanced Parameters?

Advanced parameter controls are limited only by your imagination. You can create multiple parameter controls. Parameter controls can be chained together to create linked parameters. An entire book could be written on parameter controls because they provide programming-like functionality to visualizations. Creating advanced parameter controls requires three or four steps:

1. Create the parameter control.
2. Expose the parameter control on the desktop.
3. Use the parameter in a Calculated Field (optional).
4. Use the Calculated Field in the view.

If the parameter is being directly placed in the visualization, it may be unnecessary to create a Calculated Field. The key point is that whatever the parameter is being used to change (typically a formula variable), that item must be used somehow in the visualization for the parameter control to work.

One of the most popular use cases for advanced parameter controls is to enable users to change measures or dimensions being displayed in a single view. The technique in either case is the same. Figure 4-28 shows time series charts in which a parameter is being used to change the measure being plotted.

The parameter control appears in the lower right of the Time Series charts shown in Figure 4-28. Notice that the title of the worksheet includes the parameter, and the left axis label also changes depending on the measure in view.

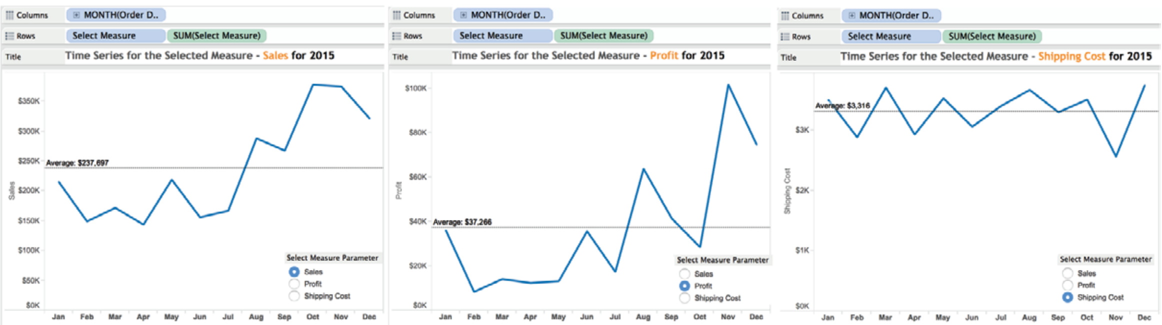


FIGURE 4-28 *Parameterizing the measure displayed in a view*

You add a Parameter Description to the title bar by double-clicking the title bar and selecting the parameter used in the view. To add the parameter name to an axis, drag the parameter from the parameters shelf to the axis. Then edit the axis and erase the static title. To erase the axis label, point at the white space in the left axis, right-click, and then erase the contents of the title editing area. Note that this technique works only if the parameter names are text.

This example also rotated the parameter label so that it appears vertically oriented. When a new selection is made from the parameter control, the time series chart will change along with the headings and reference line to reflect the selected parameter value.

Creating the Parameter Control

This is done directly in the formula editing window or by right-clicking blank space in the dimension, measures, or parameter panes and selecting Create ➢ Parameter. Doing that exposes the dialog box that is used to define the parameter as you see in Figure 4-29.

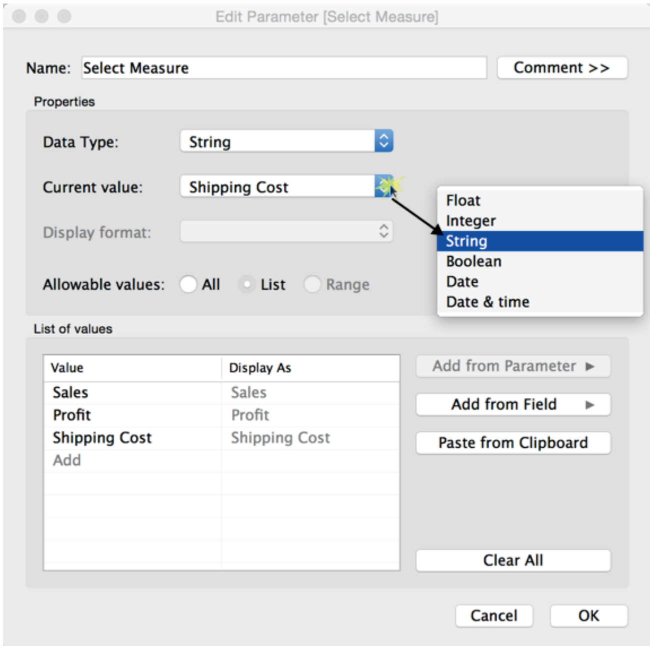


FIGURE 4-29 *Defining a parameter control*

Enter the name of the parameter as you want it to appear in the control that is placed on the desktop, and then define the data type. Parameters can be numbers (floating decimal point or integers), strings, Boolean (true/false), and date or date and time values.

In the List of Values section, you define the variables that will be contained in the parameter control. Figure 4-29 shows a list of measure names defined. While it isn't always desirable, I suggest that for this type of parameter you copy the field names of the measures exactly. This will make formula creation easier in the next step. However, if you find that the performance of your parameter is not good, use numbers in a series (1, 2, 3 . . .) as your value names in the parameter definition. It makes creating the formula in the next step a little more difficult; using numbers in the parameter definition will generally result in a more responsible parameter control. This is especially noticeable with larger datasets.

Notice that there is a Display As option. This is used to create a name alias that will appear instead of the actual field name. The options to the right of the List of Values section are not applicable to this example but are useful for cases where you might be using values from another parameter control or adding members of a particularly large set. To complete the formula definition, click OK, and the parameter will appear in the parameter pane.

Expose the Parameter in the Workspace

In order for users to access the Parameter Control, it needs to be placed on the desktop. To do this, right-click the parameter name appearing in the Parameters shelf and select the Show Parameter Control menu option.

If you access the parameter now, nothing will happen because you haven't used the control in a formula or in any other way in the visualization. The next step is to use this parameter as a variable in a formula.

Create a Formula That Uses the Parameter Control

In Figure 4-28, the Parameter Control is used to change the measure being plotted in the time series chart. This requires a formula that will link the string values defined in the parameter to measure field names in the data source. You can see the formula definition in Figure 4-30.

The formula logic associates the selected parameter string with the related field name. This is why it is a good idea to define the parameter string names to exactly match the field names you want to associate. It makes writing the formula easier if you are in learning mode. Keep in mind that if performance degrades, using sequentially ordered numeric values in the parameter definition will result in the best performance. Note that parameters are denoted by the purple color in the Calculation Editor. Clicking OK adds the Calculated Field to the Measures shelf.

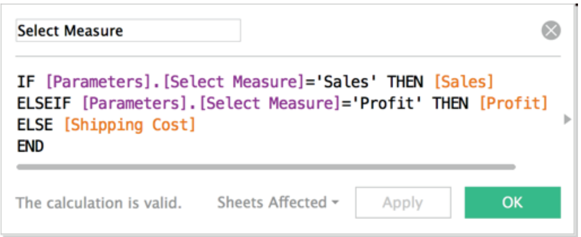


FIGURE 4-30 *Using a parameter in a formula*

Giving your parameter the same name as the related calculation makes it easier to retrace your work at a later date if you need to modify the Parameter Control to add or delete items.

Use the Calculated Field in the View

Dragging the Select Measures calculated value to the Rows shelf will activate the Parameter Control. Each selection made in the parameter will trigger changes in the select measure formula and will change the measure being displayed in the time series. Refer to Figure 4-28 to see three instances of the view, each displaying a different measure.

There are many different ways that you can use advanced parameters. The limit is your imagination. For more examples, go to Tableau Software's website and search for parameters. You should find many different forum posts, training videos, and sample workbooks related to parameters.

WHY YOU SHOULD LEARN LEVEL OF DETAIL EXPRESSIONS

Normally the granularity of the aggregation of measures expressed by Tableau is controlled by the dimensions used to create your visualization. Dropping dimensions on the Rows shelf, Columns shelf, or the Marks card exposes greater detail. Table calculations and reference lines have always provided a way for summarizing views. But they require that the view be structured to support the desired calculation.

Dashboards provide another way to expose different levels of detail. But what if you have limited space or very particular and complicated needs? Sometimes it is useful to be able to express one level of detail in a view but use a different level of detail for calculations to add more information. Introduced in Desktop

V9.0, Tableau is providing a new class of functions for controlling the granularity of calculations. Level of detail expressions require a little practice to learn. But the syntax is much easier than the SQL scripting you would have to use if they didn't exist.

Level of Detail Expression Syntax Explained

Level of Detail expressions are not difficult to write, but they do have specific syntax requirements. Understanding exactly what Tableau does with the LOD expression also requires some understanding of your dataset. Figure 4-31 shows a Level of Detail Expression.

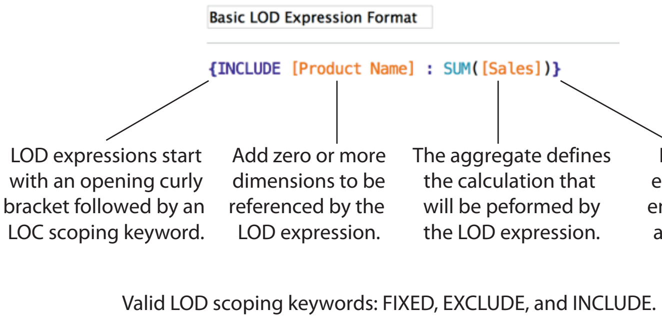


FIGURE 4-31 *INCLUDE LOD expression*

The INCLUDE expression at the start of the formula in Figure 4-31 sums the sales in the view for product name or higher-level aggregates included in the view. Note the curly brackets enclosing the expression. LOD expressions must be wrapped in curly brackets.

You can add more formula logic outside of the curly brackets to specify additional calculations. Placing additional dimensions in the expression is allowed, but you must separate the field names with commas. The best way to get a handle on these expressions is to look at examples using each type of LOD operator—FIXED, EXCLUDE, and INCLUDE.

The FIXED Expression

Applying a FIXED level of detail expression allows you to define the level of detail of the formula contained within the LOD expression. It will not vary regardless of the dimensions placed in your view. Figure 4-32 shows an example.

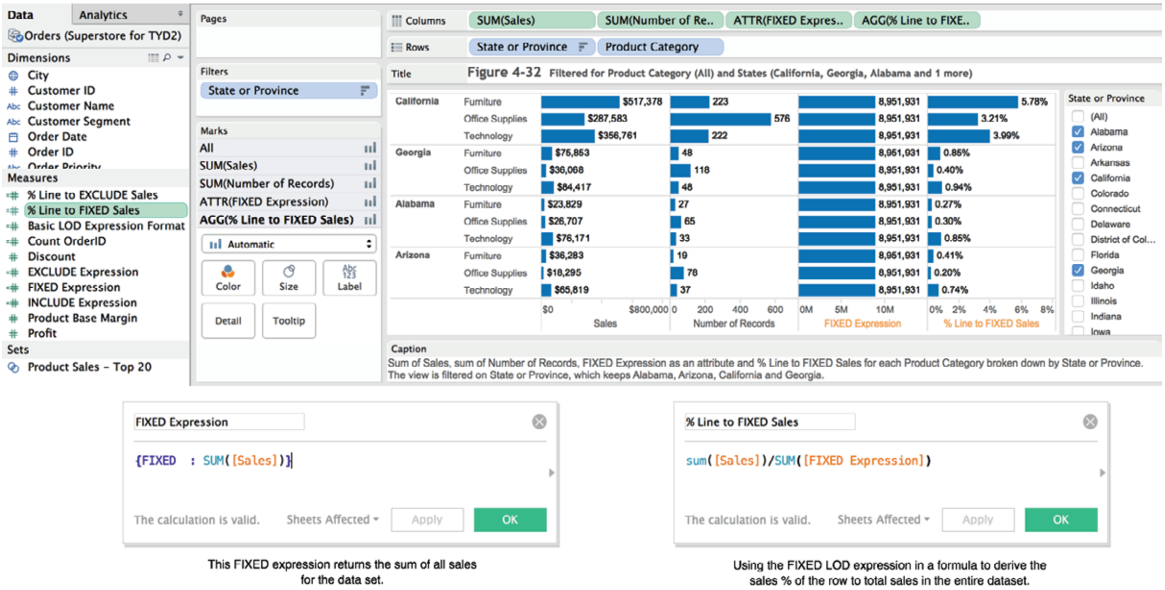


FIGURE 4-32 *Fixed LOD expressions*

The visualization in Figure 4-32 shows four different values. The left bar is plotting total sales for the State or Province and Product Category. (Notice these fields on the Rows shelf.) Then the Number of Records for the combination of State and Category are displayed. The third column shows the result of the FIXED LOD expression (8,951,931) showing the sales value for the entire data set. Even though the view includes the state and product category dimensions, the FIXED expression always returns the total sales for every record in the dataset. The result would be altered only if you were to apply a data extract that filtered out data, applied a file to the data source connection, or you added a context filter in the view.

The chart on the far right of Figure 4-32 shows the FIXED LOD expression being used in another calculated value to derive the percent of sales of total sales that the row represents. The California/Furniture sales of $517,378 represent 5.78 percent of all Superstore for TYD2 sales. Notice that the view has been filtered to include four states (Alabama, Arizona, California, and Georgia), but the FIXED LOD expression still includes all sales from everywhere. This is because dimension filters, measure filters, and table calculation filters are applied after the FIXED level of detail expression is calculated and returned.

FIXED LOD expressions are useful for computing the percent of total details. While you can create visualizations and dashboards to provide these details, the LOD expression makes this very easy to do in a single chart. Rather than showing the calculation in the view, you could include it in the tooltip as text by moving the FIXED Expression and the % Line to Fixed Expression calculations to the tooltip button on the Marks card.

The EXCLUDE Expression

Using the EXCLUDE level of detail expression allows you to define what to omit from the calculation. This expression can be used in low-level calculations using fine-grain details where there is nothing to omit. It provides a way to express the view level or any level of detail above the view level. Changing what appears in your view can change the values being calculated. Figure 4-33 provides an example.

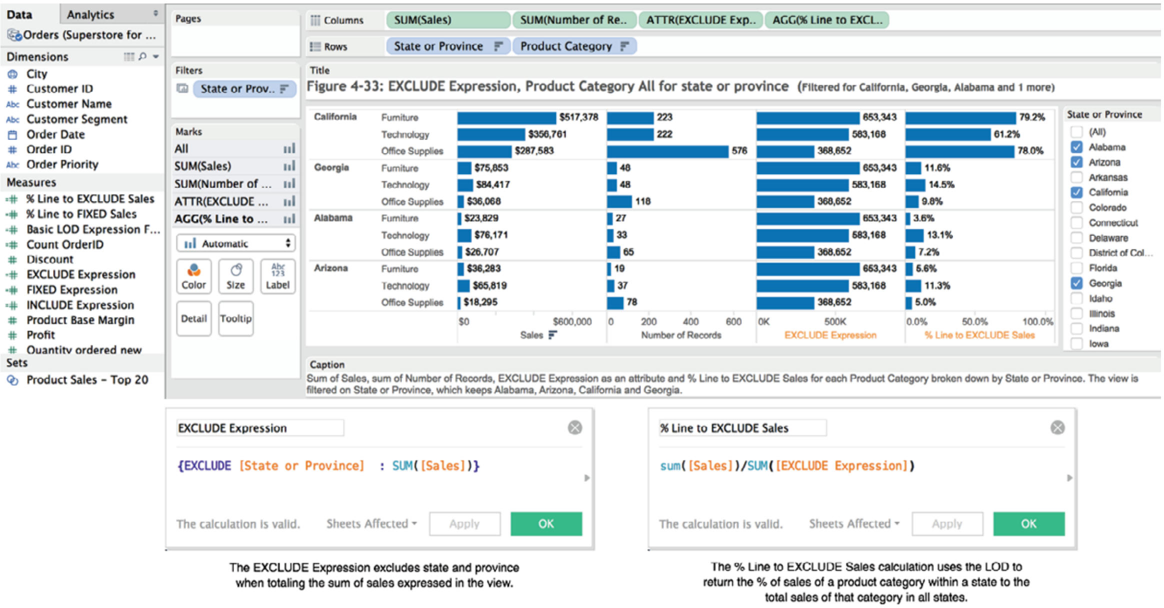


FIGURE 4-33 *EXCLUDE LOD expression*

The visualization is similar to the view displayed in Figure 4-32, but because the EXCLUDE LOD expression is being used, the result is different. The sum of sales displayed for the combination of state and product category chart on the left side of the view is unchanged, with the exception of the sort order of the product categories within each state. The third column bar chart showing the EXCLUDE expression now displays total sales for each product category in the four states filtered in the view (Alabama, Arizona, California, and Georgia). The % Line to EXCLUDE Sales at the far right is now calculating the percentage of total product category sales that the specific state represents within the group of states filtered for in the view. Figure 4-34 provides summary data for comparison.

Total Sales by Category

Product Category	Number of Records	Sales
Technology	2,312	$3,514,982
Furniture	1,933	$3,178,624
Office Supplies	5,181	$2,258,326
Grand Total	9,426	$8,951,931

Total EXCLUDED Category Sales

Product Category	Number of Records	Sales
Furniture	317	$653,343
Technology	340	$583,168
Office Supplies	837	$368,652
Grand Total	1,494	$1,605,163

FIGURE 4-34 *Total Category Sales and Excluded Category Sales*

The left text table in Figure 4-34 summarizes all sales records in the Superstore for TYD2 dataset by product category. The right text table shows the total sales by product category for only those states included in the view in Figure 4-33 (Alabama, Arizona, California, and Georgia). Compare these figures to the amounts returned in Figure 4-34.

The first column in Figure 4-34 displays California furniture sales only. Similarly, the 223 units displayed in the Number of Records column represents the row count of the California-only furniture sales. The three columns containing the EXCLUDE Expression $653,343 is the total furniture sales for the states displayed in the view (Alabama, Arizona, California, and Georgia). The far-right column, which shows the % Line to EXCLUDE Sales, calculates the percentage of sales that California represents of the same four states in the view ($517,378/$653,343 = 79.2%).

Changing the filter to include more states will change the amounts returned in the third and fourth columns that are fed by the LOD expressions, unlike the FIXED expression used to create Figure 4-32. Try adding the FIXED expression to the view in Figure 4-34 to compare the results returned by the FIXED and EXCLUDE LOD expressions.

The INCLUDE Expression

The INCLUDE LOD expression allows you to create formulas that consider more granular dimensions than the visualization displays. Similar to the EXCLUDE expression, amounts calculated using INCLUDE can change depending on what dimensions are exposed in the view. Figure 4-35 includes a bar chart of sales by state and the postal code with the maximum sales for that state. To the right of the bar chart uses the INCLUDE expression for the bar chart displaying sales by postal code for the state of Georgia. Both visualizations are filtered for the year 2015.

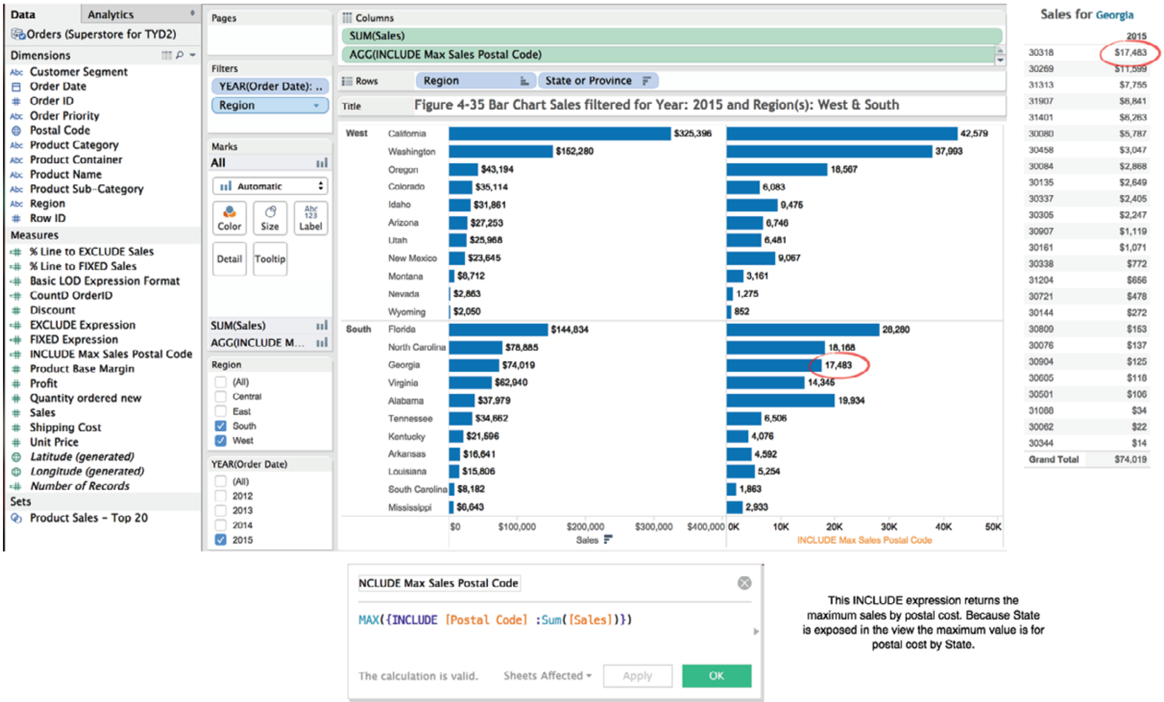


FIGURE 4-35 *INCLUDE Max Sales Postal Code*

The goal for the bar chart is to display the total sales for the state and the sales value for the postal code with the highest sales within that state. Even though the level of detail displayed in the bar charts is Region (west/south), the INCLUDE LOD expression provides a way to display the postal code that has the most sales for each state displayed. Georgia is highlighted. The total sales in 2015 for Georgia are $74,019. The postal code maximum sales in 2015 for Georgia are $17,043. This is confirmed by the text table display at the right with postal code 30318 returning the top sales value for the year 2015.

It will take you a while to get a feel for the behavior of LOD expressions. They provide precise control over the result you require without requiring that you display the level of detail in the view that you want to return. In the next section, you'll learn how different filters can affect LOD expressions.

How Filters Affect Level of Detail Expressions

Filtering views can affect the results calculated by LOD expressions. The precise outcome is the result of the type of LOD expression and the type of filter being used. Tableau executes filters at the data source, through temporary tables, and locally based on the contents of the visualization. Table 4-1 shows the order of precedence.

TABLE 4-1 Order of Precedence for Filters and LOD Expressions

FILTER TYPE	FILTER APPLIED AT	LOD EXPRESSION
Extract Filters	Data source	Filter presides
Data Source Filters	Data source	Filter presides
Context Filters	Local temp table	Filter presides
Dimension Filters	Locally	FIXED dominant
Dimension Filters	Locally	INCLUDE/EXCLUDE subordinate
Measure Filters	Locally	All LOD expressions dominant
Table Calc Filters	Locally	All LOC expressions dominant

Limitations of LOD Expressions

Level of detail expressions are not supported by every data source that Tableau can connect to. As of May 2015, the following data sources do not support LOD expressions:

- Cubes
- Google Big Query
- DataStax
- Informatica Data Services
- Microsoft Jet
- Splunk
- Actian Vectorwise

Other data sources are supported only through more recent releases. Search for “Data Source Constraints for Level of Detail Expressions” in Tableau Software’s online manual for the latest information.

Additional Resources on Level of Detail Expressions

For additional details and examples of Level of Detail (LOD) expressions, see the notes at the end of this chapter.

Using the Function Reference Appendix

Tableau provides online documentation of functions. The user forum on Tableau's website is also quite good. However, many novice users have asked for a more detailed reference for Tableau functions that provides examples and explains the formula syntax in more detail. That is what you will find in Appendix E.

Functions are listed alphabetically by function type. Each function reference entry provides a short description of the function, typical use cases, and basic, intermediate, and advanced examples. I hope you'll find the function reference a useful addition to your toolset.

In the next chapter, you learn how Tableau creates geospatial data for mapping. If your data includes country, state, or other standard geographic dimensions, you can easily plot your data in maps.

NOTES

1. Rosamund Stone Zander and Benjamin Zander, *The Art of Possibility* (New York: Penguin, 2002), 13.
2. Robin Cottiss, "Understanding Level of Detail Expressions," Tableau Whitepaper, 2015. http://www.dmgfederal.com/wp-content/uploads/2015/04/Tableau-Level-of-Detail-Expressions-Whitepaper.pdf.
3. Bora Beran, "What's New in Tableau 9.0? Part 2—Level of Detail Expressions," 2015. https://boraberan.wordpress.com/2015/01/30/whats-new-in-tableau-9-0-part-2-level-of-detail-expressions/.
4. Bethany Lyons, "Top 15 LOD Expressions," 2015. http://www.tableau.com/LOD-expressions.
5. Michelle Wallace, "Tableau 9 Beta-ers: Show Us How You're Bringing Data to a #WholeNewLevel!" 2015. http://www.tableau.com/pt-br/about/blog/2015/1/tableau-9-beta-show-us-how-bringing-data-wholenewlevel-36178.

CHAPTER 5

Using Maps to Improve Insight

Maps can reveal more than the landscape, and have for several centuries. Remember the dictum: Anything that can be spatially conceived can be mapped. The new technologies broadened the conceivable meaning of "anything" and gave mapmakers the means to call up all manner of information about people, places, and environments—anything mappable—and visualize it swiftly and in many different guises.

—John Wilford[1]

People are accustomed to using maps to find places, predict the weather, and see information regarding world events. Seeing your data displayed on a map can provide new insight. Visualizing geographic data with Tableau brings new understanding to the many "guises" that this perspective can bring. For some reason, people are always drawn to data plots on maps. Tableau's standard maps are good and render quickly. You can also replace the standard maps with customized versions provided by web mapping services. Or, if you have spatial data that is too small to fit on a map, you can replace maps with images.

Tableau provides two different map types: symbol maps and filled maps. Symbol maps place marks at the center point of standard geographic units. Filled maps color-encode standard geographic shapes using a measure or dimension to apply the color. Three standard map background image styles are provided:

- **Normal:** Off-white land forms with blue water
- **Gray:** Gray land forms with white water
- **Dark:** Dark gray land forms with light gray water

If you do not need to make a clear distinction between land and water forms, the gray map style places more emphasis on your data. The dark map style is particularly useful if you have to present maps using an older overhead projector. Try different appearance options by going to the Map menu and selecting the Map Options menu. From there you can change the map style, alter the washout of the map background, or apply map layers. Adding map layers provides context by including (Base layers, Land Cover, Streets and Highways) and United States census data. If you want to save your map option selections for use later, click the Make Default button at the bottom of the Map Options menu. Tableau's standard maps provide fine-grained geographic details when you have an Internet connection. If you don't have an Internet connection, less-detailed offline maps are provided.

Using Show Me, you can create a map visualization in less than five seconds by double-clicking any geographic dimension (denoted by the globe icon) and then double-clicking any measure. Show Me is always operating in the background and will position three pills on the appropriate shelves and then present you with a map view of your data with the measures placed in the map at the center of the geographic dimension selected.

NEW MAP FEATURES

Tableau V9 adds more geographic details for international customers. Place names are localized using the language selected for your computer's operating system. More selection and filtering tools are now available in maps. You can now search places by entering a place name. Two additional selection tools have been added: Radial search and Lasso search. Map rendering speed has been improved, and tooltips now render instantly. Beginning with Tableau V9.1 the Radial Selection tool measures and displays approximate distances (in localized units of measure) and allows you to turn off pan and zoom controls in maps.

CREATING A STANDARD MAP VIEW

Take a look at what happens when you want to plot more complex data using the Show Me menu to create a map. Using the Superstore for TYD2 dataset, create a map that uses State or Province and Product Category from the Dimensions shelf and Sales from the Measures shelf. Multi-select the fields and use the Show Me button to pick the symbol map. The resulting map is displayed in Figure 5-1.

You can see that Tableau placed six pills in various places to create the symbol map. In fact, double-clicking any geographic field will result in a map being created in which the marks plotted will show the center point of the selected entity. This is why you should rely on Show Me to build maps—it's much faster than manually dragging fields on to the Rows and Columns shelves.

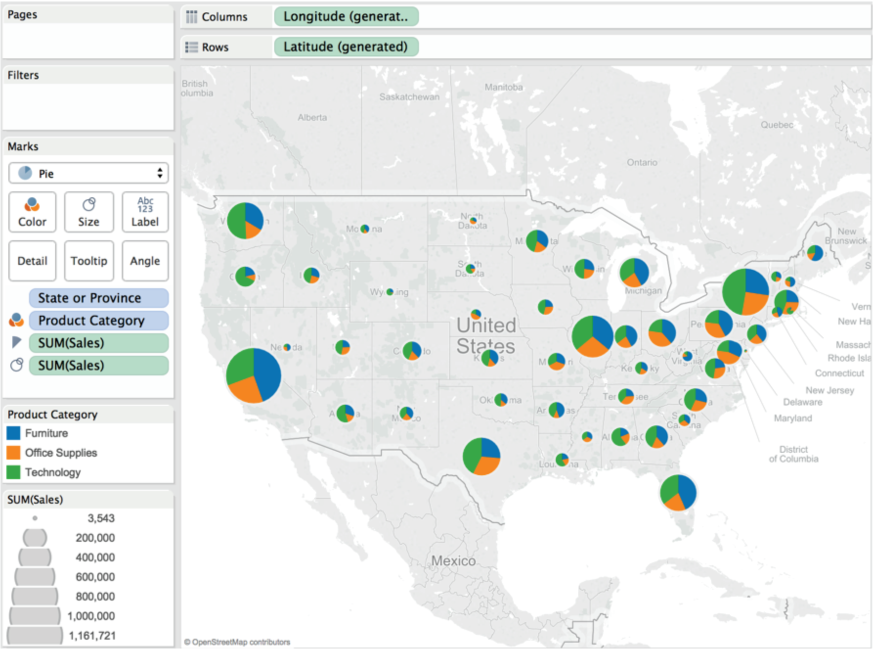


FIGURE 5-1 *Symbol map created using Show Me*

Open Show Me again and select the filled map. When you make that change, Tableau automatically moves the Product Category field to the Rows shelf. This results in the display of three maps. Each map shows sales for a specific product category. Dragging the Product Category pill from the Rows shelf to the Filters shelf collapses the view into a single map. I also added a quick filter for Product Category, changed the colors used in the Color Legend, and turned on labels for each state to display the sales amounts in thousands of dollars. Figure 5-2 shows the resulting view filtered for the Office Supplies and Technology Product categories.

The Sales amount labels were turned on by clicking the Label button on the Marks card and selecting the Show mark labels option. The number format was changed by pointing at the label, right-clicking, and then using the formatting menu to change the number style to currency with one decimal point using units of thousands. To edit the map style, use the main menu option for Map and choose the Map Layers menu.

Figure 5-3 shows the Map Layers menu on the left. Beginning with Tableau Version 9.2 you can select Map Options from the menu to expose the floating control seen on the right of Figure 5-3. Using this floating dialog you can control whether or not to enable map panning and zooming, the map search control or the Show View Toolbar.

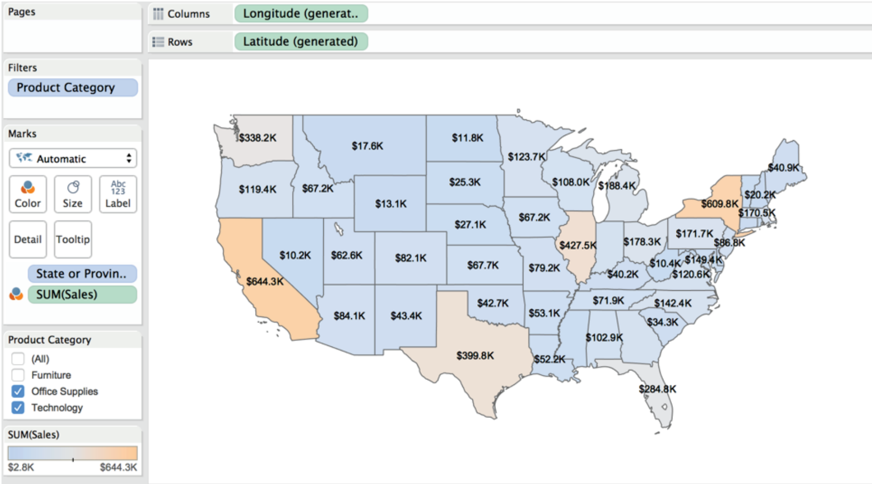


FIGURE 5-2 *Filled map created with Show Me*

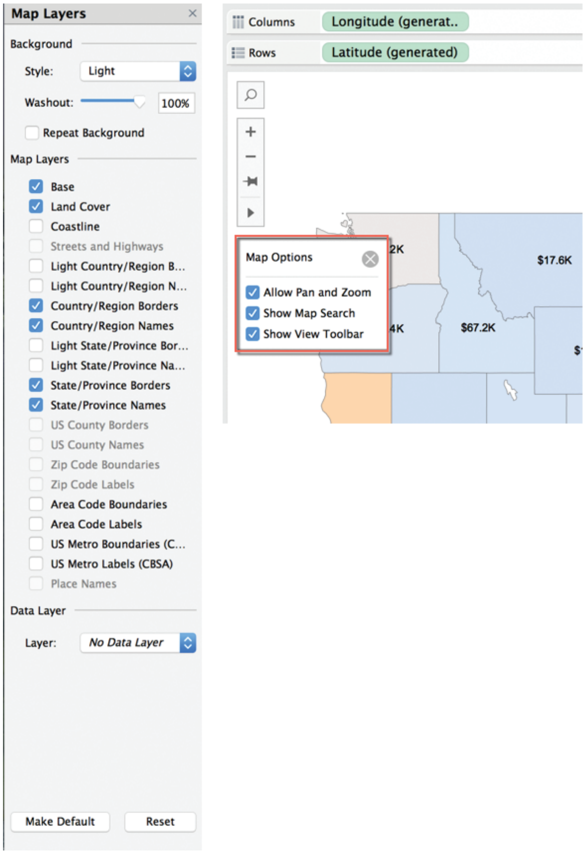


FIGURE 5-3 *Map Options menu*

For the filled map in Figure 5-2 the map washout was changed to 100 percent. This hides map features from view. The areas bordering the United States are blank. If any state lacked sales, it would be blank as well. At the top of the background section, the light map type is selected. There are two other map types: normal and dark. Figure 5-4 shows the differences between normal, light, and dark backgrounds.

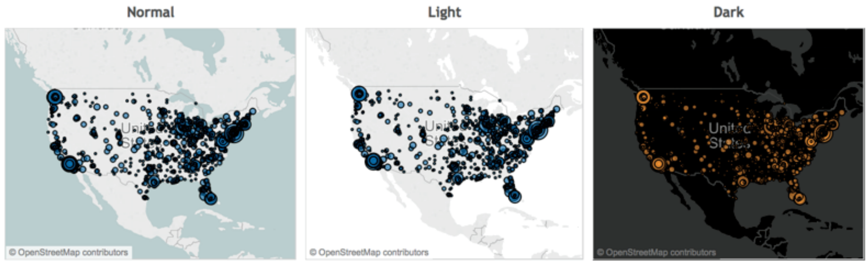


FIGURE 5-4 *Map background style options*

The map options menu in Figure 5-3 also allows you to add more details to the map by selecting additional map layers. These options allow you to color-encode geographic shapes in the map using state, county, zip code, or census block level of detail. For the United States, there are thirty different census datasets related to population, race, occupation, households, and housing. If you want to override Tableau's standard map settings with the new options you've selected, click the Make Default button at the bottom of the menu.

HOW TABLEAU GEOCODES YOUR DATA

Tableau places marks on your map, automatically positioning them at the center of the geographic unit displayed. It recognizes a large number of standard geographic entities. The United States is mapped in detail, and the geographic detail for international locations is extensive and growing with every version update. Geographic units include:

- Area code
- CBSA/MSA (USA census blocks)
- City
- Congressional District (USA only)
- Country/Region
- County

- State/Province
- ZIP Code/Post Code
- Latitude/Longitude (added to your data)

Locally stored geographic data is used to place your information on the maps. By default, Tableau uses detailed online maps.[2] If you can't get a web connection, Tableau's offline maps provide you with less detailed map images. Figure 5-5 shows online map examples for San Francisco and New York City.

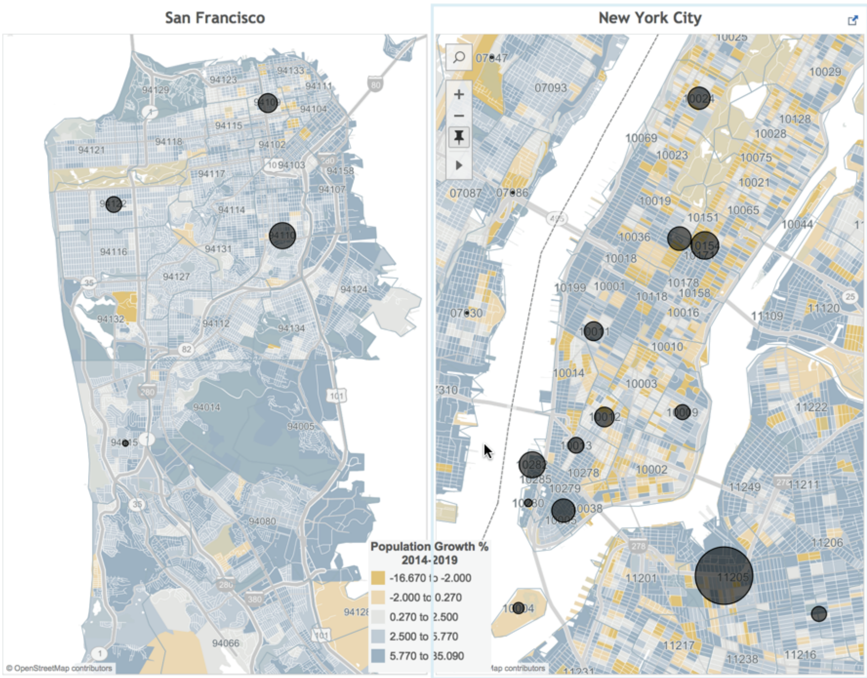


FIGURE 5-5 *Tableau online map*

Using the light map style and displaying the streets and highways map layer, Figure 5-5 also includes projected population growth at the ZIP code level of detail. ZIP code boundaries and titles are also displayed. The black marks (circles) display sales data by ZIP code. The marks in Figure 5-5 are plotted at the geographic center of the ZIP code. Tableau doesn't include international census data at this time, but international maps do include extensive road details. Figure 5-6 shows four international cities using the normal map style.

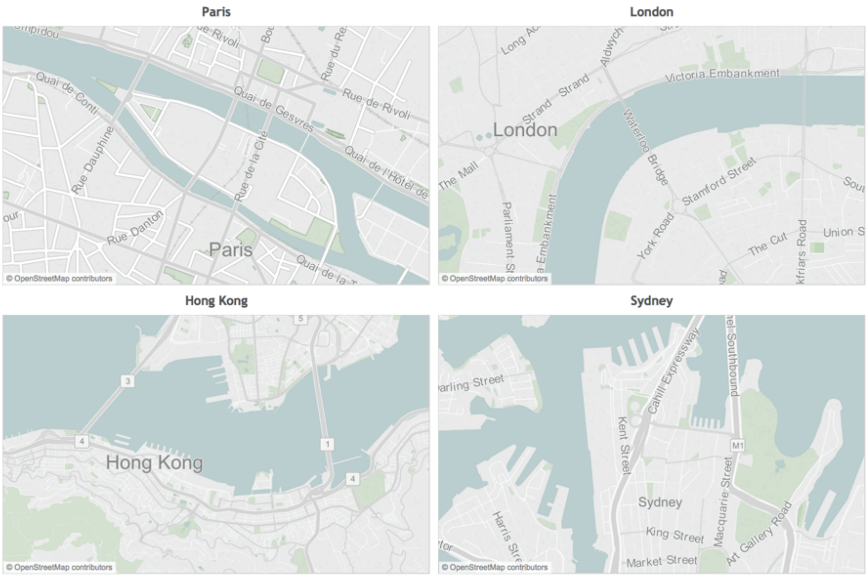


FIGURE 5-6 *International maps*

Tableau balances the rendering speed of maps with good map details so that you can find relevant reference points quickly. This is accomplished by displaying more granular details as you zoom into smaller areas. Tableau maps provide more than 300,000 municipalities globally. Detailed map views are available for most locations.

SEARCHING FOR ITEMS IN MAPS

After building a map of your data, it's normal to want to drill down into specific parts of the map. Tableau provides zooming, selecting, and filtering tools inside the map. Figure 5-7 highlights the tools.

The plus and minus symbols in the upper-left side of Figure 5-7 are basic tools for zooming in and out on the map in view. When you zoom, the pushpin symbol in the map and in the icon bar will appear to be depressed. Clicking the right-pointing arrow exposes more tools for zooming, a rectangular selection tool, a radial selections tool, and the lasso selection tool. The image in Figure 5-7 shows the result of using the lasso selection tool for the highlighted marks on the map. Get familiar with how these tools operate by testing them in a map view you create.

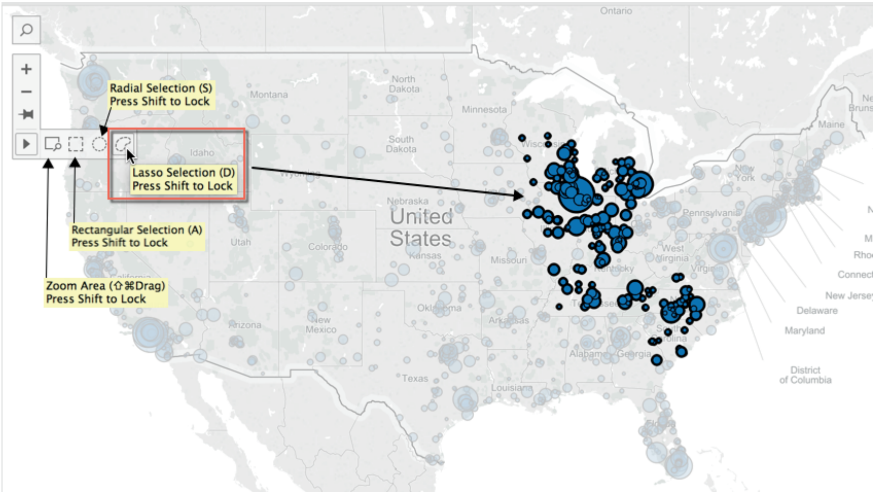


FIGURE 5-7 *Map zoom and filtering tools*

The Radial and Lasso Select tools were added in V9.0. Tableau also added the location text filtering tool that you see in Figure 5-8.

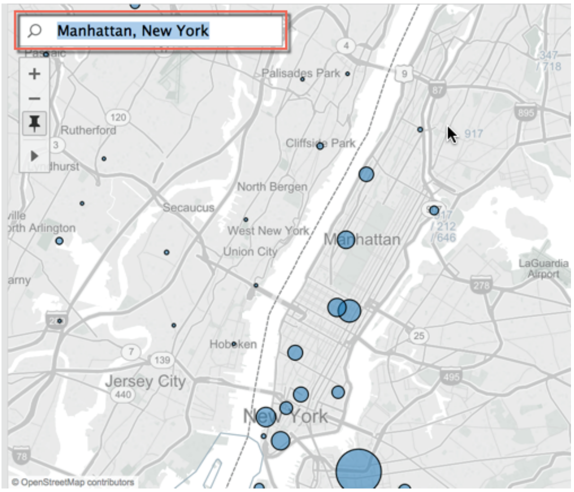


FIGURE 5-8 *Text filtering tool*

As you can see, the map has been filtered in Figure 5-9 by typing in Manhattan, New York. You can filter for any standard geographic unit that Tableau recognizes using this filter.

TYPICAL MAP ERRORS AND HOW TO DEAL WITH THEM

It isn't unusual to have some missing or erroneous data, especially when you are investigating data for the first time. Fortunately, Tableau helps you identify non-conforming details and make corrections quickly without having to edit the data source directly. Figure 5-9 shows a filled map. The color encoding of the map displays the relative sales value of each state. You can see that there is something wrong with the view because Missouri is blank. This could be caused if an abbreviation was used for Missouri (MO) if the other state names are not abbreviated. In this case, the problem was caused by a misspelling of the Missouri state name in the data source.

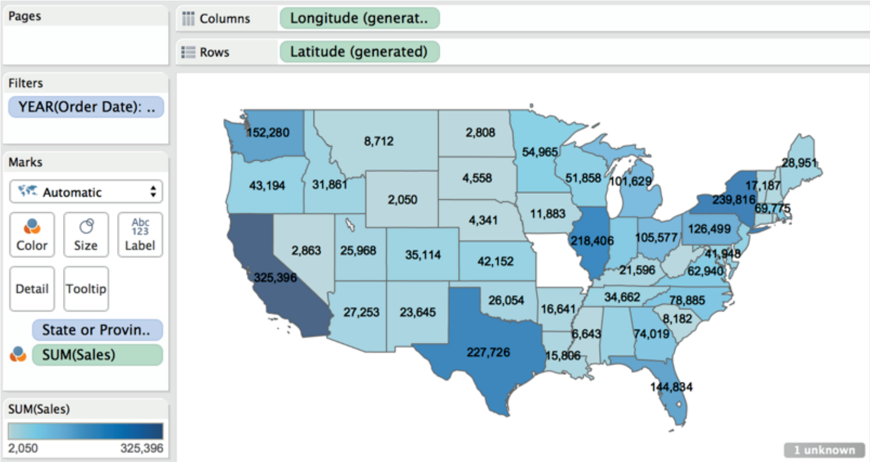


FIGURE 5-9 *Filled map missing data*

In the lower-right area of the map in Figure 5-9 there is a gray pill that includes the text (1 unknown). This indicates that one geographic record is missing or unrecognized in the dataset. Clicking the pill opens the special values menu, which provides three options for dealing with the unknown record:

- Editing the locations (to correct the error)
- Filtering the data (to exclude the record)
- Showing the data at the default position (this means zero)

Clicking the Editing the locations option exposes the special values menu you see in the upper-left area of Figure 5-10.

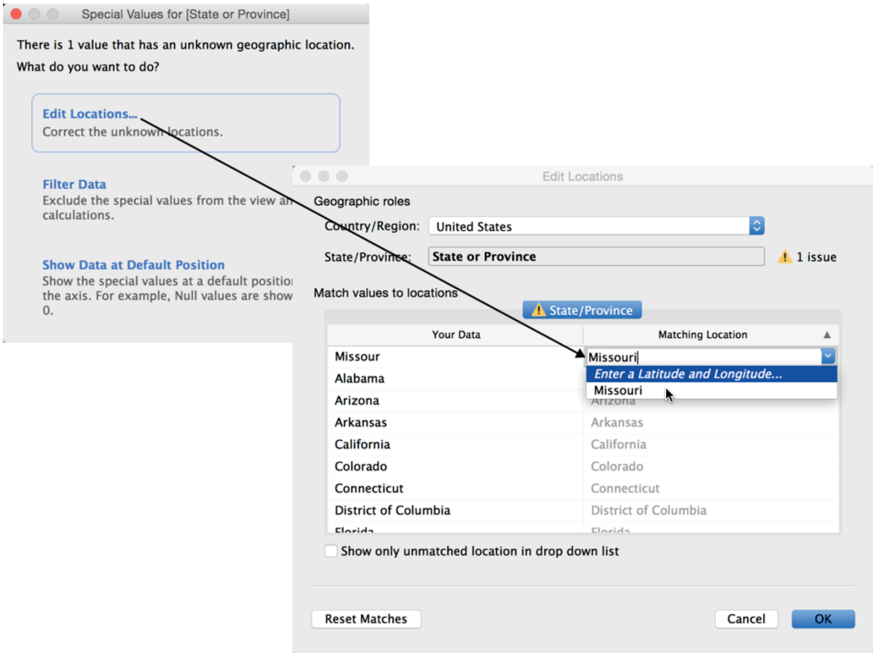


FIGURE 5-10 *Correcting place name errors*

Selecting the Edit Locations option exposes the menu on the right of Figure 5-10. Tableau identified that the state of Missouri was misspelled. This is why there is no color fill for Missouri in the map. You fix the error by typing the correct spelling into the Matching Location area. After typing a few letters, Tableau narrows the list of candidates to Mississippi or Missouri. Selecting Missouri aliases the state name in Tableau with the correct spelling and fixes the problem. The source data set is still wrong, but Tableau's name alias will correct the problem in the map. Lock in the change by clicking OK. Tableau recognizes different place name variations (abbreviations) and will edit other geographic entities as well (city, county, province, and so on). This ability to quickly identify and correct non-conforming records will save you time and make it easy for you to provide detailed feedback to correct errors in your source data.

PLOTTING YOUR OWN LOCATIONS ON A MAP

It would be impractical for Tableau to monitor and save every possible location in the world. If you have specific places you want to plot on maps that Tableau doesn't automatically recognize, you can enable this using two different methods. You can add the specific longitude and latitude to your source data, or you can import custom geocode lists into Tableau.

Adding Custom Geocoding to Your Data Source

You must obtain geographic coordinates in the form of longitude and latitude values that you provide to your dataset in order to add custom geocoding to your data source. There are many free web-based geocoding services that provide this information. Figure 5-11 shows a list of jazz clubs in the metro New York area that have the necessary latitude/longitude data added.

1	ID	Venue Name	Address	City	Phone	State	Type	latitude	longitude
2	1	Paramount Center for the Arts	1008 Brown Street	Peekskill	877-840-0457	NY	Jazz Played	41.2899838	-73.9197845
3	2	Metropolitan Museum of Art	1000 Fifth Avenue	New York	212-535-7710	NY	Jazz Played	40.7791544	-73.962697
4	3	Buckingham Hotel	101 W. 57th St.	New York	212-999-5585	NY	Jazz Played	40.7645904	-73.9773501
5	4	Vox Pop Brooklyn	1022 Cortelyou Road	Brooklyn	718 940 2084	NY	Jazz Played	40.63932	-73.967996
6	5	Oceana Hall	1029 Brighton Beach Ave	Brooklyn	718-513-6616	NY	Jazz Played	40.5783085	-73.9584269
7	6	Oceana Ballroom	1029 Brighton Beach Ave.	Brooklyn	347-462-2810	NY	Jazz Played	40.5783085	-73.9584269
8	7	North Square Lounge	103 Waverly Place	New York	212-254-1200	NY	Jazz Played	40.7325	-73.998692
9	8	Jazz Museum in Harlem	104 E. 126th Street	New York	212-348-8300	NY	Jazz Played	40.80528	-73.938056
10	9	Lansky Lounge & Grill	104 Norfolk Street	New York	212-677-9489	NY	Jazz Played	40.7143528	-74.0059731
11	10	Cafe Sabarsky	1048 5th Avenue	New York	212-628-6200	NY	Jazz Played	40.781219	-73.960228
12	11	Havana - New Hope	105 South Main Street	New Hope	215-862-1933	PA	Jazz Played	40.3616309	-74.9504278
13	12	The Village Quill	106 Franklin Street, Second Floor	New York	212-226-0442	NY	Jazz Played	40.7143528	-74.0059731
14	13	Arturo's Restaurant	106 West Houston Street	New York	212-677-3820	NY	Jazz Played	40.727459	-74.000351
15	14	Village Ma	107 Macdougal Street	New York	212-529-3808	NY	Jazz Played	40.729911	-74.000945
16	15	Clemente Soto Velez Cultural Center	107 Suffolk Street	New York	212.260.4080	NY	Jazz Played	40.7190801	-73.9861613

FIGURE 5-11 *Custom location list*

The sample data partially in view in Figure 5-11 includes more than 500 addresses. You can see the coordinates in the far-right columns. Using the custom latitude and longitude data allows you to plot each location on its specific address. Figure 5-12 shows the custom plot with a tooltip exposed displaying additional location details.

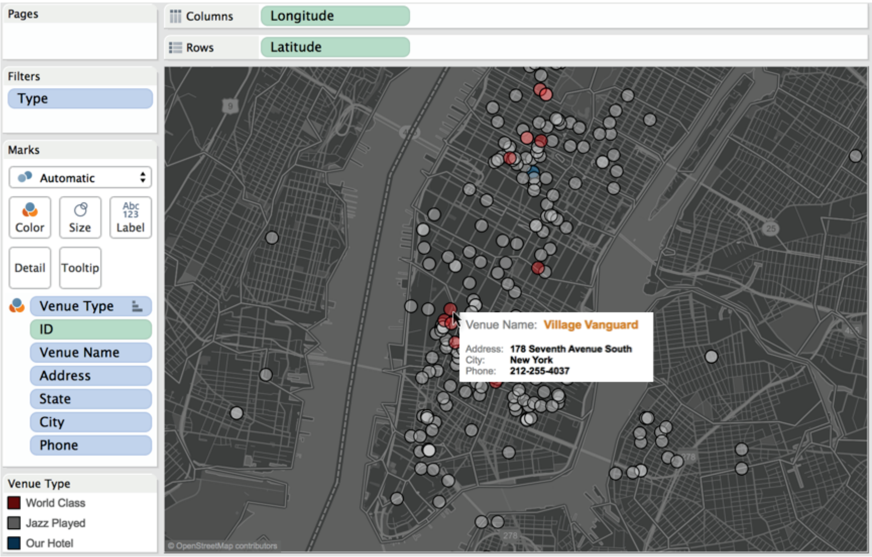


FIGURE 5-12 *Custom geocoding*

The column and row shelves hold the custom latitude and longitude coordinates that place the marks on the map. The tooltips were customized to show the street address, phone number, and venue information. The dark map style was modified slightly by employing a 25 percent washout via the Map ➢ Map Options menu.

Importing Custom Geocoding into Tableau

If you go to the trouble to add customized geographic coordinates, wouldn't it be nice to make that available to other people within your organization? This is done by importing the address coordinates directly into Tableau Desktop. After they are imported, the custom locations behave like Tableau's default geographic units. The import file should have the following characteristics:

- Give each location record a unique identifier (key record).
- Save the location file in comma-delimited CSV format.
- Label the coordinate fields Latitude and Longitude (these key words must be used).

It's best not to include a lot of additional dimensional data in these import files. Be sure that you have only one instance of Tableau open when you perform the import. If you want to share this custom data with other people building reports in Tableau, store the list in a shared network directory so that other users can import from the same list as well. Figure 5-13 is an example of a custom geocode table properly formatted for import.

1	Whse	Latitude	Longitude
2	1	40.157319	-75.403442
3	2	41.816565	-87.735664
4	3	33.409758	-84.727439
5	4	32.622213	-97.305182
6	5	33.975119	-118.42637

FIGURE 5-13 *Custom geocode table*

After saving the custom list in comma-delimited CSV format, the custom geographic data can be imported into Tableau. Initiate the import using the main menu (Map ➢ Geocoding ➢ Import Custom Geocoding). A small file will require only a few seconds to load. Large lists can take several minutes. Running the import will create a new data file on your computer within (My Tableau Repository/Local Data). The custom data is stored as a Tableau data source file (`.tds`) with the same name as the source CSV file that was imported.

Using Custom Geographic Units in a Map

After importing custom geocodes into Tableau Desktop, you can use them with other data files to build maps as long as those files contain the same location key record defined in your data import table. The custom data file used in the rest of this example looks like Figure 5-14.

	A	B	C	D	E	F
1	Whse	Street	City	State	Zip	Headcount
2	1	3123 Ridge Pike	Eagleville	PA	19403	100
3	2	4500 West 42nd Place	Chicago	IL	60632	150
4	3	365 Walt Sanders Memorial Drive	Newnan	GA	30265	50
5	4	1425 Forum Way South	Fort Worth	TX	76140	105
6	5	13031 West Jefferson Blvd.	Los Angeles	CA	90066	225
7						

FIGURE 5-14 *Warehouse location table*

Notice there is no longitude or latitude data in the table displayed in Figure 5-14. The Whse field is being used to link the imported data from the custom geocode table displayed in Figure 5-13. The following are the steps to use custom geocoding in this example:

1. Attach Tableau to a data source.
2. Alter the geographic role of the key records (use the imported custom geography).
3. Use the key record in your view to plot the location on a map.

Altering the geographic role of the warehouse location is done by right-clicking the Whse field and selecting Geographic Role ➢ Whse as you see in Figure 5-15.

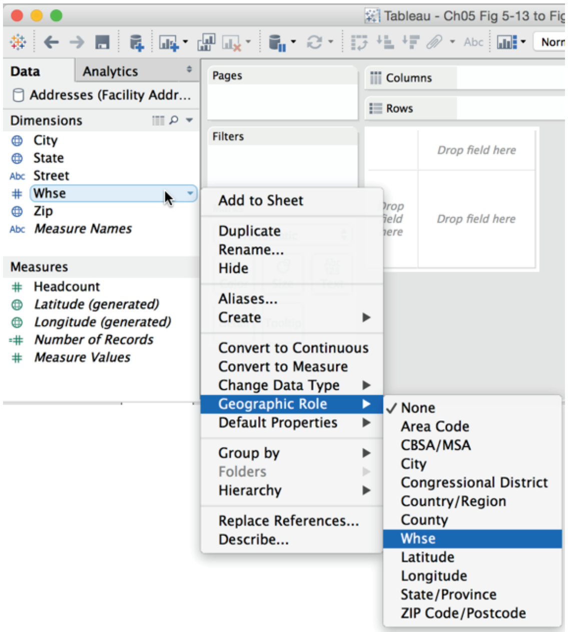


FIGURE 5-15 *Altering the geographic role*

The Whse field icon will change to an icon that is similar to a standard geographic icon but with a small list in front of the globe. Tableau will now recognize the imported geographic data. Figure 5-16 shows a map plot of the custom geocoded warehouse locations.

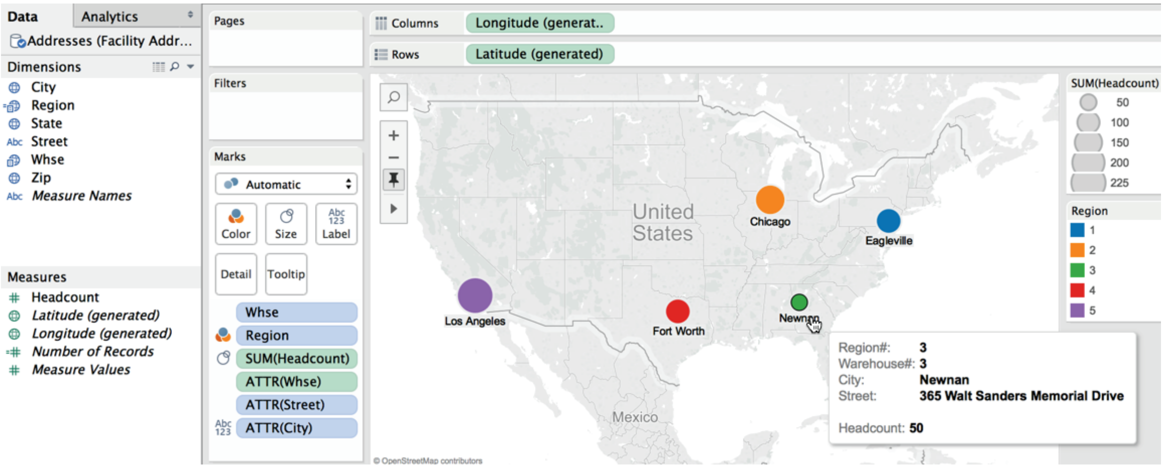


FIGURE 5-16 *Warehouse location map*

Notice that the Columns and Rows shelves are using the (generated) latitude and longitude fields from the imported custom locations to place the warehouse location marks on the map. Hovering over the Newnan location causes a customized tooltip to appear showing particulars about the warehouse. Zooming in on the Los Angeles location in Figure 5-17, you can see that the mark is more precisely placed than Tableau's standard geocoding can provide. This geographic placement is provided by the imported geocodes shown in Figure 5-13, and tooltip details are supplied from the spreadsheet shown in Figure 5-14.

The order of the steps for enabling custom geocodes at the beginning of this section isn't the only way to achieve the result. You can import the custom geocoding at any point—even after a map plot is created using standard geographic plotting.

Now that you've learned how to precisely plot any location on a map with custom geocoding, in the next section you learn how to replace the standard maps provided by Tableau with customized map sources from the web.

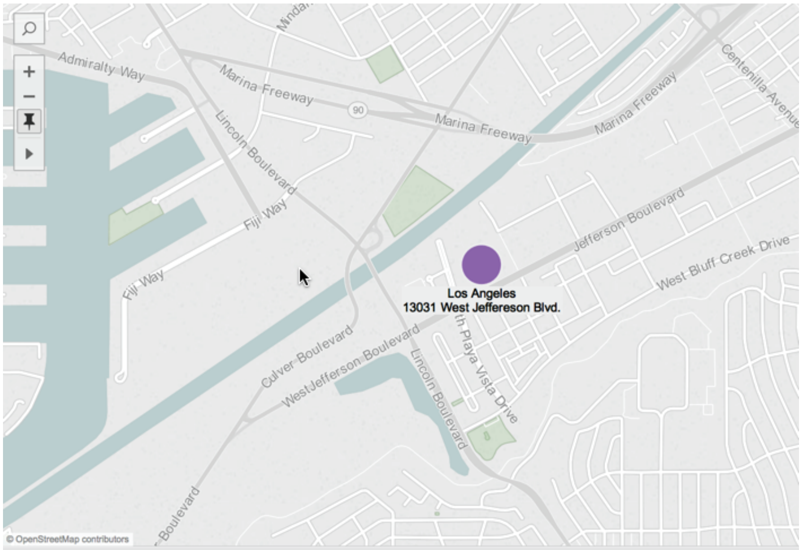


FIGURE 5-17 *Los Angeles warehouse*

REPLACING TABLEAU'S STANDARD MAPS

Tableau Maps visualizations are special scatter plots that use map images for backgrounds and special measures (longitude and latitude) to plot marks on the map. This implies you can replace Tableau's standard map images with other image files.

Why Replace Tableau's Standard Maps?

Tableau's dynamic map files are designed to balance high-quality graphic details while optimizing map rendering speed. But if standard maps don't provide the detail you require, Tableau makes it easy to replace the standard maps with custom maps provided by map services. With a little more effort, you can even use image files for map backgrounds.

There could be many reasons you might want to replace Tableau's standard map files. Custom maps can include non-standard geographic units or demographics particular to your organization. Perhaps you have spatial data that isn't large enough to be seen on a standard map. Examples might include warehouse or office layouts, retail planograms, building complexes, circuit board schematics, web pages, or even the human body. Or, the scale of your spatial data might be microscopic.

As long as you can define the vertical (y) and horizontal (x) coordinates within the spatial layout, it is possible to place data points on image files precisely. This process can be trivial—requiring only a few clicks—or require significant effort.

Replacing Tableau's Standard Maps to Enhance Information

The easiest way to replace Tableau's standard maps is with a web map service. Tableau can seamlessly integrate maps that adhere to an open source map protocol developed by the Open Geospatial Consortium. Many web mapping services adhere to the Web Mapping Service (WMS) protocol. Replacing Tableau's standard maps with WMS services is easy because the service provides the coordinate dimensions of the map, thereby eliminating the need for you to define the map coordinates. Tableau saves these map imports as Tableau Map Source (`.tms`) files.

If you have specific needs and the resources to set up your own WMS service, refer to the map section of Tableau's online manual. The sections on using WMS servers and exporting a WMS server are concise. Tableau's online manual provides clear instructions, so I won't review that method. I don't recommend using free WMS mapping services for critical reporting. These services come and go. The quality of the map files provided from free web mapping services is generally not very good, and the rendering speed can be unacceptably slow. If you need map images that are more responsive, investigate paid services.

Creating a Custom TMS File

Connecting to other map services that do not support the Web Mapping Server (WMS) protocol is also possible with a little more effort. Tableau provides a knowledge base article that explains how to create a Tableau Mapping Service (TMS) file. A blog post by Zak Gorman on the InterWorks blog provides some additional resources that will help you locate additional map styles.[3]

Tableau stores its standard map files in the Mapsources folder contained within the My Tableau Repository directory on your hard disk. Refer to the knowledge base article for more details. You can replace Tableau's standard maps with maps from another map service as long as the map server complies with the following requirements:

- Maps must be returned as a collection of tiles.
- The tiles must be Web Mercator projections.
- The tiles can be addressed by URL using the same numbering scheme as common web mapping services.

Even if you aren't an XML coder, you can create your own TMS file as long as you can find a suitable map service and modify a few variables. To do this, you'll need a text editor. Windows comes with a basic text editor called Notepad. The Mac's default text editor is simply named TextEdit. My favorite text editor is Sublime Text, and it is available in both Windows and Mac versions. You can even use Word to create the TMS file, as long as you save it as a plain-text (.txt) file. Start by copying the XML you see in Figure 5-18 into your text editor.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <mapsource inline="<boolean>" version="8.1">
3 <connection class="OpenStreetMap" port="80" server="<server-url>" url-format="<url-format>" />
4 <layers>
5 <layer display-name='Base' name='base' show-ui='false' type='features' request-string='/' />
6 </layers>
7 </mapsource>
```

FIGURE 5-18 *Copy this XML*

You need to modify three variables:

- The Boolean (true or false)
- The server-url
- The url-format

You can see the Boolean placeholder <boolean> in line 2 of Figure 5-18. In line 3, you will replace <server-url> and <url-format> with the specific values needed to render the base map layer. What exactly do these lines of script do? Line 2 defines whether the specified TMS configuration will be saved within the workbook. Answering true means that it will. A false answer means it will not.

If the false Boolean answer is provided, Tableau must have access to the TMS file saved in the Mapsources folder to display maps from the specified map server. Unless your environment requires otherwise, replacing the <boolean> script with true will provide the most flexibility—allowing your custom map to work whether you publish to an internal instance of Tableau Server or to Tableau Pubic. For this example, answer using the true variable.

Line 3 includes the server address and formatting information for the map tiles. There are at least three variables that need to be defined. To provide the server-url and url-format variables, you need a public map source that meets the map requirements defined earlier. Zak Gorman's blog post provides an excellent web resource for finding maps at http://mc.bbbike.org. Open your browser and navigate to that site; then type 1 and hit Enter on the keyboard. This will filter the view to include just one map instead of the four-map view that the site normally displays. Figure 5-19 shows the BBBike.org website with the OSM Toner Retina display map base layer and the related XML script that is needed for the line 3 variables.

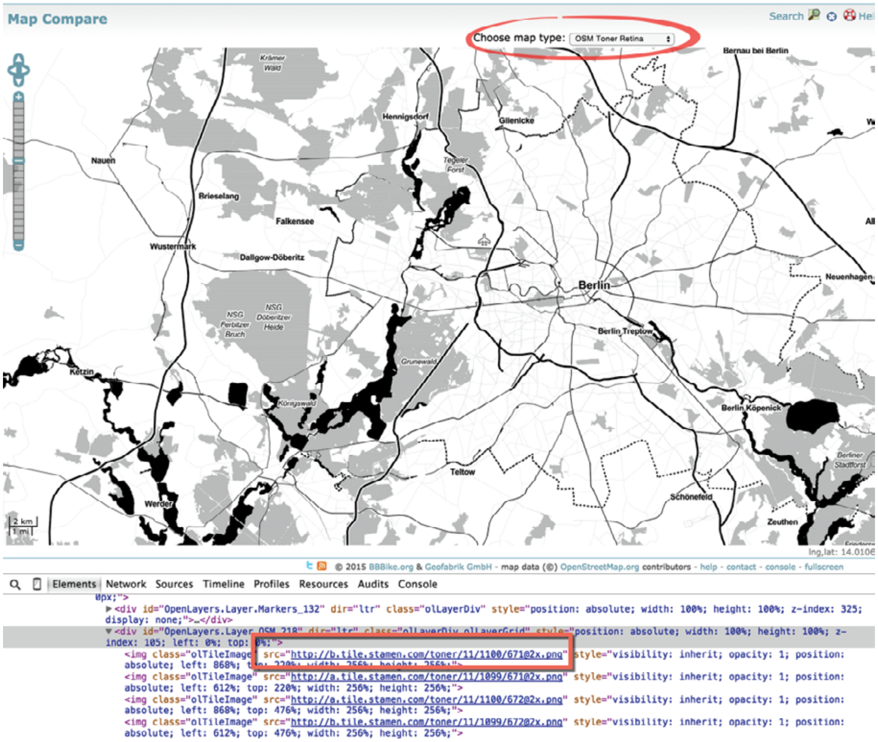


FIGURE 5-19 *OSM toner map with exposed XML*

Note that many other maps styles are available for view at this website. The parts of the script you need for the TMS file are contained in the red box highlight in Figure 5-19. The XML that follows the `scr= script http://b.tile.stamen.com` is the `server-url` information. The script immediately following that is the `url-format` information. You need to replace the hard-coded numbers (11/1100/671) with the `url-format` variables `/{Z}/{X}/{Y}`. The completed script should look exactly like Figure 5-20.

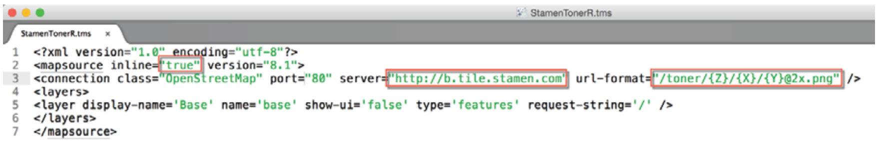


FIGURE 5-20 *Completed TMS file*

When you have completed the text file, as you see in Figure 5-20, save it with the `.tms` extension to the Mapsources folder on your computer. The default location differs depending on your environment for this folder:

- **Windows:** `C:\Users\<user>\Documents\My Tableau Repository\Mapsources`
- **Mac:** `/Users/<user>/Documents/My Tableau Repository/Mapsources`
- **Tableau Server:** `C:\Program Files\Tableau\Tableau Server\<version>\vizqlserver mapsources`

Once the custom TMS is saved in the correct repository on your computer's hard drive, you can replace the standard Tableau map with the StamenTonerR map layer. To create a customized map using the new custom TMS file, build a standard map view of the data and then replace the standard background map with the custom TMS file. Figure 5-21 shows a map of the Superstore for TDY2 data source, filtered for the state of New York, and then zoomed to the New York City area.

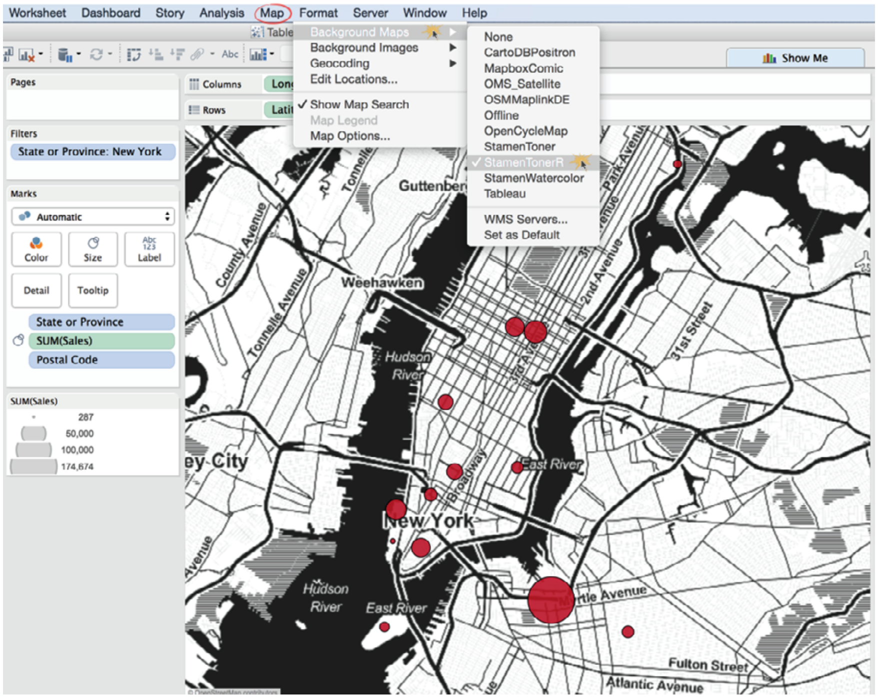


FIGURE 5-21 *Custom TMS file map*

As you can see in Figure 5-21, selecting the map menu/background maps menu option permits you to select the StamenTonerR map style you just created. My iMac has eight other custom TMS map layers. These files are included with the companion website files for this chapter. You can examine those files with your text editor or just place them into your Mapsources folder to experiment with other custom map layers. Figure 5-22 shows six different custom maps.

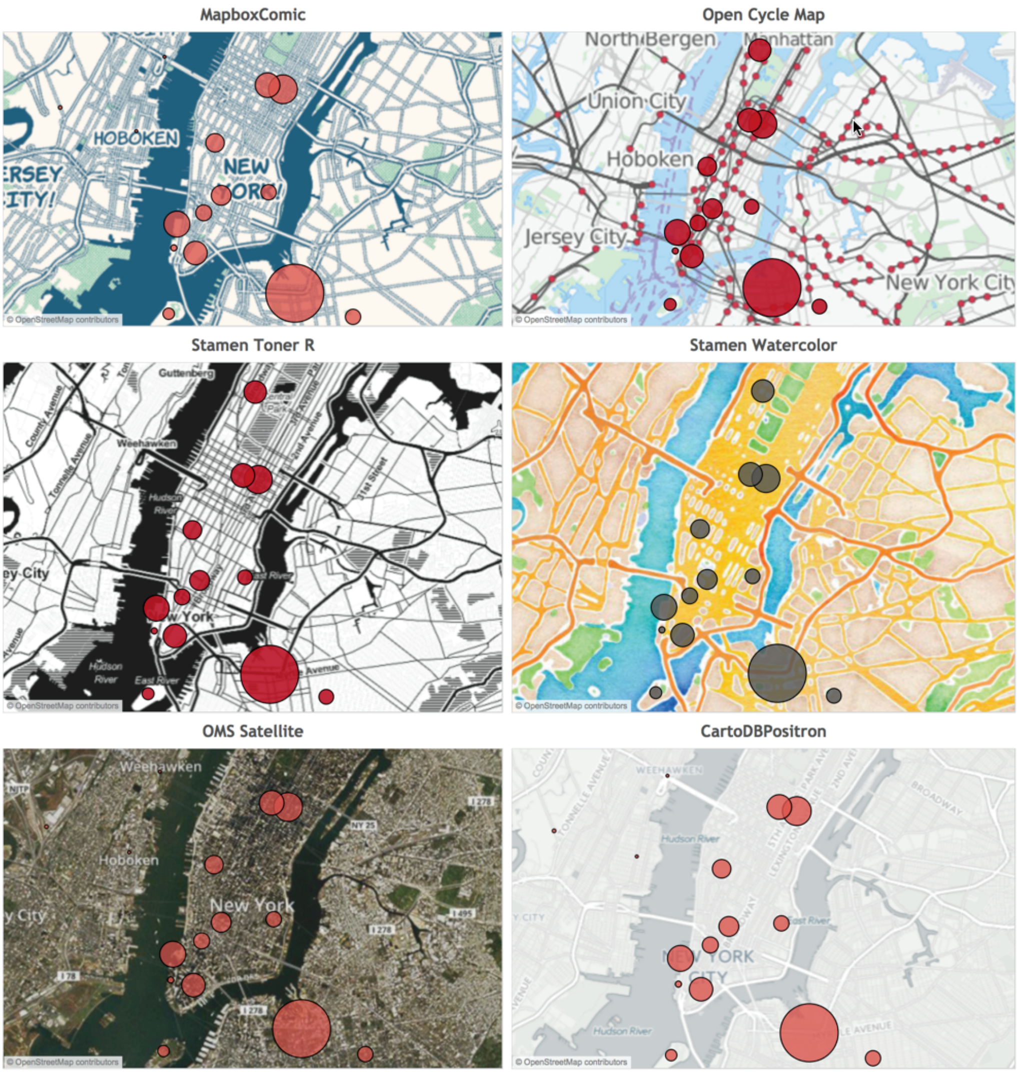


FIGURE 5-22 *Six custom maps*

Custom maps are fun. If you are doing a blog post and need some kind of special map to add visual interest to your map view, customizing your base map layer can add value. Tableau's standard maps are very good and perform well. Avoid using customized map layers to ensure the fastest possible load speeds. What if the space that you want to map is much smaller? Maps in Tableau are just scatter plots with a map background and a predefined coordinate system. Next, you'll learn how to create maps using any background image.

USING CUSTOM BACKGROUND IMAGES TO PLOT SPATIAL DATA

If the spatial data you want to plot isn't adequately portrayed on a map, you can also use image files as backgrounds. This option offers the advantage of infinite flexibility. But it requires more effort to implement because you will have to define the image boundary coordinates and the point coordinates for the items you want to place on the image.

Why Are Non-Standard Plots Useful?

Analyzing spatial data that's too small to be meaningful on a map may still yield interesting insight. Alternatively, if you know that your audience won't have access to the web, you can import a custom map image that contains specific details that are normally available only using online maps or not at all. For example, if you work in a large office and you want to analyze activity within the office, plotting employee movement over time within that space could help you improve the office layout. Retail merchandise managers are interested in tracking how the placement of products on shelves affects sales. Casino managers might be interested in seeing how the placement of cash machines within the casino affects revenue generation in different wagering areas. The options for spatial analysis in Tableau are limited only by your imagination.

Steps Required to Build a Custom Spatial Plot

Creating spatial analysis with image files requires additional steps that aren't needed when using images from web mapping services because the boundary coordinates on a map are based on the longitude (y-axis) and latitude (x-axis) of the map. Image files don't have the built-in coordinates that are provided by map services. The steps to use an image for a spatial plot are as follows:

1. Find or create an image file (`.jpeg` or `.png` formats work well).
2. Trim the image to include only the details you need.
3. Define the image boundaries (using any metric you desire).
4. Add point coordinates to your dataset.
5. Tweak point coordinates to precisely position marks on the image.

Assume you want to design a small office floor plan for use as a background image to map employee movement within the space. The level of precision you can achieve is dependent on your capture system. Figure 5-23 includes a floor plan image file that will be used to create a custom map background.

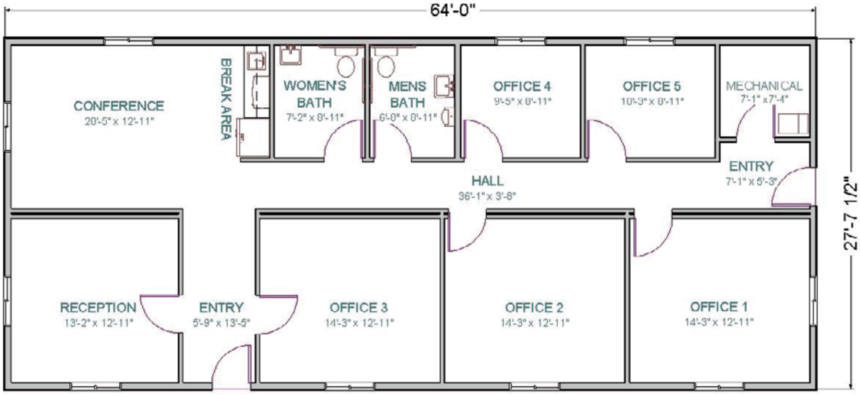


FIGURE 5-23 *Office floor plan image*

You can see that Figure 5-23 includes some peripheral areas outside of the floor plan in the image file. These areas are not part of the floor plan. Including areas outside of the floor plan in the image file complicates the image layout later because the dimensions of the office encompass only the office space. For this reason, it makes sense to trim the image to include only the actual floor plan image and not the surrounding white space. The example floor plan dimensions are 64 ft 0 in. × 27 ft 7.5 in. It makes sense to define the layout coordinates in inches. This will provide for precise placement of marks within each location desired within the office space.

Positioning Marks on a Non-Standard Map

Positioning the points within images can take a little time. A point coordinate system allowing for at least one position in each room in the floor plan provides the necessary level of detail for this example. Figure 5-24 includes the dataset with locations that will need to have point coordinates. The initial estimated point coordinates are in the table.

	A	B	C	D
1	Location	LocID	X	Y
2	Reception	1	120.0	72.0
3	Conference	2	120.0	264.0
4	Womans Bath	3	318.0	300.0
5	Mens Bath	4	330.0	300.0
6	Mechanical	5	720.0	312.0
7	Break Area	6	228.0	300.0
8	Office 1	7	696.0	72.0
9	Office 2	8	576.0	72.0
10	Office 3	9	300.0	72.0
11	Office 4	10	540.0	300.0
12	Office 5	11	648.0	300.0
13				

FIGURE 5-24 *Estimated office point coordinates*

The goal of this visualization will be to place the marks in a way that won't obscure the place labels that are included in the office layout image. The steps to finish a map of the office plan are as follows:

1. Connect to a dataset that includes the data in Figure 5-24.
2. Disaggregate the measures (so that each individual office location appears).
3. Add an image of the floor plan from Figure 5-23.
4. Edit the (X-Y) point coordinates to precisely position the marks.

After connecting to the dataset, the (X-Y) coordinate measure values should be placed on the column and row shelves. This will result in a scatter plot view with one mark. Tableau will express the sum of the (X-Y) coordinate values. The measures need to be disaggregated to display all of the rows in the dataset. This will cause one mark to appear in the view for each location in the floor plan. To do this, de-select the aggregate measure option from the Analysis menu. Figure 5-25 shows the view before and after disaggregating the measures.

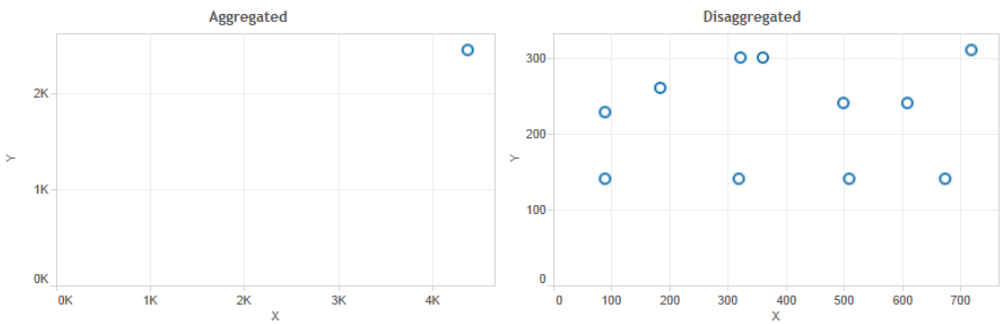


FIGURE 5-25 *Connecting to the data*

Next, the background image will be added to the floor plan by accessing the maps/background images menu and setting the boundary coordinates for the background image. Figure 5-26 shows the menus used to enter the coordinates and define how the image will be displayed.

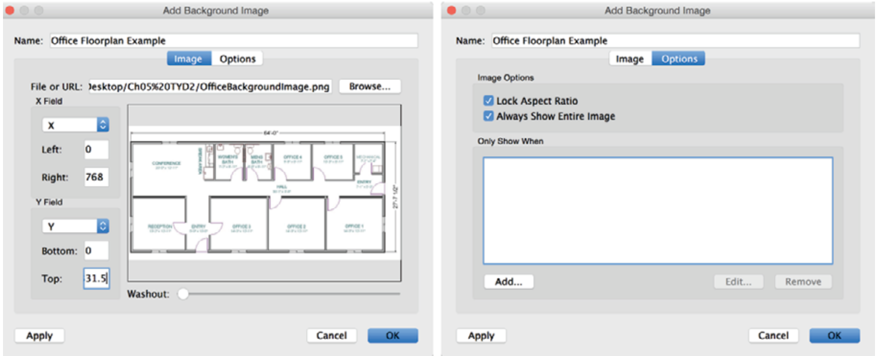


FIGURE 5-26 *Defining the image boundaries*

On the left side of Figure 5-26, you see the Add Background Image dialog box. Access this menu from the Map menu ➢ Background Images option. This is where the coordinates for the (X-Y) axis ranges are entered. The values are defined in inches. Selecting the options menu exposes the menu shown on the right of Figure 5-26. The selected options you see on the right of Figure 5-26 ensure that the image will not be distorted if its overall size is changed. Clicking OK will add the image to the view shown in Figure 5-27.

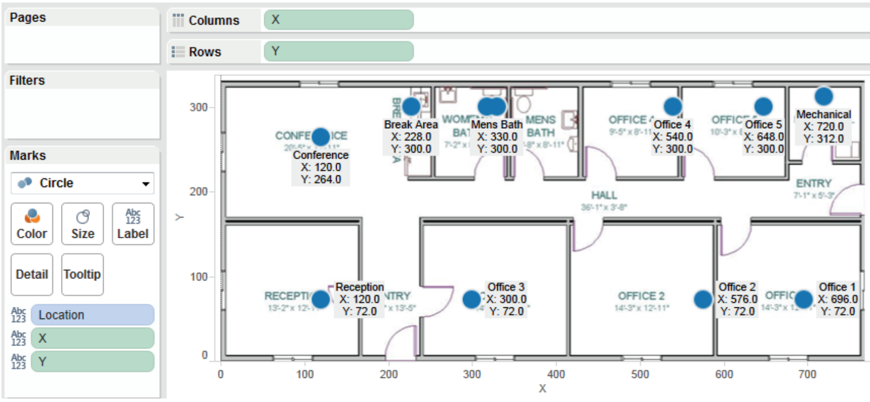


FIGURE 5-27 *Initial floor plan image*

As you can see in Figure 5-27, the initial estimates for the mark coordinates are a little off. A good way to reposition the marks is to open the source file next to your Tableau workbook. (It helps if you have a large monitor or dual-monitor setup when you do this.) With the source file opened next to the visualization, enter revised coordinate values in the source file (save it), and then refresh the Tableau view by right-clicking the data source in Tableau's data window. You will see the position of the mark change. Figure 5-28 shows the final adjusted coordinate layout.

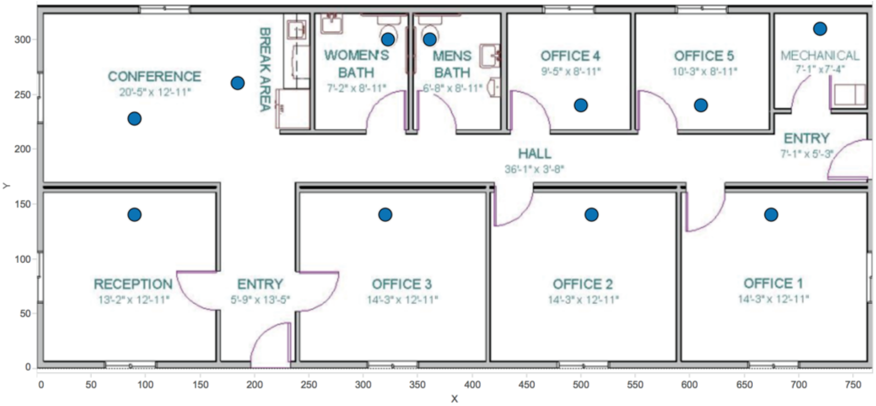


FIGURE 5-28 *Adjusted point coordinates*

See how precisely each mark is placed? The bathroom marks are right on the toilet seats. It normally requires a few tries to get the marks centered exactly because it's largely a trial-and-error process to position marks precisely on image files. Using point annotation on the marks also helps when you perform this task. With an appropriate capture system, the point coordinate data could be provided by a real-time system that captures staff position with time stamps to create the possibility of making an animated view of staff movement. This technique can be used in many different settings.

Publishing Workbooks with Nonstandard Geographies

If you use a custom map or image file and share a Tableau workbook file (`.twb`) with other people, you must also share the custom source map file (`.tms`) with them or they won't be able to see the custom map. Alternatively, you can distribute the workbook as a Tableau Packaged Workbook file (`.twbx`). Tableau Packaged workbooks save all your data and any custom (`.tms`) image files together—eliminating the need to provide the (`.tms`) file separately.

Shaping Data to Enable Point-to-Point Mapping

Mapping point-to-point details on maps requires that your data supports plotting and linking each point. Possible use cases for this style of presentation might include delivery truck routes, subway line activity, or city traffic flow. Similar presentations could be made using image files for spatial plots of areas too small for maps. Real-world applications may require automation to collect time-stamped geographic points and could require millions of records. To plot customized points on a map, your data must include:

- A unique key record for each location
- Location coordinates (longitude and latitude)
- Other interesting measures or facts related to the data

The next example will map point-to-point travel between two office locations. A line connecting each point will be color-encoded to express the duration in minutes at normal speeds required to traverse each segment between points. Figure 5-29 shows a sample dataset with the necessary details.

	A	B	C	D	E	F	G	H	I	J
1	Point ID	Location Name	Type	Latitude	Longitude	Start Time	End Time	Elapsed Time	Seg Duration	Comment
2	1	Stillwater Office	Office	36.105116	-97.104061	13:00:00	13:00:30	0.0	0.0	Starting point in front of the Stillwater office
3	2	Stillwater entrance	Roadpoint	36.105144	-97.104984	13:00:30	13:01:00	0.1	0.1	Turn right on to S. Sangre Road
4	3	OK-51 and S Sangre Rd	Roadpoint	36.11603	-97.105223	13:01:00	13:02:00	1.0	0.9	Turn left on to OK-51
5	4	I-35 Ramp, South	Roadpoint	36.115686	-97.345358	13:02:00	13:16:00	15.0	14.0	Turn right on exit ramp to I-35 South
6	5	I-35 South	Roadpoint	35.812803	-97.416398	13:16:00	13:36:00	35.0	20.0	Continue on I-35 South
7	6	Highway	Roadpoint	35.609447	-97.425155	13:36:00	13:49:00	48.0	13.0	Continue on I-35 South
8	7	Highway	Roadpoint	35.544237	-97.458242	13:49:00	13:51:00	50.0	2.0	Take I-44 West toware Lawton/Amarillo
9	8	Harrison Ave	Roadpoint	35.529691	-97.514078	13:51:00	13:59:00	58.0	8.0	Task the south exit to 1-235 South
10	9	NE 4th Street, OKC	Roadpoint	35.473929	-97.508934	13:59:00	14:01:00	60.0	2.0	Turn right on Harrison Avenue exit
11	10	Walker Avenue, OKC	Roadpoint	35.471846	-97.511866	14:01:00	14:04:00	63.0	3.0	Veer right on 4th Street
12	11	OKC Office	Office	35.472169	-97.520441	14:04:00	14:06:00	65.0	2.0	Ending point - OKC office

FIGURE 5-29 *Point-to-point details*

The route in Figure 5-29 starts in Stillwater, Oklahoma, at point 1 and finishes in Oklahoma City, Oklahoma, at point 11. If there were multiple records for each location, a combination of the location, the key record, and the time stamp could be used to identify unique points. In that case, the measures would need to be disaggregated (by accessing the analysis menu and deselecting aggregate measures) so that the plot displays all the different times each location was visited. Because the sample dataset includes only one record per location, there is no need to disaggregate the measures to display all the records. Figure 5-30 shows the completed point-to-point plot.

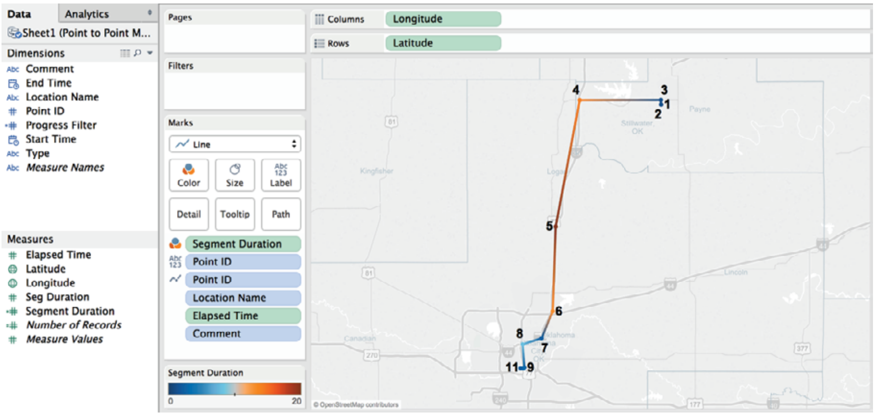


FIGURE 5-30 *Map view of the route*

Notice that the line mark type is selected on the Marks card. Point ID defines the order of the route and must be placed in the path button so that the line connects each point in the correct order. Placing the Point ID on the label shelf causes the location point ID number to display in the map as well. The line connecting the route is color encoded by segment time duration. Because this covers a large area, it would be helpful to provide two additional map views that zoom into the local areas surrounding the origin and destination, making more granular street-level detail visible, as in Figure 5-31.

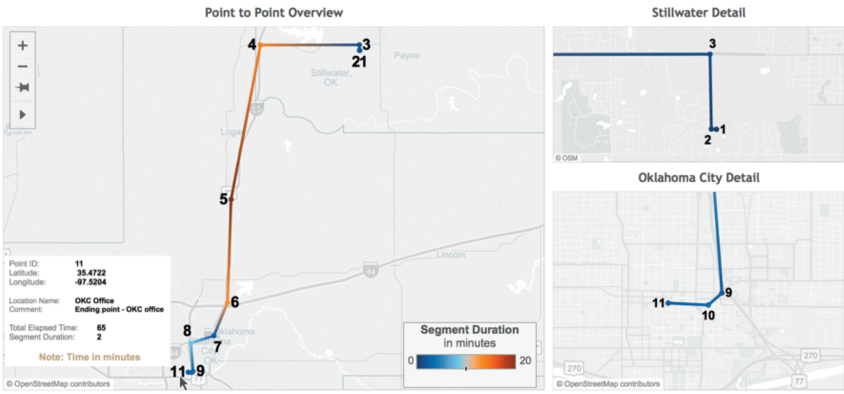


FIGURE 5-31 *Route dashboard*

The maps on the right of the dashboard show more road details around the starting and ending points. Pointing at any mark exposes a tooltip with additional information about the location. The main map view on the left shows the tooltip related to the end point of the route.

Animating Maps Using the Pages Shelf for Slider Filters

The most convenient way to animate views is to utilize a date/time dimension on the Filters shelf (or the Pages shelf) to increment time forward and backward. Creating a quick filter using a continuous dimension presents the user with a slider-type filter that will work well for animating the view. The Pages shelf goes beyond quick filters by enabling an auto-incrementing filter. The Pages shelf works well in Tableau Desktop and Reader; however, it is not supported in Tableau Server.

The example in Figure 5-31 doesn't include a date/time dimension, but there is a single key record for each point ID. The example map can be animated by placing the Point ID field on the Filters shelf. Figure 5-32 shows the Point ID added to the desktop as a continuous slide filter.

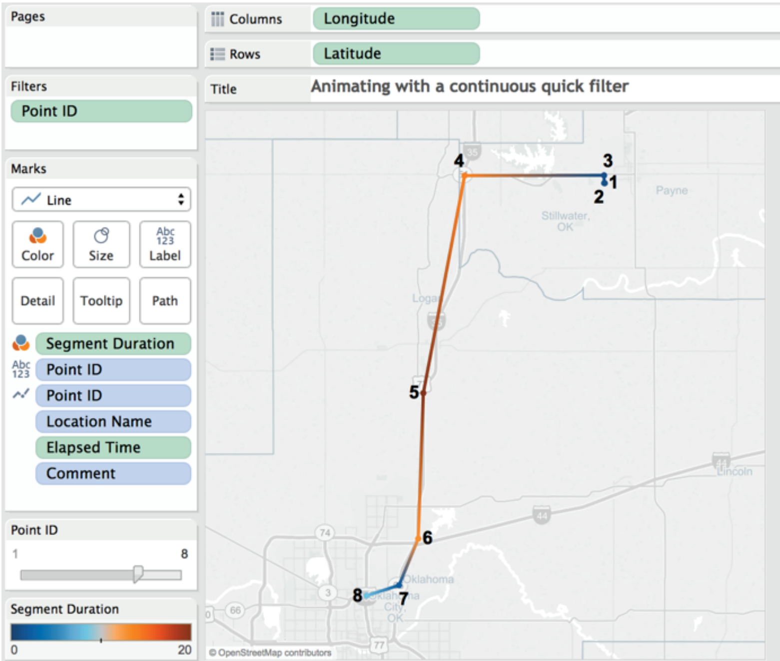


FIGURE 5-32 *Animating a map*

The route line now ends at point 8, as specified in the quick filter. Dragging the slider to the left or right animates the route map manually. Notice that the Point ID pill on the Filters shelf is green. This color indicates that the point ID was changed to a continuous dimension. The Point ID field was initially a discrete dimension. Using a discrete dimension for the quick filter would not facilitate animating the view. The filter was changed from discrete to continuous by right-clicking the Point ID pill on the Filters shelf and selecting Continuous.

Tableau's standard maps and automatic geocoding should meet your needs most of the time. Through the use of custom geocoding and custom maps, you'll be able to create even more detailed geographic analysis. And, by using custom map backgrounds from web services or image files, you can fully customize the detail and appearance of map backgrounds.

In the next chapter, you learn how to create an ad hoc analysis environment with Tableau.

NOTES

1. John Noble Wilford, *The Mapmakers* (New York: Vintage, 2001) 411.
2. "Connecting to the Tableau Map Service," last modified May 28, 2015, accessed May 31, 2015, `http://kb.tableau.com/articles/knowledgebase/connect-to-tableau-map-service`.
3. Zak Gorman, "Bringing a Custom Map into Tableau in 10 Minutes or Less," The InterWorks Blog, May 11, 2015.

CHAPTER 6

Developing an Ad Hoc Analysis Environment

The creative process, in essence, is an individual in dialogue with themselves and the work. The painter, when at a distance from the easel, can assess and analyze the whole of the work from this vantage. He scrutinizes and listens, chooses the next stroke to make, then approaches the canvas to do it. Then, he steps back again to see what he's done in relation to the whole. It is a dance of switching contexts, a pitter-patter pacing across the studio floor that produces a tight feedback loop between mark-making and mark-assessing.

—Frank Chimero[1]

In Chapter 2, you learned that Tableau connects to a wide variety of Data sources, and this is further extended through the Data Interpreter and Data blending. Chapter 3 introduced the Show Me button, trend lines, reference line, and how filters, sets, grouping, and hierarchies can be used to present information meaningfully—for facts and dimensions that are included in views.

In this chapter, I discuss how Tableau's design enables discovery work, how discovery differs from reporting and analysis, and how this combination of creative and analytical discovery can lead to insight in a more engaging flow than what is provided by traditional business information tools.

You will also learn about Tableau capabilities that encourage ad hoc analysis via easy-to-use forecasting, building flexible data views using parameters, and allowing information consumers in Tableau Server to change existing views or make their own analysis from secure access in a web browser.

DATA DISCOVERY AS A CREATIVE PROCESS

When discussing discovery using Tableau, it's important to distinguish between data analysts whose role is predominantly performance analysis and information consumers who normally don't do analytical work. Analysts have a better grasp of the underlying data structures. Information consumers are the operational experts. When Tableau enters an organization, it is not unusual for people with operational responsibilities to take on more of the analysis and discovery work. This benefits the information technology support staff that is burdened with managing the technical infrastructure and normally is not as knowledgeable about the operational details.

A painter has knowledge of the subject, the setting, and the tools that will be used to create the painting. Similarly a good data analyst has knowledge of the source data, the domain needs, and the analytical tools used for the discovery work. Tableau's design enables the analyst to perform analysis in the same way a painter approaches the creation of artwork by providing immediate feedback when something is added to a view. How fast and how competently one can do discovery work is dependent on the person's knowledge of the data, knowledge of Tableau, and the quality, granularity, and completeness of the data sources being used.

Learning to do discovery work using Tableau is attainable to everyone with the inclination to do it. Tableau isn't a barrier. It's designed for nontechnical staff to use. Its ability to provide immediate feedback in the flow of analysis accelerates learning and enables staff that doesn't come from technical backgrounds to perform.

PREPARING YOUR TEAM FOR SUCCESS

Providing training to the people in your company is the best way to ensure success. The training should include the less technical staff that may not possess deep knowledge of the existing systems, and it should also include the most experienced technical resources that are responsible for the source data within the company. Tableau's design is fundamentally different from traditional business information tools. Including your technical staff in basic training will help them understand how the underlying database design may need to change in order to facilitate analysis.

Knowledge of database schema, coding, and your company's security requirements doesn't always correspond with the ability to build great dashboards, create meaningful reports, or perform insightful analysis. If it did, Tableau wouldn't be as popular as it is. Knowledge about the business context is important.

By including a broad range of analysts, technical staff, and end users in your initial Tableau deployment team, you provide the necessary foundation for success. Less technical staff need to learn the basics related to your data environment. The more technical resources need to learn how Tableau differs from the tools they are accustomed to using.

Everyone needs to develop an understanding of how Tableau can enable a much larger number of people to explore data and make actionable discoveries.

QUALITIES OF A GOOD DATA ANALYST

A good Tableau analyst doing discovery analysis with Tableau doesn't need to be a technical expert, but there are some personality qualities and skills that are normally part of the makeup of a good analyst who proves to be successful using Tableau. These include:

- Curious
- Detail-oriented
- Skeptical
- Open-minded
- Artistic

Discovery is undefined analysis. People who are naturally curious like going where others haven't been. This is important because the most important discoveries can involve a novel use of disparate data sources. There won't be a set of procedures to follow. A good analyst doing discovery work with Tableau needs to be comfortable working with loose definitions and undefined deliverables. The best discoverers are excited by the prospect of diving into a large unknown set of data.

Being detail-oriented and a little skeptical when creating the initial round of visualizations and dashboards is healthy. False assumptions can lead to inaccurate or misleading conclusions. A good analyst is diligent and digs a little deeper to validate what the discovery work initially indicates by applying additional rigor to the analysis.

Good analysts are open about the data sources being used, the completeness of their vision, and other possible explanations that could explain any outliers, trends, or correlations that are uncovered by their analysis. And, before they present their findings to a larger group, good analysts seek feedback from the most knowledgeable technical or operational resources in the domain of data that they are working with. They are not too proud to use good ideas that come from other sources.

Finally, when they are ready to present their finding to a broader audience, good analysts take the time to polish the appearance of their work. More people will consume the findings in any discovery work if the data is presented in an interesting and presentable format.

DOING EFFECTIVE DISCOVERY WORK

There is no single blueprint for doing discovery work with Tableau. After using Tableau for nearly eight years, I have my own way of doing discovery work that generally follows a few basic steps. First, I get access to a granular data set of sufficient size and diversity to provide the potential for an interesting discovery. If one to two years of granular history is available, that's desirable.

The data should also have sufficient dimensionality, including time, geographic features, and descriptive information about a service, product, vendor, staff, or process. Sufficient dimensions regarding the location, business unit, and organizational hierarchy all add grist for discovery.

My preference is to start the analysis by using Tableau measure names and measure values to display all of the dimensions and measures available in the dataset in a text table. This provides a high-level view of the range of values across the domain of dimensions and provides a foundation for assessing the reasonableness of any indications that come from later analysis. After that, I proceed to build as many different views as the dataset supports. After doing that, it's a good idea to step away for a few minutes. Then, return and start to assemble small multiple views and interactive dashboards in order to find interesting patterns and outliers in the data. This is an open-ended process. Within a couple of hours, some stories will begin to emerge.

Some of the best success stories from Tableau customers are discovery stories. Tableau's visual metaphor and its ability to provide quick and understandable visual feedback are engaging to analysts and end users.

WHAT IT CAN DO TO HELP

The more technical resources in your organization have a very important role to play in encouraging discovery work. First, they need to ensure that the data people are using is complete and accurate. They also need to ensure that the data playground is secure but not so locked down that people can't access the data on very short notice. Creating a very rapid response capability (a user hotline) is a great way to leverage limited technical resources across a much larger group of less technical users. For this to be effective, the response must be truly rapid. That means minutes, not hours, days, or weeks. And the response times should be measured and reported to the larger Tableau community. In this way, the IT team will be perceived as enabling instead of as a road block to be avoided. For those of you working in IT, it is important to consider the staffing of this help line to ensure that the response is timely.

SPREADING DISCOVERY TO INFORMATION CONSUMERS

Once your analysts start generating interesting discoveries using Tableau, a natural demand will build, and more people will want access to the tool. In most cases, the majority of people won't need desktop licenses. Tableau Server provides plenty of capability for end users to analyze data. There are also techniques for analysts building dashboards and interactive workbooks that facilitate discovery by information consumers. The remainder of this chapter introduces Tableau's forecasting capabilities, parameter controls, and instructions on how information consumers can create their own formulas and build new views within Tableau Server.

GENERATING NEW DATA WITH FORECASTS

Creating forecasts is easily accomplished by dragging the forecast tool from the Analytics tab in the Data pane. The resulting figures can be exported, revised, and possibly even added to your data source—providing a fast and easy way to model the future based on past history.

How Tableau Creates Forecasts

Tableau generates forecasted values by using time-series data that is included in your worksheet. Figure 6-1 shows a time series chart that includes forecasted values.

Forecasted values are presented in a lighter color than the actual values. The forecast values in Figure 6-1 can be added by right-clicking within the worksheet and selecting the Forecast ➢ Show Forecast menu option. It's even easier to add the forecast by selecting the Analytics tab in the Data pane and then dragging the Forecast option within the Model section of that pane into the view.

Forecasting Options

Tableau can forecast data in a variety of ways and will automatically select what it feels is the best method. If you don't want to accept the default, edit the forecast model by right-clicking the worksheet and selecting the Forecast Options menu. Tableau provides the following forecast trend models:

- Forecast Length
- Source Data
- Forecast Model
- Prediction Intervals

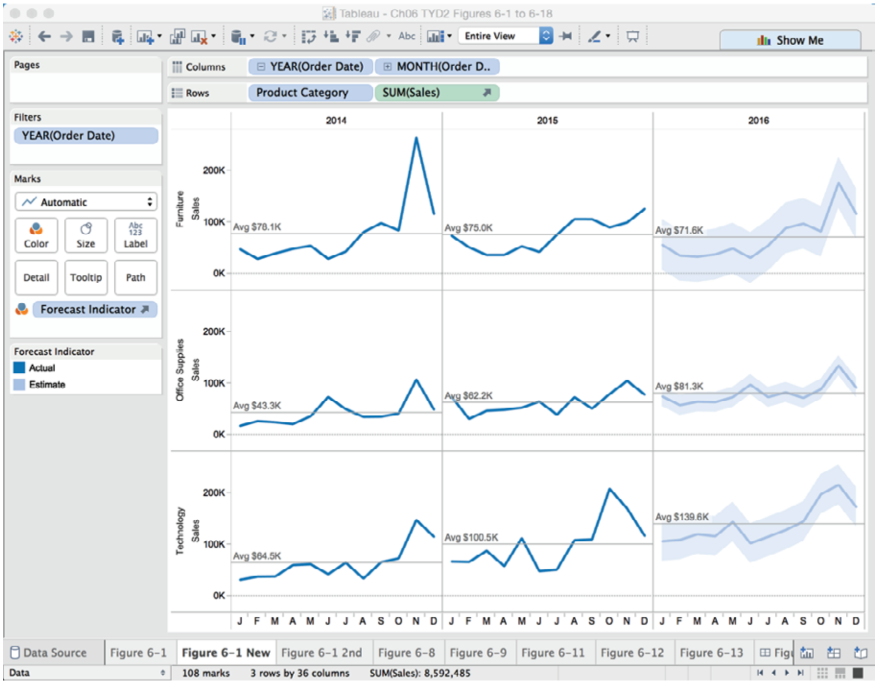


FIGURE 6-1 *Time series with forecast*

Depending on the amount and granularity of the historical data, each option will generate different results. The Forecast Options menu includes several other variables that can be adjusted. You can see the Forecast Options menu in Figure 6-2.

At the top of Figure 6-2 you can see that, by default, Tableau will generate a 12-month forecast, but it is possible to forecast a specific number of periods into the future. The number of periods that Tableau forecasts is dependent on the date range in your view and the data aggregation level presented. The Ignore Last setting allows you to omit incomplete historical data so that it won't skew the forecast results. Checking the Fill in Missing Values with Zeros box will prevent null values from corrupting the forecast.

Review and Present Forecast Quality Metrics

You can examine the quality of the forecast that Tableau generates by right-clicking within the workspace and selecting the Forecast ➢ Describe Forecast option. Figure 6-3 shows the Summary tab.

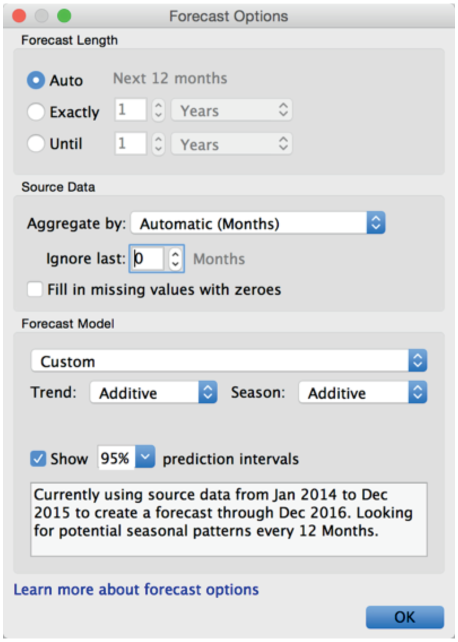


FIGURE 6-2 *The Forecast Options menu*

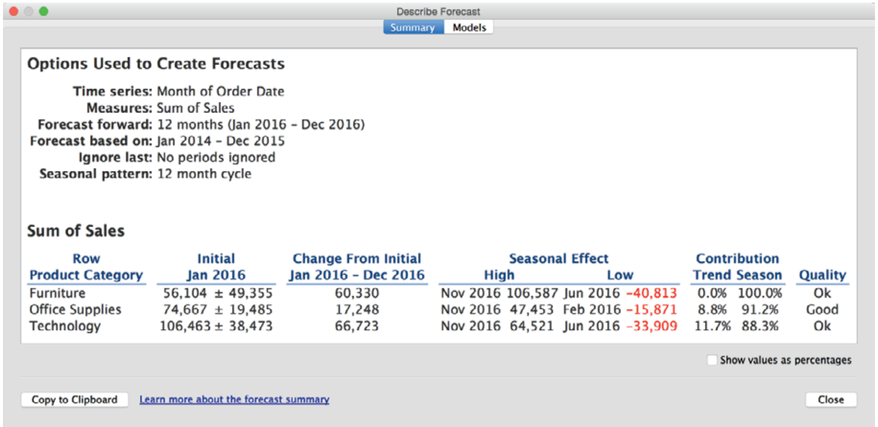
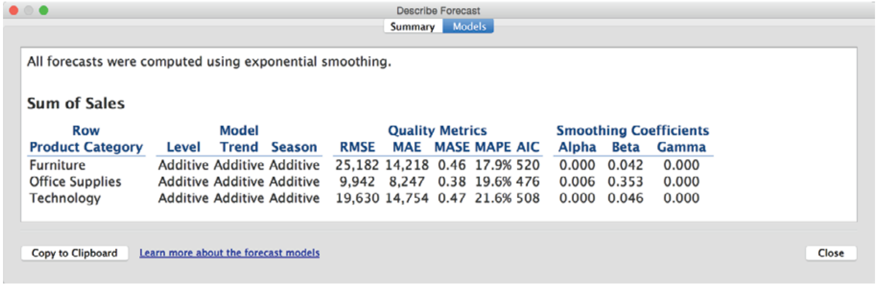


FIGURE 6-3 *Describe Forecast Summary*

The Summary tab provides details regarding the precision and quality of the forecasted values. You can choose to express forecast precision as number or percentage ranges. Forecast quality is described as poor, OK, or good. Clicking the Models tab exposes more detailed quality metrics, as you see in Figure 6-4.


Describe Forecast
Summary Models

All forecasts were computed using exponential smoothing.

Sum of Sales

Row	Model			Quality Metrics					Smoothing Coefficients		
Product Category	Level	Trend	Season	RMSE	MAE	MASE	MAPE	AIC	Alpha	Beta	Gamma
Furniture	Additive	Additive	Additive	25,182	14,218	0.46	17.9%	520	0.000	0.042	0.000
Office Supplies	Additive	Additive	Additive	9,942	8,247	0.38	19.6%	476	0.006	0.353	0.000
Technology	Additive	Additive	Additive	19,630	14,754	0.47	21.6%	508	0.000	0.046	0.000

Copy to Clipboard Learn more about the forecast models Close

FIGURE 6-4 *Describe Forecast model*

Tableau's forecasting model weights recent history more heavily. The statistical models for the different quality metrics presented in Figure 6-4 are defined in detail in the Tableau desktop manual. The smoothing coefficients for alpha (level smoothing), beta (trend smoothing), and gamma (seasonal smoothing) refer to the amount of smoothing applied. Values closer to one are smoothed less than the lower values. If the value is very close to zero, a lot of smoothing was performed.

Adding Quality Metrics to Tooltips in Visualizations

By dragging and dropping the forecasted measure from the Measures shelf in the Data window to the Marks card detail window, this allows you to modify information contained in the tooltip to include quality and precision metrics. Figure 6-5 shows the placement.

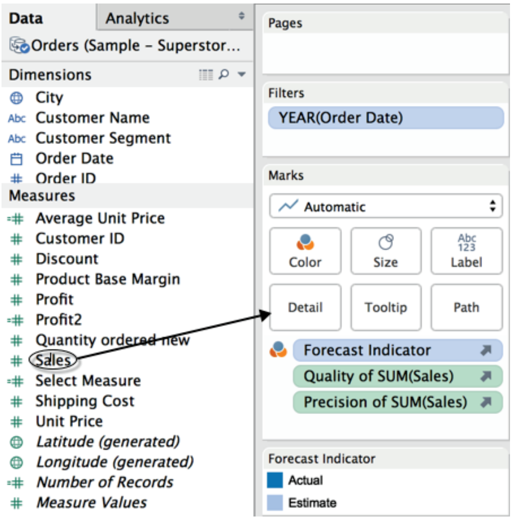


FIGURE 6-5 *Adding sales to the Detail button*

Once sales are dropped into the Detail button (see Figure 6-5), you can modify the value presented by right-clicking the pill in the Marks card and making a selection, as you see in Figure 6-6.

FIGURE 6-6 *Enabling quality metrics*

Figure 6-6 shows the selection options for adding quality and precision metrics so that they become available for tooltips in the chart. Each individual selection desired in the view will require another sales measure to be dropped on the Details button. In this example, two metrics were added—Quality and Precision. Figure 6-7 shows the resulting tooltip.

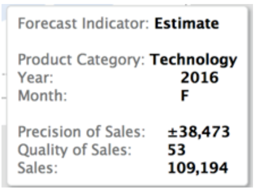


FIGURE 6-7 *Tooltip with Quality and Precision metrics*

This tooltip is exposed to users when they hover their mouse over the mark. Figure 6-7 shows that a Quality of Sales forecast metric and a Precision of Sales forecast metric have been added to the tooltip. The Quality metric range is from 0 to 100 (higher numbers mean better quality). The Precision metric is expressed as a value range and provides the 95 percent prediction interval for the forecast—a measure of the potential volatility of the Forecast value. In this example, the Value Range refers to a Sales dollar range of 49,611.

Exporting Forecasts

Exporting Tableau-generated forecasts can be a time-saver for developing more nuanced forecasts. One way to accomplish this might be to duplicate the original view in Figure 6-1 as a text table and then export the view using the menu option for Worksheet ➢ Export ➢ Text table to Excel. Figure 6-8 shows the resulting spreadsheet values.

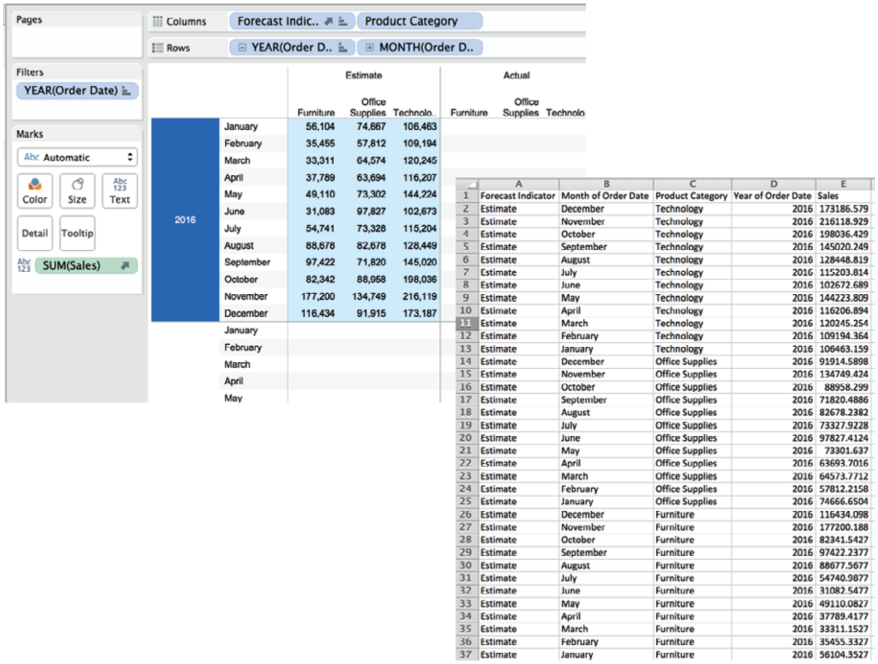


FIGURE 6-8 *Exported forecast in a spreadsheet*

Alternatively, it is also possible to go to the menu option Worksheet ➢ Export ➢ Data to an Access database. Using either method enables you to adjust the forecasted values more specifically and perhaps add those altered figures into your main database in their own field. Tableau's forecasting model isn't intended to replace sophisticated statistical forecasting tools. It provides an easy-to-use way to create forecasts along with quality and precision metrics to access the quality and precision of the resulting estimates.

Creating forecasts in views that you publish to Tableau Server and share with others is one way to stimulate an ad hoc analysis environment for users who only have access to the view via Tableau Server. In the next section, you learn how to create parameter controls that enable Tableau Server users to change measures and dimensions in views or dashboards.

PROVIDING SELF-SERVICE AD HOC ANALYSIS WITH PARAMETERS

Parameters enable those consuming reports to change the context of views with Quick Filter–like controls. Report builders design parameters into views when the report is created in Tableau Desktop. Parameters create a pathway for non-technical consumers to conduct ad hoc analysis by changing what and how facts and dimensions are displayed—within the boundaries of the designer's intended usage. Concerns regarding the efficacy of self-service analysis are minimized because the report designer controls what changes are permitted.

WHAT ARE PARAMETERS?

Parameters are variables that allow users to alter the content of a formula or change a dimension or measure contained in the view. Parameters create a powerful means for changing normally static values into dynamic entities that facilitate ad hoc discovery without the need for changing the design of the view.

HOW CAN PARAMETERS BE USED?

The ways in which parameters can be used are limited only by your imagination. Tableau provides some basic parameter controls by building them into different contexts that commonly benefit from the use of a variable. Creative report designers can dream up a myriad of other ways to use this powerful feature by building formula variables that control the facts in view, the dimensions that appear, or the length and granularity of time series data. Anywhere that you can place a field in Tableau Desktop is a potential repository for a parameter control.

BASIC PARAMETER CONTROLS

Parameter controls first appeared in Tableau several years ago, and they have become a popular feature. To make parameters easier for novice users, Tableau created basic parameter types that are built into typical use cases that benefit from variables. These include:

- Reference line parameters
- Bin size parameters (for histograms)
- Ranking parameters (in value comparison views)

The steps required to add basic parameter controls to a view are straightforward provided you know that they exist. Figure 6-9 is a scatter plot that includes basic parameter controls for the vertical and horizontal reference lines.

There are two reference line parameter controls in Figure 6-9 that enable a user to change the location of the reference lines. These variable controls are built into the same dialog box where standard fixed reference lines are created. To create the parameter, right-click the axis and select Add Reference Line to expose the reference line dialog box, and then in the Value box select the Create a New Parameter option. Figure 6-10 shows the added reference dialog box on the left. Selecting the Create a Parameter Option in the Value menu exposes the Edit Parameter window on the right of Figure 6-10.

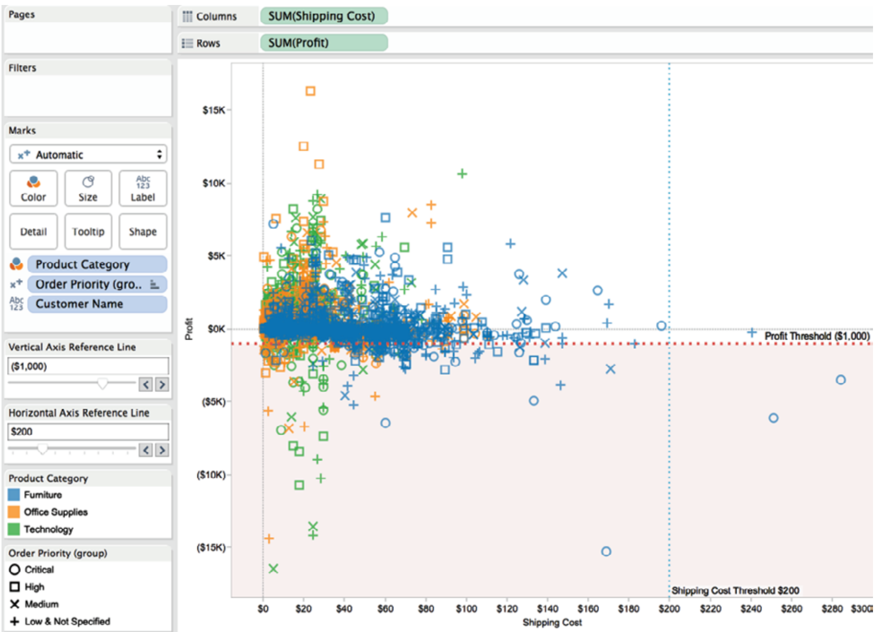


FIGURE 6-9 *Reference line parameters*

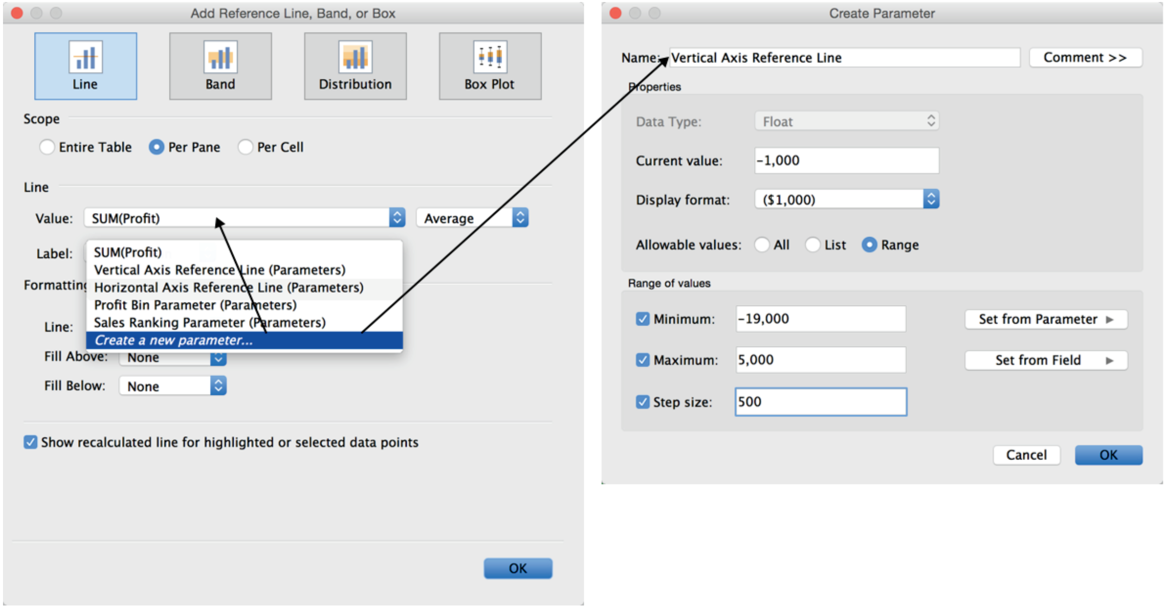


FIGURE 6-10 *Creating a reference line parameter*

In the Create Parameter window, you name and define the parameter. A comment field can be used to hold notes describing the parameter. The Properties section is used to define the parameter type. In this example, a floating decimal point value is selected, the display format is currency, and the parameter is defined for a range of values with a specified increment defined by step size. Clicking OK adds the parameter control. Refer to Figure 6-9 again and notice that the parameters allow users to move a slider to change the position of the reference lines.

A second basic parameter type is available for making variable bin sizes in histograms. The view seen in Figure 6-11 was initially constructed by selecting the Profit field and then picking the histogram chart type from the Show Me menu. This resulted in a bin size of $5,000 with a very large concentration of items in only two bins.

Right-clicking the Profit Bin dimension that Show Me automatically created enables you to select a parameter option for bin size in a manner similar to the last example. By defining a smaller step size for each bin, you can see more granular views of the profit bins. This is always desirable when viewing histograms. The view in Figure 6-12 shows the Profit Bin Parameter set to a smaller value.

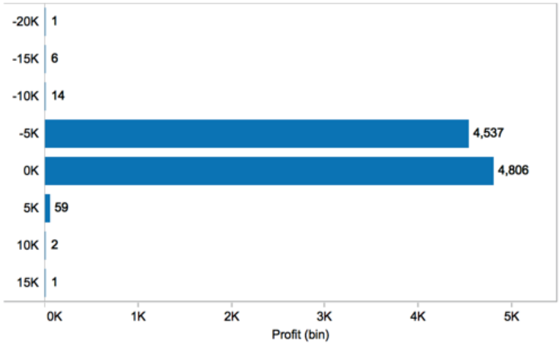


FIGURE 6-11 *A basic histogram*

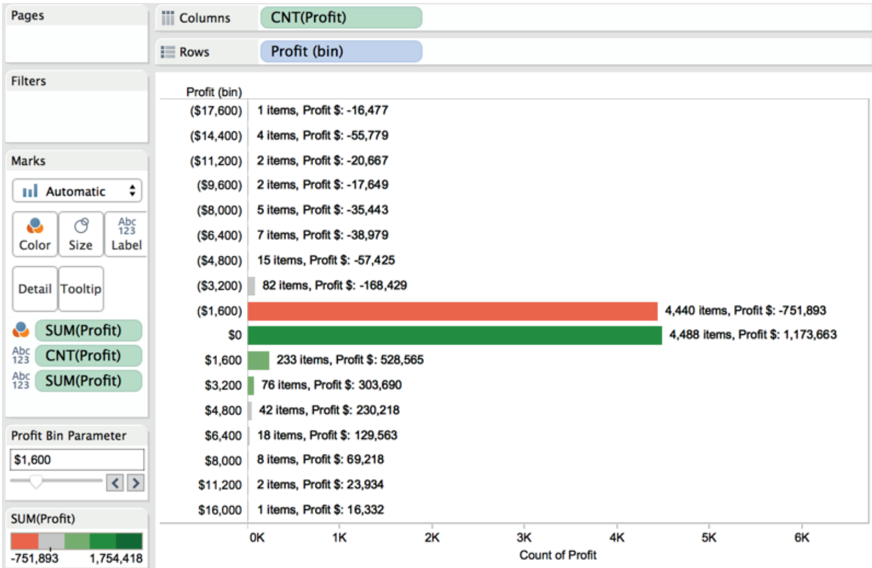


FIGURE 6-12 *Parameterize bins in a histogram*

The capability to vary bin sizes within histograms can be very useful. As you can see in Figure 6-12, labeling was also added to each bar, providing an item count and the total profit or loss expressed in each bin.

Another type of basic parameter is built into charts for creating variable-sized rank lists. Figure 6-13 shows a bar chart comparing sales values by customer with a year filter. Notice the Parameter Control provides a variable rank list size for the top customers.

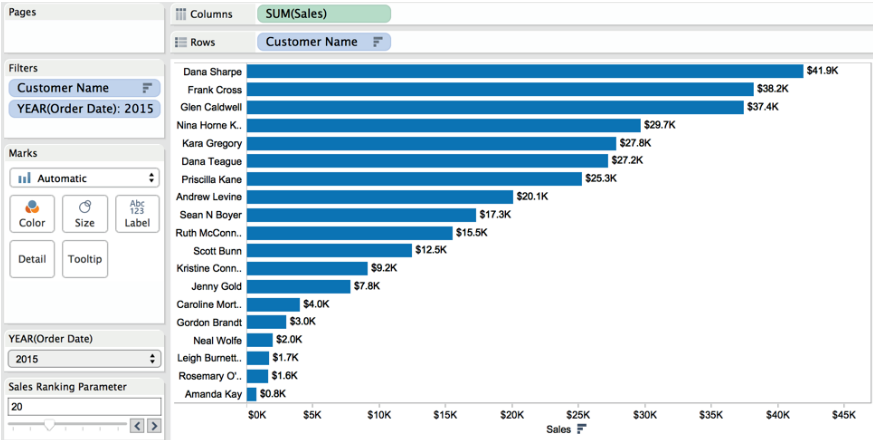


FIGURE 6-13 *A top rank parameter*

In this example, the Parameter Control was invoked by right-clicking the customer name pill on the Rows shelf, selecting the Top filter tab, and defining the parameter range value for the number of customers you want to display in the bar chart. This isn't the only way you can create a flexible rank list, but it is one of the easiest methods. Figure 6-14 shows the ranking parameter menu used to create the Sales Ranking Parameter control for the bar chart.

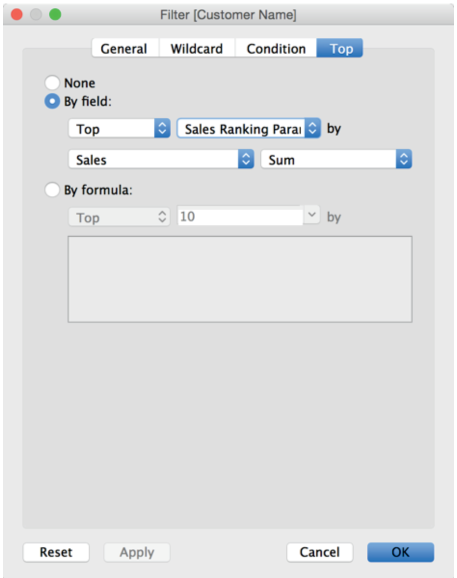


FIGURE 6-14 *Quick filter with sales parameter*

By placing the sales parameter into the normally static top rank definition dialog box, this creates a flexible rank list. This isn't the only way you can create a flexible rank list, but it is one of the easiest methods.

ADVANCED PARAMETER CONTROLS

More advanced parameter controls can be created that provide greater flexibility. The steps required to create advanced parameters are:

1. Create the parameter control.
2. Expose the parameter control on the desktop.
3. Create a calculated value using the parameter control.
4. Use that calculated value in the view.

Advanced parameters do require a little more effort, but they are easy to build once you become familiar with the process. One of the most common use cases for advanced parameter controls is to permit users to alter the measure being plotted in a view. Figure 6-15 shows a time series chart that is currently displaying sales over time, but with a parameter control that allows the end user to change the measure in view to profit, discount, order quantity, or shipping cost.

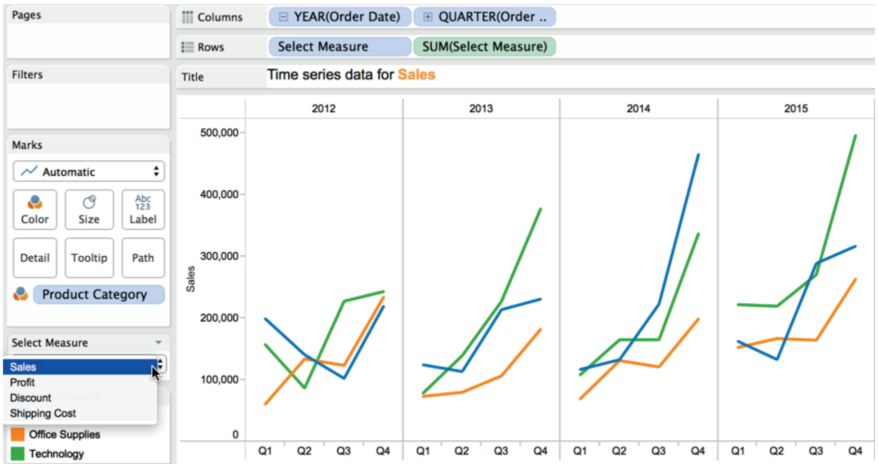


FIGURE 6-15 *A parameterized time series chart*

The Parameter Control contains strings identifying each different measure. Selecting the Parameter Control's drop-down menu exposes each measure—enabling the user to change the measure displayed in the time series chart. Also notice that the view contains parameterized headings for both the report title and the axis label. Take a look at the step-by-step creation of the Parameter Control for this example.

Create the Parameter Control

You can create the parameter control from the calculation menu or directly from the Data Shelf by right-clicking within the Dimensions shelf and selecting the Create Parameter menu option. This exposes the Parameter menu from which you can enter the options you see in Figure 6-16.

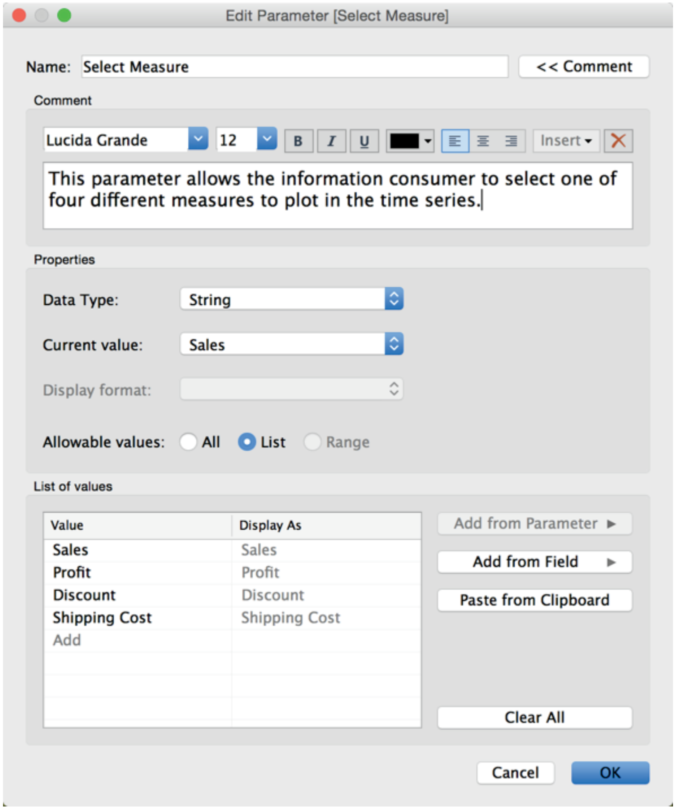


FIGURE 6-16 *Defining a string parameter*

Notice the parameter is named Select Measure, which is what appears in the parameter Quick Filter title. It may seem counterintuitive that the parameter definition is for a String type versus a number. A String is necessary to contain the field names of the measures that will be enabled in the view. This step only defines the filter box that is exposed on the desktop.

Exposing the Parameter Control

To make the Parameter Control available to information consumers, it must be exposed to the worksheet. This is done by pointing at the parameter title that appears on the Parameters shelf, right-clicking, and selecting the menu option Show Parameter Control. After this step, the control is available on the desktop to make selections.

Create a Calculated Field Using the Parameter Control

Making a Calculated Field that uses the Parameter control brings it to life. Figure 6-17 shows the completed Calculated Field.

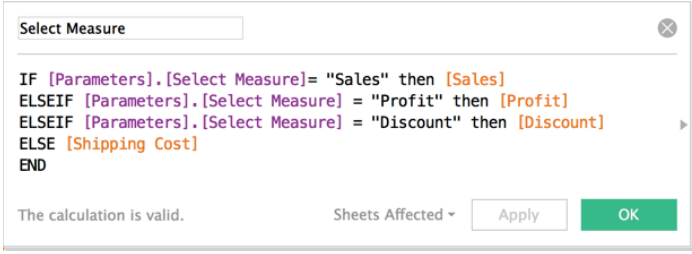


FIGURE 6-17 *Creating the parameter calculation*

The calculation uses an if/then/else logical statement to evaluate each string contained in the parameter and then associates the selected string with a specific measure field. With the completion of this Calculated Value, only one step remains to activate the parameter within the view.

Use the Calculated Value in the View

Placing the Select Measure calculated value on the Rows shelf activates the parameter. Figure 6-18 shows the completed view.

Adding the Select Measure value to the view connects the Parameter Control to the view and changes what is communicated to the data source when a different measure is selected within the Parameter Control. The Select Measure filter now allows users to change the measure to any of the items added in the parameter and calculated value. Notice that the report title contains the parameter name. Also, the axis label is variable. Enabling these refinements requires a couple of extra steps. First, edit the report title and insert the parameter name into the title block. Then, drag the parameter from the Parameters shelf to the axis and add the name variable there as well. By editing the axis label and removing the row heading, you can achieve a clean look that flexibly names the report and the axis.

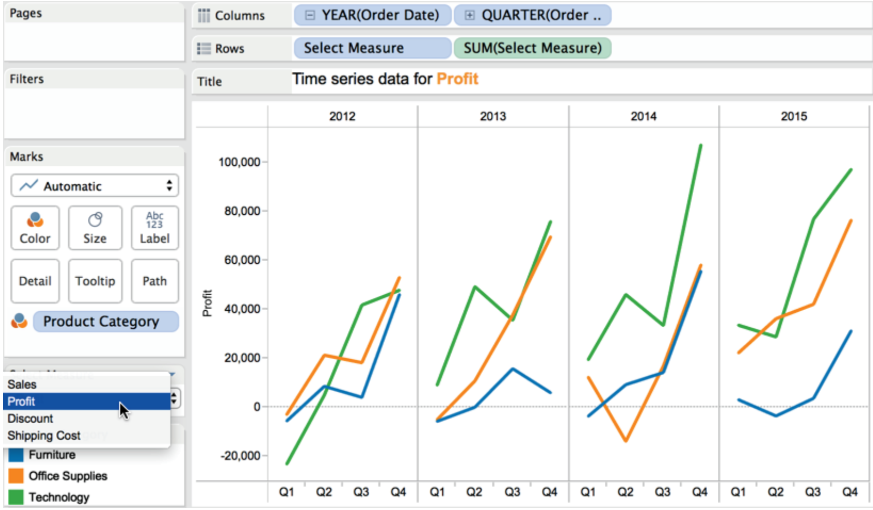


FIGURE 6-18 *A parameterized time series chart*

There are many other ways you can use Advanced Parameters, but the basic process for all of them follows these four steps. As you gain familiarity with Tableau's calculation functions, you will think of many different ways to leverage Advanced Parameter controls.

EDITING VIEWS IN TABLEAU SERVER

Another way that Tableau enables ad hoc analysis is through a facility called Web Authoring in Tableau Server. This is a powerful feature released in Tableau Version 8 that allows Tableau Server information consumers to alter and create visualizations from within Tableau Server without the need for anything other than a web browser and access rights to the view.

Formula Editing in Server

Your Tableau Server administrator makes this feature available by enabling a permission for editing views. When this feature is activated, the user will find an Edit Menu option in their browser session window. Figure 6-19 shows a typical report window within Tableau Server.

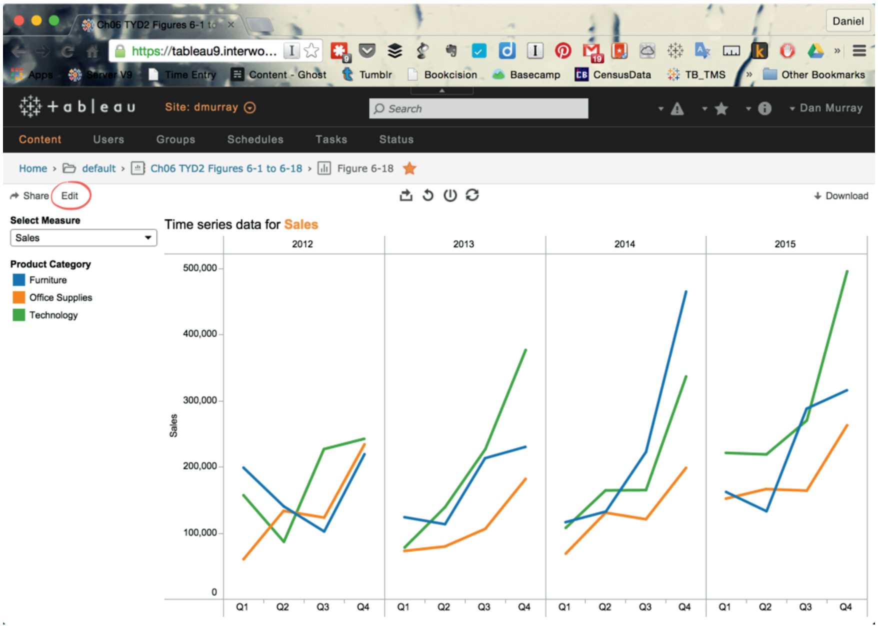


FIGURE 6-19 *Enabling editing in Tableau Server*

Notice the Edit menu option in the upper-left side of the window. This menu option appears in Tableau Server if permissions are set to allow the user to edit views. Selecting the Edit view exposes the data shelf, Marks cards, and the Rows and Columns shelves. Users can't do everything that a Desktop user can do, but within this specific workbook, they have the ability to edit existing views or create new worksheets and build new views. Figure 6-20 shows the controls that are exposed when editing is turned on.

You can see that a new menu bar appears at the top of Figure 6-20. These controls are optimized to work in the browser and will work when using a tablet computer to access the workbook as well. Full drag-and-drop capability exists—facilitating true ad-hoc analysis for end users in a workbook that is based on data vetted and controlled though the server administrator.

Formula Creation in Tableau Server

You can also build new views from scratch by dragging fields from the Dimensions and Measures shelves to the Columns and Rows shelves. And Tableau Server V9 allows you to create calculated values. Figure 6-21 shows the creation of an Ad-hoc Calculated Field.

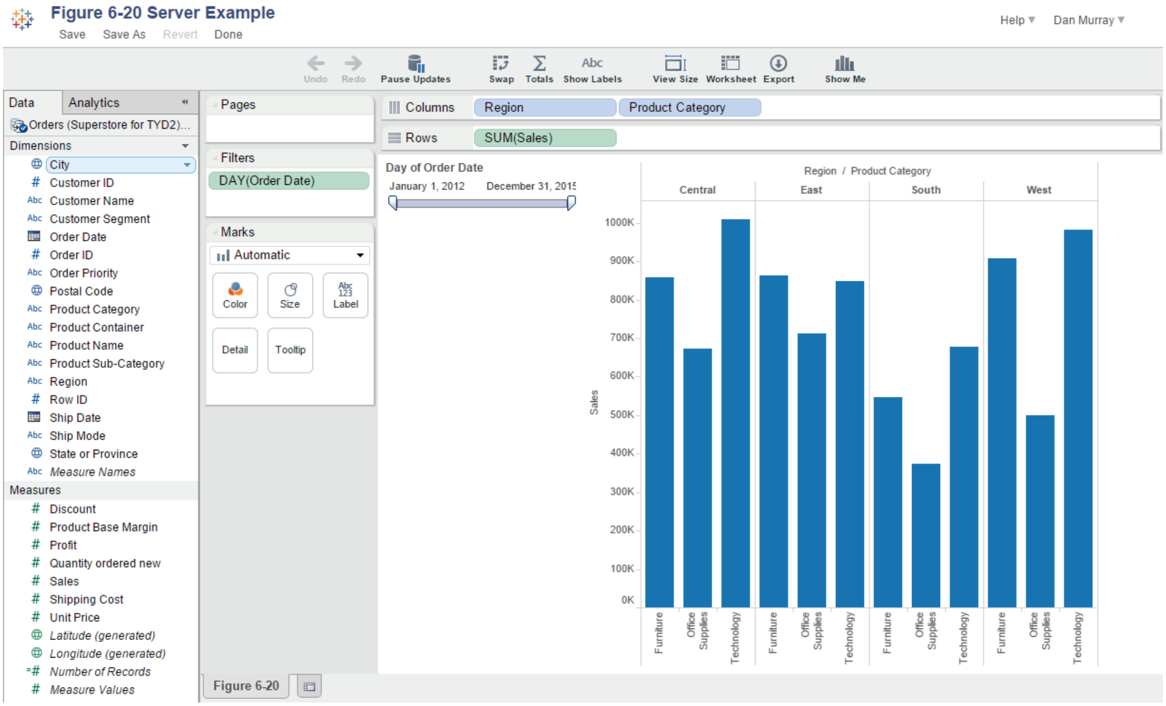


FIGURE 6-20 *Tableau Server editing mode*

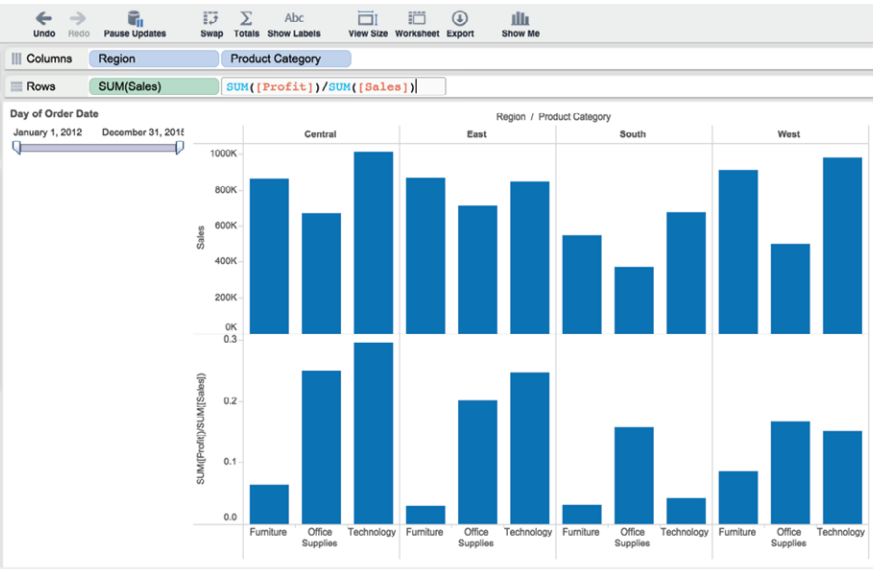


FIGURE 6-21 *Ad hoc formula creation in Server*

The design and build capabilities in Tableau Server are not as robust as Tableau Desktop, but for many information consumers who want to do analysis or create their own views, it provides the basics.

Formula Creation in Mobile

Information consumers viewing Tableau Server workbook dashboards and worksheets can also build formulas and create views from mobile devises. See Chapter 9 for more details.

Tableau's ad hoc analysis capabilities should reduce the workload on the technical resources within your organization and reduce the amount of time required for managers to make new inquiries of your data.

In the next chapter, you learn some tips and tricks that will help you build views more quickly and create visualizations that pack more data into small spaces.

NOTE

1. Frank Chimero, *The Shape of Design*, Kickstarter Project (Minnesota: Shapco Printing, 2013), 21.

CHAPTER 7

Tips, Tricks, and Timesavers

Mastering the basics of building visualizations and dashboards isn't difficult or time-consuming. Most people achieve good results without having to spend a lot of time learning the nuances of data visualization or mastering more advanced techniques.

In this chapter, you learn timesaving tips for building views, altering the default formats of fields and axis headers, creating new fields, and customizing the content and appearance of tooltips. I explain a trick for using legends to change the order data in views. After I cover customizing shapes, colors, and fonts, I present useful advanced chart types that demonstrate how to create more advanced chart types that aren't directly supported using the Show Me menu. The chapter closes with an introduction to some simple methods for creating subtle behavior in dashboards that will set the table for a more extensive treatment of dashboard-building in Chapter 8.

SAVING TIME AND IMPROVING FORMATTING

There are typically several ways to accomplish desired results in Tableau. Becoming faster at achieving the outcome takes a little practice. Knowing shortcuts that save seconds when you are creating an individual view can add up to hundreds of hours per year. If your team has many people using Tableau, the time savings can be significant.

DOUBLE-CLICK FIELDS TO BUILD FASTER

Double-click any field to quickly create a view or add it to an existing view. If you are working with a file-based data source (Excel or Access), you can utilize the Measure Names and Measure Values fields to quickly create an overview of an unfamiliar dataset. Warning: Do not use this technique if you are connecting to a very large database as it may overload your system. Start your analysis by double-clicking on the Measure Names field. As a result, every measure contained in the data window is displayed as a text tables—providing a quick view of the facts contained in the dataset. Add a time element to see value breakdowns. Figure 7-1 required only three mouse selections to generate.

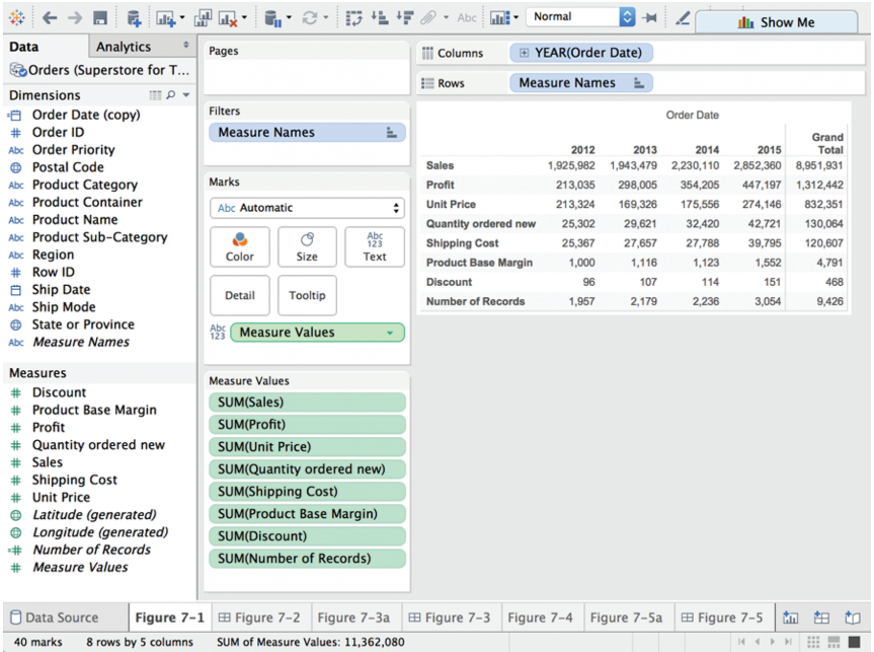


FIGURE 7-1 *Double-click to review all measures.*

When you dive into a dataset for the first time, knowing measure totals, the number of records in the set, and the breakdown over time helps you reconcile amounts in your views against source data batch totals. Figure 7-1 was created by:

1. Double-clicking Measure Names
2. Clicking the Swap icon
3. Double-clicking Order Date
4. Selecting the menu option Analysis ➢ Totals ➢ Show Row Grand Totals

With a little practice, you'll be able to create that type of view in under six seconds. When diving into a file-based dataset for the first time, it's the fastest way to get some benchmark information.

REDUCE CLICKS USING THE RIGHT MOUSE BUTTON DRAG

To save time when you want to display dates, numbers, or text, use the right mouse button when you drag fields into a view. Using this method to place the field pills opens a dialog box that gives you access to more presentation options and significantly reduces the number of mouse clicks required to customize the result. Figure 7-2 shows the three different dialog boxes that are provided.

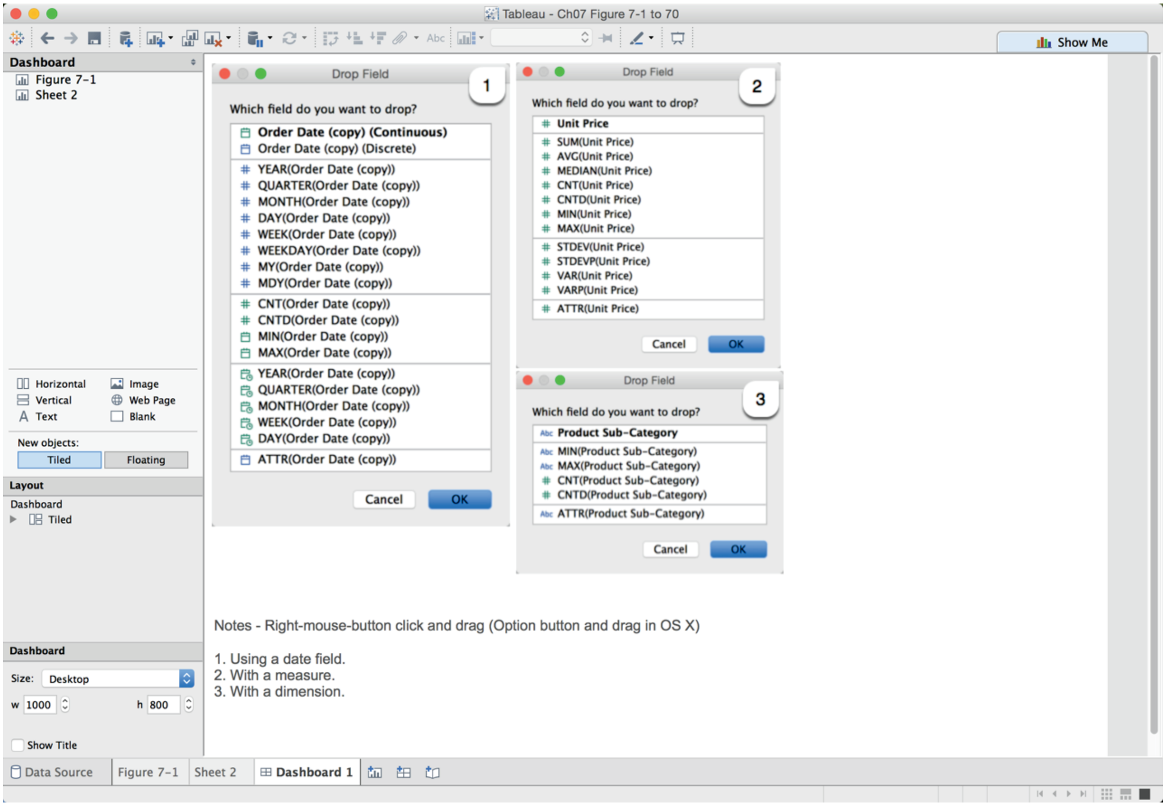


FIGURE 7-2 *Exposing field options*

The first option in Figure 7-2 is the dialog box presented when a date field is placed with a right-click drag. Option two is for measures, and option three is for a non-date dimension. In each case, the right-click drag and drop provides direct access to all the available options for expressing time, measure aggregation, and different ways strings can be expressed.

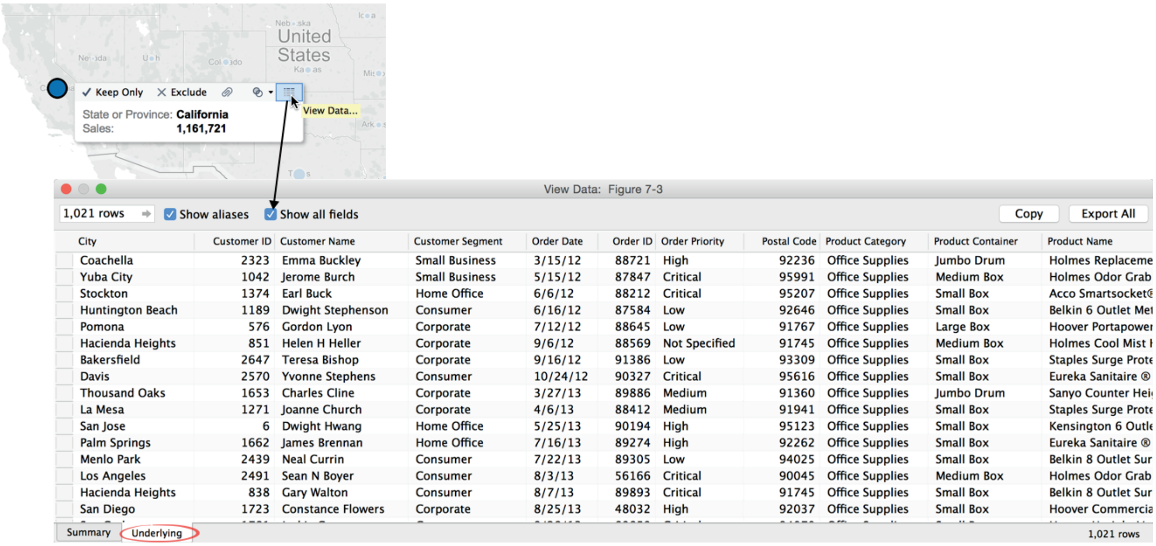
QUICK COPY FIELDS WITH CONTROL-DRAG

Holding the Control button down while dragging an active field causes a copy of that pill to be created wherever it's placed. This is particularly helpful if you want to build a Table Calculation using an active field or if you want to use a measure or dimension that is expressed in the Rows or Columns shelf on the marks card as well.

REPLACE FIELDS BY DROPPING THE NEW FIELD ON TOP

Dropping any measure or dimension on top of a field already expressed in the view will result in that field being replaced with the new selection. This is particularly useful if you are exploring a dataset for the first time and want to cycle through a variety of measures using the same view. After creating an initial view and then duplicating that chart, you can use this technique to quickly create a series of charts, each displaying a different measure.

Using tooltips to drill into details exploring marks within a view generates questions when you find outliers. Figure 7-3 shows how you can use a tooltip to expose the underlying source data.



City	Customer ID	Customer Name	Customer Segment	Order Date	Order ID	Order Priority	Postal Code	Product Category	Product Container	Product Name
Coachella	2323	Emma Buckley	Small Business	3/15/12	88721	High	92236	Office Supplies	Jumbo Drum	Holmes Replaceme
Yuba City	1042	Jerome Burch	Small Business	5/15/12	87847	Critical	95991	Office Supplies	Medium Box	Holmes Odor Grab
Stockton	1374	Earl Buck	Home Office	6/6/12	88212	Critical	95207	Office Supplies	Small Box	Acco Smartsocket
Huntington Beach	1189	Dwight Stephenson	Consumer	6/16/12	87584	Low	92646	Office Supplies	Small Box	Belkin 6 Outlet Met
Pomona	576	Gordon Lyon	Corporate	7/12/12	88645	Low	91767	Office Supplies	Large Box	Hoover Portapower
Hacienda Heights	851	Helen H Heller	Corporate	9/6/12	88569	Not Specified	91745	Office Supplies	Medium Box	Holmes Cool Mist
Bakersfield	2647	Teresa Bishop	Corporate	9/16/12	91386	Low	93309	Office Supplies	Small Box	Staples Surge Prote
Davis	2570	Yvonne Stephens	Consumer	10/24/12	90327	Critical	95616	Office Supplies	Small Box	Eureka Sanitaire ®
Thousand Oaks	1653	Charles Cline	Corporate	3/27/13	89886	Medium	91360	Office Supplies	Jumbo Drum	Sanyo Counter Hei
La Mesa	1271	Joanne Church	Corporate	4/6/13	88412	Medium	91941	Office Supplies	Small Box	Staples Surge Prote
San Jose	6	Dwight Hwang	Home Office	5/25/13	90194	High	95123	Office Supplies	Small Box	Kensington 6 Outle
Palm Springs	1662	James Brennan	Home Office	7/16/13	89274	High	92262	Office Supplies	Small Box	Eureka Sanitaire ®
Menlo Park	2439	Neal Currin	Consumer	7/22/13	89305	Low	94025	Office Supplies	Small Box	Belkin 8 Outlet Sur
Los Angeles	2491	Sean N Boyer	Consumer	8/3/13	56166	Critical	90045	Office Supplies	Medium Box	Holmes Odor Grab
Hacienda Heights	838	Gary Walton	Consumer	8/7/13	89893	Critical	91745	Office Supplies	Small Box	Belkin 8 Outlet Sur
San Diego	1723	Constance Flowers	Corporate	8/25/13	48032	High	92037	Office Supplies	Small Box	Hoover Commercia

FIGURE 7-3 *Use the tooltip to expose the data.*

The tooltip contains a button on the far right that can be used to expose a summary of the mark's makeup, all of the details contained in the dataset pertaining to that mark, or selected details. Rearrange columns within the exposed table by dragging them manually. You can also sort the rows by clicking any column to toggle between ascending or descending sorts of the data included in the column selected. If the tooltip doesn't include the details you want to answer your question, this technique will provide access to the all of the dimensions and measures available in the source dataset.

RIGHT-CLICK TO EDIT OR FORMAT ANYTHING

If you don't like the appearance of any element contained in a view, a quick way to get to the appropriate formatting option menu is to point at the objectionable element, right-click, and select Format. A context-specific formatting menu will appear in place of the data shelf area on the left side of the workspace. Figure 7-4 shows how flexible formatting can be.

	2014					2015					
	Q1	Q2	Q3	Q4	Total	Q1	Q2	Q3	Q4	Total	All Years
Furniture	116,656	132,448	222,514	464,984	936,603	162,320	132,751	288,402	316,202	899,675	**1,836,277**
Office Supplies	68,915	131,066	120,853	198,627	519,461	152,269	166,770	164,102	263,030	746,171	**1,265,632**
Technology	107,875	164,765	164,812	336,594	774,046	221,560	219,071	270,163	495,720	1,206,514	**1,980,561**
Grand Total	**293,446**	**428,279**	**508,179**	**1,000,206**	**2,230,110**	**536,149**	**518,592**	**722,667**	**1,074,952**	**2,852,360**	**5,082,470**

FIGURE 7-4 *Right-click formatting*

Special formatting in Figure 7-4 has been applied to rows, columns, panes, totals, and subtotals. Year headers are in a green font. The headings for each quarter, the subtotal heading for each year, and the grand total heading for the column displaying the grand total for both years are colored blue. The "All Years" text was edited from the default "Grand Total" heading text. A custom red color was applied to the year total panes, and a custom black bold font was applied to the column totals at the bottom of the text table by applying a custom font to the pane and header of the Grand Total row. Finally, two different shades of red were applied to the row banding in the pane and in the header. While there is more than one way to apply these customizations, the easiest way is to point at the screen element and right-click, and Tableau will present the appropriate set of formatting controls on the left side of the workspace.

EDITING OR REMOVING TITLES FROM AXIS HEADINGS

Sometimes it is desirable to edit axis titles or remove them entirely. This can be done by pointing at the axis (white space or header) and selecting the Edit Axis option. Figure 7-5 shows the menu that is displayed.

Not only does the Edit Axis menu allow you to edit or remove the axis title (without removing the axis header), but you can also modify the title or erase the title in the Titles box you see in Figure 7-5. Later in this chapter, you'll see how a range selection can be used to create a Sparkline chart.

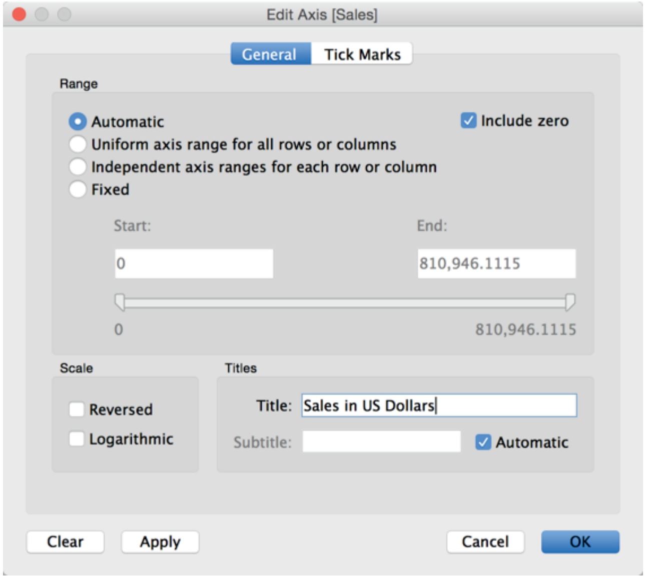


FIGURE 7-5 *Edit axis menu*

QUICKEN YOUR PRESENTATION PAGE VIEWS

Making your presentations truly interactive by replacing static slide decks with interactive visualizations provides a powerful and flexible story. If your Tableau workbook has many different worksheets and dashboards, loading each new worksheet can cause delays as each worksheet or dashboard is materialized. Avoid delay by preloading your dashboard views.

You preload views by accessing the multiple worksheet view (the PowerPoint slide deck style view) from the tab in the upper right of the screen. Figure 7-6 shows all of the worksheets and dashboards contained within the workbook.

Right-clicking in the worksheet window exposes a menu option—Refresh All Thumbnails—that triggers Tableau to query the data source(s) used for all the worksheets and dashboards in the workbook. Now as you run through your presentation, each worksheet and dashboard will be preloaded and materialize instantly.

You can also trigger a query of all of the data sources via the Filmstrip view of worksheets, as shown in Figure 7-7.

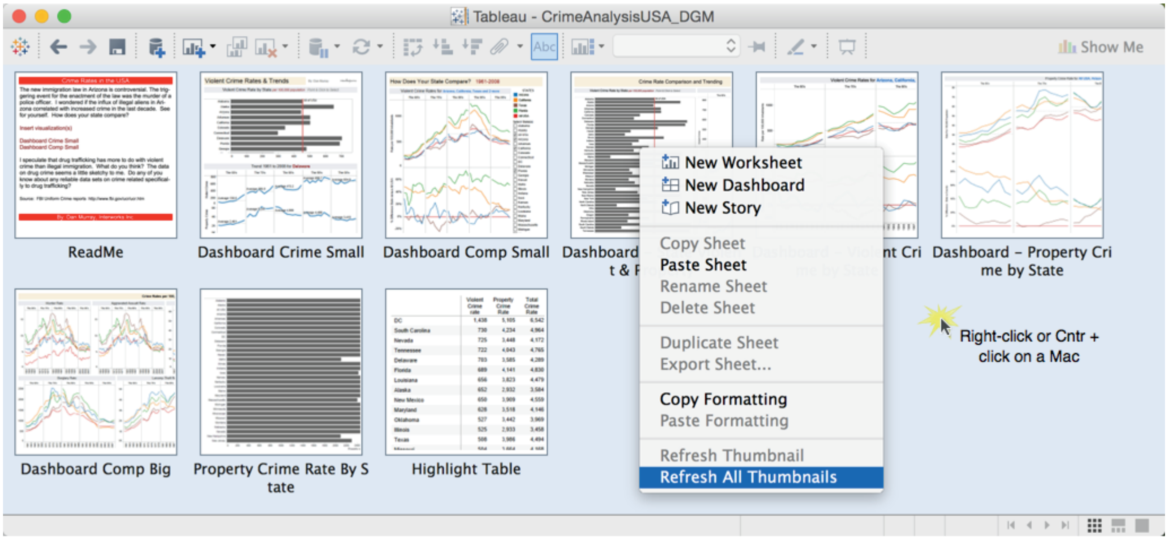


FIGURE 7-6 *Worksheet window*

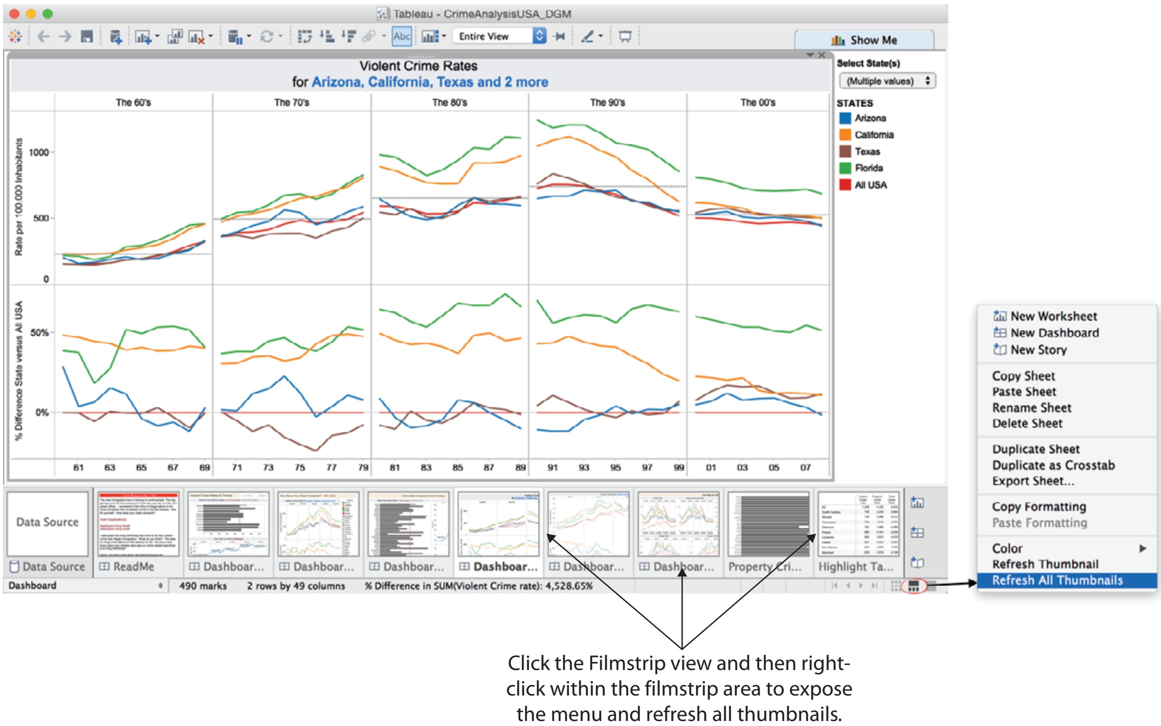


FIGURE 7-7 *Filmstrip view*

Turn on the Filmstrip view by clicking the small up and down Show Filmstrip option in the lower right of the worksheet. Trigger the worksheet by right-clicking within the filmstrip sheet area shown on the right side of Figure 7-7.

A FASTER WAY TO ACCESS FIELD MENU OPTIONS

Hovering over a field pill that is placed anywhere in your worksheet will expose a drop-down arrow located on the right side of the pill. Clicking on the drop-down arrow exposes menu options related to the measure or dimension. Figure 7-8 shows the exposed menu.

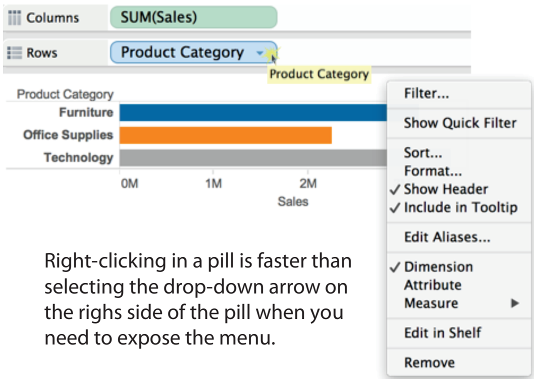


FIGURE 7-8 *Exposing a pill menu*

An easier way to expose the same menu is to point anywhere at the field pill and click the right mouse button. The same menu will be exposed in a way that requires less precise pointing.

ZOOMING THE FORMULA DIALOG BOX

Did you know that you can zoom inside the Calculated Field formula dialog box? Try creating a new calculation. After you get some text in the Calculated Field dialog box, press the Ctrl button and use your mouse scroll wheel to increase or decrease the size of the text. On the Mac you do this by pressing the Command button and swiping your mouse to the left or right.

DRAG A FIELD INTO THE FORMULA DIALOG BOX

A shortcut for entering fields into the Formula dialog box is to drag a dimension or record into the Formula dialog box and drop it. If you do this while pointing at a field in the Formula dialog box, you will replace that field with the new one.

SWAP DATA IN PANE AND REFERENCE LINE FIELDS

If you have a visualization that includes related measures—one being used for the data plot and another being used to plot a reference line—you can swap the two measures so that the Reference Line field is used as the source for the marks in the view and the plotted measure is now used for the reference line.

IMPROVING APPEARANCE TO CONVEY MEANING MORE PRECISELY

Your dashboard and worksheet designs need to fit in the available space. For this reason, headings, instructions, and details related to your views—conveying the information while using as little space as possible—is desirable. The techniques are space-efficient without compromising meaning.

CHANGING THE APPEARANCE OF DATES

Alter date formats that appear on an axis by pointing at the date header, right-clicking, and selecting the Format Menu option. This exposes many different date formats—including a custom formatting option, as shown in Figure 7-9.

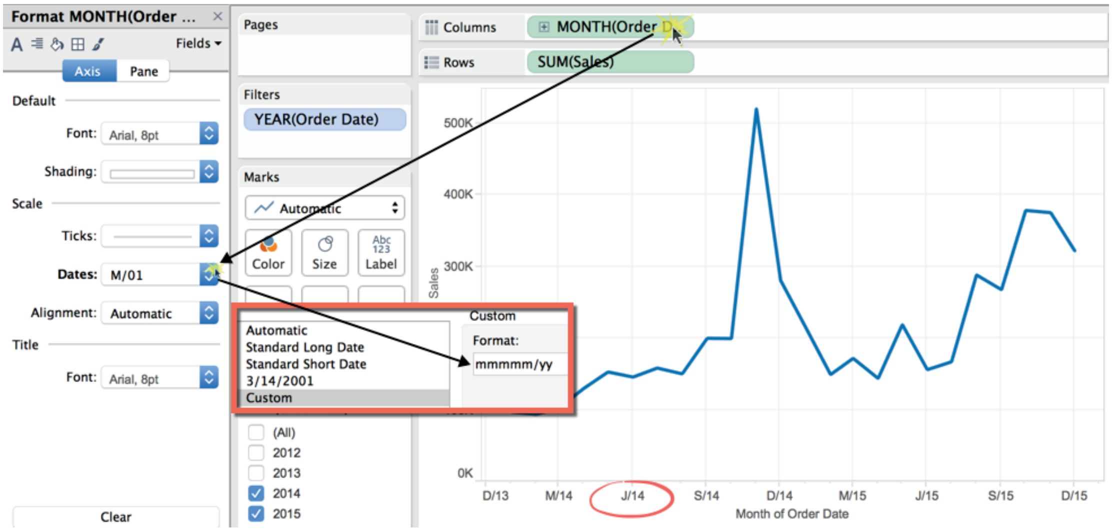


FIGURE 7-9 *Customizing date formats*

The specific date formatting available will vary depending on the type of date being expressed (continuous or discrete). Continuous dates provide more formatting options than discrete dates.

FORMATTING TOOLTIP CONTENT

Tooltips in worksheets and dashboards can be improved by adding fields that are not included in the view, formatting text font and color, and adding instructions. Edit your tooltip from the main menu by selecting Worksheet ➢ Tooltip. Figure 7-10 shows a modified tooltip that uses custom colors, custom font sizes, field name revisions, and explanatory text along with contact information.

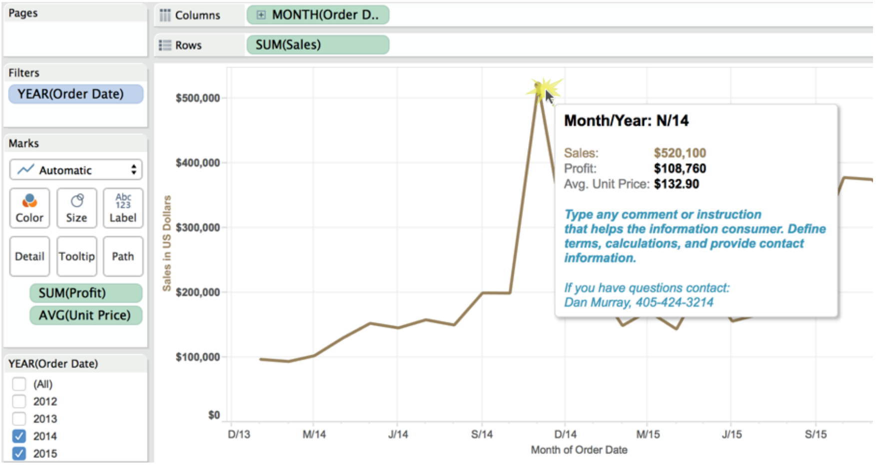


FIGURE 7-10 *A customized tooltip*

Note that any fields included on the Marks cards can be added to the tooltip. Tooltips are a space-efficient way to add details on-demand to worksheets and dashboards.

CHANGE THE ORDER OF COLOR EXPRESSED IN CHARTS

When using colors to express members of dimensions, comparing different members in the set is easier if the item you want to focus on starts at the same point on the axis. Figure 7-11 shows a stacked bar chart that compares the sales mix percentage of product categories in different date aggregations (month, quarter, and year) by using a quick Table Calculation and color to express the relative sales for each product category.

Dragging the Furniture color to the bottom of the color legend, as shown on the right of Figure 7-11, enables more precise comparison of the furniture product category.

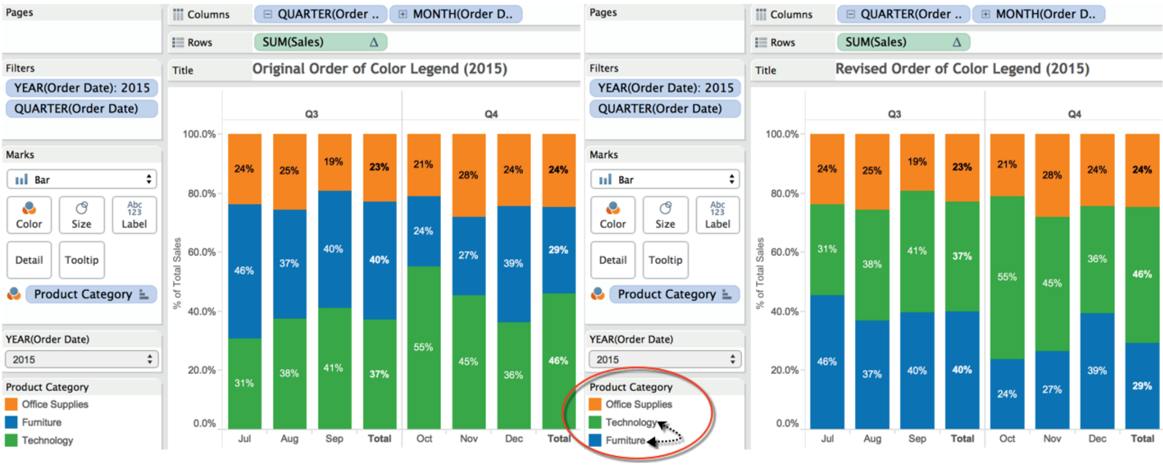


FIGURE 7-11 *Reordering the color legend*

EXPOSING A HEADER IN A ONE-COLUMN TEXT TABLE

Adding a small text table in a dashboard can provide an effective means for triggering a filter action. For this reason, you may want to create a very basic text table, as you see in Figure 7-12.

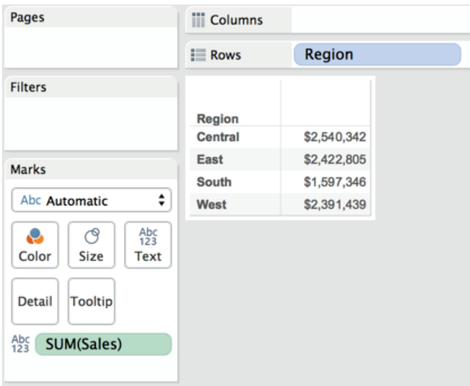


FIGURE 7-12 *No heading over the sales values*

The chart was created using the Superstore dataset. Building Figure 7-12 requires two steps:

1. Double-click the Region field in the Dimensions shelf.
2. Double-click the Sales field in the Measures shelf.

This is fast and easy, but what if you want to add a header directly over the sales values to create a well-labeled text table without having to add a worksheet title? Worksheet titles consume additional pixel height, which may take more vertical space than you have available.

At this point, there is a row label over the region names in Figure 7-12 but no row header over the sales value. Tableau's default behavior doesn't provide a row label when only one measure is included in the view. To get a header to appear immediately above the sales values, double-click any other field included on the Measures shelf (except for the geocoding measures used for mapping) and then point at the column heading of the second measure and right-click to hide the measure. Alternatively, right-click the Measure Values pill (which automatically appeared on the Marks card when the second measure was added) and filter out the new measure so that the Sales field is the only measure remaining in view. The text table should now look like the one in Figure 7-13.

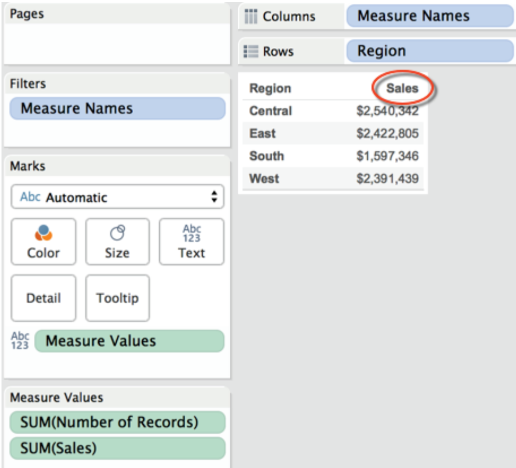


FIGURE 7-13 *Text table with a sales header*

Figure 7-13 presents a very compact view of the sales by region with headers directly above the field values. This text table could be placed into a dashboard requiring the same amount of space as a multi-select filter but providing a little additional data. Another way to build the same text table is to use Measure Names and Measure Values directly to build the view. Follow these steps:

1. Double-click the Region field on the Dimensions shelf.
2. Double-click the Measure Names field on the Dimensions shelf.

3. Right-click the Measure Values pill on the Marks card.
4. Filter out all of the measures leaving only the Sales selected.

The key point in this example is that Tableau will not provide a header over the measure when only one measure is in view. You will use a text table like this in a dashboard example that you build in Chapter 8.

UNPACKING A PACKAGED WORKBOOK FILE

Unpacking a Tableau Packaged Workbook (`.twbx`) file allows you to view the original data source. Unpacking is useful if your data source is file-based (Excel/Access/CSV). To open this type of file, point at it and then right-click and select the Unpackage option. Tableau creates a data folder that contains a copy of the file source.

MAKE A PARAMETERIZED AXIS LABEL

Using parameters to alter the measure plotted in a view is an excellent way to make one chart serve many purposes. But the default axis label isn't very informative, as you see in Figure 7-14.

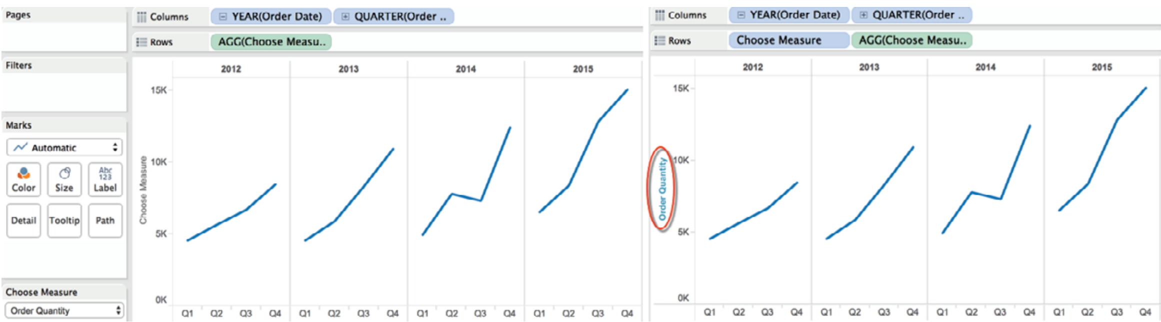


FIGURE 7-14 *Axis label default*

The time-series chart on the left of Figure 7-14 displays the default axis label for the parameter control Choose Measure. To enable a dynamic parameterized label for the axis, follow these steps:

1. Drag the parameter from the parameters shelf to the axis.
2. Drop the parameter on the axis.
3. Remove the default axis title by right-clicking on the axis.
4. Erase the default title in the titles area.

5. Right-click the parameter heading and hide the field label.
6. Rotate the parameter label by right-clicking it and selecting Rotate.

USING CONTINUOUS QUICK FILTERS FOR RANGES OF VALUES

When your worksheet or dashboard contains a continuous Quick Filter, many people don't realize you can restrict the range of values and then drag them from within the range to scroll. Figure 7-15 shows a bar chart that displays sales by customer and a Quick Filter using profit.

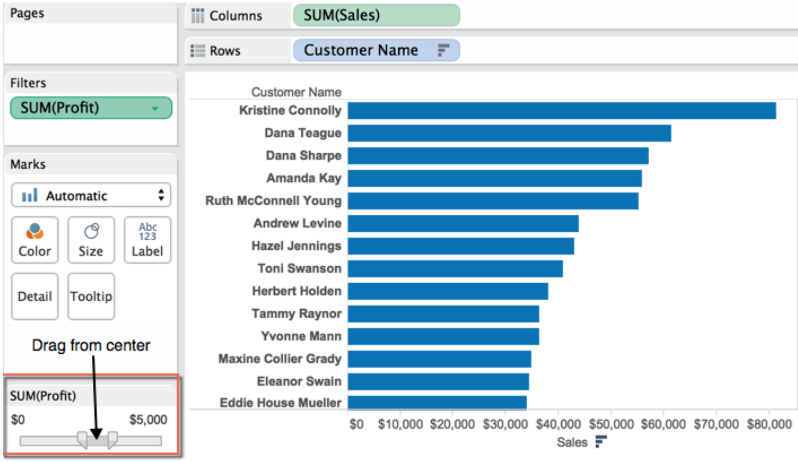


FIGURE 7-15 *Filtering for a range of values*

Restrict the range by dragging the bar handles in or by typing specific values in the filter values. You can see that the range has been restricted from $0 to $5,000. To scroll, point at the gray area in the filter bar and, while pressing your left mouse button, drag the range to the left or right to move through the entire set in $5,000 profit range increments.

CREATE YOUR OWN CUSTOM DATE HIERARCHY

Tableau's automatic data hierarchies save a lot of time, but what if you don't want to display all of the hierarchy that Tableau provides? By creating custom dates, you can combine them into hierarchies that meet your specific needs. Figure 7-16 shows a bar chart comparing sales values for specific dates.

The custom hierarchy includes discrete year and quarter values and nothing more. Notice that the Date Year pill can be expanded by clicking the plus sign, but the grouping of the custom Date Year and Date Quarter overrides the normal date hierarchy structure within Tableau.

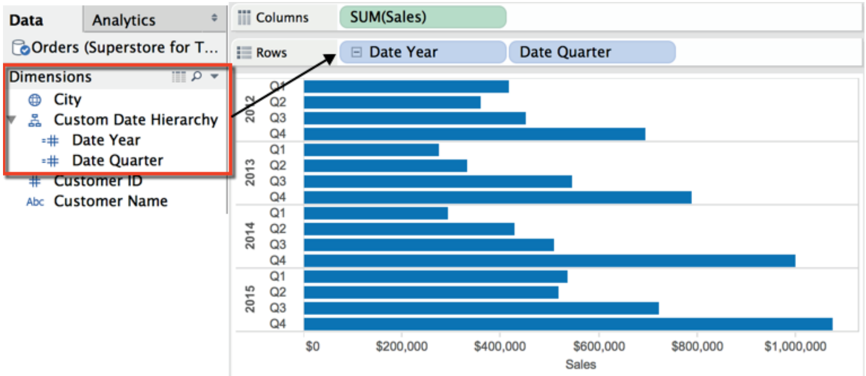


FIGURE 7-16 *Custom date hierarchy*

To create custom date hierarchies, follow these steps:

1. Point at a date field in the dimension shelf and right-click.
2. Select the Create Custom Date option.
3. Edit the date as you require.
4. Drag one date field on top of another to create the custom hierarchy.
5. Use the custom hierarchy in your view.

Figure 7-17 shows the custom date dialog box being accessed from the menu.

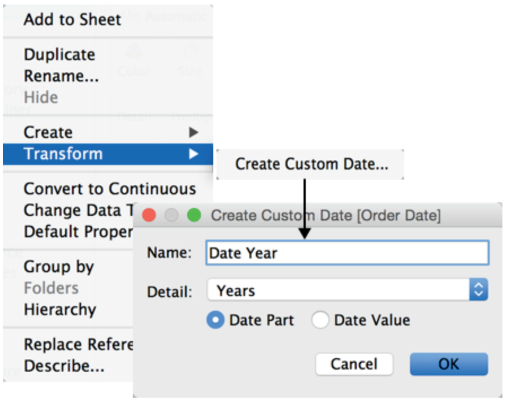


FIGURE 7-17 *Building a custom date hierarchy*

Complete the date by giving it a specific name. Use the Detail drop-down selector to pick the exact date granularity you desire. The radio buttons below that define whether the date is a discrete date (date part) or a continuous date (date value).

After the custom dates are defined, drag one on top of another in the dimension window to create your custom date hierarchy. You can right-click and edit the name of the hierarchy as desired. This technique is particularly useful in dashboards when you might need to limit the expansion of the hierarchy so that the chart fits into the available space nicely.

CONCATENATING TO MAKE CUSTOM FIELDS

This is a favorite easy formula hack for creating key records on the fly if your data source doesn't really include a truly unique key record. To create a new field that is the combination of two or more fields, use the Formula Editor. Figure 7-18 shows a concatenation formula.

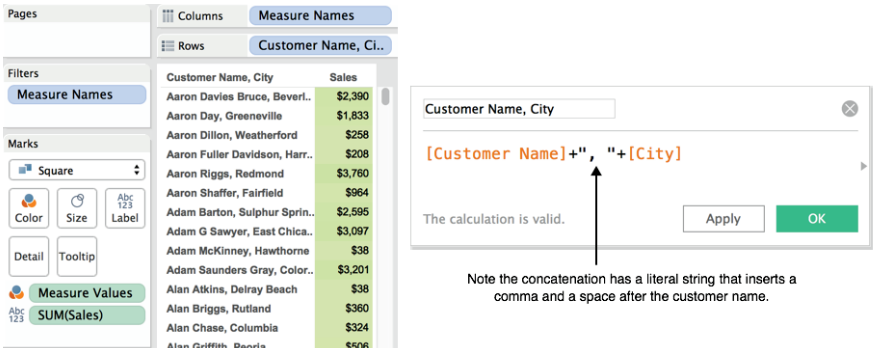


FIGURE 7-18 *Concatenating fields*

Using the + sign between each field creates a concatenated (joined together) field that will be available in the Dimensions shelf. This is also useful when you want to assemble addresses from discrete fields to create mailing lists. In Figure 7-18, the formula also inserts a literal string including a comma and a space between the Customer Name and City fields. If you experience performance degradation using this technique, try combining sets. Refer to Chapter 3 for more details.

USING LEGENDS TO BUILD HIGHLIGHT ACTIONS

Color or shape legends can be used to create highlight actions. Activate a color action by selecting the highlighting tool in the legend, as you see in Figure 7-19, and then click any color.

Similarly, the shape legend in Figure 7-19 can be used to create another highlight action. The resulting action in the dashboard will use the combination of color and shape when selecting marks from the scatter plot, as you see in Figure 7-20.

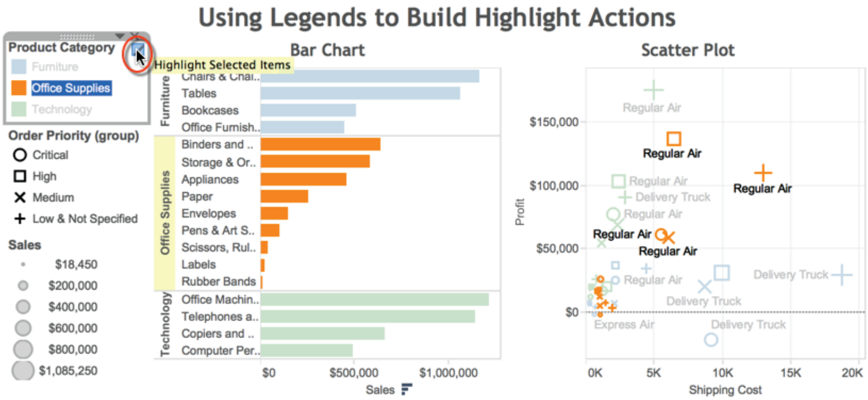


FIGURE 7-19 *Creating a highlight action from a color legend*

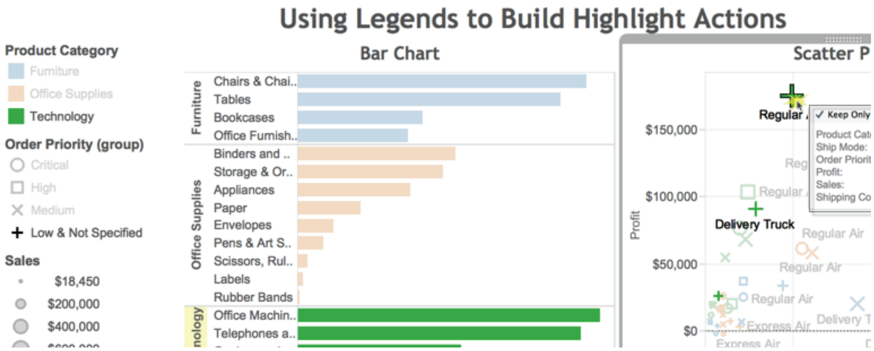


FIGURE 7-20 *Highlighting using actions generated by the color and shape legends*

Selecting a blue circle in the scatter plot triggers the highlight action—changing appearance of the scatter plot and bar chart. The combination of Order Priority (shape) and Product Category (color) are highlighted. Tooltips for both items have been displayed together in Figure 7-20 to expose the details for you to review. Tableau normally displays only one tooltip at a time.

You can view the Action definitions by going to the Dashboard menu, selecting the Actions option, and then selecting Edit. Figure 7-21 displays the menu details.

This time-saving technique will be used in the dashboard example you create in Chapter 8.

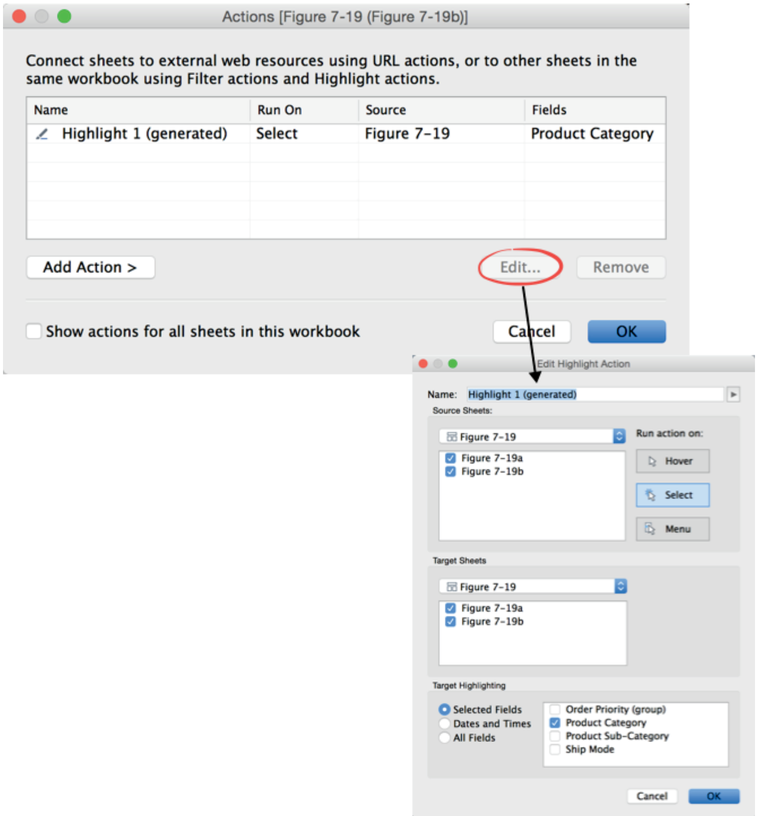


FIGURE 7-21 *Highlight action menu*

FORMATTING NULL VALUE RESULTS

Table Calculations use your visualization to create new values. If the calculation defined results in a null value, Tableau provides a variety of formatting options that allow you to control exactly how the null results are presented in the resulting chart. Figure 7-22 shows six time series visualizations. Area 1 in the upper left is an unedited plot of the data using a Table Calculation. The remaining panes show different ways that you can control how null values are displayed.

The dialog box displayed in Figure 7-22 (area 1) shows the unedited plot of a quick Table Calculation for a three-month moving average. The actual quick table definition is shown in Figure 7-23.

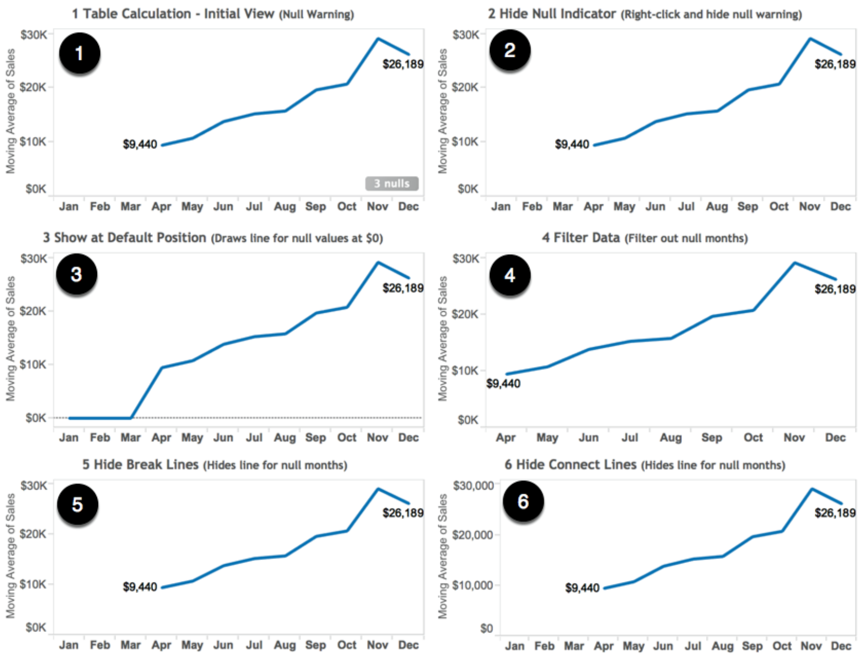


FIGURE 7-22 *Time series with null values*

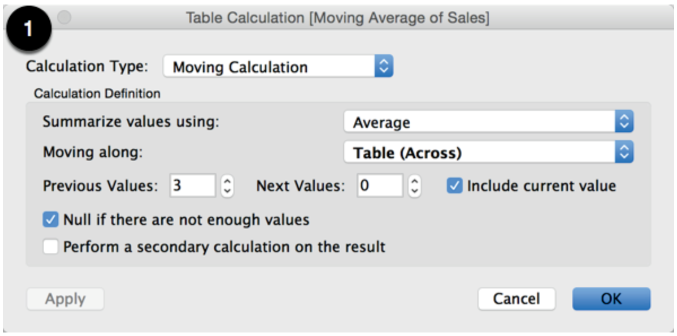


FIGURE 7-23 *Three-month moving average*

Note that the indicator in the dialog box in Figure 7-23 (Null if there are not enough values) is checked. Selecting this tells Tableau not to plot marks if there is insufficient data to correctly calculate the result. This means that no mark will be plotted if any month included in the time series does not have data for the three preceding months. As a result, area 1 of Figure 7-22 isn't plotting any marks for January, February, or March because the dataset doesn't include date for the preceding October through December time period.

One way to deal with the null warning that is displayed in Figure 7-22 (area 1) is to right-click the 3 nulls pill to expose the Hide Indicator control shown in Figure 7-24.

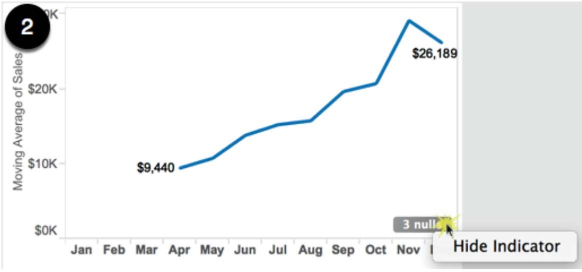


FIGURE 7-24 *Hiding the null indicator*

Selecting the Hide Indicator option merely removes the null warning pill from view without defining how additional null values should be treated. If your source data is being updated regularly, this selection hides the null indicator without providing any additional formatting rules for Tableau to use if new null values appear in the data. You see the result in Figure 7-22 (area 2).

If the (nulls) pill is selected using the left mouse button, the dialog box shown in Figure 7-25 is displayed.

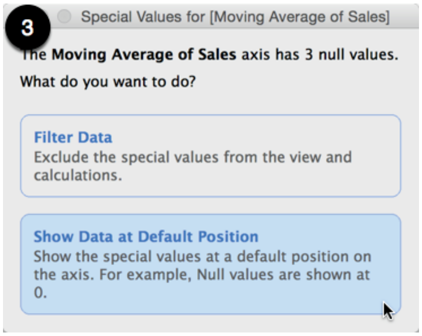


FIGURE 7-25 *Show at default value*

Showing the data at the default position causes Tableau to draw the line for the months with null values at zero, as you see in Figure 7-22 (area 3). If the Filter Data option is selected, as you see in Figure 7-26, Tableau will filter the null value months from view.

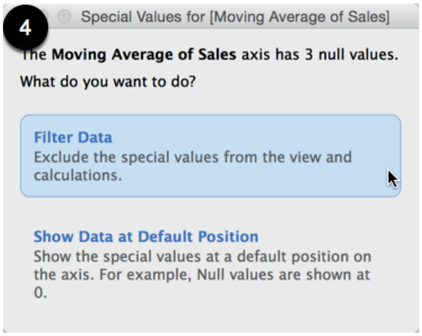


FIGURE 7-26 *Filtering out nulls*

Notice the axis for Figure 7-22 (area 4) starts in April. This option might be misleading if the source data includes gaps in the middle of the time series. For this reason, Tableau provides two additional options to format null values.

The bottom charts in the dashboard Figure 7-22 (areas 5 and 6) look similar to the chart in area 2 only because the null values in this example occur in the first three months of the time series. If the null values had occurred in the middle of the time series, these options provide slightly different treatments of the data breaks in the plot. To access the Special Values (for example, Null) formatting dialog box, right-click the field pill that you are using to express the Table Calculation and select Format. This exposes the formatting menu for the pane, as shown in Figures 7-27 and 7-28.

Hiding break lines results in the view you see in Figure 7-22 (area 5).

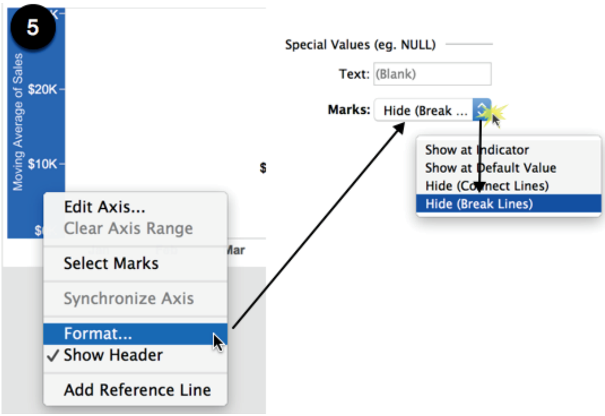


FIGURE 7-27 *Hiding break lines*

Figure 7-28 shows another option for handling nulls. You can see the result in Figure 7-22 (area 6).

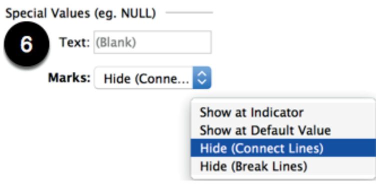


FIGURE 7-28 *Hide connect lines*

Because the null values appear at the beginning of the date range, the result looks the same using these options. Experiment with these options when your data includes nulls within the middle of the range. One will filter out the month entirely from view, and the other will show the month but without connecting the lines.

Table Calculations offer many options for deriving new information from your source data. Tableau's formatting options for null values provides for nuanced treatment of missing values in your source data so that information consumers are not misled by gaps in your source data.

WHEN TO USE FLOATING OBJECTS IN DASHBOARDS

Tableau supports the use of Floating objects, and this can be a great way to add information to your dashboards efficiently. This facility should be used with care. Think about how the underlying visualization can change and ensure that the floating object doesn't obscure the data contained in the view. Figure 7-29 is an example of a suboptimal use of floating objects.

The floating year filter and color legend in Figure 7-29 are space-efficient, but the Office Furnishing mark is obscuring the color legend. Floating objects in this chart are not a good choice unless you can be certain that the products in the top third of the view won't extend into the floating controls. Controlling this is impossible because your audience's computer resolutions will be different and the resulting plots on their computer screens will be different. Figure 7-30 shows a better use case for floating objects.

Presuming that sales occur only in the lower 48 states, the Floating objects in Figure 7-30 take advantage of the white spaces contained in the map to display color and size legends as well as a time series chart. A filter action could be added to the map and the time series to filter the view for selections made by the user—creating a more compact view than would otherwise be possible with the use of non-floating controls and quick filters. This use of floating legends is less susceptible to the problem highlighted in Figure 7-29.

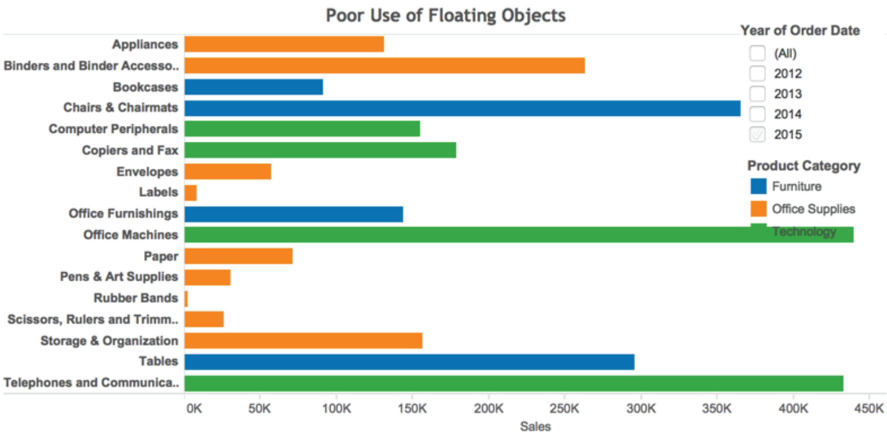


FIGURE 7-29 *A bad use of Floating objects*

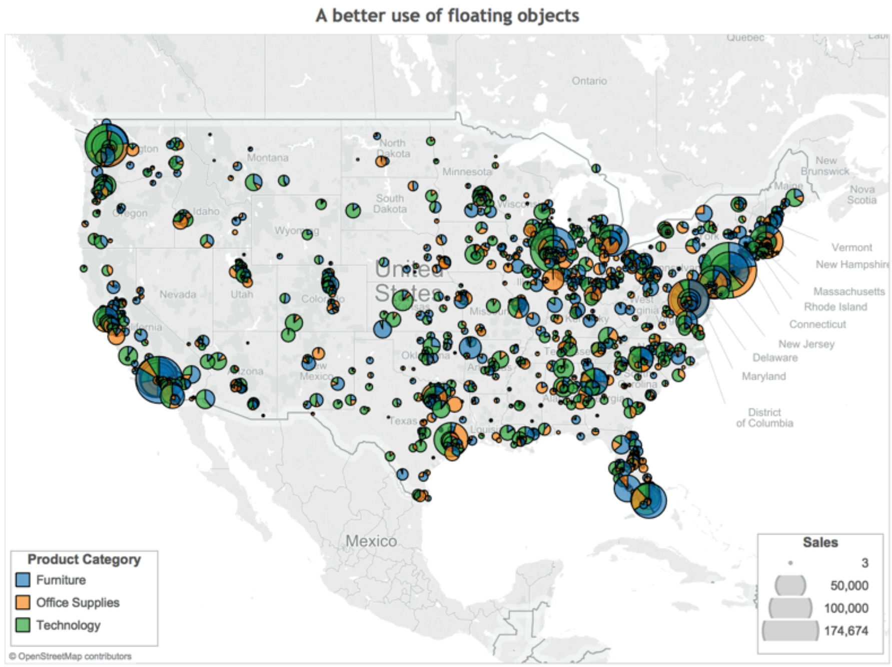


FIGURE 7-30 *A better use of Floating objects*

COMBINED AXIS SHADING IN A SCATTER PLOT

Shading from a single axis in a scatter plot is easy. But what if you want to shade using a combination of the vertical and horizontal axis? Figure 7-31 shows four possibilities.

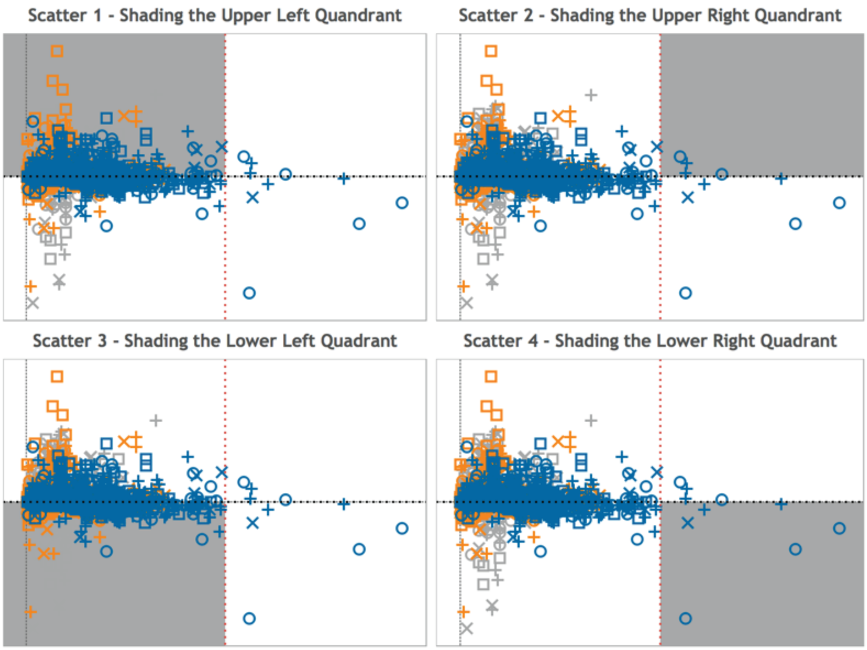


FIGURE 7-31 *Scatter plot shading*

The scatter plots use two reference lines to color the quadrants: one reference line for the vertical axis and another reference line for the horizontal axis. The key to getting this appearance is knowing how to apply the color fill for the reference lines in each example. Figure 7-32 shows the combination of fill options to achieve each view.

The quadrant color effect is created by the overlapping shading from the vertical and horizontal axes. This isn't the only way you could achieve this kind of output, but the cheat sheet provides a reliable shortcut as a starting point. Figure 7-33 shows you the settings used to create Scatter 1 in Figure 7-31.

Scatter Plot Shading Cheat Sheet			
#	Reference Line Fill Area	Vertical Reference Line	Horizontal Reference Line
1	Fill above	**Pick a color**	White
	Fill below	None	None
2	Fill above	**Pick a color**	None
	Fill below	None	White
3	Fill above	None	White
	Fill below	**Pick a color**	None
4	Fill above	None	None
	Fill below	**Pick a color**	White

FIGURE 7-32 *Shading cheat sheet*

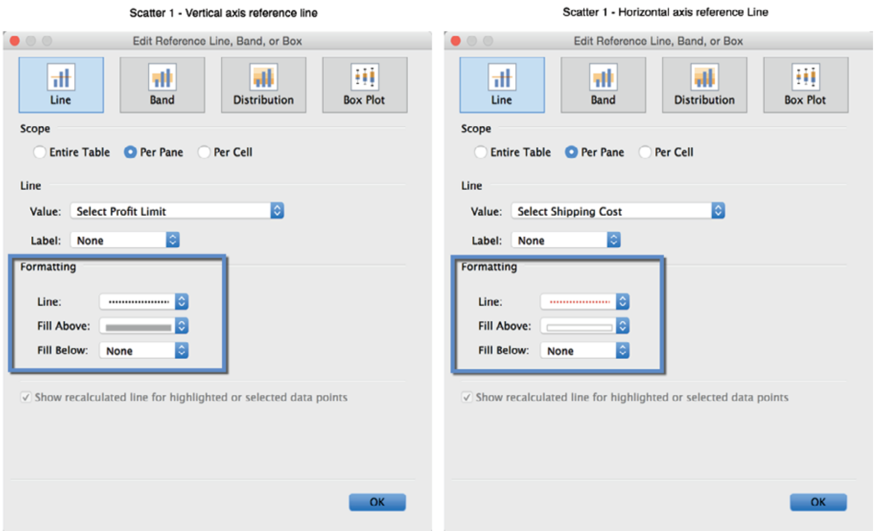


FIGURE 7-33 *Settings for Scatter 1*

Try building your own scatter plot using the Superstore for TYD2 dataset. Make your vertical axis a plot of Profit and the horizontal axis a plot of Shipping Cost. Place Product Category on the color Marks card, Order Priority on the Shapes Marks card, and Customer Name on the Detail Marks card. Next add a vertical reference line by selecting the Analytics tab in the Data pane and then dragging the Reference Line into the view. Pick either Table or Pane while selecting the Sum (Profit) area in the Reference Line dialog box. Alternatively, right-click on the vertical axis and select the Add Reference Line option. Use the shading cheat sheet in Figure 7-32 to assign the correct settings for Fill Above and Fill Below. Repeat the procedure for the horizontal axis. See if you can build all four of the plots displayed in Figure 7-32.

CREATING FOLDERS TO HOLD FIELDS

If you have to deal with database information, there can be hundreds of fields to search through. Tableau provides a search function at the top of the data pane to make it easier to find a specific field. Hierarchies can be useful for grouping fields in the data pane, but what if you don't want the behavior of a hierarchy in your visualizations?

Folders provide a way for you to group related fields within the data pane without having a hierarchy. To group fields in the Data pane, right-click in the white space within the Dimensions shelf (to group dimensions) or the Measures shelf (to group measures), and then select the Create Folder menu option and name the folder as you see in Figure 7-34.

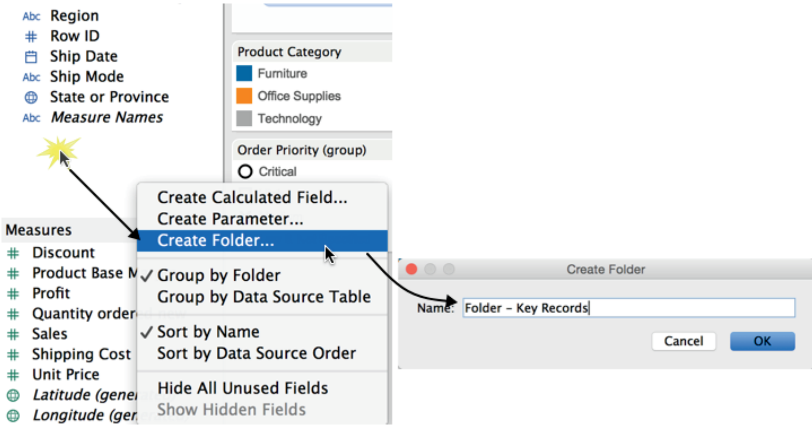


FIGURE 7-34 *Creating a folder*

Grouping related fields in folders is a timesaver if you find yourself using related fields to create analysis.

CUSTOMIZING SHAPES, COLORS, FONTS, AND IMAGES

Tableau comes with a wide variety of predefined shapes, colors, and fonts, but you can style these objects to meet your specific needs.

CUSTOMIZING SHAPES

There is nothing wrong with using the default shapes, as you see in Figure 7-35.

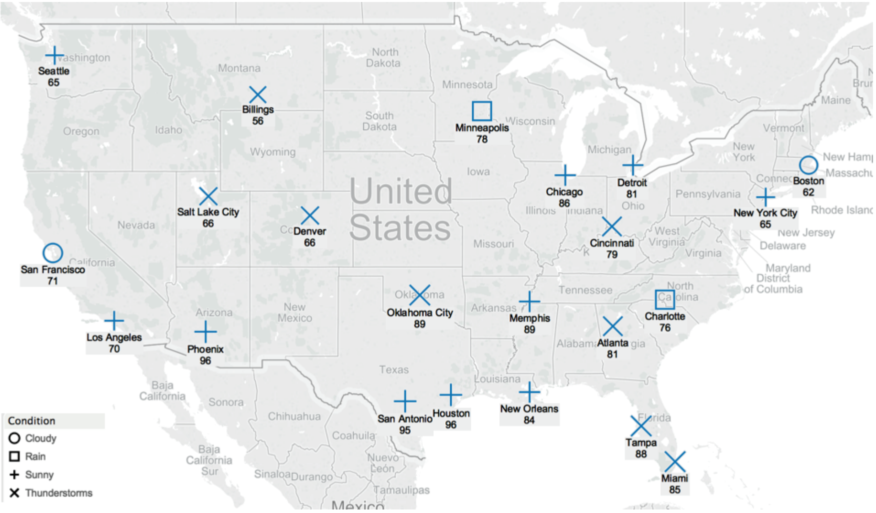


FIGURE 7-35 *Map with standard shapes*

Customizing the shape used to plot weather conditions provides more immediate understanding. Figure 7-36 shows the same map, but with weather images to depict weather conditions.

The use of customized images in Figure 7-36 conveys weather conditions more intuitively. This example was created using one of the available standard shapes provided in Tableau's shape palette. Editing shapes is done by accessing the Shape menu from the Shape legend you see in Figure 7-37.

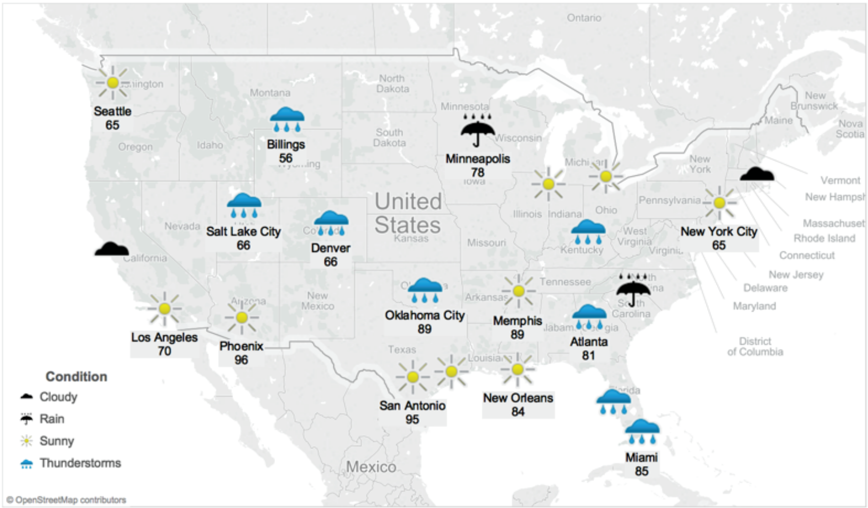


FIGURE 7-36 *Map with weather images*

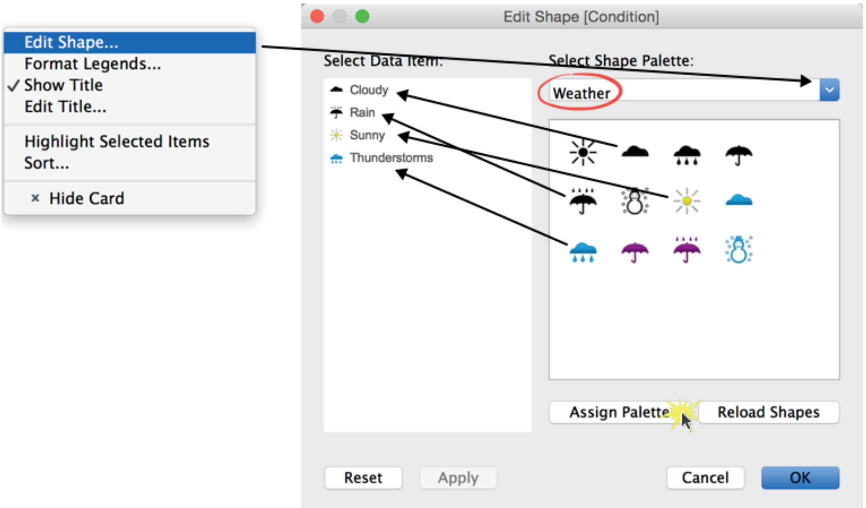


FIGURE 7-37 *Customizing shapes*

If Tableau's standard shape legend or palettes do not fit your requirements, import custom shape files (`.png`, `.jpeg`, `.bmp`, `.gif`) and make them available to use in your views by following these steps:

1. Create a folder to hold the image files under My Tableau Repository.

2. Give the folder a one- or two-word name (Tableau uses This Name).
3. Create a view that uses shapes.
4. Edit the standard shape by selecting the imported custom shape.

The best results are achieved using images that are sized at (32 × 32) pixels.

CUSTOMIZING COLORS

Creating customized colors for individual marks can be done easily using the Color button in the Marks tab. Make a custom color by clicking the Color button and selecting the More Colors option. This exposes the window shown in the upper left of Figure 7-38.

You can also pick from a wide variety of border color options by modifying the border using the Marks card ➢ Border menu, shown in Figure 7-38 just below and to the right of the Color dialog box. Select the Add To Custom Colors option to expose a wide variety of color tools for border color customization.

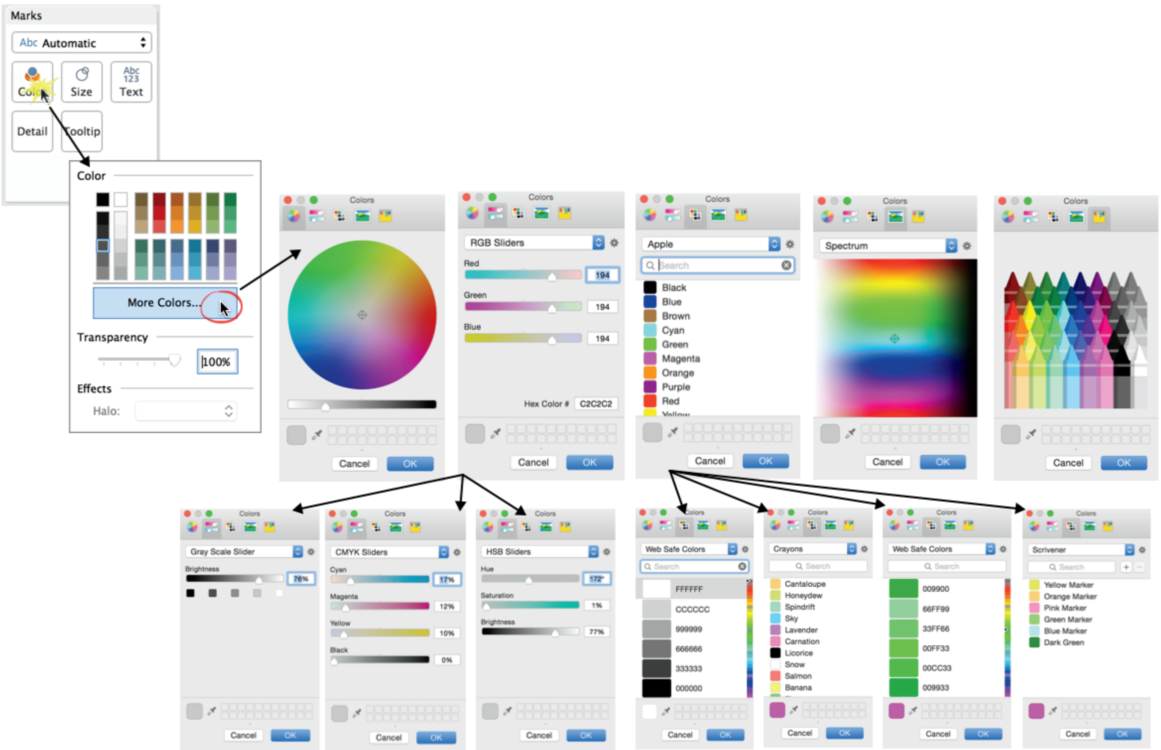


FIGURE 7-38 *Customize an individual color.*

It's also possible to create completely customized color palettes. Tableau took great care to create default color palettes that effectively communicate. They considered factors such as color blindness—providing gray scale and a specific color-blind–friendly palette. If you have a specific need that the available color palettes don't fulfill, try mixing colors from different standard palettes. If you do have a very specific need (perhaps matching a logo color scheme), creating a completely customized palette is possible, but you have to modify Tableau's preferences file located in `\My Documents\My Tableau Repository\Preferences.tps`.

Search Tableau's website for a knowledge base article called "Creating Custom Color Palettes" for specific details. You'll need to use a text editor (such as Windows Notepad) to add the custom palette by adding an XML script that defines the palette name (as you want it to appear) and then define the color values.

CUSTOMIZING FONTS

Tableau provides a wide range of fonts. You can customize the font style, size, color, boldness, and underlining for every element of text contained in headings, axis labels, mark labels, and tooltips. In most cases, the standard font selections work fine. Changing the font style of dynamic title elements is a very common use and helps people notice that values in dashboards change when selections are made. The most common need for customizing fonts is to modify the title font in a pane or dashboard. Double-clicking in the area opens a text editing dialog box, as you can see in Figure 7-39.

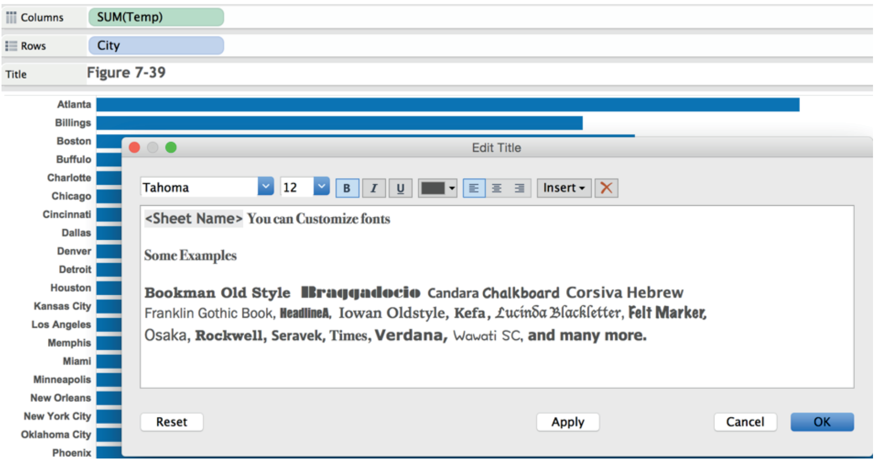


FIGURE 7-39 *Customizing fonts in titles*

The text in the editing area of the dialog box has been modified to display many different font styles. You change fonts by highlighting existing text and selecting another font style from the menu at the top. Figure 7-40 shows custom font colors in a title but also includes a dynamic title element, `<Sheet Name>`.

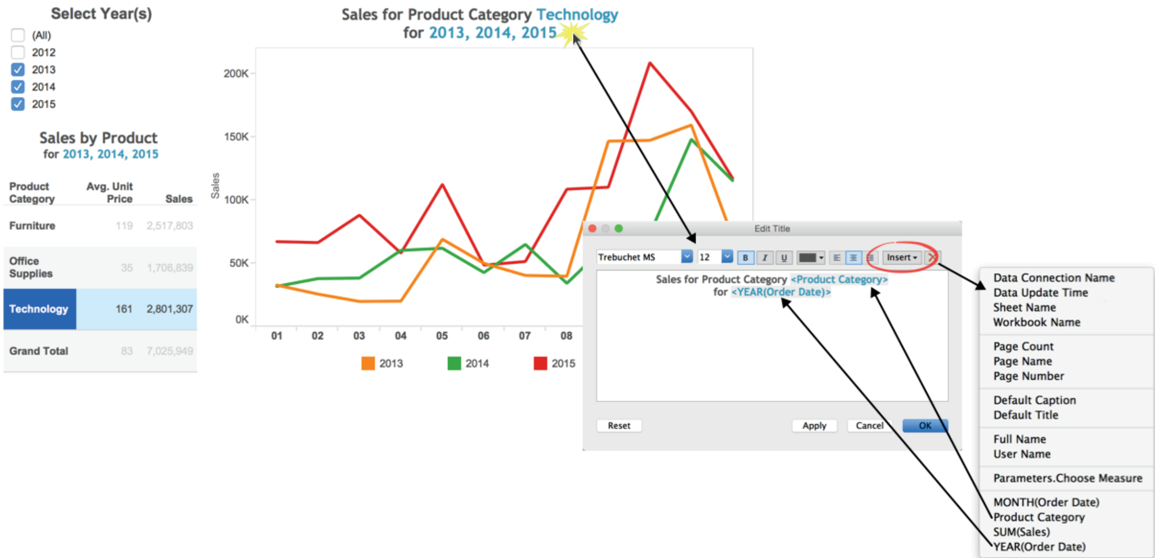


FIGURE 7-40 *Dynamic title elements*

In Figure 7-40, the Year Filter in the left section of the view filters both charts contained in the dashboard for year. You see that 2013, 2014, and 2015 are selected, and the title for each pane in the dashboard reflects those years in the title block. You add dynamic title elements by double-clicking the title and using the insert button to select the field. For the field to be available, you must use it in the view somehow or include it on the Marks card.

CUSTOMIZING IMAGES IN DASHBOARDS

The most typical use for an image in a dashboard is to add a company logo. By using the image object, logos can be placed and sized to fit in the title space. There are a couple of tricks you should be aware of that will help you fit images precisely. The InterWorks logo is a standard JPEG file. After placing the image object into the dashboard at the desired location, select the Pick Image option by pointing to the upper-right corner of the object to expose the menu you see in Figure 7-41.

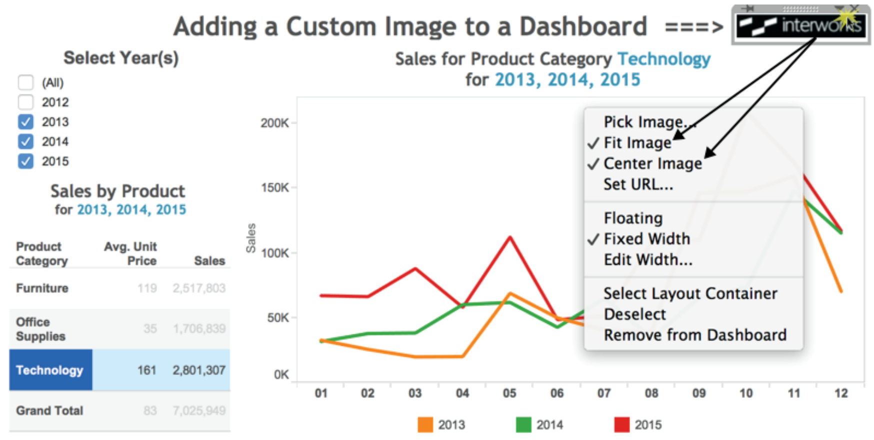


FIGURE 7-41 *Defining an image file for an image object*

After selecting the image, click the Fit Image and Center Image options. These force the image file to resize automatically if you alter the image object size.

ADVANCED CHART TYPES

Tableau provides a complete range of chart styles. You really don't even have to understand why a particular chart is better. If you rely on the Show Me button, Tableau will provide an appropriate chart based on the combination of measures and dimensions you've selected.

There are some useful variations to the default chart types that require a little more knowledge to create. Knowing what default settings to modify makes all the difference. This section reviews six of the most commonly used non-standard chart types.

BAR-IN-BAR CHART

The bar-in-bar chart you see in Figure 7-42 provides another way to compare values.

In this example, color and size denote actual and budgeted sales. The height of each bar expresses the values of each measure for a particular region. The key to building this chart is to understand how to use color and size while altering Tableau's default bar-stacking behavior. To build this example using the coffee chain sample dataset, follow these steps:

1. Multi-select Market, Budget Sales, Sales.

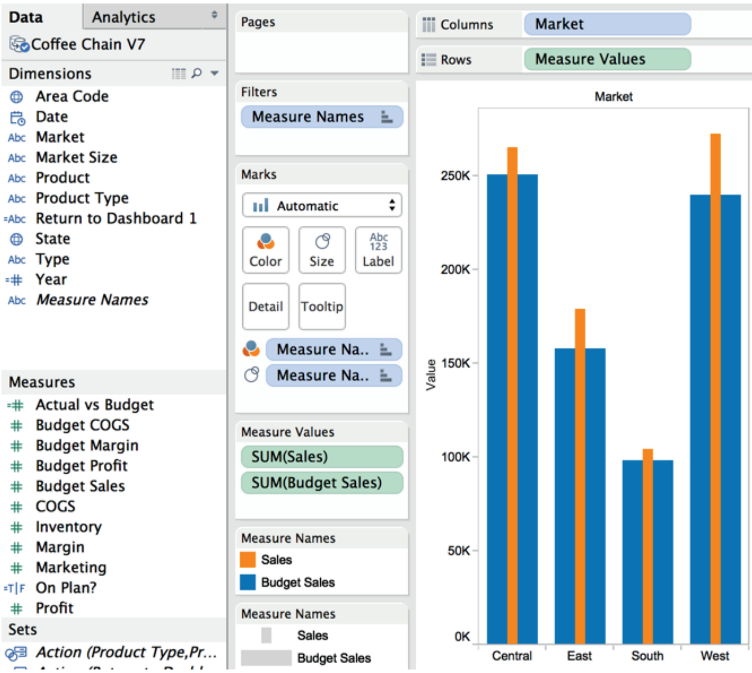


FIGURE 7-42 *Bar in bar chart*

2. Using Show Me, select the Side-by-Side Bar Chart.
3. Move the Measure Names field pill from the Column shelf to the Size button in the Marks card.
4. If you prefer budgeted sales to be the wider bar, drag the SUM (Budget Sales) below the SUM (Sales) pill in the Measure Values card. Alternatively, reorder the Measure Names color legend to accomplish the same thing.
5. Go to the main menu analysis/stack mark and select the Off option.

The bar-in-bar chart has more limited use than a bullet graph, which I cover at the end of the chapter, but this chart type also packs a dense amount of information into a small space. It is particularly useful when you want to compare a small number of measures across a larger number of dimensions.

PARETO CHARTS

Known as the 80–20 Rule, the Pareto Principle was developed by Vilfredo Pareto in 1906 to describe the unequal distribution of wealth in his country.

In general, the (80–20) principle states that 20 percent of the inputs account for 80 percent of the outputs. For example, 80 percent of profits come from 20 percent of the products. Figure 7-43 shows a Pareto chart that displays profit by product. The following example was built using the Superstore Sales for TYD2 sample dataset. You will learn how to create a Pareto chart that plots the cumulative profit generated by each distinct product that Superstore sells.

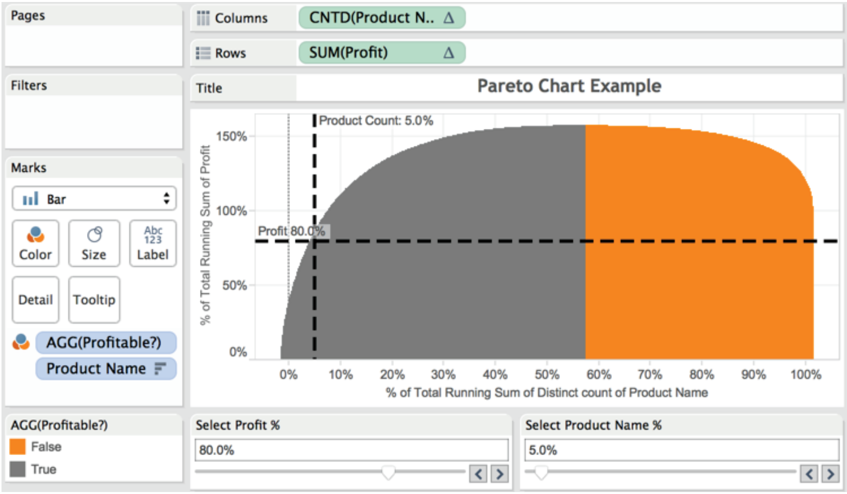


FIGURE 7-43 *Pareto chart—profit by item*

The vertical axis plots the cumulative profits expressed as a percentage of the total profits generated by the business. The horizontal axis plots the contribution of each individual product (item). Color encoding is being used to display positive and negative profit items as discrete groups. Parameterized reference lines are included, which allow the information consumer to move the lines on both the horizontal and vertical axis. In this way, the user can determine how closely the sample conforms to the Pareto Principle. In the case of Figure 7-43, you can see that the sample dataset has 80 percent of product profits being generated from a mere 5 percent of products. This is a much greater concentration than would normally be expected.

The trick to building this chart type is to understand how Table Calculations can be used to express the axis values as percentages of the total values. The steps required to build this chart are as follows:

1. Drag the Product Name dimension to the Columns shelf.
2. Drag the Profit measure to the Rows shelf.

3. Sort the Product Name by descending profit (highest profit to lowest profit item).
4. Change the view from Normal to Entire View using the control on the menu icon bar.
5. Change the SUM (Profit) field on the Rows shelf to a two-stage Table Calculation by right-clicking the field pill and editing Quick Table Calculation, as you see in the top half (section A) of Figure 7-44.

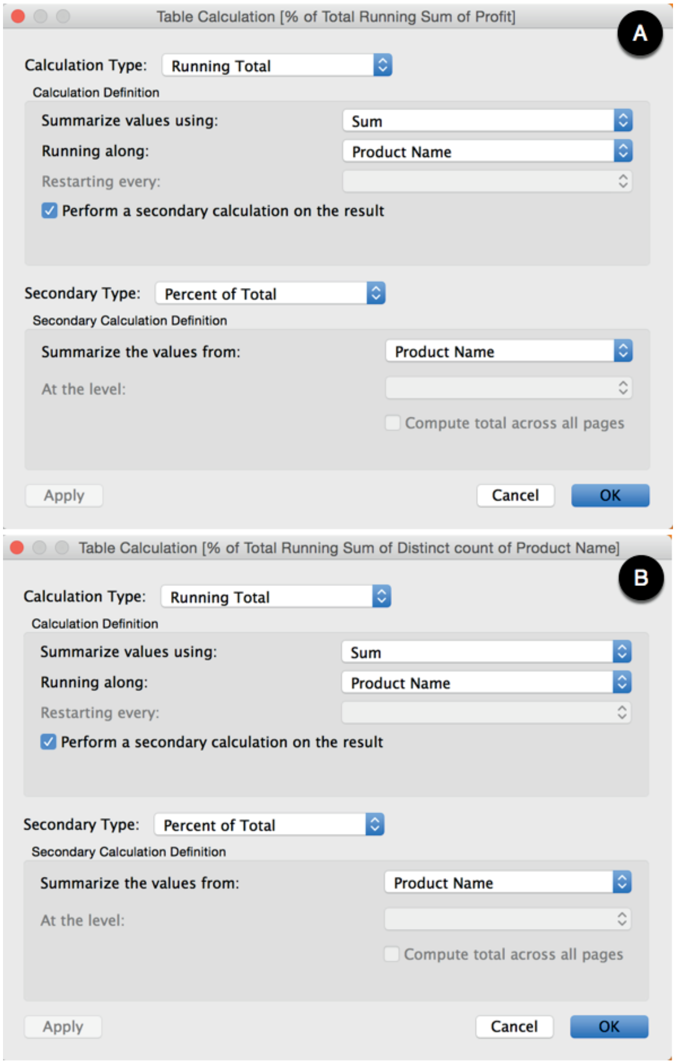


FIGURE 7-44 *Defining the Table Calculations*

6. Drag the Product Name field from the Dimensions shelf to the Marks card.
7. Edit the Product Name field just placed in step 6 by right-clicking the field pill and selecting Measure ➢ Count Distinct.
8. Add a two-stage Table Calculation to the field editing in step 7 by right-clicking the pill and selecting Add Table Calculation. It should look like the definition in the lower half of Figure 7-44 (section B).
9. Drag the new Table Calculation created in step 8 to the Columns shelf and place it to the right of the Product Name pill. Then drag the Product Name field pill from the Columns shelf to the Marks card. Your chart will momentarily look broken. Don't worry—it isn't.
10. Change the mark type in view on the Marks card from Automatic to Bar.
11. Create a calculated value called (Profitable?) to determine if profits are greater than zero. Use this formula: `SUM(Profit)>0`.
12. Place the (Profitable?) calculated value on the Color button located on the Marks card.
13. Add parameterized reference lines on each axis that allow the information consumer to change the location of the reference line from 0 to 100 percent in .01 increments. Refer to Figure 7-45 to view the setting used to create the vertical reference line (section A). The horizontal reference requires a second definition that should be initiated from the horizontal axis, as you see in Figure 7-45 (section B).
14. Edit the color scheme to match the gray/orange colors that indicate profitability.

Once the parameterize reference lines are completed, the only remaining work is repositioning the screen elements to your task. The parameter controls in Figure 7-43 are positioned below the Pareto chart to better utilize the worksheet by reducing the amount of unused white space.

Don't be discouraged if it takes a few tries for you to get this chart type comfortably mastered. There are several ways you could build the chart. You may find another way to create the same effect.

The last two visualizations that you learn about in this chart are closely related to the next chapter on dashboards. Sparklines and bullet graphs work well in dashboards because together they convey a lot of information even when space is restricted.

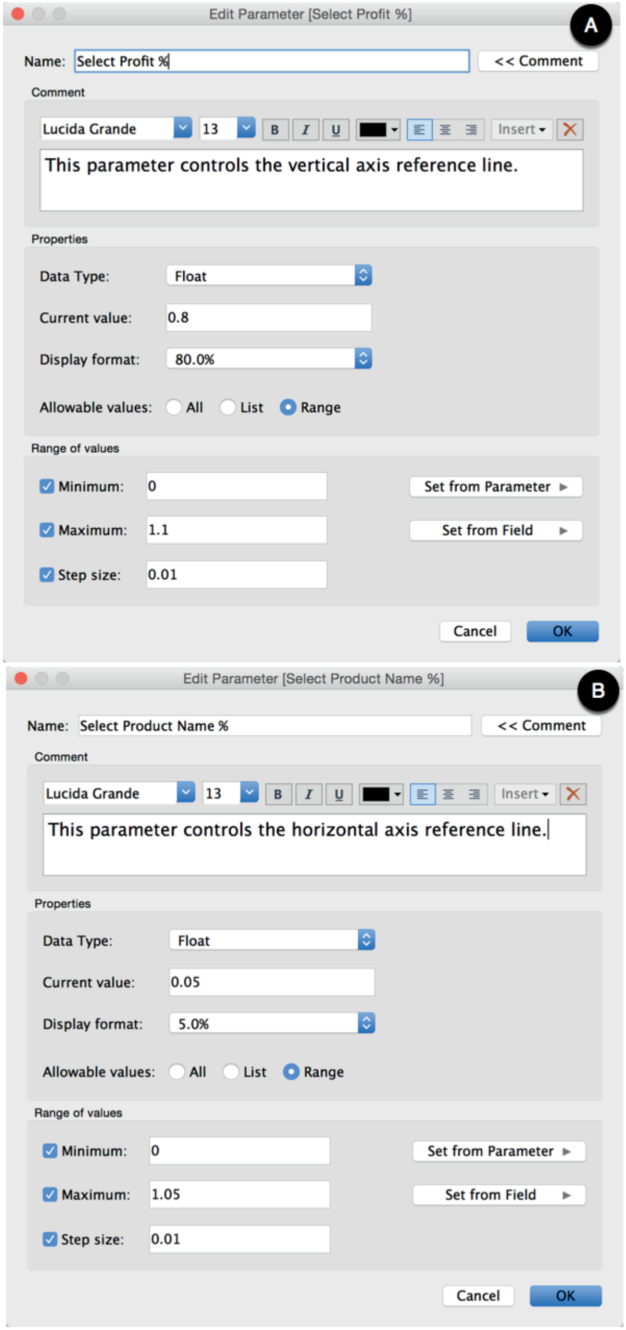


FIGURE 7-45 *Parameterized reference lines*

SPARKLINES

Edward Tufte conceived sparklines in his wonderful book *Beautiful Evidence* (Graphics Press, 2006). He referred to them as "intense, simple, word-sized graphics." Sparklines can provide very effective time series charts in dashboards. When pixel height and width are constrained, you'll find that sparklines can convey a good deal of information in much less space than Tableau's default time series charts. Build sparklines using the following steps:

1. Create a standard time series chart using the Coffee Chain starter workbook.
2. Edit the axis and make each axis range independent.
3. Remove the axis headings.
4. Drag the right edge of the chart to the left.
5. Drag the chart bottom up.
6. Reduce the mark size from the Marks card and remove the borders from the chart while adding shading on alternate rows, as you see in Figure 7-46. You access this from the format menu.
7. Hide the field labels for the rows by pointing at the heading at the top of the first column and selecting the option to hide them.
8. If necessary, emphasize change using a table calculation. You may also need to define how Tableau should handle null values when you define the table calculation.

If you follow these steps correctly, your view should look like Figure 7-46.

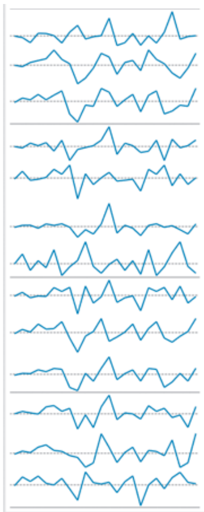


FIGURE 7-46 *Coffee chain sales sparkline*

In this example, it was necessary to use a Percent Change table calculation to emphasize the change in sales month over month. Why? The data was boring and contained very minimal dollar changes, resulting in the "dead man EKG" effect, or flat lines on every row of the time series when the view was compressed. A nice feature of employing a table calculation for percent change is that a very light gray dotted line appears in each chart denoting the zero change level. In addition, some of the normal formatting elements have been removed from the view—axis titles, row and column headings—and the lines separating each product cell have been de-emphasized with a very light gray color.

You will build a sparkline in combination with a bullet graph as part of an exercise in the next chapter on dashboard technique.

BULLET GRAPHS

Bullet graphs were developed by Stephen Few as another means for efficiently comparing metrics in a limited space. Bullet graphs are bar charts (comparing one-to-many relationships) with the addition of comparative reference lines and reference distributions. Bullet graphs, in combination with sparklines, are an excellent combination in dashboards because they are space efficient and insightful. Look closely at the bullet graph in Figure 7-47.

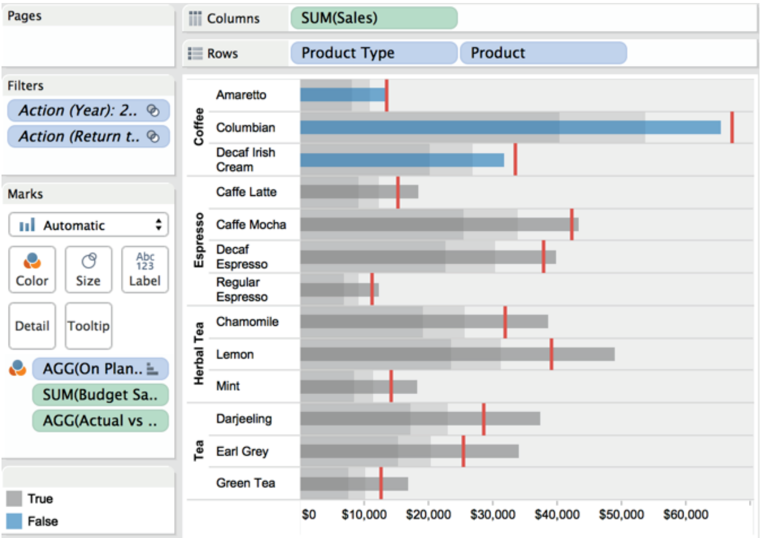


FIGURE 7-47 *Bullet graph*

The bars in the bullet graph have been color-coded to reflect the result of a Boolean (true/false) calculation that evaluates Actual versus Planned Sales. Products that are encoded in blue are below plan. The cell-level reference lines in red reflect the budgeted sales value. The gray encoding of the reference distribution behind the bars reflects levels of performance versus the budget as well (60 percent, 80 percent of budget). Also notice that the color of the actual sales bars has been faded to 6 percent using the Color button on the Color shelf. So this bullet graph was built using Show Me but includes several appearance tweaks to enhance understanding. The steps required to build the example in Figure 7-47 included the following:

1. Open the coffee chain sample database.
2. Multi-select Sales, Budget Sales, Product Type, and Product.
3. Click the Show Me button.
4. Check that the bars use Actual Sales.
5. Check that the reference line uses Budget Sales.
6. Items 4 and 5 will be wrong. Right-click the bottom axis and choose the Swap Reference Line Fields.
7. Create a Boolean calculation `sum([sales]) < sum(budget sales)]`.
8. Drop the Boolean calculation result on the Color button.
9. Style the reference line to taste.
10. Style the reference distribution color scheme to taste.

The bars in bullet graphs should reflect the Actual Value. The reference line should reflect the Comparative Value (budget, prior year, and so on). Tableau doesn't try to determine the actual versus target value when the graph is created automatically using the Show Me button. You may have to use the Swap Reference Line Fields option, which is accessed by right-clicking within the white space of the bottom axis. This swaps the pill placed in the Column shelf and the Marks card. It should make sense by now that the pill being expressed in the Column (or Row shelf) is plotted using the bar. The pill contained in the Marks card is used to create the reference line.

The combination of sparklines and bullet graphs in dashboards provides a very space-efficient way to compare performance to plan and performance versus prior years (if you add reference lines for that). The sparkline provides a very dense information-packed display of performance over time. Figure 7-48 shows them aligned in a dashboard.

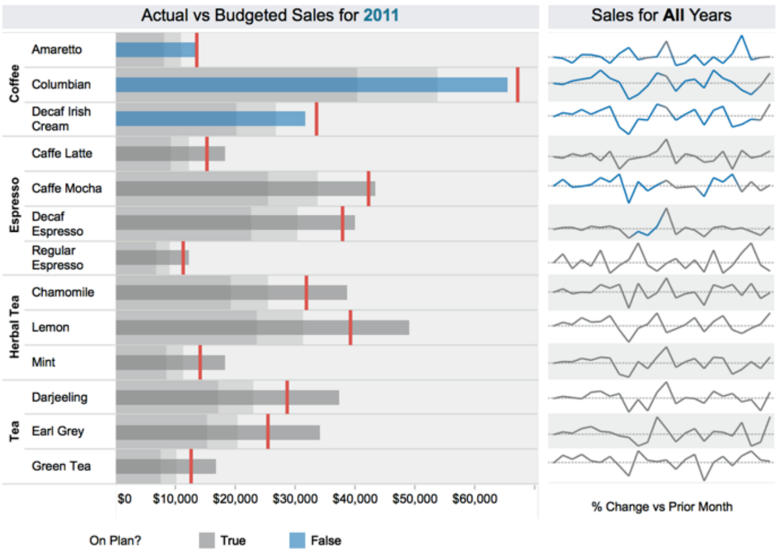


FIGURE 7-48 *Bullet graph and sparkline*

In the next chapter, you learn about best practices for dashboard design—using a bullet graph and sparkline along with other visualizations to create a compact, information-rich dashboard design.

CHAPTER 8

Bringing It All Together with Dashboards

Storytelling is the creative demonstration of truth. A story is the living proof of an idea, the conversion of idea to action.

Robert McKee[1]

An essential element of Tableau's value is delivered through dashboards. Allowing the audience to interact with a dashboard and change the details being displayed provides a means to shift context—leading to new and potentially important discoveries. Assembling dashboards in Tableau is fun for the designer, and good dashboard design can delight information consumers.

HOW DASHBOARDS FACILITATE ANALYSIS AND UNDERSTANDING

When reviewing reports or creating new analytical reviews of data, you are looking for a story—something of value that you can share with others to enable change for the better. Dashboards fortify this storytelling by providing complementary views of the data and turning the data into actionable information that is supported by facts.

Well-designed dashboards are also visually interesting and draw the user in to play with the information, providing details on-demand that enable the information consumer to understand what, who, when, where, how, and perhaps even why something has changed.

HOW TABLEAU IMPROVES THE DASHBOARD-BUILDING PROCESS

Only three things really matter when it comes to business information—speed, accuracy, and the ability to make a new inquiry. Tableau delivers on all three. Tableau's capability to directly connect to a variety of data sources and render the data using the appropriate visualizations provides three distinct advantages over traditional data analysis tools:

- Reduced dashboard development time
- Reduced technical resource involvement
- Better visual analytics

Tableau reduces the need for technical staff in the dashboard development process by providing a user-friendly environment that doesn't require knowledge of database schema, SQL scripting, or programming. Creating dashboards with Tableau is primarily a drag-and-drop operation. When individual chart panes are placed into the dashboard workspace, filtering and highlighting between panes are also accomplished with point-and-click efficiency.

Publishing your dashboard for consumption on personal computers, tablets, or the Internet requires no technical programming skills either. After learning a few basic principles, you will be creating compelling visual analytics in dashboards more quickly than was ever possible with older tools. Something magical happens the first time people use Tableau and gain a new insight. They begin to understand the potential unlocked when the tool disappears and the information becomes the center of attention.

In this chapter, you learn field-tested techniques that will help you to build dashboards that effectively communicate to your audience. The chapter covers the following:

- Recommended best practices for building dashboards with Tableau
- The mechanics of the dashboard shelves and design objects
- Using actions to filter, highlight, and embed web pages
- Publishing dashboards to Tableau Server or Tableau Online
- Performance tuning dashboards for fast load and query times

You will learn these techniques by building a dashboard that adheres to best practices using sample data included with your Tableau Desktop software. Before building this example, let's discuss the wrong and right ways to build a dashboard using Tableau.

THE WRONG WAY TO BUILD A DASHBOARD

Traditional providers of reporting tools have been companies that have a core competency in data collection and storage. These entities attract people who are very knowledgeable in the technical aspects of database building, data quality, and data storage, but not data presentation.

Traditional buyers of business information systems tend to be people from finance and accounting. The information technology staff are normally involved in the procurement process because the IT staff possess the technical knowledge of database design, data collection, and data governance. Plus, IT is usually responsible for installing and maintaining the system.

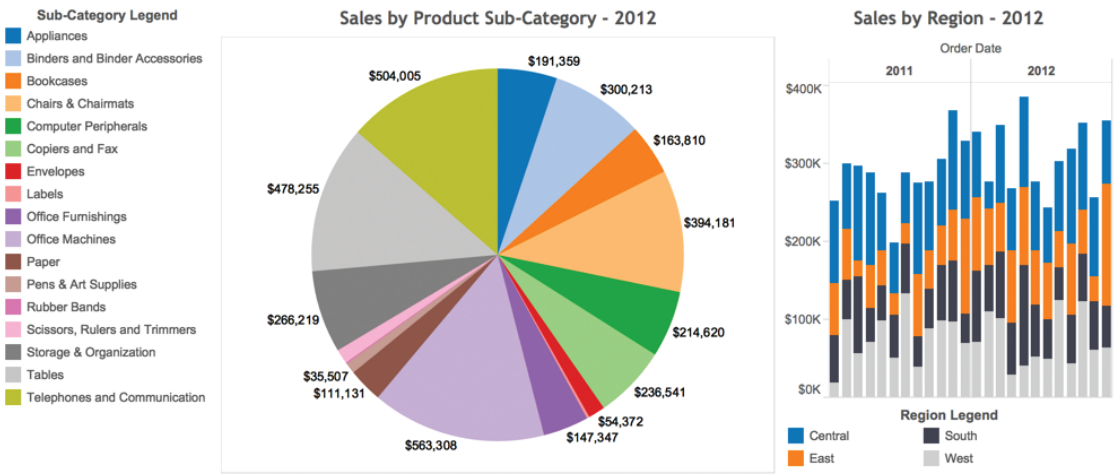
Neither group possesses knowledge of the best practices related to data visualization. The IT group's knowledge of charting typically comes from the commonly available spreadsheet programs, which often provide a lot of unnecessary and inappropriate chart styles. Historically, older business information (BI) tools that information technology staffs are familiar with for report building have been more adept at data creation and storage—not information visualization.

Good report builders from both of these groups develop time-saving techniques that work well for creating dashboards in old-style tools. Unfortunately, those techniques are more concerned with the technical challenges of building the report, not the aesthetic qualities of the user experience.

Why would an experienced designer use overly complex graphics? One possible reason is that dashboards created with legacy tools are more difficult to build—requiring more time and effort to produce. Often, with legacy tools it makes sense to place as much information as possible into a single view to save time. This practice can lead to visualizations that are complex and difficult for end users to understand. Also, internal customers ask for what is familiar (grids of numbers), so that is what they receive. Unfortunately, these techniques are exactly the wrong way to build dashboards in Tableau.

Relying on grids or overly complex individual charts generally accomplishes two undesirable outcomes in Tableau. First, the dashboard doesn't communicate effectively. Second, it doesn't load as quickly as it should.

For example, a sales report displaying 12 months of history for 20 products ($12 \times 20 = 240$ data points) does not help the information consumer see the trends and outliers as easily as a time-series chart of the same information. Also, the quality of the data won't matter if your dashboard takes five minutes to load. Dashboard viewing is an activity that resembles browsing a website. Web browsing isn't very useful if you have a slow connection. Viewing a dashboard isn't either if it takes a long time to load or if the interactivity is slow. The dashboard shown in Figure 8-1 displays some common pitfalls—overly dense and complicated charts and inappropriate chart types. Note the pie chart for comparing sales by product subcategory. The stacked bar chart uses a different (conflicting) color legend to display sales by region. The pie chart has too many slices, and performing precise comparisons of each product subcategory is difficult. The crosstab at the bottom requires that the user scroll to see all the data.



Sales and Profit Trend by Producdt Type and Product - 2012

		Sales					Profit %				
		2012					2012				
		Q1	Q2	Q3	Q4	Total	Q1	Q2	Q3	Q4	Total
Furniture	Bookcases	$47,632	$47,442	$20,874	$47,862	$163,810	-9.3%	-6.9%	-17.1%	-26.5%	-14.6%
	Chairs & Chairmats	$118,987	$85,165	$65,797	$124,233	$394,181	18.4%	2.1%	8.0%	4.6%	8.8%
	Office Furnishings	$28,965	$27,787	$61,818	$28,778	$147,347	18.6%	4.0%	13.5%	8.9%	11.8%
	Tables	$141,715	$85,051	$138,484	$113,005	$478,255	-4.1%	-5.9%	-12.0%	-5.3%	-7.0%
	Total	$337,299	$245,445	$286,972	$313,878	$1,183,593	5.0%	-2.2%	-2.3%	-3.3%	-0.5%
Office Supplies	Appliances	$23,976	$65,083	$26,192	$76,108	$191,359	-4.3%	20.9%	-17.2%	19.9%	12.1%
	Binders and Binder Accessori..	$95,656	$68,732	$53,832	$81,993	$300,213	34.2%	29.6%	24.9%	34.0%	31.4%
	Envelopes	$8,094	$20,040	$12,656	$13,581	$54,372	27.2%	35.1%	19.5%	22.0%	27.0%
	Labels	$1,726	$1,815	$2,758	$2,008	$8,306	35.2%	41.1%	39.4%	27.5%	36.0%
	Paper	$28,227	$26,325	$32,324	$24,255	$111,131	10.5%	19.7%	6.8%	5.4%	10.5%
	Pens & Art Supplies	$9,272	$9,286	$9,791	$7,158	$35,507	2.6%	0.1%	5.4%	3.1%	2.8%
	Rubber Bands	$855	$602	$1,350	$1,162	$3,969	0.8%	10.8%	12.5%	2.0%	6.6%
	Scissors, Rulers and Trimmers	$1,707	$900	$2,706	$41,508	$46,821	-10.9%	-28.5%	-25.9%	-1.4%	-3.7%
	Storage & Organization	$71,770	$93,764	$75,590	$25,095	$266,219	1.0%	-7.7%	3.1%	-22.1%	-3.6%

FIGURE 8-1 *A poorly designed dashboard*

The dashboard fails to convey important information quickly. Presenting the data this way can also lead to performance problems if there are a large number of rows being displayed in the product crosstab.

Fixing these problems is typically not difficult. Tableau is designed to supply the appropriate graphics by default. Understanding why a dashboard loads slowly and how to ensure good speed requires only a basic understanding of how Tableau renders the information. We will dive into those details at the end of this chapter.

THE RIGHT WAY TO BUILD A DASHBOARD

How can you improve the previous dashboard and ensure that it loads quickly? Can the crosstab be eliminated in order to reveal what is important in this data? A more effective dashboard conveys the information with less noise and provides details on demand.

The dashboard shown in Figure 8-2 uses a bar chart to provide a more precise comparison of sales by product subcategory (color-encoded bars). The time-series combination line and bar chart at the bottom provides sales by month (the vertical bars are color-encoded gray and black for a 5 percent profit threshold) and year-to-date sales by product category (color-encoded lines matching the bar colors in the bar chart above). The small crosstab in the upper-right corner of the dashboard provides summary information by region. The use of grayscale to depict a profit ratio threshold in the time-series combination line and bar chart provides additional insight into overall profitability. Darker gray is used to highlight product categories and months in which profit ratio is under 5 percent.

The headings contain dynamic elements denoting the regions, product categories, and subcategories that have been selected using filter actions embedded in the bar chart and region text table. The dashboard has been filtered for the central and south regions as well at the technology and office supply product categories. These selections are highlighted in the bar chart and region crosstab.

This dashboard communicates more effectively by removing clutter and unnecessary details. The audience for this dashboard might include senior managers and regional sales staff. This design would serve both groups.

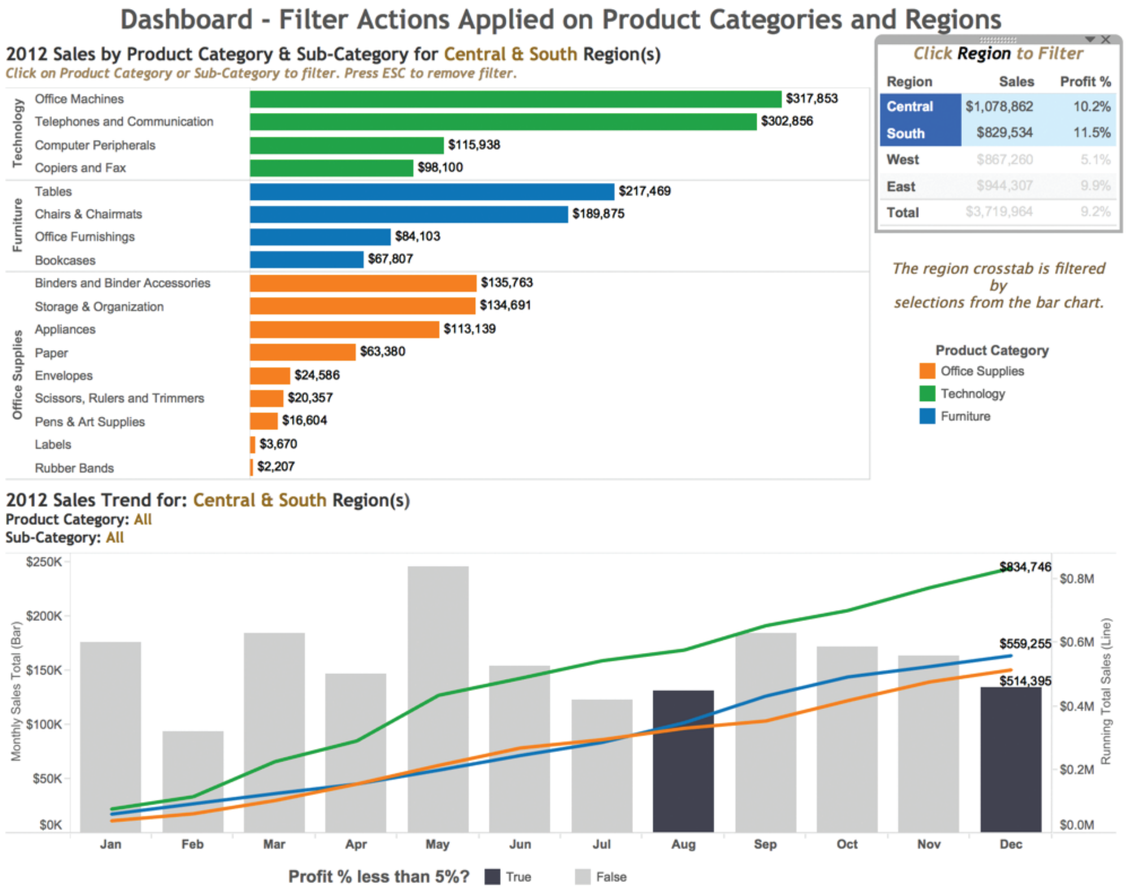


FIGURE 8-2 *A dashboard using simpler views*

BEST PRACTICES FOR DASHBOARD BUILDING

After you have analyzed some data and determined what information you need to share, adhering to these principles will help you create better dashboard designs:

- Size the dashboard to fit in the worst-case available space.
- Employ four-pane dashboard designs.
- Use Actions to filter instead of Quick Filters.
- Build cascading dashboard designs to improve load speed.
- Limit the use of color to one primary color scheme.

- Use small instructions near the work to make navigation obvious.
- Filter information presented in text tables to provide relevant details on-demand.
- Remove all non-data ink.
- Avoid one-size-fits-all dashboards.

Work to achieve initial dashboard load times of less than 10 seconds. These principles come from personal lessons learned building dashboards in a wide variety of use cases. They work well for 90 percent of the use cases across industry, government, and education.

You may find specific use cases for which violating one or more of these best practices performs well and communicates the information effectively. By all means then, do what works best for your specific case.

SIZE THE DASHBOARD TO FIT THE WORST-CASE AVAILABLE SPACE

Dashboard building would be easy if everyone had the best computer with high-resolution graphics. Unfortunately, this normally isn't the case, so you must design your dashboard to fit comfortably in the available space by determining the pixel height and width of the worst-case dashboard consumption environment. Tableau provides defaults for the typical sizes you will need or allows you to define a custom size. Doing a lot of design work without knowing the consumption environment is a recipe that results in unhappy information consumers and extra work for the designer.

Will the dashboard be consumed on laptops via Tableau Reader? If so, do you know the range of screen resolutions that are being used? Are tablet computers used? Is the dashboard going to be consumed via Tableau server, or will you have to embed the dashboard in a website? You need to understand the specific height and width of your dashboard space. For laptop consumption, this can be as little as 800 × 600 pixels. For desktop computers or better resolution laptop monitors, 1000 × 800 pixels normally works well. Web-embedded dashboards can be smaller but a typical worst-case minimum size might be as little as 420 × 420 pixels. Tableau has predefined sizes to help you lay out the dimensions of your dashboard. Tableau also makes it easy to define custom size ranges if the default values don't meet your needs.

EMPLOY FOUR-PANE DASHBOARD DESIGNS

Four individual visualizations will fit well on most laptop and desktop computer screens, as shown in Figure 8-3. This style of presentation naturally highlights the upper-left pane because people in western societies have been taught to read from the upper left to the lower right of a page. Figure 8-3 shows a four-pane design intended for laptop or desktop consumption.

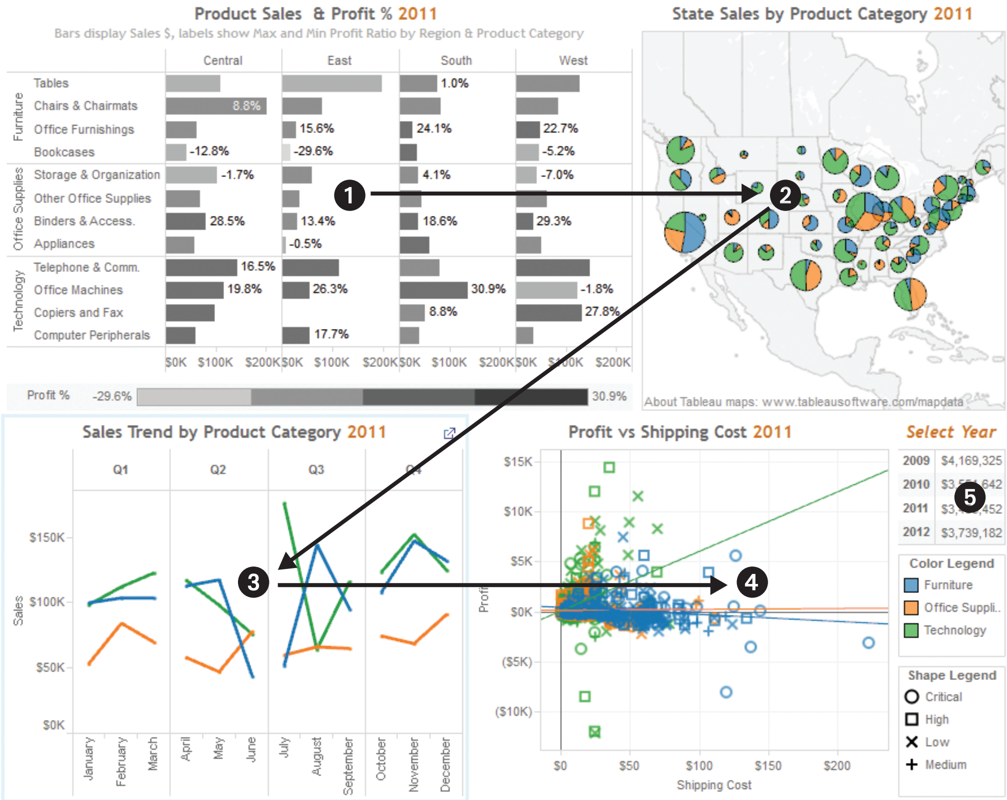


FIGURE 8-3 *A four-pane design*

A four visualizations design style will generally be read from the upper left to the lower right in a *Z* pattern unless you do something to grab attention elsewhere. Note that this design actually includes five panes—but the fifth pane, (the small Select Year crosstab) acts as a filter for the rest of the dashboard. Ordinarily, a quick filter would be used to permit the audience to select the year in view. Instead, the example in Figure 8-3 uses a small text table to trigger a filter action. The advantage of using a text table instead of a quick filter is that additional information is provided (total sales for each year) in the same amount of space that a multi-select quick filter would have required. The design employs a fifth data pane (an apparent contradiction to the best practice) but in a way that is consistent with the recommendation. Another reason to use a text table for this purpose leads to the next best practice recommendation for using filter actions in place of quick filters.

Different rules apply when designing dashboards for tablet computers. Designing for tablet computers will be covered in detail at the end of this chapter.

USE ACTIONS TO FILTER INSTEAD OF QUICK FILTERS

Using actions in place of quick filters provides a number of benefits. First, the dashboard will load more quickly. In order to visualize quick filters, Tableau must scan the source table from your database. If the table you are scanning is large, it can take some time for Tableau to render the filter. Tableau has improved quick filter load performance over the last several releases, but you may opt to use filter actions for another reason—aesthetics. In the same space that is required to display a multi-select filter, you can provide a small visualization with a filter action that enables filtering, but in a way that also enhances content included in the dashboard.

Employing multiple quick filters in a dashboard is also potentially confusing to the audience. My personal worst-case scenario involved a client dashboard with two data panes and thirteen quick filters. The source database was very large (billions of records). It required 6 minutes and 30 seconds to load—all but 8 seconds of that time was required to visualize the quick filters. Not only was it difficult to find the right filters, it was slow loading. By altering the design to a series of four-pane dashboards and replacing the quick filters with filter actions, load time for each dashboard was reduced to less than 8 seconds. This leads to the next best practice recommendation.

BUILD CASCADING DASHBOARD DESIGNS TO IMPROVE LOAD SPEEDS

Achieving fast load times can be challenging if the source data is very large. In the case mentioned in the preceding section, the load speed of the dashboard was terrible because many of the 13 quick filters used in the original design were scanning massive tables. The executive who requested the dashboard needed to see the data summarized globally first, but also wanted to be able to drill into much more detailed subsets of the data. Unfortunately, the initial design was slow-loading and didn't provide much insight.

Replacing the use of quick filters in the original dashboard with a design that employed a series of four-pane dashboards that used filter actions in place of the quick filters dramatically-improved the dashboard's load speed and made the information presented is much easier to understand.

The redesigned primary dashboard provided a good overview of operations by showing a bar chart (comparing different products), a map (to show data geographically), a scatter plot (to provide outlier analysis), and a small text table with very high-level metrics. Filter actions were added to these visualizations, which allowed the executive to see more detailed information in other dashboards that were pre-filtered by the selections made on the main dashboard. This cascading dashboard style provided all the information requested, but in a way that improved load speed and understandability.

The final design replaced the original dashboard (that had 13 quick filters) with four cascading, four-panel dashboards (filtering being provided through filter actions within the dashboard panes). The top-level dashboard provided a summary view, but included filter actions in each of the visualizations that allowed the executive to see data for different regions, products, and sales teams. None of the new dashboards required more than 8 seconds to load.

If you employ this recommendation and you are experiencing slow performance, Tableau's Performance Recorder provides visibility of the technical details you will need to troubleshoot the issues that may be degrading performance. You'll learn more about the Performance Recorder at the end of this chapter.

LIMIT THE USE OF COLOR TO ONE PRIMARY COLOR SCHEME

Too much color on a dashboard is confusing. Try to limit the use of color to expressing one dimension or one measure. You can effectively add a secondary use of color in the same dashboard if that secondary use of color employs a more muted color scheme. The dashboard in Figure 8-2 used two colors more effectively than the dashboard in Figure 8-1 because the secondary use of color expressed a limited set of values (true/false) and the color was expressed using a muted shade of gray. According to data visualization expert and author Stephen Few, up to 10 percent of males and 1 percent of females have some form of color blindness.[2] The most prevalent form of color blindness limits the ability to distinguish red and green. Take this into consideration if your dashboard will be utilized by a large population. To avoid potential problems, apply grayscale or blue-orange color palettes. These are visible to most color-blind people. Tableau also provides a color-blind palette with ten colors. You may also consider building color-blind–specific dashboards if you have a very large population of information consumers.

USE SMALL INSTRUCTIONS NEAR THE WORK TO MAKE NAVIGATION OBVIOUS

Quick filters are obvious. Actions are not. Because actions are triggered by selecting elements of your visualizations, they will not be obvious to your audience unless you provide instructions within the dashboard. Placing instructions in the title bar of the worksheet that triggers the action is a good way to remind people of the availability of the action.

Use a consistent font style and color for these instructions in your dashboards so that your audience learns that style denotes an instruction. The instructions used in the Figure 8-2 dashboard are highlighted through the use of a brown italic font.

Another alternative is to place the instructions in tooltips that appear when the user hovers over marks, as shown in Figure 8-4. This method offers the advantage of having the instructions appear in more complete text without crowding the dashboard space.

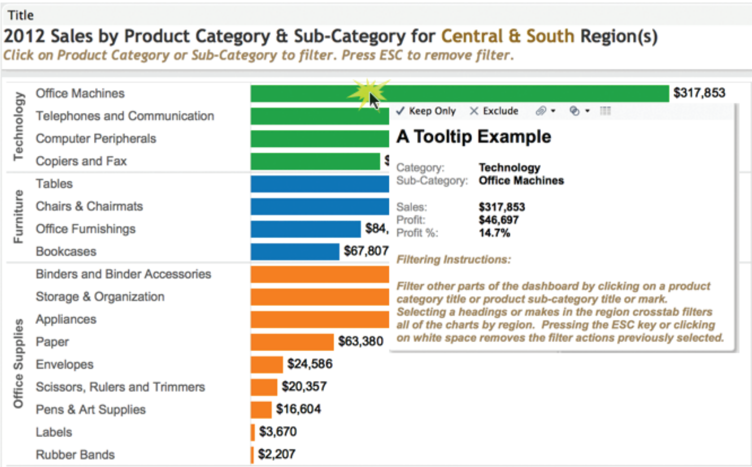
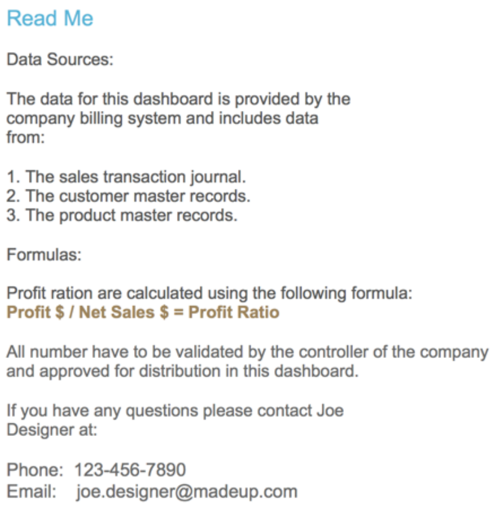


FIGURE 8-4 *Instruction in a tooltip*

Note that the format of the instruction matches the color, font, and style of the instructions in the headers of the dashboard.

Give your audience even more explanatory information by adding a separate Read Me dashboard that includes additional details regarding the data sources used, formulas used, and navigation tips. You can even include links to websites that provide even more information, as shown in Figure 8-5.



Read Me

Data Sources:

The data for this dashboard is provided by the company billing system and includes data from:

1. The sales transaction journal.
2. The customer master records.
3. The product master records.

Formulas:

Profit ration are calculated using the following formula:
Profit $ / Net Sales $ = Profit Ratio

All number have to be validated by the controller of the company and approved for distribution in this dashboard.

If you have any questions please contact Joe Designer at:

Phone: 123-456-7890
Email: joe.designer@madeup.com

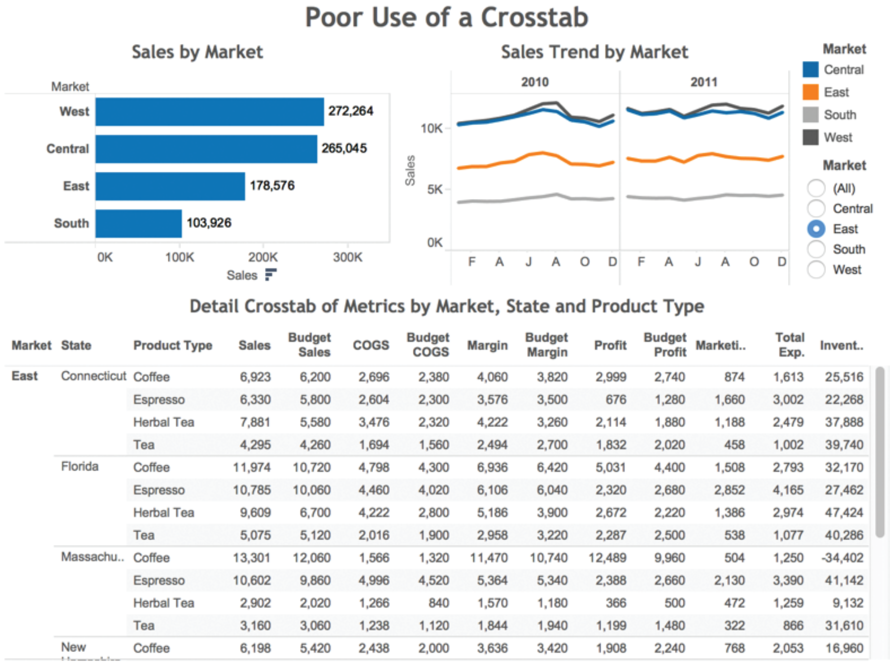
FIGURE 8-5 *A Read Me dashboard*

The time required to add this information to your dashboard will more than pay for itself in reduced phone calls and confusion for the people who are using your dashboard. Finally, provide your contact information so that people can easily ask any other questions that may not be anticipated in your design.

FILTER INFORMATION PRESENTED IN CROSSTABS TO PROVIDE RELEVANT DETAILS-ON-DEMAND

Crosstabs are useful visualizations for looking up specific values when you know exactly what you're looking for. Crosstabs are not the best visual style for quickly discovering trends and outliers. Figure 8-6 shows the poor use of a text table view. Even though the text table in view has been filtered for a specific dimension, vertical scrolling is still necessary in order to see all of the state values.

There is also a lot of white space generated by the column headers for Market and State in Figure 8-6. This dashboard could be improved by creating a filter action from the bar chart that could restrict the market displayed in the bar chart to a single market, but even with a filter action for market, the text table would still require scrolling to see all the values.



Market	State	Product Type	Sales	Budget Sales	COGS	Budget COGS	Margin	Budget Margin	Profit	Budget Profit	Marketi..	Total Exp.	Invent..
East	Connecticut	Coffee	6,923	6,200	2,696	2,380	4,060	3,820	2,999	2,740	874	1,613	25,516
		Espresso	6,330	5,800	2,604	2,300	3,576	3,500	676	1,280	1,660	3,002	22,268
		Herbal Tea	7,881	5,580	3,476	2,320	4,222	3,260	2,114	1,880	1,188	2,479	37,888
		Tea	4,295	4,260	1,694	1,560	2,494	2,700	1,832	2,020	458	1,002	39,740
	Florida	Coffee	11,974	10,720	4,798	4,300	6,936	6,420	5,031	4,400	1,508	2,793	32,170
		Espresso	10,785	10,060	4,460	4,020	6,106	6,040	2,320	2,680	2,852	4,165	27,462
		Herbal Tea	9,609	6,700	4,222	2,800	5,186	3,900	2,672	2,220	1,386	2,974	47,424
		Tea	5,075	5,120	2,016	1,900	2,958	3,220	2,287	2,500	538	1,077	40,286
	Massachu..	Coffee	13,301	12,060	1,566	1,320	11,470	10,740	12,489	9,960	504	1,250	-34,402
		Espresso	10,602	9,860	4,996	4,520	5,364	5,340	2,388	2,660	2,130	3,390	41,142
		Herbal Tea	2,902	2,020	1,266	840	1,570	1,180	366	500	472	1,259	9,132
		Tea	3,160	3,060	1,238	1,120	1,844	1,940	1,199	1,480	322	866	31,610
	New	Coffee	6,198	5,420	2,438	2,000	3,636	3,420	1,908	2,240	768	2,053	16,960

FIGURE 8-6 *A poor use of a text table*

In Figure 8-7, the text table is much more compact. The Market and State dimensions are being displayed in the title dynamically, and the orientation of the text table has been changed to place the measures (11 fields) in rows and the Product Type (4 fields) in columns. This reduces white space and eliminates the need for scrolling.

The unfiltered version of the dashboard on the left of Figure 8-7 clearly shows all the information without any scrolling. The filtered version on the right of Figure 8-7 shows more granular data in both the time series and the text table. This is accomplished using filter actions triggered from the bar chart and map—providing details on-demand for the markets and states of interest. The use of dynamic titles in the time series and text table visualization (highlighted in Figure 8-7) communicates the information more effectively in less space.

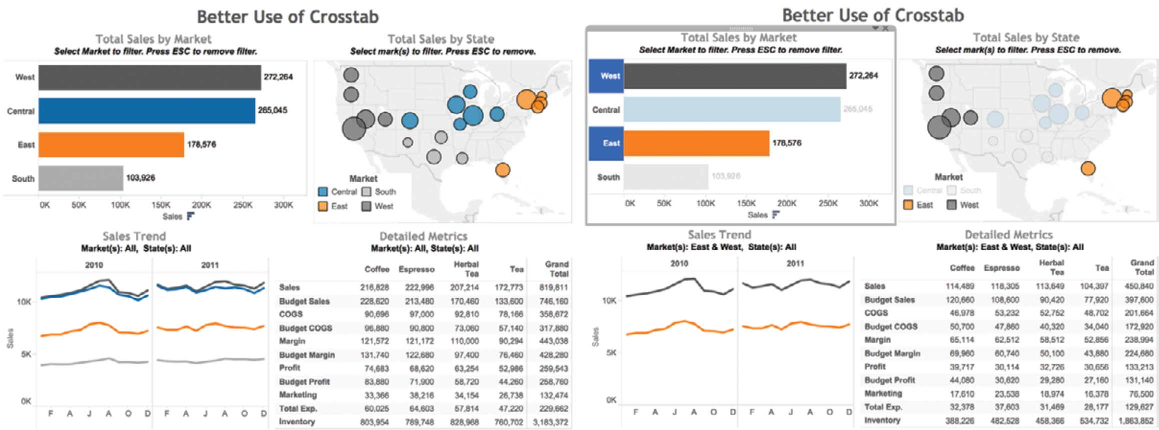


FIGURE 8-7 *A better use of a text table*

REMOVE ALL NON-DATA-INK

This best practice rule is inspired by Edward Tufte, author of *The Visual Display of Quantitative Information*.[3] Remove any text, lines, or shading that doesn't provide actionable information. Remove redundant facts. If a company logo isn't required for promotion purposes, remove it. Ruthlessly eliminate anything that doesn't help your audience understand the story contained in the data.

AVOID ONE-SIZE-FITS-ALL DASHBOARDS

Trying to save time by making one dashboard serve many purposes will not result in the best performing dashboard or save design time. It is so easy to build dashboards and apply data restrictions within data extracts that I recommend making your dashboards fit the particular purpose of each audience. Generally, executives need to see high-level data across multiple geographies, product lines, and markets. Regional staff need more granular data but for restricted geographies, products, and customers.

While it is possible to make one dashboard that works for both groups, it normally doesn't produce the best possible format or the best-performing dashboard for either. Strive to provide the best possible experience for each audience even if that requires a little extra effort.

WORK TO ACHIEVE DASHBOARD LOAD TIMES OF LESS THAN TEN SECONDS

Fast load times are dependent on the size and complexity of your data as well as the type of data source you are using. Slow-loading dashboards can also be caused by poor dashboard design. There are several ways that the dashboard design itself can contribute to slow load speeds. Including high granular visualizations (that plot a large number of marks) can consume resources and slow load times. Using too many quick filters or trying to filter a very large dimension set can slow the load time because Tableau must scan the data to build the filters.

Tableau includes built-in tools for both Tableau Desktop and Tableau Server that help you identify performance issues. At the end of this chapter, you'll learn about the desktop version of Tableau's Performance Recorder. The server version is covered in Chapter 11.

BUILDING YOUR FIRST ADVANCED DASHBOARD

Creating dashboards with Tableau is an iterative process. There isn't a single best method. Starting with a basic concept, discoveries made along the way lead to design refinements. Feedback from your target audience provides the foundation for additional enhancements. With traditional BI tools, this is a time-consuming process. Tableau's drop-and-drag ease of use facilitates rapid evolution of designs and encourages discovery.

INTRODUCING THE DASHBOARD WORKSHEET

After creating multiple, complementary worksheets, you can combine them into an integrated view of the data using the dashboard worksheet. Figure 8-8 shows an empty dashboard workspace.

The top-left half of the dashboard shelf displays all of the worksheets contained in the workbook. The bottom half of the same space provides access to other object controls for adding text, images, blank space, or live web pages into the dashboard workspace. The worksheets and other design objects are placed into the dashboard by dragging the selected object into the "Drop sheets here" area. The bottom-left dashboard area contains controls for specifying the size of the dashboard and a checkbox for adding a dashboard title.

In this chapter you will step through the creation of a dashboard using a data source that ships with Tableau called Coffee Chain. You will create the dashboard by employing the best practices recommended earlier in the chapter.

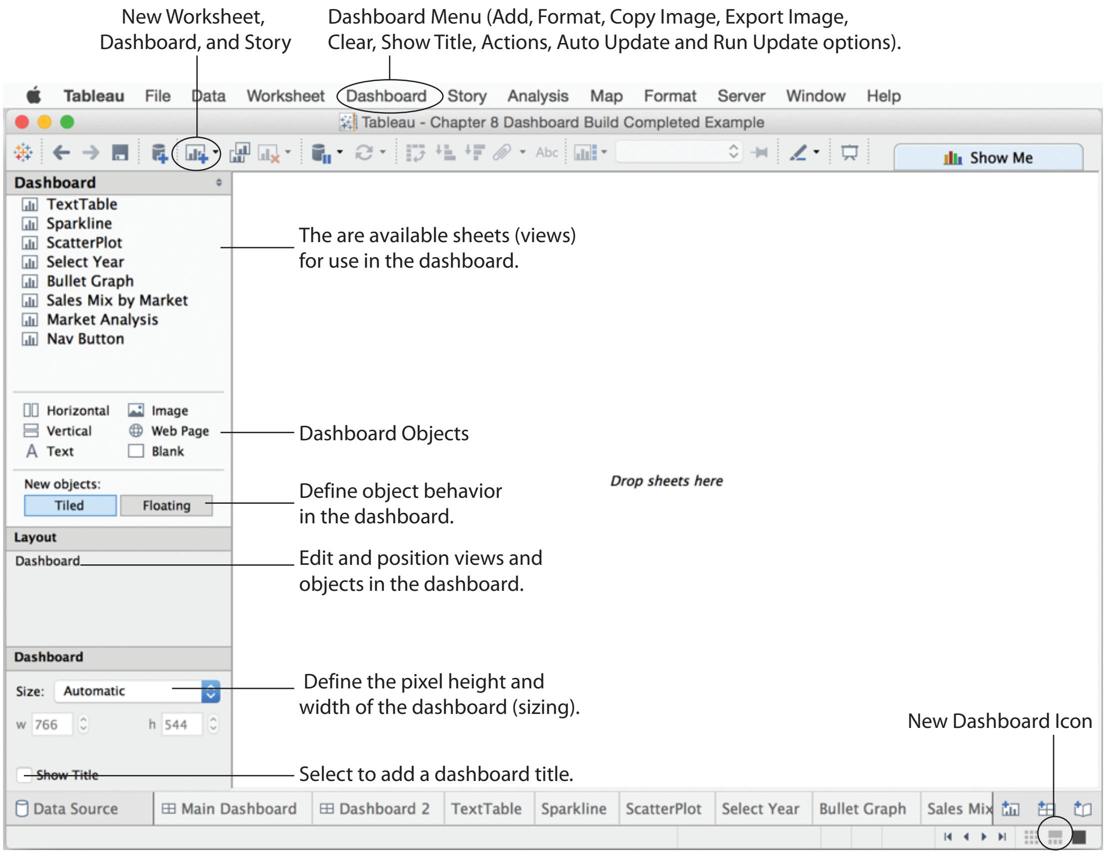


FIGURE 8-8 *Tableau's dashboard worksheet*

The example dashboard is suitable for a weekly or monthly recurring report. The specifications have been defined and are demanding. The example utilizes a variety of visualizations, dashboard objects, and actions. It will include a main dashboard and a secondary dashboard that will be linked together via filter actions.

Read through the rest of the chapter first to get an overview of the process. Then step through each section and build the dashboard yourself. When completed, your dashboard should look like Figure 8-9.

The dashboard follows the four-pane layout recommended earlier in the chapter, in the section "Best Practices for Dashboard Building," but is actually a five-pane design with the small Select Year crosstab acting as a filter via a filter action. The dashboard example also includes a second dashboard that you see in Figure 8-10.

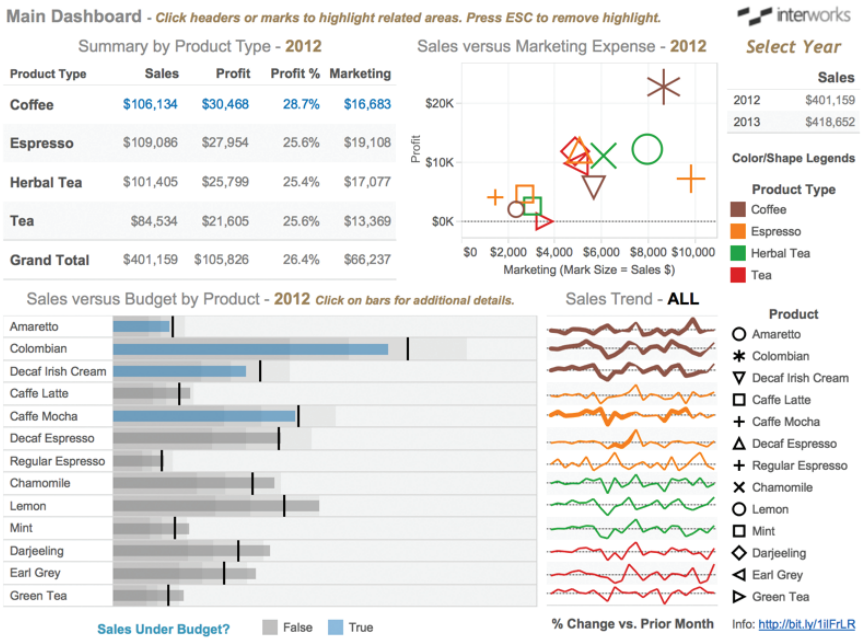


FIGURE 8-9 *Completed dashboard example*

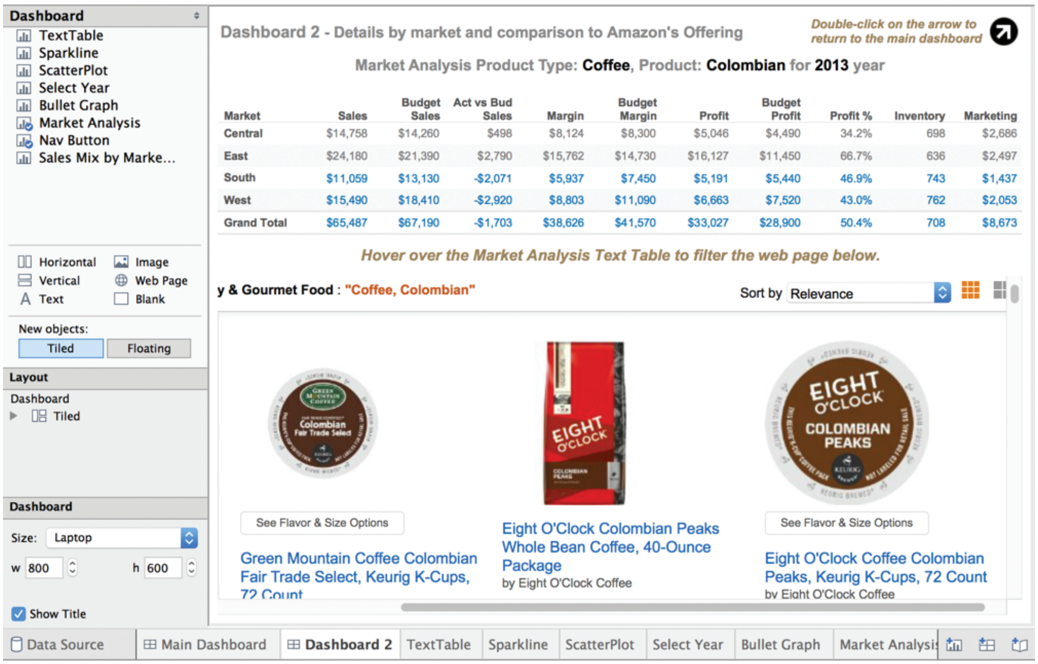


FIGURE 8-10 *Dashboard 2 example*

The dashboard in Figure 8-10 is accessed through a filter action from the Main Dashboard shown in Figure 8-9. This cascading style is another best practice that really helps to speed up load times when your data source contains very large datasets.

The Main Dashboard and Dashboard 2 contain many different views and objects. Figures 8-11 and 8-12 show exploded views of both dashboards.

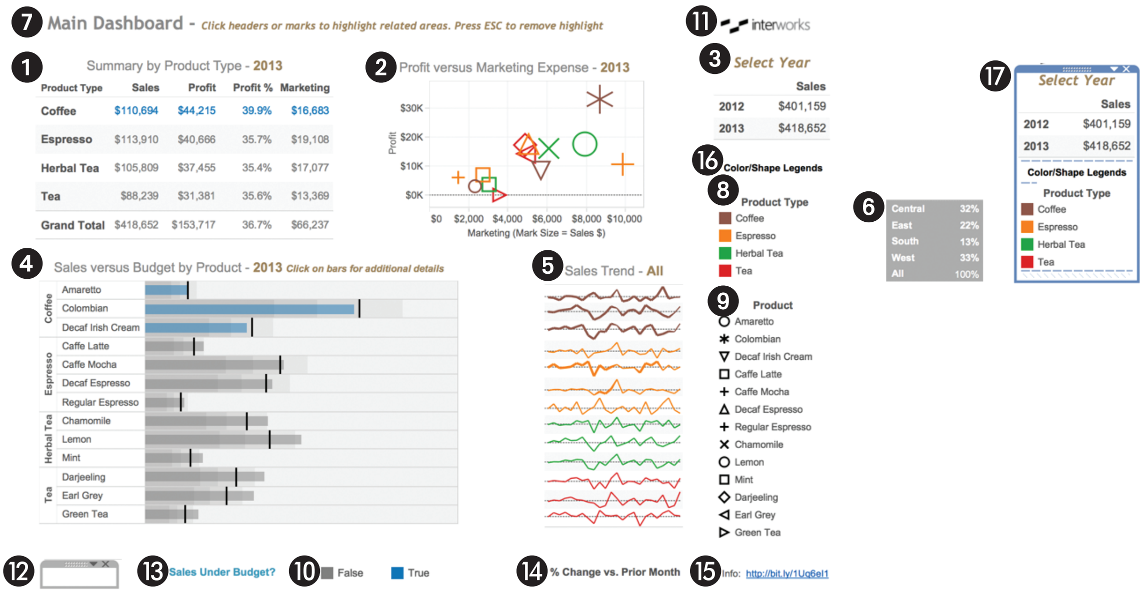
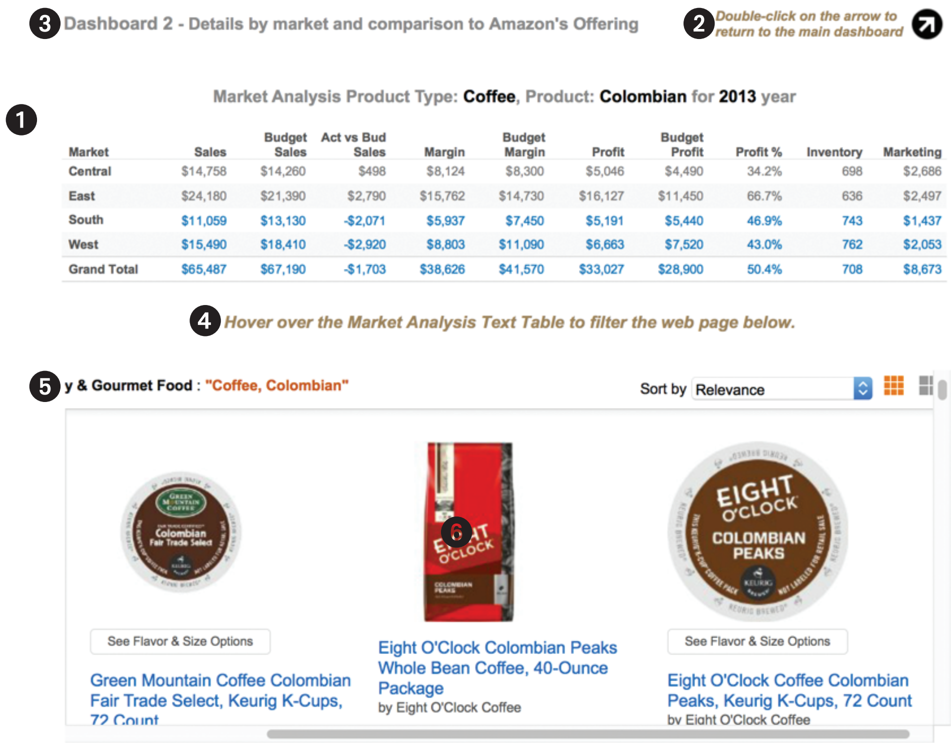


FIGURE 8-11 *Main Dashboard contents*

The Main Dashboard includes 17 different objects:

- **Items 1–6:** Worksheet panes (views)
- **Item 7:** Dashboard title (with descriptive text added)
- **Items 8–11:** Color and shape legends
- **Item 12:** Blank object (used for spacing)
- **Items 13–16:** Text objects (to annotate content and provide a weblink)
- **Item 17:** Vertical layout container (to control alignment of objects)



Market	Sales	Budget Sales	Act vs Bud Sales	Margin	Budget Margin	Profit	Budget Profit	Profit %	Inventory	Marketing
Central	$14,758	$14,260	$498	$8,124	$8,300	$5,046	$4,490	34.2%	698	$2,686
East	$24,180	$21,390	$2,790	$15,762	$14,730	$16,127	$11,450	66.7%	636	$2,497
South	$11,059	$13,130	-$2,071	$5,937	$7,450	$5,191	$5,440	46.9%	743	$1,437
West	$15,490	$18,410	-$2,920	$8,803	$11,090	$6,663	$7,520	43.0%	762	$2,053
Grand Total	$65,487	$67,190	-$1,703	$38,626	$41,570	$33,027	$28,900	50.4%	708	$8,673

FIGURE 8-12 *Dashboard 2 contents*

Dashboard 2 contains five different objects:

- **Items 1–2:** Worksheet panes (views)
- **Item 3:** Dashboard title (with descriptive text added)
- **Item 4:** A Text object (to provide instructions)
- **Item 5:** A URL object (containing a live website)

This example is designed to use many of Tableau's advanced dashboard features included in Tableau Desktop Version 9.2. The major steps required to complete this example are as follows:

1. Download the Chapter 8, "Dashboard Exercise," workbook from the book's companion website. Refer to Appendix F for additional details.
2. Define the dashboard size and position the dashboard objects in the dashboard workspace.

3. Enhance title elements, improve axis headers, and place image and text objects into the primary dashboard.
4. Create a secondary dashboard with a detailed text table, web page object, and navigation pane.
5. Add filter, highlight, and URL actions to the dashboards.
6. Finish the dashboard by enhancing the tooltips and testing all filtering and navigation. Add a Read Me dashboard to explain how the dashboard is intended to be used, data sources, and any calculations created that may not be obvious to the audience.

The exercise follows the best practice recommendations made at the beginning of this chapter in the context of using as many different advanced dashboard techniques as the sample data supports. You can build very functional dashboards without using many of these methods. The goal is to provide methods and alternative approaches that are not taught in the Tableau Public training courses.

When assembling worksheets in a dashboard, you should consider the available space that your audience has to view the dashboard. Will it be viewed on an old overhead projector with limited resolution and brightness? Is the dashboard going to be embedded in a company reporting website? Or will the audience consume the dashboard on a personal computer or a tablet computer? For this exercise, assume that the majority of people will be viewing the dashboard on laptop computers. A small number of people will view it on desktop computers. The easiest way to start a dashboard is to click the new dashboard tab. Figure 8-8, shown earlier in the chapter, highlights the new dashboard icon at the bottom of the workspace.

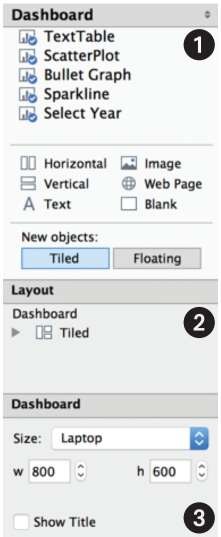


FIGURE 8-13 *Dashboard design shelves*

POSITION THE WORKSHEET OBJECTS IN THE DASHBOARD WORKSPACE

Placing worksheets into the dashboard workspace can be done by double-clicking the worksheet objects at the top of the dashboard shelf. Tableau will automatically place them into the view. Alternatively, drag the worksheet object into the view and place it in the exact position you desire. Tableau provides a light gray shading as you drag objects into the workspace, indicating the space that it will occupy when you release your mouse button.

Unless custom titles were added in the worksheets, the titles that are displayed in the dashboard for each worksheet reflect the worksheet tab names. A variety of dashboard objects can be accessed and placed into the dashboard workspace using the dashboard and layout objects displayed in Figure 8-13.

Dashboard area 1 includes worksheet views that are available in the workbook; objects for controlling the orientation of groups of objects, Horizontal and Vertical; and objects for adding text, images, live web pages, or blank space. By default, Tableau uses the New objects option Tiled to place objects in their own panes. Selecting the Floating option makes objects float over other objects already in the workspace. As you add worksheet objects to the dashboard, a small blue circle with a checkmark will appear on its icon. Refer to Figure 8-9 to see how the shelves appear in the finished dashboard.

Layout area 2 includes objects that have been added to the dashboard as well as layout options. Dashboard area 3 at the bottom allows you to define the sizing of the entire dashboard and for the individual objects included in the workspace. Before any worksheets are added into the workspace, define the dashboard size to accommodate the worst-case scenario that the dashboard will be viewed—(800 × 600) pixels. The Laptop option in the menu provides this exact size.

To view more options, click the size shelf, as shown in Figure 8-14, to see additional ways that size can be controlled.

- **Automatic:** Expands the dashboard to fill the available screen space
- **Exactly:** Allows you to lock the dashboard width and height
- **Range:** Enables the designer to define minimum and maximum limits

Exactly mode allows you to set the worst-case parameters for space. After completing your design, you may want to change the size mode to Range and define specific limits that the dashboard can expand to fill.

Automatic mode expands or contracts the dashboard to fill the available screen resolution of each computer viewing the dashboard. If any of your audience has a high-resolution graphics card, the dashboard might look out of place. The Range option allows you to define specific maximum limits so that dashboards designed for compact spaces don't look too sparse on large monitors. If someone is using a very low-resolution monitor to view the dashboard, minimum limits can be set for the dashboard pixel height and width. Once the dashboard size has been defined, you are now ready to add individual worksheet objects to the dashboard. Figure 8-13, displayed earlier, shows five different worksheet views that are available to add to the dashboard. There are two ways to add objects into the dashboard. Double-clicking a worksheet object causes

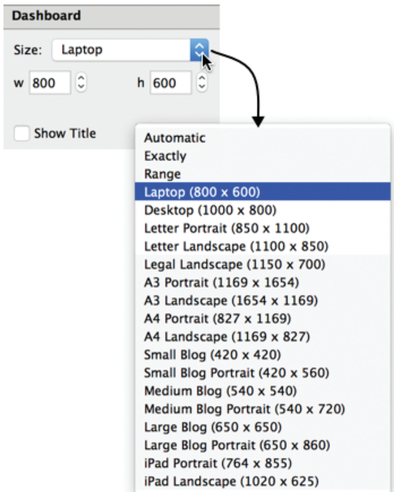


FIGURE 8-14 *Dashboard layout size definition*

Tableau to place that object into the workspace automatically. To control the placement of an individual object more precisely, drag the object into the view. As long as your left mouse button is depressed, Tableau will preview the area that the object will occupy by shading it in gray.

Double-clicking each worksheet object in the order that it appears in the dashboard shelf will result in the worksheet views being displayed in the dashboard, as you see in Figure 8-15.

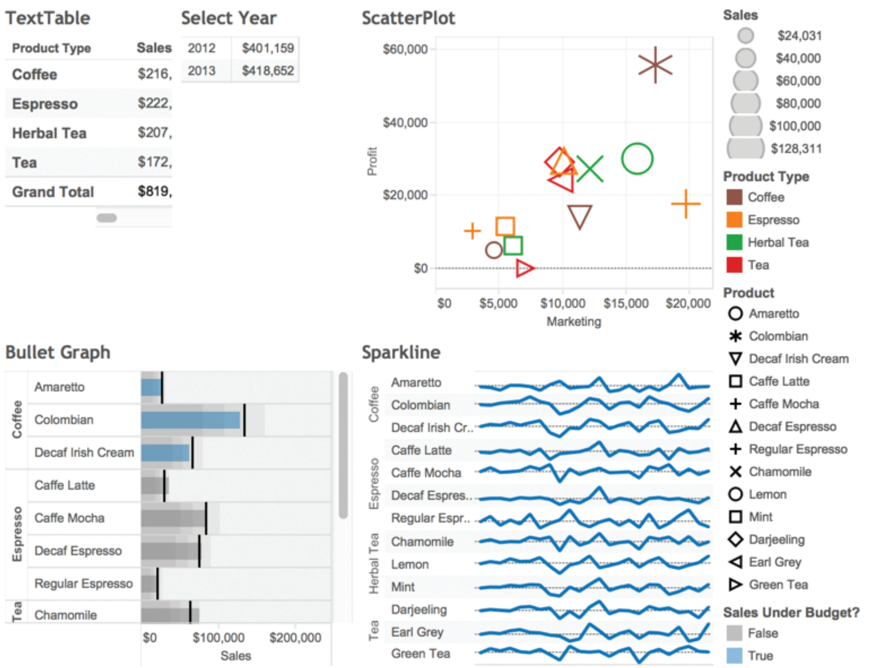


FIGURE 8-15 *Initial layout of the coffee chain dashboard*

Each worksheet has been added into the dashboard, but the placement of the individual views can be improved. Reposition the Select Year Text Table pane by clicking inside the Text Table pane to activate it. Then, using the handle at the top and center of the object, drag it into the upper-right area of the workspace above the sales size legend. Figure 8-16 shows the Select Year pane being activated and the cursor as it appears when you have located it over the handle for the pane.

Next, delete the Sales size legend. When these steps are completed, the dashboard pane should look like Figure 8-17.

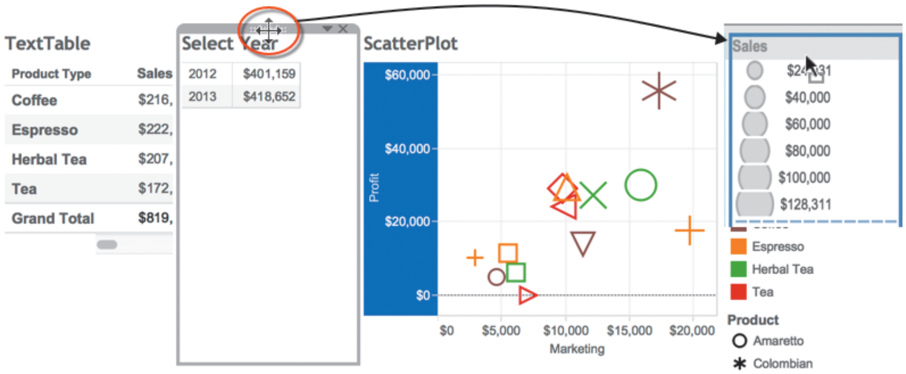


FIGURE 8-16 *Repositioning a sheet pane*

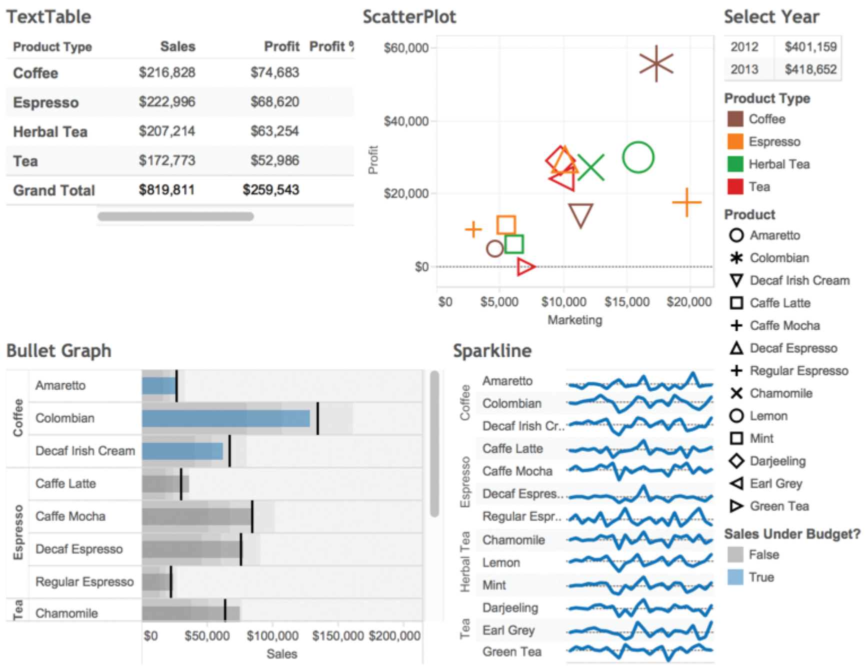


FIGURE 8-17 *Repositioned dashboard objects*

Add a title to your dashboard by selecting the Show Title option in the bottom left of your dashboard shelves. The default title will be the name of the dashboard worksheet that was created by Tableau. Edit the title text by double-clicking the default name, and type in **Main Dashboard**. Edit the title font to Trebuchet MS, 16-point, and select a light gray color. Make sure that the title is left-justified. After adding the title, it should appear as you see in Figure 8-18.

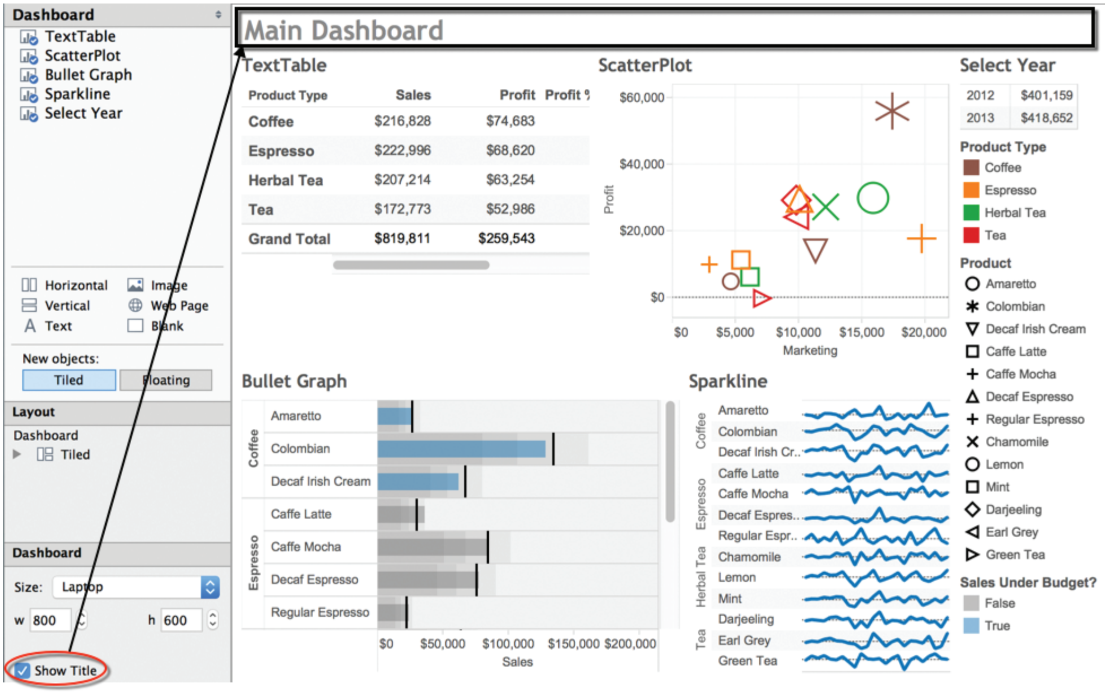


FIGURE 8-18 *Dashboard with title object added*

USING LAYOUT CONTAINERS TO POSITION OBJECTS

Layout containers allow you to group objects horizontally or vertically within the dashboard workspace.

Use a Horizontal Layout Container for the Dashboard Title

In Figure 8-19, the InterWorksLogo file is aligned horizontally to the right of the dashboard title. You can download this logo from the companion website or use your own logo.

The title and logo alignment in Figure 8-19 was achieved using the following steps:

1. Drag a horizontal layout container to the top of the dashboard.
2. Drag the title object into the horizontal container.
3. Place the logo image file into the right side of the layout container.

4. Adjust the height of the layout container.
5. Position the title and image within the layout container.
6. Associate a URL with the logo.

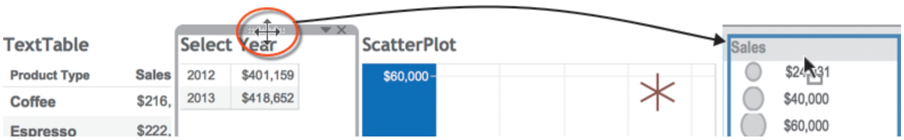


FIGURE 8-19 *Title and logo aligned*

Add the horizontal layout container to the dashboard by dragging the Horizontal object from the dashboard shelf to the area above the title bar, as you see in Figure 8-20.

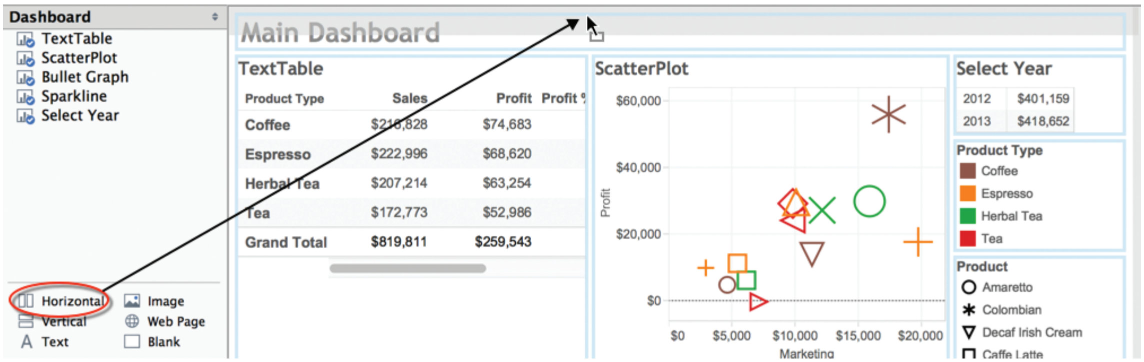


FIGURE 8-20 *Adding a horizontal layout container*

Before you let go of the object, be sure that the gray area highlights the full width of the dashboard at the top. This will ensure that the title object occupies the entire width at the top of the dashboard. After releasing the mouse button, don't worry if the vertical space occupied by the layout container is very large—you can reposition it by dragging up from the bottom edge of the layout container. Then drag the title object into the horizontal layout container.

Now that the title is placed inside the horizontal layout container, you can drag an Image object into the layout container in the dashboard, as you see in Figure 8-21.

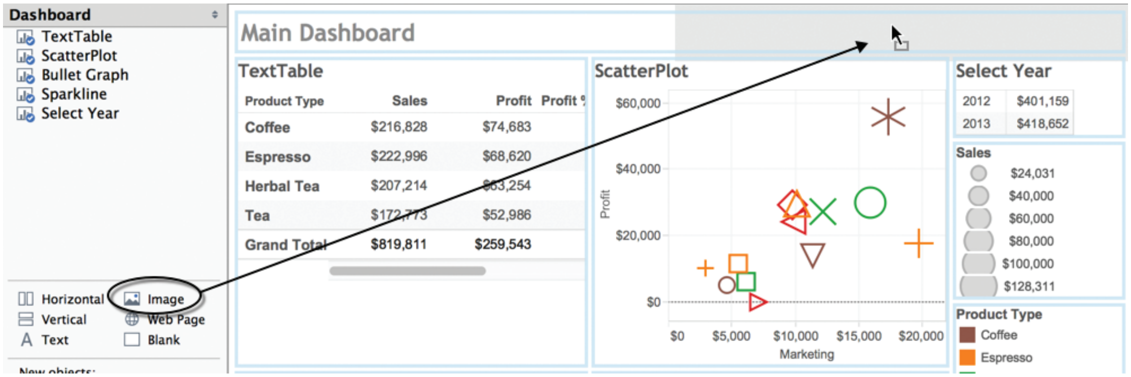


FIGURE 8-21 *Place an image object in the layout container*

After dropping the image object into the horizontal layout container, select an image file to add into the space. Use any image file you prefer for the logo. The example uses the InterWorksLogo file provided on the companion website.

Reposition the title and image objects within the layout container by clicking in the title object space. Then, point the mouse at the right edge of the title object until your pointer changes to a horizontal pointer. Drag the edge to the right to align the logo with the left edge of the vertical space occupied by the Year Filter crosstab object. Your logo should now be positioned directly over the vertical space on the right over the legends.

Make the title bar narrower by pointing at its bottom edge and dragging up. The logo probably isn't centered within the image object. You will notice that the logo image does not fit into the space well. To fit and center the logo in the Image object, click in the object to access the drop-down arrow and expose the object's menu, as you see in Figure 8-22.

Select Fit Image and Center Image. Your logo should now be resized to fit in the space.

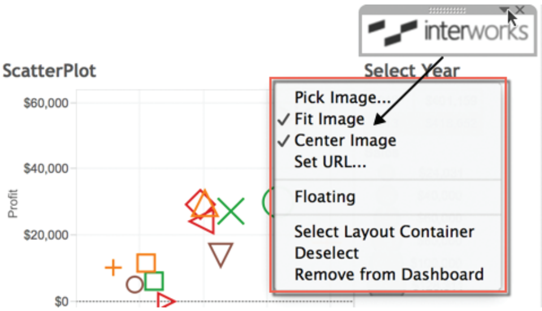


FIGURE 8-22 *Fit and center the logo*

To complete the title area, add the URL associated with the logo to the image pane. Set the website address by clicking the image pane to activate the menu; pick the Set URL option and type in the website address. Now, when the logo is clicked and web access is available, a browser session will open and the website will be displayed. Figure 8-23 shows the settings used to create the link for the InterWorks logo.

Now that the dashboard title is complete, turn your attention to the area on the right side of the dashboard containing the Select Year Filter text table along with the color, shape, and size legends.

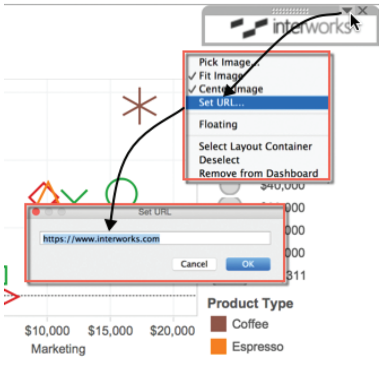


FIGURE 8-23 *Setting the logo image URL*

POSITIONING THE SELECT YEAR TEXT TABLE AND LEGENDS

Look at the completed dashboard again in Figure 8-9 and notice that the color legend for Sales Under Budget? has been repositioned below the bullet graph, the Sales size legend is gone, and a text box containing a website link has been added to the bottom. At this point, your legend area should look similar to Figure 8-24.

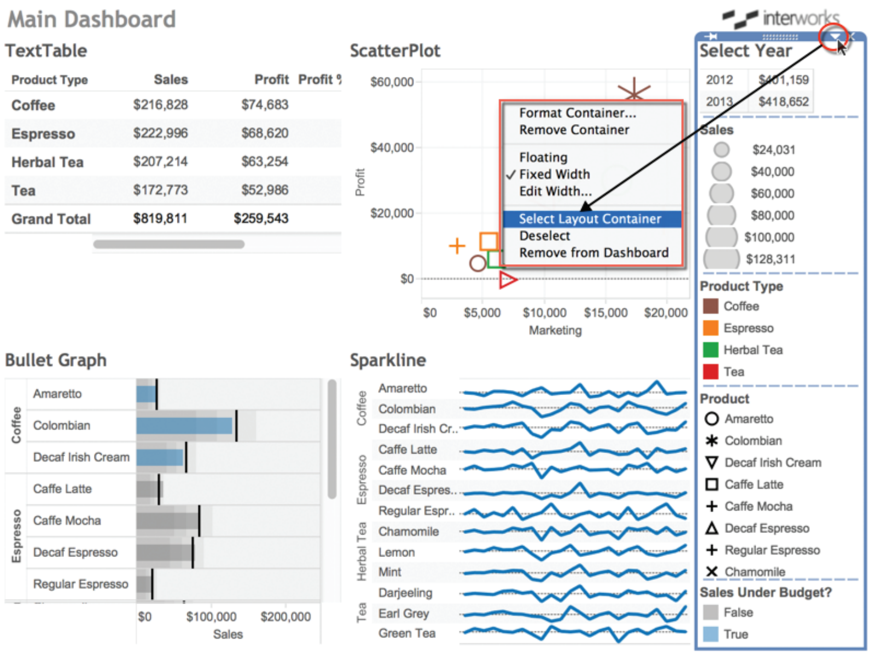


FIGURE 8-24 *Right vertical layout container*

Figure 8-24 shows the Vertical Layout Container and the menu you use to expose it. To highlight the layout container, press the down arrow and then the Select Layout Container menu option. This will cause Tableau to highlight the layout container. Tableau places a vertical layout container on the right side of dashboards that have legends or quick filters. Notice that all of the elements on the right side of the dashboard are in that space except the image pane containing the logo that was added to the horizontal layout container next to the title.

FIGURE 8-25 *New Text object*

INSERTING AND MOVING TEXT OBJECTS

Insert a text object below the Select Year text table and type in **Color/Shape Legends**. Format the text as Arial 9 point bold and center it. After completing that, you should have a text object inserted between the Select Year pane and the Product Type color legend, as you see in Figure 8-25.

Next, reposition the Sales Under Budget? legend by placing it below the Bullet Graph in the lower-left area of the workspace. From the legend drop-down menu, select Arrange Items > Single Row option. Don't worry about the title; it will be fixed in the next section. Figure 8-26 shows the relocated color legend for the bullet graph.

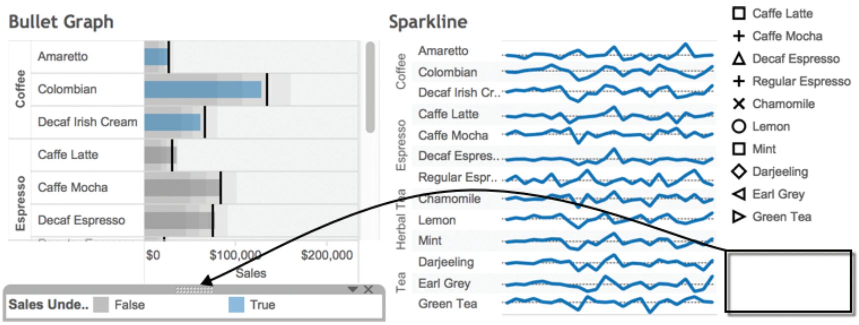


FIGURE 8-26 *Relocated color legend*

This color legend comes from a Boolean calculation placed on the Color Marks card in the Bullet Graph view that compares actual and budgeted sales. The bars in the bullet graph that are colored blue are below the Budget Sales field included in the data.

Use the additional space in the bottom right of the dashboard to drag another text object below the Product shape legend. Text objects can hold live web addresses if you include a valid web prefix. I used a web service to shorten the address of my personal website `http://bit.ly/1ilFrLR`. Adding a web address with the "http" or "https" prefix in a Text object is another way active website links can be placed into a dashboard. Precede the website address with `Info:`, use an 8-point font, and center the text. The new text object should look like Figure 8-27.

Open the layout container by clicking the Select Year area; then click the menu drop-down arrow to pick the Select Layout Container menu option. Go to the bottom of the container and drag the text object to the bottom of the layout container, as you see in Figure 8-27. Complete the layout container edits by making these changes:

- Edit the font of Select Year to 12-point, bold, italics.
- Center the Select Year title.
- Edit the Select Year data pane to fit the entire view.
- Center the remaining headings in the layout container.
- Reduce the horizontal space taken by the layout container.

When you complete these steps, your dashboard should look like Figure 8-28.

To edit the Select Year title font, double-click the title to expose the Edit Title dialog box, as you see in Figure 8-29, and make the defined changes.

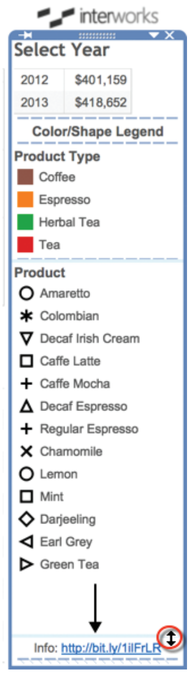


FIGURE 8-27 *Text object with live weblink*

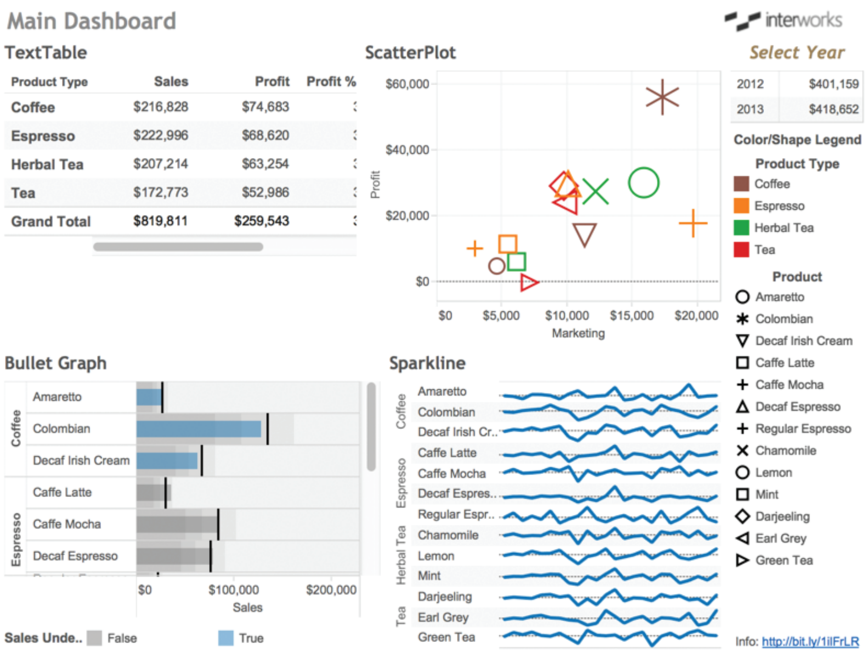


FIGURE 8-28 *Dashboard editing progress*

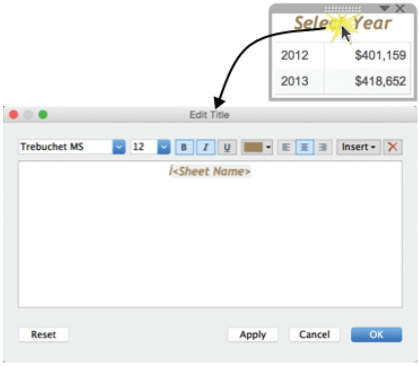


FIGURE 8-29 *Editing the Select Year title font*

Resize the fit of the Select Year text table in the upper right of the dashboard so that it fills the entire view. Access the Fit menu you see in Figure 8-30 to perform the editing.

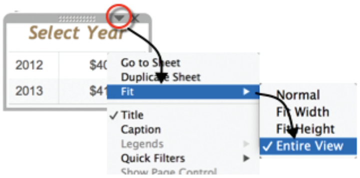


FIGURE 8-30 *Fitting the Select Year pane*

Finally, reduce the amount of horizontal space used by the legend area by dragging the left edge of the vertical layout container to the right. Be careful not to obscure any of the legend text. You are done with the layout container styling for now. If necessary, you can come back and make additional refinements later. Your dashboard should now look like Figure 8-31.

The dashboard is starting to take shape, but the data panes don't utilize the available space well. The title text used in the color legend below the Bullet Graph is partially obscured. In addition, the Bullet Graph and Sparkline objects are displaying identical row headers—creating redundant data ink that can be removed if you ensure that the rows are sorted the same way. In the next section, you'll learn how to deal with these issues so that the dashboard utilizes the available space more effectively.

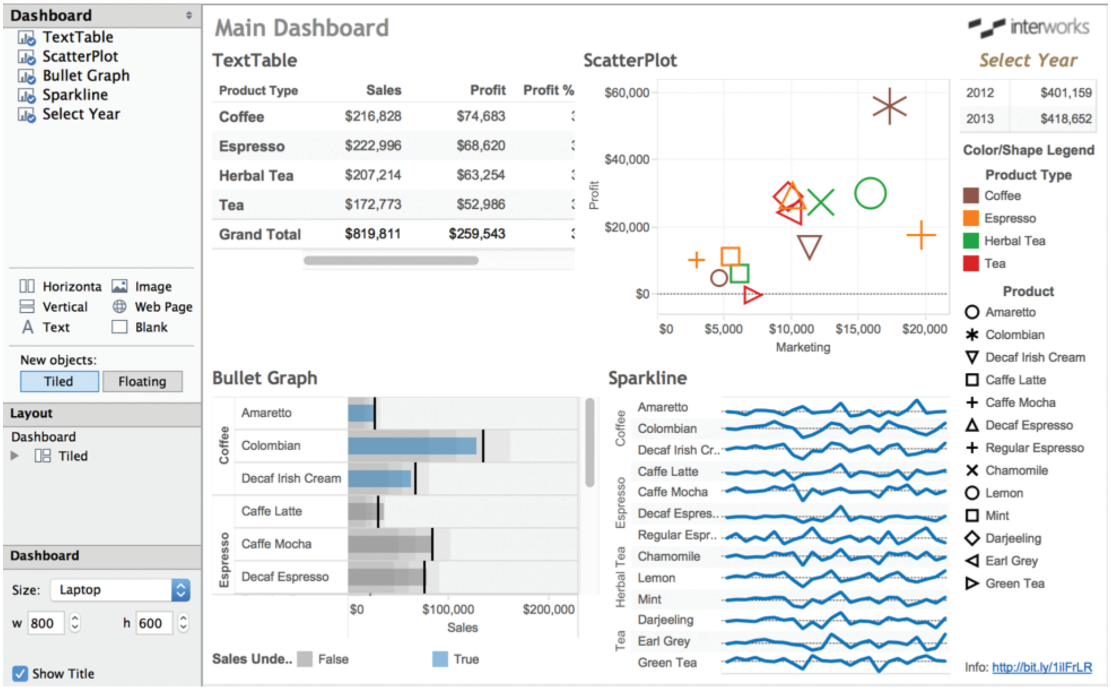


FIGURE 8-31 *Dashboard with improved legends*

POSITIONING AND FITTING THE DASHBOARD OBJECTS

The general layout of this dashboard is good. The upper-left quadrant contains a text table overview of performance. The table isn't fitting into the space, so that needs to be addressed. The ScatterPlot shows how promotional spending relates to profits and sales (although it isn't clear that the size of the marks in the scatter plot provide relative sales amounts). The Bullet Graph and Sparkline provide complementary views of actual sales performance. Color is used in different ways. Editing the use of color in the Sparkline and the Text Table could provide additional insight and understanding. In the ScatterPlot, color is used to distinguish Product Type. In the Bullet Graph, color indicates if a Product is below budgeted sales. To make this dashboard communicate the information more effectively, follow these steps:

- Ensure that each worksheet pane fits its entire view.
- Create more descriptive titles for each pane.
- Make the Product Shape Legend, Bullet Graph, and Sparkline sort in the same order.

- Hide the redundant row headers in the Sparkline chart.
- Reposition the worksheet objects to better utilize space.

ENSURE THAT EACH WORKSHEET OBJECT FITS ITS ENTIRE VIEW

Start by changing the fit within the Bullet Graph pane. The most straightforward method to access the fit menu is the one you used to fit the Select Year pane—clicking the title block of the pane and exposing the pane's menu. Alternatively, you can expose the same controls from the layout area by selecting the Bullet Graph and right-clicking. Figure 8-32 shows that you can access the same menu using either method.

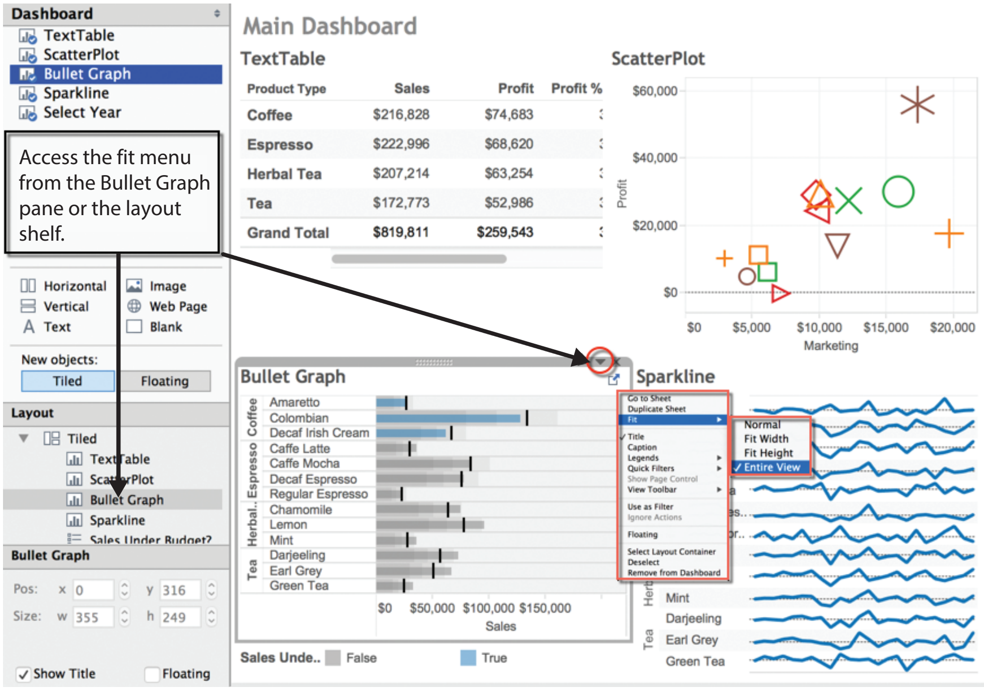


FIGURE 8-32 *Fitting the Bullet Graph*

Change the Bullet Graph fit from Normal to Entire View. You should see the graph fill the pane completely.

Notice when you click the Bullet Graph pane in the layout area, the context of the Position and Size changes to display the values for the Bullet Graph pane. You now see the pixel positions and size of that particular pane. Note that the

Show Title option is selected and the Floating option is not. If the Floating pane option were selected, this pane could be placed on top of other panes in the dashboard—floating over the area. This choice wouldn't be appropriate for the bullet graph. Later in the exercise, you'll utilize a floating pane. Repeat the same process for all of the data objects so that all of them fill the available space.

CREATE MORE DESCRIPTIVE TITLES FOR EACH DATA PANE

Adding more descriptive data object titles will make it easier for the audience to interpret the dashboard. Edit the titles by double-clicking each view's title and replacing the `<sheetname>` text with the following title text:

- **Bullet Graph:** Sales versus Budget by Product
- **Sparkline:** Sales Trend
- **Crosstab:** Summary by Product Type
- **ScatterPlot:** Sales versus Marketing Expense

Figure 8-33 shows the dashboard after completing the title editing.

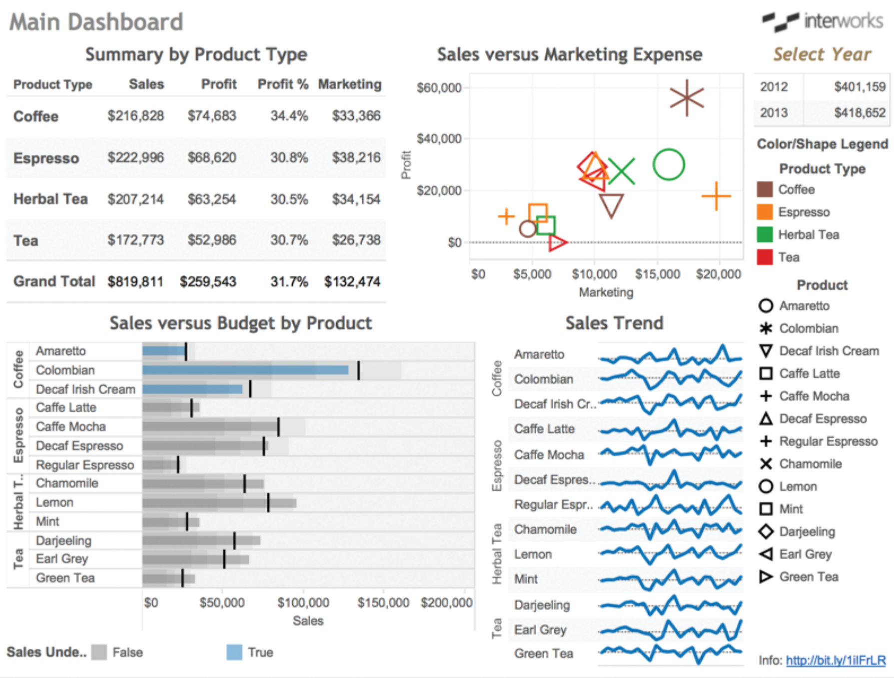


FIGURE 8-33 *Improved dashboard titles*

The size, color, and style of the title font used for the Select Year text table in the upper right serves a particular purpose. The brown color and italic font is used to indicate an instruction. Later in the chapter, you will add filter and highlight actions. The Select Year pane will be used to filter the rest of the dashboard using a Filter Action. Next, you'll learn about creative ways to use sorting, text objects, and mark labels to improve the legibility of the dashboard.

IMPROVING THE BULLET GRAPH AND SPARKLINE CHARTS

In Figure 8-32, you can see the Bullet Graph and Sparkline have the same row headings. The duplicate headers are an inefficient use of space. The charts are meant to be used together to see performance versus budget and trends over time, but they are not perfectly aligned. The title of the color legend below the Bullet Graph is partially obscured and needs to be edited so that it is legible. Apply these improvements with the following steps:

1. Make the row sort order in both charts and the Product legend identical.
2. Hide the row labels in the Sparkline.
3. Turn on mark labels and hide the header in the Bullet Graph.
4. Improve the color legend below the Bullet Graph.
5. Precisely align the Sparkline and Bullet Graph rows.

Make the Row Sort Order in Both Charts Identical

Hover your mouse over the bullet graph title. This will expose the Go to Sheet navigation control. Click on the small box with the arrow (see Figure 8-34) to jump to the Bullet Graph worksheet.

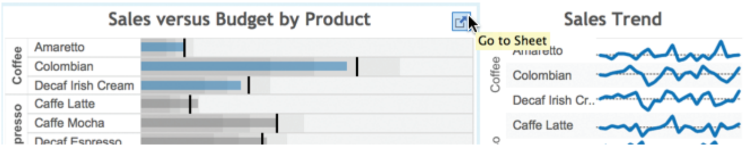


FIGURE 8-34 *Jump to the bullet graph worksheet.*

Edit the sort order of the Product Type and Product field pills on the Rows shelf so that the rows in both charts sort identically. Access the sort menu for each field by right-clicking the field pill on the Rows shelf, and then select the Sort menu/Manual sort option, as displayed in Figure 8-35.

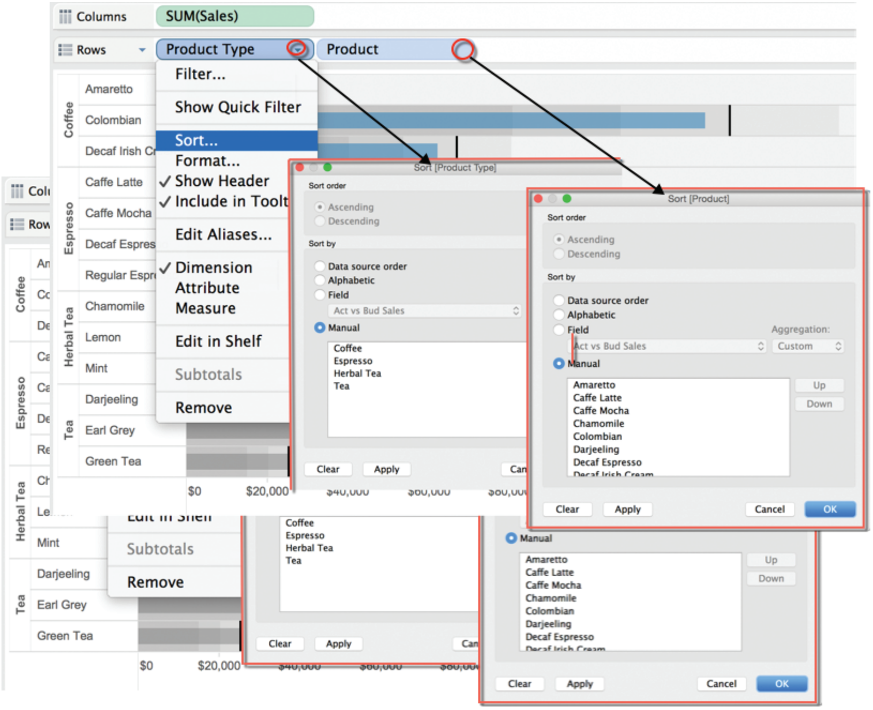


FIGURE 8-35 *Editing row sorting*

Repeat the same steps in the Sparkline worksheet. When this step is completed, Tableau provides a visual cue in the Product Type and Product field pills confirming that a sort has been applied to each field in both worksheets. The cue is a small bar chart that appears in the right side of each field pill. Now that the Bullet Graph and Sparkline are sorted the same way, you can hide the product type and product row labels in the Sparkline worksheet—saving space and eliminating redundant data ink. Right-clicking the Product Type and Product pills on the row shelf exposes the menu you see in Figure 8-36.

FIGURE 8-36 *Hiding the Sparkline row headings*

Hide the Sparkline row headings by selecting any row header, right-clicking, and clearing the Show Header option. Do this for both Product type and Product.

Although the Shape legend doesn't belong to the Sparkline, it might be useful later if the order of the products displayed in the shape legend matched the order of the Bullet Graph and Sparkline. Edit the sort order of the Product legend by clicking the legend, selecting Sort on the dashboard view menu (down pointing arrow), and doing a Manual sort as you did for sorting fields in the Bullet Graph and Sparkline. This change has already been added to the sample workbook. If you are building the example yourself, you will need to edit the shape legend to match the order of the graphs.

Turn on Mark Labels and Hide the Axis Header in the Bullet Graph

The Bullet Graph can be edited to provide more vertical space by hiding the axis header at the bottom of the chart. These axis labels provide valuable context. If the dashboard were going to be printed and consumed on paper, it would not be a good idea to remove the axis header.

When dashboards are consumed interactively on a computer, mark labels can be used to replace axis headers by presenting important details on demand—when a mark or heading is selected. Mark labels can always be displayed, but in this case space would be better utilized if the labels displayed only when the user wants to see them. To make the mark labels appear on demand, go to the bullet graph worksheet and click the Label button on the Marks card to expose the menu shown in Figure 8-37.

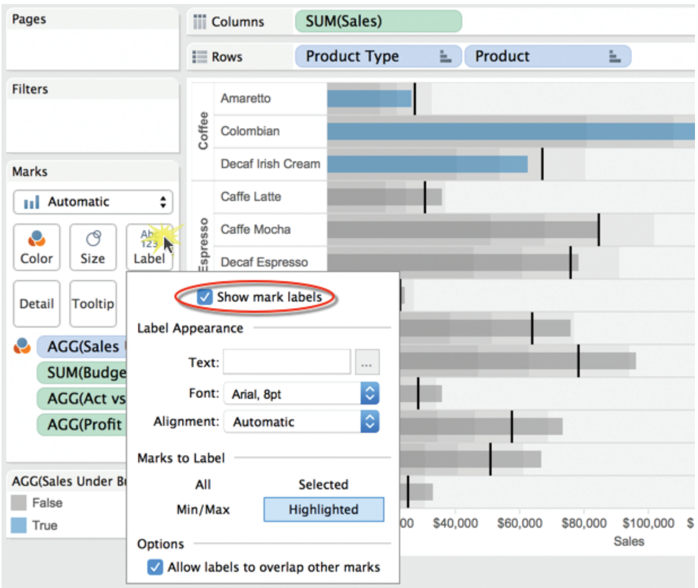


FIGURE 8-37 *Show mark labels*

The axis header at the bottom of the Bullet Graph can be hidden by pointing at the axis header area, right-clicking, and unchecking the Show Header option. The view on the left side of Figure 8-38 shows the menu selection, and the resulting appearance of the Bullet Graph is shown on the right.

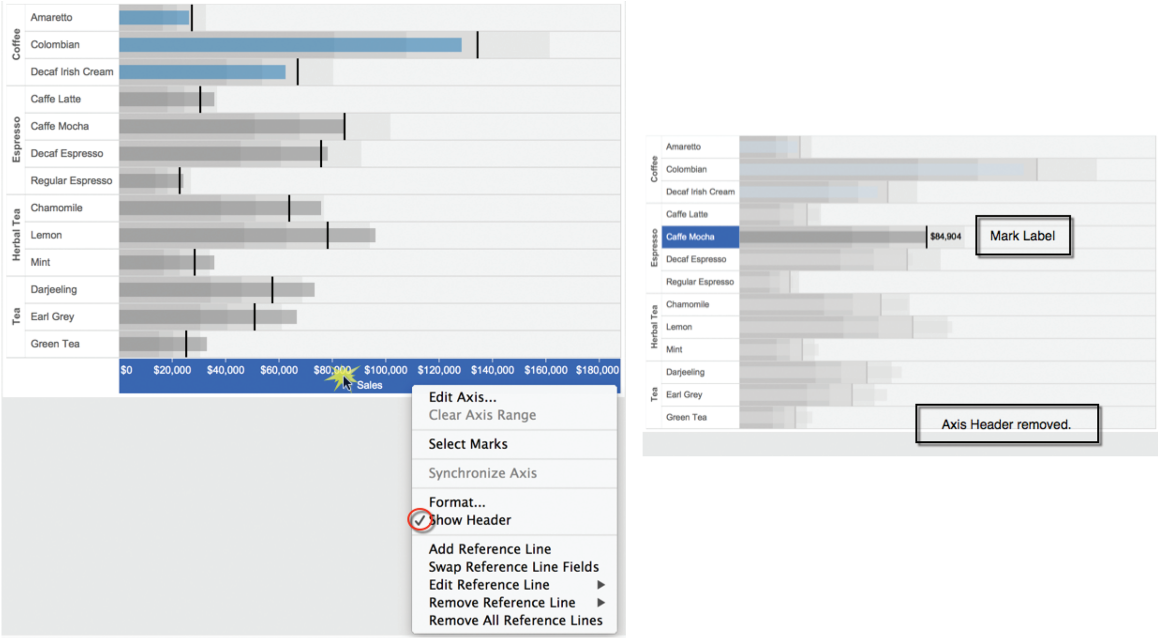


FIGURE 8-38 *Hiding the Bullet Graph heading*

Removing the axis heading in the Bullet Graph is a compromise that the mark labels enable by providing sales details via point-and-click selection of the mark or row header. Given the limited space requirements for this dashboard, this is an acceptable compromise.

Improve the Color Legend Below the Bullet Graph

One more item near the Bullet Graph needs to be addressed. In the view on the right side of Figure 8-33, you can see that the color legend title is partially obscured. This can be addressed by erasing the legend title and then adding a text object to the left of the legend. This technique allows for more precision in the alignment and positioning of the text to describe the legend colors. It also provides a means for centering the legend below the Bullet Graph.

Erase the legend title by accessing the legend menu via the drop-down arrow that appears when the legend is clicked; then clear the Show Title option. Follow these three steps to finish formatting the Bullet Graph:

1. Drag a Text object to the left of the color legend.
2. Enter **Sales Under Budget?** and apply the font and color you see in Figure 8-39.
3. Reposition the color legend by dragging the True color closer to the False color.

The height of the legend area may make posting the Text object difficult. To make it easier to place the text object in that space, click on the Bullet Graph pane and drag the edge of the pane up to provide more space. After placing the Text Object to the left of the color legend, you can drag the bottom edge of the Bullet Graph down to minimize the vertical height of the legend. Figure 8-39 shows the steps.

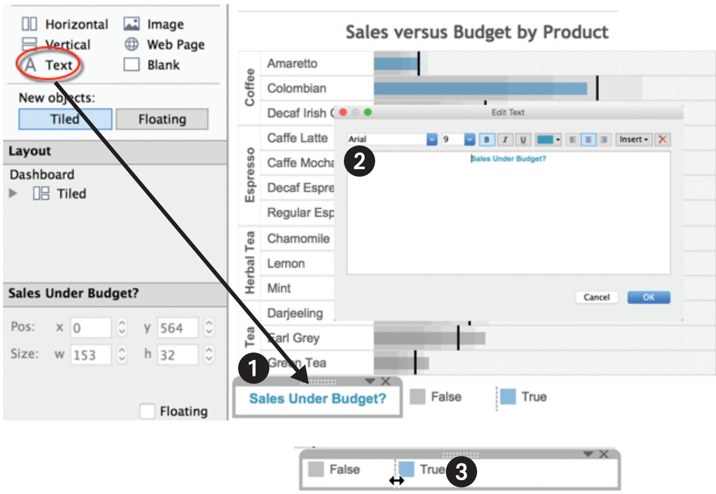


FIGURE 8-39 *Replace the legend title with a text object.*

These changes have reduced the amount of space required for the Bullet Graph.

Now turn your attention to the Sparkline. Because the row headings were hidden in the Sparkline, it's important that the Bullet Graph and Sparkline be sorted in the same way. While you did that sort earlier in the chapter, it would be good to provide additional visual cues regarding the Product Type represented by each mark in the Sparkline. You can use color to denote the

Product field in the Sparkline just like it was used in the scatter plot displayed in the upper right of the dashboard. It would also look better if the size of the line marks were reduced in the Sparkline. Figure 8-40 shows the Product Type field being added to Color in the Marks card.

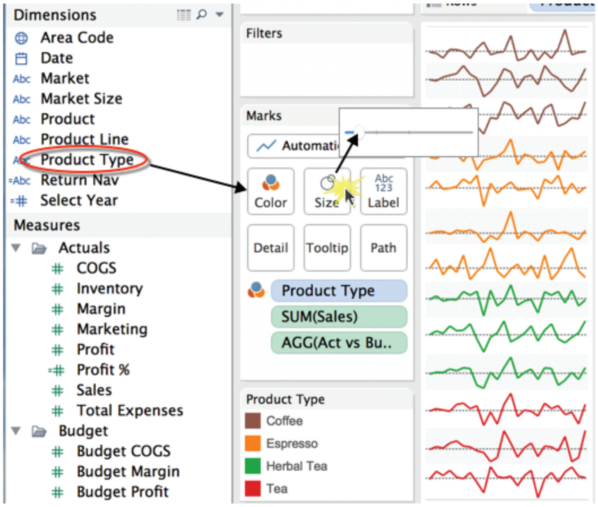


FIGURE 8-40 *Editing the Sparkline appearance*

Both of the edits are reflected in Figure 8-40. Placing the Product Type field on the Color button in the Marks card changes the line colors. Selecting the Size button and then dragging the slider to the left allows you to reduce the thickness of the lines.

Precisely Align the Scatterplot and Bullet Graph Rows

The Sales Trend Sparkline is not actually plotting the monthly sales value. It is showing the percent change versus the prior month to accentuate the pattern of monthly changes. Refer to Chapter 7 for a full explanation of the reasons why presenting the data in this way may be helpful. This information should be communicated to the audience.

As you can see in Figure 8-41, the color legend below the Bullet Graph is also causing a misalignment of the charts.

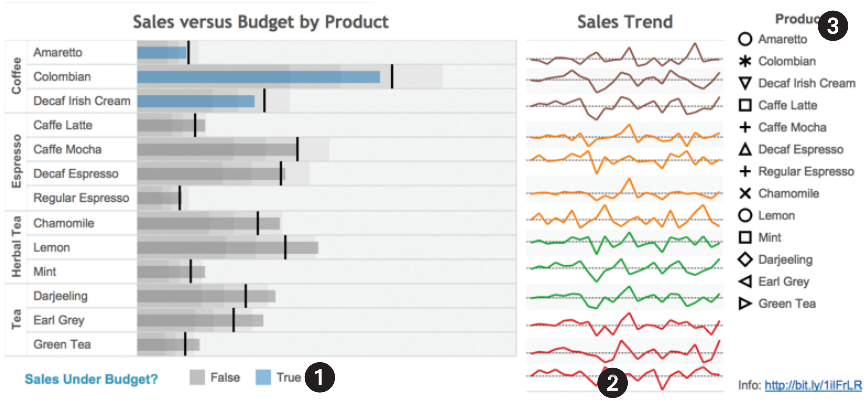


FIGURE 8-41 *Misaligned charts*

Formatting in this part of the dashboard can be improved in the following ways:

- Center the Sales Under Budget? Color legend under the Bullet Graph.
- Add a Text object under the Sales Trend Sparkline to align the rows with the Bullet Graph and provide additional information about what is being plotted in the graph.
- Align the Product Shape Legend with the Sparkline and Bullet Graph.

Center the color legend under the Bullet Graph by placing a Blank object to the left of the Sales Under Budget Text object. If you run into any trouble getting the Text object in that space, temporarily reposition the Bullet Graph to allow more space to place the object, and then drag the bottom edge of the Bullet Graph back down to minimize the vertical space taken by the legend, as you did once before.

Now add the Text object to the bottom of the Sparkline, type **% Change vs. Prior Month**, and center the text. Drag the bottom edge of the Sparkline down to align the last row in the Sparkline with the last row in the Bullet Graph.

Finally, position the Product Shape legend with the top and the bottom of the Bullet Graph and Sparkline rows. You will probably have to use a combination of dragging in the Vertical layout container on the right side of the dashboard and repositioning the main dashboard data panes.

Figure 8-42 shows what your dashboard should look like after you have completed these steps.

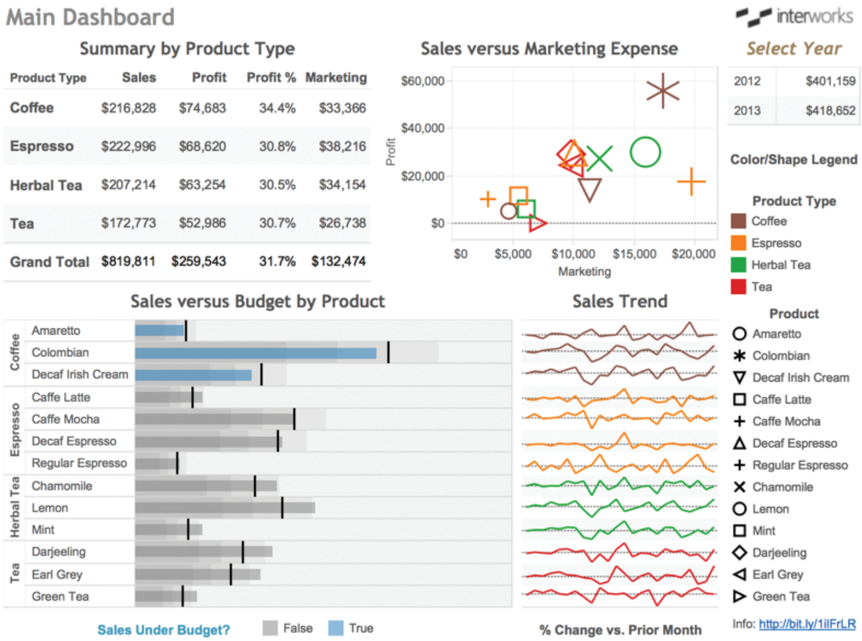


FIGURE 8-42 *The updated dashboard*

The bottom half of the dashboard looks very good now. The color legend title is centered, and the legend title is clear and legible. The rows in the Bullet Graph, Sparkline, and Product shape legend are aligned so users can see product Actual Sales compared to the Budget Sales and the related sales trend.

Although this wasn't discussed in this chapter, the color applied in the Bullet Graph to indicate sales that are above and below Budget Sales is achieved by creating a new field using a calculation. The field Sales Under Budget? is part of an example created in Chapter 7. To see the calculation, go to the Bullet Graph worksheet. Figure 8-43 shows the calculation.

In the next section, you will focus on the top half of the dashboard. This Boolean calculation will be utilized there as well.

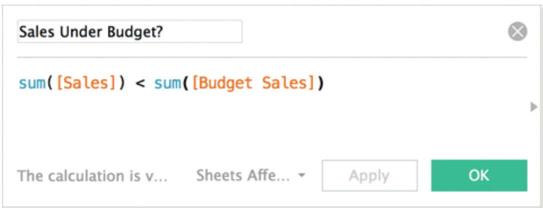


FIGURE 8-43 *Boolean calculation*

IMPROVING THE TEXT TABLES AND SCATTER PLOT

The dashboard is beginning to look finished, but there is some refinement that can be done to improve the two text tables and the scatter plot. Adding the Boolean calculation applied to the Bullet Graph (refer to Figure 8-43) is a good way to highlight sales performance problems in the Summary by Product Type and Select Year tables. Add the Sales Under Budget? calculated value to the Color button on the Marks card in each of the worksheets. After you do that, you should see the Coffee row in the Summary by Product Type turn blue.

Because the size legend for the Sales vs. Marketing Expense scatter plot was removed earlier to conserve space, it would be useful to find a way to let information consumers know that the size of the marks in the scatter plot denotes the relative sales value of each mark. These issues will be addressed with following steps:

1. Adjust the horizontal space used by the row headings in the crosstab chart and reposition the horizontal space allocated to each chart.
2. Change the format of the values displayed on the vertical axis of the scatter plot to thousands.
3. Edit the scatter plot's horizontal axis label to provide information related to the size legend that was removed earlier.

The two circles marked 1 in Figure 8-44 show you where to point your mouse in order to reposition the header in the Summary by Product Type to reduce the vertical space required for the headings and for moving the right edge of the Summary by Product Type chart to the right to provide more horizontal room for the Grand Total row label in the chart.

To reposition the title space and the size of the pane, point at the area, depress the left mouse button, and drag the edge to the desired position.

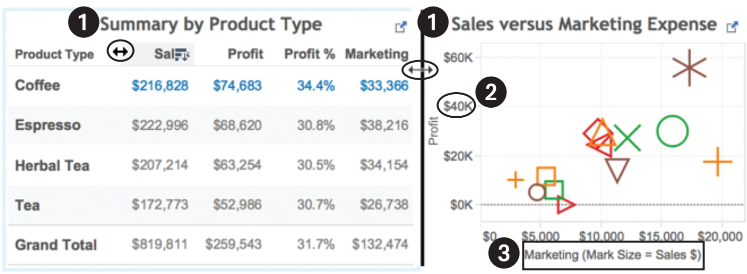


FIGURE 8-44 *Editing the top half of the dashboard*

Changing the format of the axis labels marked area 2 in Figure 8-44 is done by pointing at one of the axis labels and right-clicking to expose the formatting menu. Figure 8-45 shows that menu. It's important to select the `SUM(Profit)` measure when you apply the formatting, as you see in Figure 8-45.

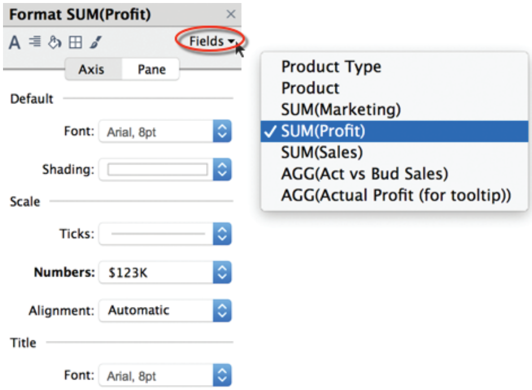


FIGURE 8-45 *Formatting the axis labels*

The axis title at the bottom of the scatter plot can be edited by pointing at the axis title, right-clicking, and then selecting the Edit Axis menu option. This will expose the menu you see in Figure 8-46.

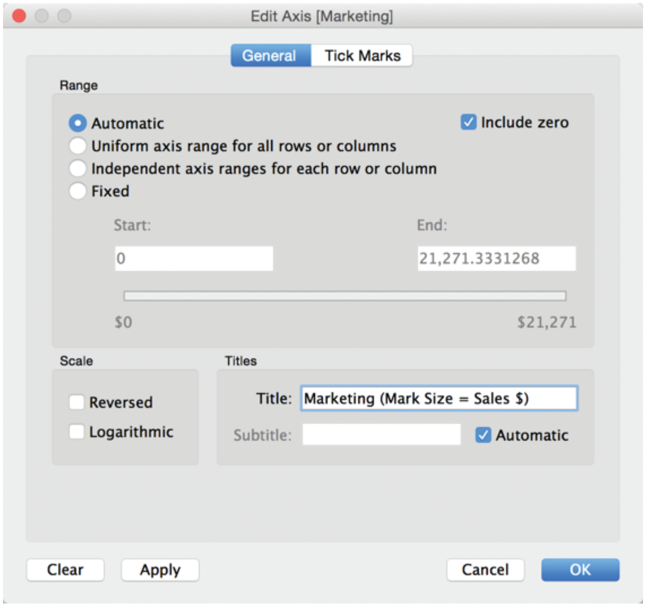


FIGURE 8-46 *Editing the scatter plot axis title*

Add the text **Mark Size = Sales $** to the Title dialog box in the Edit Axis menu, as you see in Figure 8-46. Clicking OK locks in the change. After completing these steps, the dashboard will look like Figure 8-47.

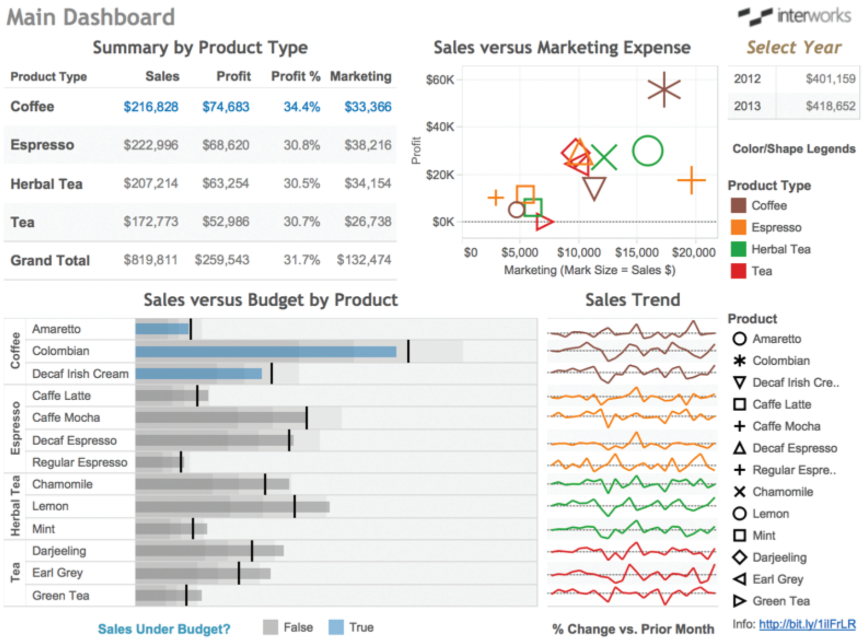


FIGURE 8-47 *Dashboard appearance after changes*

The dashboard looks finished now. In the next section, you learn how to use the data in the main dashboard to create actions for filtering and highlighting related information.

USING ACTIONS TO CREATE ADVANCED DASHBOARD NAVIGATION

Tableau's quick filters provide an easy method for filtering dashboards and worksheets. Refer to Tableau's manual for details on quick filters. This dashboard example purposefully avoids using quick filters because Tableau actions provide even more flexibility and in many instances also provide better initial load speed than quick filters—consistent with the best practices recommended earlier in the chapter.

Actions facilitate discovery by altering the context of the dashboard based on selections made by the audience. In this section, you create actions that utilize all of the available ways Tableau can invoke actions. You'll build actions that:

- Filter and highlight the main dashboard

- Facilitate navigation to a supporting dashboard
- Filter a detailed text table in the new supporting dashboard
- Call and filter an embedded website in the supporting dashboard
- Return the audience to the main dashboard from the supporting dashboard

USING THE SELECT YEAR TEXT TABLE TO FILTER THE MAIN DASHBOARD

The Select Year text table in the upper-right side of Figure 8-47 is titled using a different font style and color to make it stand out and provide a brief instruction identifying that the text table serves as a filter. Building filter actions in Tableau can be done in as few as three clicks. Create a filter action using the Select Year text table with these steps:

1. Click in the Select Year text table to select the view.
2. Select the drop-down arrow to expose the menu.
3. Pick the Use as Filter menu option to create the filter action.
4. Edit the filter action (Dashboard ➢ Actions) so that the Sales Trend Sparkline isn't filtered.

Figure 8-48 shows the menus related to steps 1 through 3.

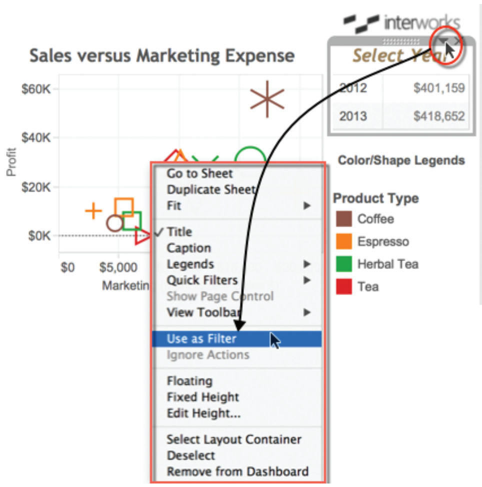


FIGURE 8-48 *Making a filter action*

After finishing step 3, you should be able to click one of the years in the text table, and every chart in the dashboard will be filtered to show only the selected year.

Creating filter actions this way is very easy, but it would be better if the Sales Trend Sparkline always showed both years. Tableau generated a filter action from the Use as Filter option. If you don't want the filter action to apply to the Sales Trend Sparkline graph, you must edit the generated filter action.

To edit the filter action generated by Tableau, access the Dashboard menu option, and then select the Actions menu to expose the Actions dialog box. You can see the unedited filter action on the left side of Figure 8-49.

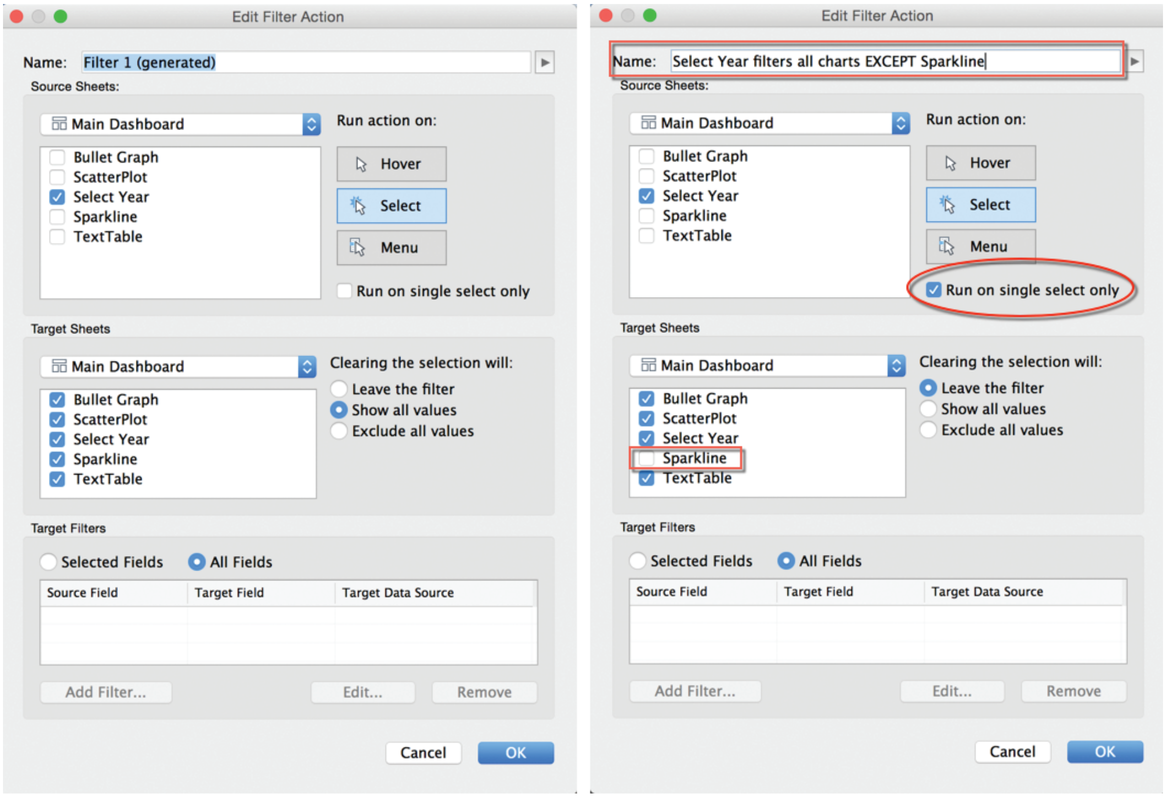


FIGURE 8-49 *Editing the filter action*

The Edit Filter action that Tableau generated applies to all of the Target Sheets in Dashboard 1 (now renamed to Main Dashboard). The Edit Filter Action dialog box on the right side of Figure 8-49 shows the changes applied in that screenshot.

- **Recommended:** Give the action a more descriptive name.
- **Required:** Uncheck the Sparkline in the Target Sheets area.
- **Required:** Select the Leave the Filter clearing option.
- **Recommended:** Check the Run on single select only option.

Naming the filter action with a descriptive title makes it easier to identify the exact purpose of the action. This is useful in two ways. First, if you need to come back months later to edit the action, a specific name makes it much easier to locate and understand its function. Second, if Run Action On uses the Menu option, the Name field will appear in tooltips or when users click on the heading. The name of the menu action can then be used to invoke the action. The importance of this will become clearer as we continue through this example. You will see this type of action later in the chapter.

Unchecking the Sparkline in the Target Sheets area means that the Sparkline will no longer be filtered by year. Because the Sparkline requires very little space to clearly display two years of data, it makes sense to leave that chart unfiltered.

The Leave the Filter option causes the filter action to remain in place when the action is removed. For example, if the year 2013 is selected (and then the filter action is removed by pressing the Escape key [ESC] or by clicking white space within a chart), the chart objects affected by the filter action will continue to display only the year 2013—until another selection is made in the Select Year text table.

Finally, checking the Run on single selection option means that only one year (and not both years) will be displayed when the filter is invoked.

ADDING A COLUMN HEADING TO SELECT YEAR

The Select Year text table needs a column heading that identifies the reported values as actual sales amounts for each year. We covered how to add a column heading to a single-column text table in Chapter 7. Recall that Tableau does not supply a column heading if only one field is being displayed. Go to the Select Year sheet and add the Profit % field to the view and then hide that field. If you don't remember how to do that, refer to the example in Chapter 7. When this is done, the Select Year pane will include a heading that describes the values, as you see in Figure 8-50.

Select Year

	Sales
2012	$401,159
2013	$418,652

FIGURE 8-50 *New column heading*

Adding the field name to the heading provides a necessary detail for your audience.

ADDING DYNAMIC TITLE CONTENT

The dashboard now includes a filter action that gives users the ability to filter for the year 2012 or 2013. Another helpful visual cue for your audience would be to include the year in the titles for each of the charts. Dynamic titles are a good way to provide visual confirmation that the objects in the dashboard have been filtered correctly. Figure 8-51 shows the full dashboard filtered for the year 2013 with dynamic titles that include the year.

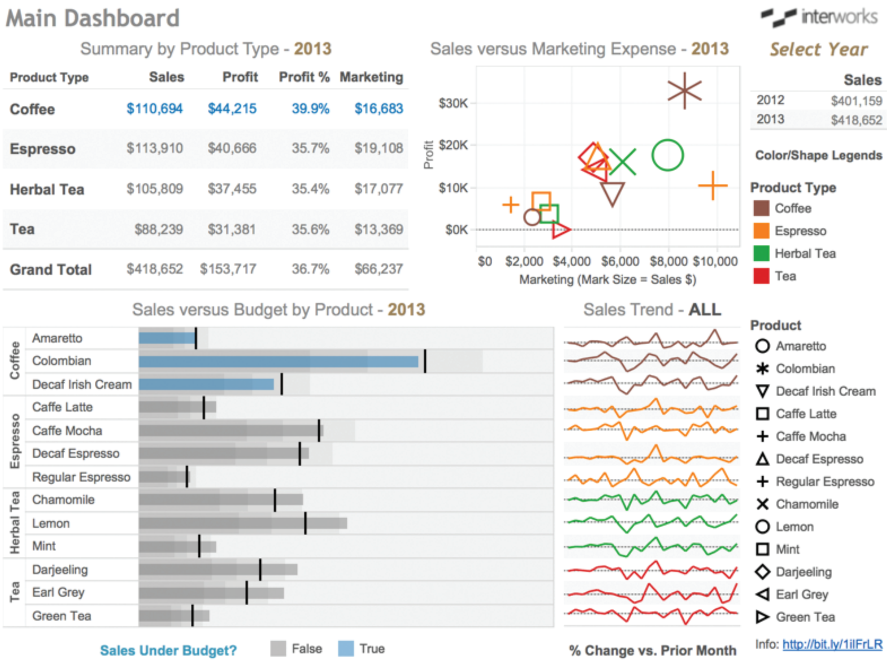


FIGURE 8-51 *Dashboard with dynamic titles*

You can see that the text table, scatter plot, and bullet graph are filtered for the year 2013 while the sparkline continues to display both years. Matching the font of the dynamic title elements to the Select Year text table title color provides a visual link for the audience. Add changeable title elements by inserting fields from your data into the title. Edit the title by double-clicking the title; then select the Insert menu option to position the Select Year field into the title, as you see in Figure 8-52.

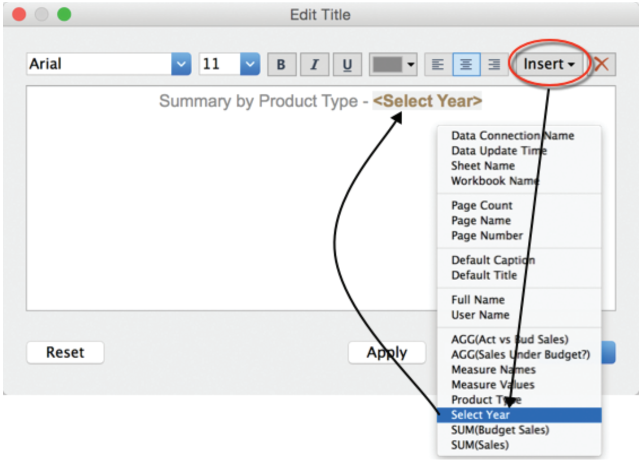


FIGURE 8-52 *Adding a dynamic field to chart titles*

Alternatively, you can directly type the field name in as long as you include the wrappers (`<Select Year>`) before and after the field name. Note that the `<Select Year>` added in the title is not the name of the text table. It is a similarly named custom date field that was added to the source data. Refer to Chapter 3 for more details about adding dynamic titles.

A custom date is being used in the dashboard to override Tableau's default behavior, which gives users the ability to display Tableau's default date hierarchy (year, quarter, month, and so on). Because of the space limitations imposed for this dashboard, it is desirable to limit the display of dates in the Select Year text table to show only the year. Refer to Chapter 7 for the details on how to create a custom date.

In the next section, you learn how to use the color and size legends to create highlight actions.

AUTO-GENERATING HIGHLIGHT ACTIONS FROM LEGENDS

Highlighting helps users see related information in dashboards more easily. Users can generate highlighting from legends by activating the highlighting tool that appears when you point at a legend, as shown in Figure 8-53.

Highlighting this way is effective for looking at one dimension at a time. In Figure 8-53, the Product Type Herbal Tea is being highlighted in the dashboard. This highlighting can be invoked from within any of the data views or from within the color and shape legends.

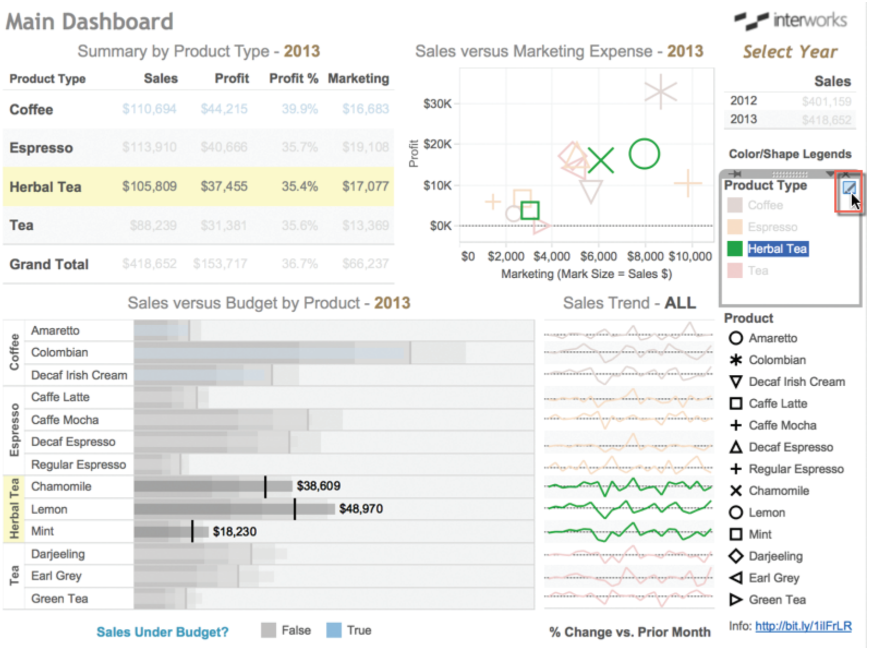


FIGURE 8-53 *Highlighting from a legend*

When these selections are initiated from a legend for the first time, Tableau creates a Highlight Action automatically. By activating highlighting from both the Product Type and Product legends, Tableau generates a Highlight Action that is the combination of both the color and shape legends. To see the action that Tableau creates, access the Dashboard menu and Actions submenu just as you did for the filter action. Figure 8-54 shows the Actions dialog box and the Edit Highlight Action dialog box, which is accessed via the Edit button.

When Tableau creates these highlight actions, it will automatically apply them to every worksheet object in the dashboard. In this case, uncheck the Select Year text table in the Target Sheets area so that the highlight action isn't applied to that object. In the Target Highlighting area of the Edit Highlight Action dialog box, I edited that to include the Selected Fields option and both the Product (color legend) and Product Type (shape legend) fields.

This action will be available for anyone consuming the dashboard. Highlighting will now occur if marks or headings are selected within the dashboard using either of the fields, as you can see in Figure 8-55.

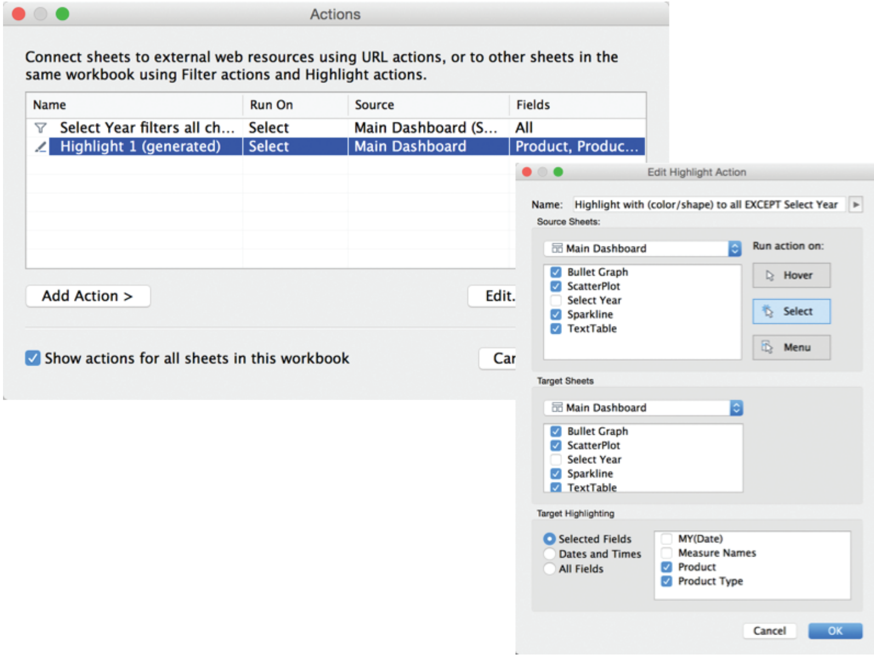


FIGURE 8-54 *Editing the generated Highlight action*

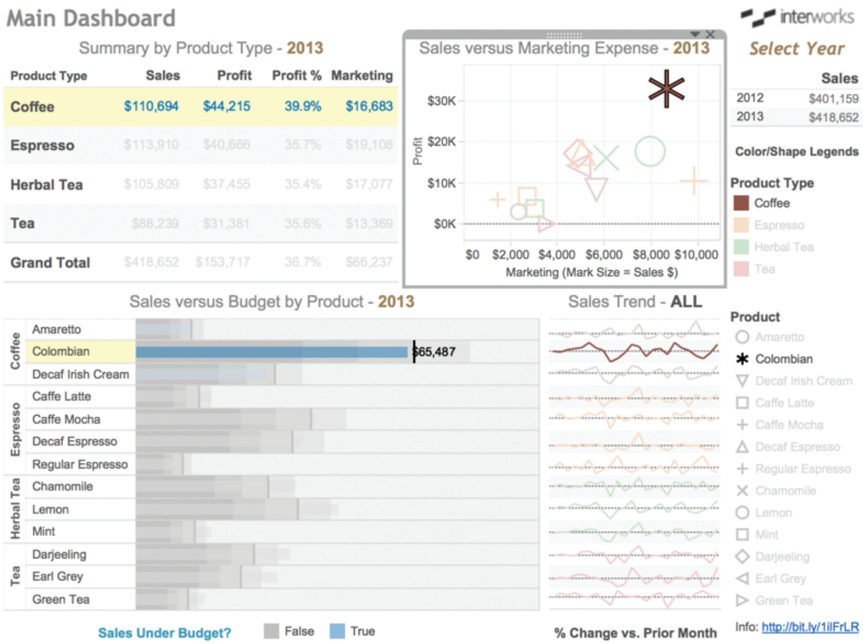


FIGURE 8-55 *Highlight from a mark*

The highlight in Figure 8-55 was triggered by selecting a mark in the scatter plot. The highlight action uses the combination of color and shape to highlight Product Type and/or Product in all of the other charts with the exception of the Select Year text table that was removed from the action. Try clicking marks in different parts of the dashboard to see how the highlighting changes.

Because Tableau has built simple controls for triggering actions, it is possible to build them even if you don't understand exactly how they work. If you know how to edit Actions, you can advance your knowledge through experimentation. If you want to understand more about the available options for Actions, details about the Action dialog box are covered next.

UNDERSTANDING THE ACTION DIALOG BOX

Actions can be applied to one or more dashboards or worksheets. This capability enables the creation of elegant cascading dashboard designs by using filter actions to link the contents of one dashboard to another related dashboard.

The steps to define filter actions and highlight actions are similar, but there are differences in how the data needs to be expressed for highlighting to work properly. For example, highlighting requires exact field names that are visually differentiated in each view, whereas filters don't have this requirement. Figure 8-56 shows the Filter and Highlight Action dialog boxes that include the definitions for the two Actions created so far in the example.

The Filter/Highlight action screens are each comprised four main areas:

- **Name:** Defines the name of the action as it appears in the dialog box and tooltips or Header menus if the action is run using a menu
- **Source Sheets:** Controls where and how actions are invoked
- **Target Sheets:** Defines where actions are applied and their behavior when cleared (only for filter actions)
- **Target Filters/Highlighting:** Limits what fields are used to apply the actions

Name

Tableau automatically assigns action names sequentially by type. While this naming convention keeps things organized during the design process, it isn't helpful if you need to revise your design later.

This exact name text is also used for triggering the action when the "Run action on – Menu" option is chosen. The name text appears inside the tooltip or when the cursor is pointed at the related row heading in the source sheet.

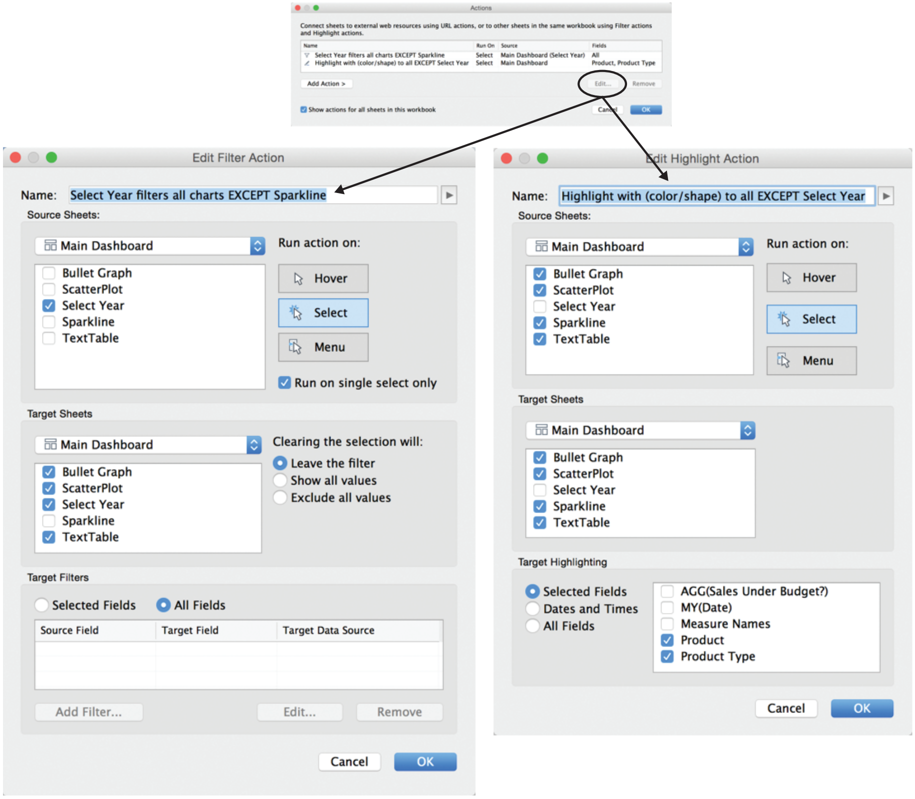


FIGURE 8-56 *Add and Highlight action menus*

Source Sheets

The Source Sheets area contains a drop-down menu that allows you to select any of the worksheets included in your workbook as the source of the action. Main Dashboard is the source sheet in this example. The checked items indicate the panes from which the action will be triggered. Unchecked panes do not invoke the action. Run Action On specifies how the action is invoked.

- **Select:** Action will run using a point and click.
- **Menu:** Runs via point and click in a tooltip or by right-clicking the dimension heading in the pane.
- **Hover:** Action runs as your mouse pointer hovers over a mark.

The examples you have seen so far have used the Select method to run on. Later in the chapter, you create actions that run on the Menu and Hover methods.

Target Sheets

This defines where the action will be applied—what dashboard or worksheet and what individual worksheet object(s). The radio buttons on the right side of this area define Tableau's behavior when the action is deselected by the user.

For example, in Figure 8-51 presented earlier, a filter action for the year 2013 has been applied. In that example, pressing the Escape (ESC) key, or clicking on blank space in a worksheet object, will deselect the filter action. However, because the "Clearing the selection will" option is defined using the Leave the Filter option, the dashboard will remain filtered for the year 2013. In Figure 8-51, clicking Escape clears the 2013 selection in the Select Year pane but leaves the action filter in place as indicated in the titles of the table, scatter plot, and bullet graph.

The "Clearing the selection will" area defines what happens when the action is cleared. Please note that this particular control applies only to filter actions and does not exist for highlight actions. The three options are:

- **Leave the filter:** The filter action keeps only the last selected filter action.
- **Show All Values:** Returns the worksheet or dashboard to an unfiltered state within the context of the dashboard or worksheet.
- **Exclude All Value:** Excludes the data from the view so the worksheets that use the filtered data will not display any information when the filter action is removed.

Target Filters and Target Highlighting Options

Tableau's normal behavior for Highlight Actions types is to use any possible common field existing between the source and the target sheets to apply the action. For Filter Actions, it is the fields that make up the mark selected in the source sheet that drive what fields are included in the filter.

The Target Filters area enables you to specifically restrict the fields that Tableau uses to apply the action if you choose the Selected Fields option. For example, in the highlighting example presented earlier in the chapter, the Highlight Action was restricted to the Product Type and Product fields.

Filter and highlight actions enable you to use the visualizations and legends in your views to create interactive dashboards that respond to selections users make—even if the source and target locations reside in different worksheets. These types of actions are confined to a single workbook.

Tableau provides a third kind of action that allows you to pass the data from your workbook to an external website. This website can be displayed via a separate browser session or embedded into a Tableau dashboard. These are referred to as URL Actions. Figure 8-57 shows the Edit URL Action dialog box.

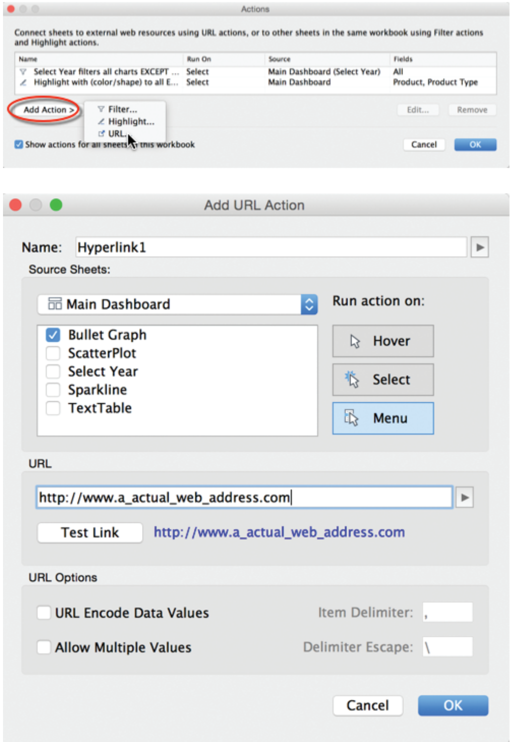


FIGURE 8-57 *Add URL Action dialog box*

At the top of Figure 8-57 is the menu used to open the Add URL Action menu. The name, Hyperlink1, is assigned by default, which you can override with text that is more descriptive, just as you can for Filter and Highlight actions.

The Name and Source Sheets sections are similar to Filter and Highlight actions. In the URL area, you place the web page URL that you want to use as the target of the URL Action. The small arrow on the right side of the URL field allows you to insert field values from your data into the web address. This feature enables you to control what is displayed by the website based on selections made in Tableau. This *never fails* to impress people when they see it for the first time. People will assume that a very skilled hacker was needed to create this type of action. When you get the hang of building URL actions, you can make them in under 30 seconds.

The URL Options at the bottom of the dialog box allow you to deal with characters that may not be understood by the target URL via the URL Encode Data Values. The Allow Multiple Values option gives you a way to pass a list of values (such as a list of products) as parameters to the URL. When passing multiple values, you will also need to define how to separate each record (Item Delimiter) and the Delimiter Escape if the selected delimiter character is used anywhere in your data values.

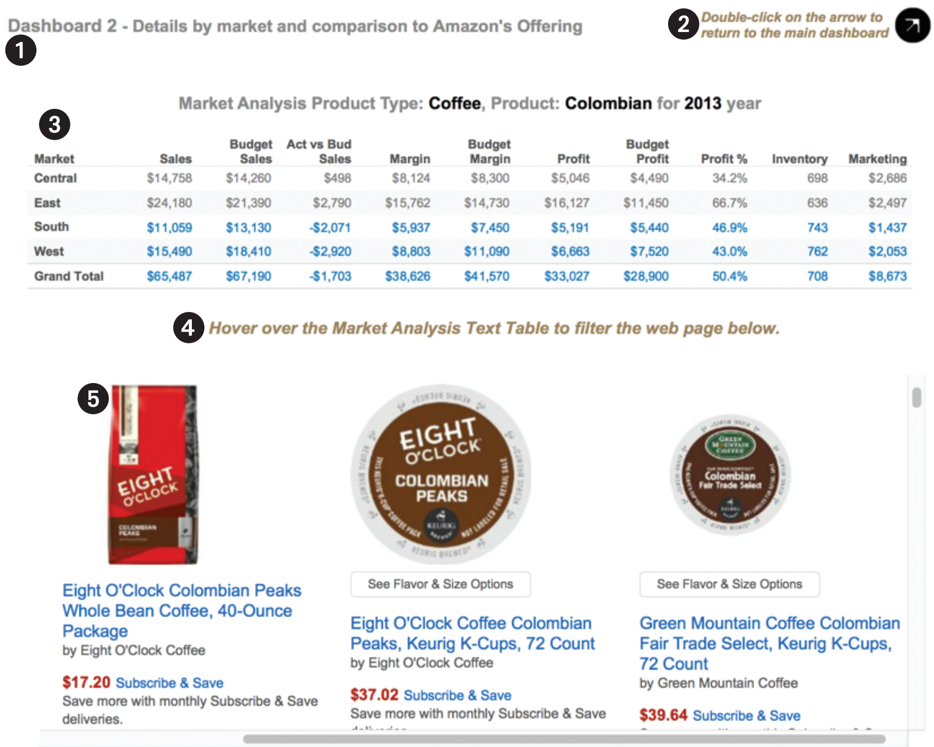
In the next section, you will create another dashboard that allows the information consumer to navigate from the Main Dashboard to the new dashboard by simply pointing and clicking a mark of interest. This new dashboard will include a URL action that will be used to filter a live website embedded into the view.

EMBEDDING A LIVE WEBSITE IN A DASHBOARD

When completed, the next dashboard will include two primary objects—a text table with regional market metrics and an embedded website object. It should look similar to Figure 8-58.

This dashboard has the same size constraints as the Main Dashboard. Navigation to and from the dashboard will be provided by filter actions. It will also include a URL Action to search an embedded website. This URL action will be triggered by hovering over the Market Analysis text table. Figure 8-59 shows an exploded view of Dashboard 2.

FIGURE 8-58 *Completed Dashboard 2*



Market	Sales	Budget Sales	Act vs Bud Sales	Margin	Budget Margin	Profit	Budget Profit	Profit %	Inventory	Marketing
Central	$14,758	$14,260	$498	$8,124	$8,300	$5,046	$4,490	34.2%	698	$2,686
East	$24,180	$21,390	$2,790	$15,762	$14,730	$16,127	$11,450	66.7%	636	$2,497
South	$11,059	$13,130	-$2,071	$5,937	$7,450	$5,191	$5,440	46.9%	743	$1,437
West	$15,490	$18,410	-$2,920	$8,803	$11,090	$6,663	$7,520	43.0%	762	$2,053
Grand Total	$65,487	$67,190	-$1,703	$38,626	$41,570	$33,027	$28,900	50.4%	708	$8,673

FIGURE 8-59 *Dashboard 2 exploded view*

The objects included in Dashboard 2 are:

1. Dashboard title (with added instruction text)
2. Text table (to provide navigation back to the Main Dashboard)
3. Text table (with dynamic titles passed from a filter action from the Main Dashboard
4. A Text object (providing instructions about an action triggered from the Market Analysis text table)
5. A live website (controlled by a URL action invoked from within the Market Analysis text table)

To create Dashboard 2, follow these steps:

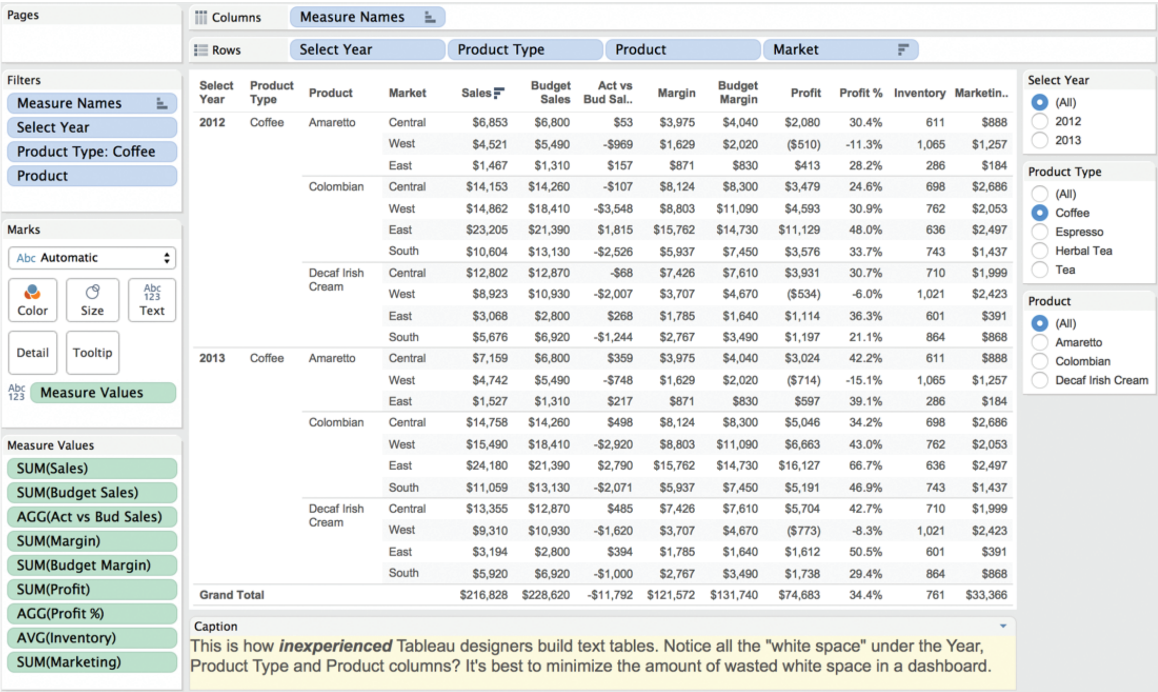
1. Build the Market Analysis text table.
2. Create the Market Analysis dynamic title.
3. Make a calculated value that includes the navigation button text.

Once this new content is finished, you will assemble the dashboard and add a filter action that enables the information consumer to go from the Main

Dashboard to Dashboard 2. You will embed a live website and add another filter action that is invoked when the user moves their mouse over the Market Analysis text table. This action will filter the embedded website by passing a combination of the Product Type and Product information from Tableau. A final action will be added using a text table that includes a string of text and a shape to enable navigation back to the Main Dashboard.

Building the Market Analysis Text Table

Start by adding a text table that you name Market Analysis Step 1 (see Figure 8-60).



Select Year	Product Type	Product	Market	Sales	Budget Sales	Act vs Bud Sal..	Margin	Budget Margin	Profit	Profit %	Inventory	Marketin..
2012	Coffee	Amaretto	Central	$6,853	$6,800	$53	$3,975	$4,040	$2,080	30.4%	611	$888
			West	$4,521	$5,490	-$969	$1,629	$2,020	($510)	-11.3%	1,065	$1,257
			East	$1,467	$1,310	$157	$871	$830	$413	28.2%	286	$184
		Colombian	Central	$14,153	$14,260	-$107	$8,124	$8,300	$3,479	24.6%	698	$2,686
			West	$14,862	$18,410	-$3,548	$8,803	$11,090	$4,593	30.9%	762	$2,053
			East	$23,205	$21,390	$1,815	$15,762	$14,730	$11,129	48.0%	636	$2,497
			South	$10,604	$13,130	-$2,526	$5,937	$7,450	$3,576	33.7%	743	$1,437
		Decaf Irish Cream	Central	$12,802	$12,870	-$68	$7,426	$7,610	$3,931	30.7%	710	$1,999
			West	$8,923	$10,930	-$2,007	$3,707	$4,670	($534)	-6.0%	1,021	$2,423
			East	$3,068	$2,800	$268	$1,785	$1,640	$1,114	36.3%	601	$391
			South	$5,676	$6,920	-$1,244	$2,767	$3,490	$1,197	21.1%	864	$868
2013	Coffee	Amaretto	Central	$7,159	$6,800	$359	$3,975	$4,040	$3,024	42.2%	611	$888
			West	$4,742	$5,490	-$748	$1,629	$2,020	($714)	-15.1%	1,065	$1,257
			East	$1,527	$1,310	$217	$871	$830	$597	39.1%	286	$184
		Colombian	Central	$14,758	$14,260	$498	$8,124	$8,300	$5,046	34.2%	698	$2,686
			West	$15,490	$18,410	-$2,920	$8,803	$11,090	$6,663	43.0%	762	$2,053
			East	$24,180	$21,390	$2,790	$15,762	$14,730	$16,127	66.7%	636	$2,497
			South	$11,059	$13,130	-$2,071	$5,937	$7,450	$5,191	46.9%	743	$1,437
		Decaf Irish Cream	Central	$13,355	$12,870	$485	$7,426	$7,610	$5,704	42.7%	710	$1,999
			West	$9,310	$10,930	-$1,620	$3,707	$4,670	($773)	-8.3%	1,021	$2,423
			East	$3,194	$2,800	$394	$1,785	$1,640	$1,612	50.5%	601	$391
			South	$5,920	$6,920	-$1,000	$2,767	$3,490	$1,738	29.4%	864	$868
Grand Total				$216,828	$228,620	-$11,792	$121,572	$131,740	$74,683	34.4%	761	$33,366

FIGURE 8-60 *Market analysis, Step 1*

The view in Figure 8-60 includes additional information, but it is an inefficient layout. Including the Select Year, Product Type, and Product fields as columns in the Columns shelf consumes too much vertical and horizontal space. The design won't work well in Dashboard 2 because we are limited to 800 × 600 pixels of space. The white space in this layout would be better utilized by data ink. You can address this problem by moving these fields into a title that utilizes dynamic title objects for those fields. Make a copy of this sheet and modify the copy to look like Figure 8-61.

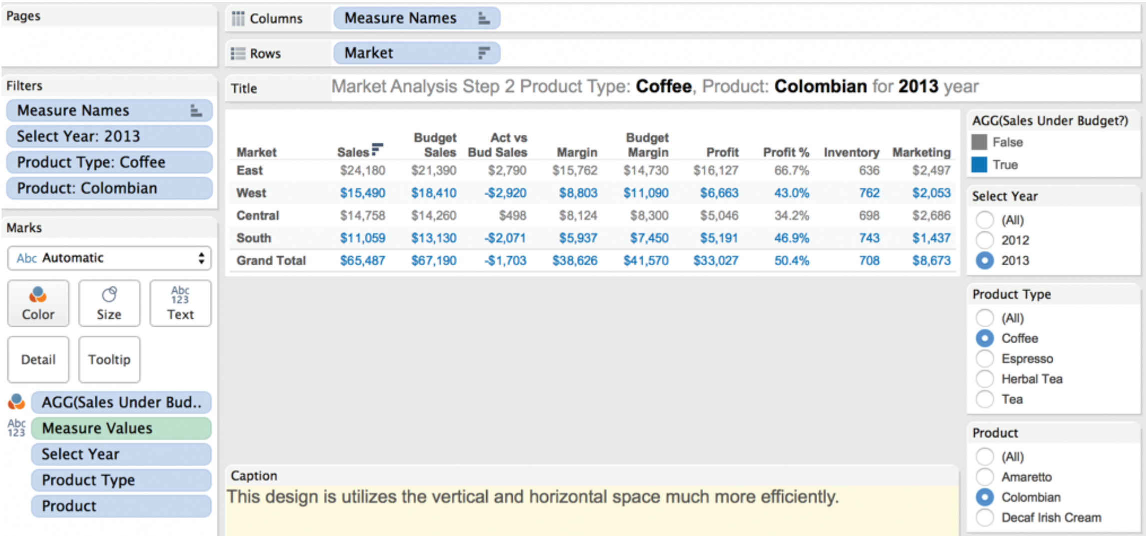


FIGURE 8-61 *Market analysis, Step 2*

As you can see in Figure 8-61, this layout is much more efficient. Adding the Boolean calculation (the Sales Under Budget? field) to the Color button on the Marks card is consistent with the use of color in the Summary by Product Type text table and Sales versus Budget by Product panes in the Main Dashboard. Remember that the blue color indicates below budget sales.

The addition of the dynamic title objects is a more space-efficient way to display the Select Year, Product Type, and Product fields. Figure 8-62 shows how you should edit the title of this sheet.

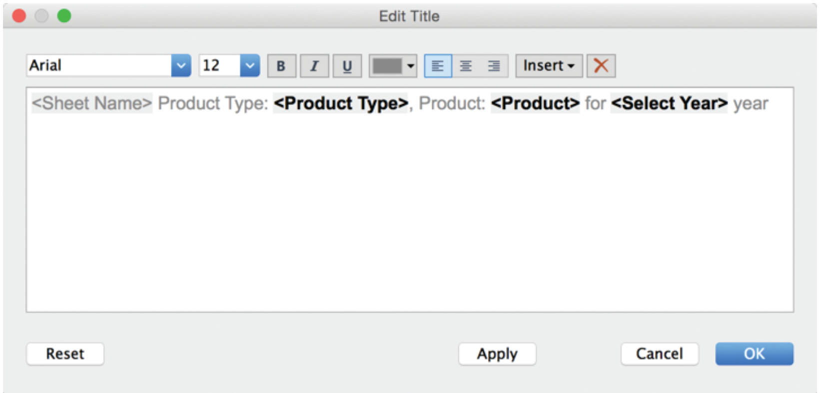


FIGURE 8-62 *Market analysis title*

After you have completed this view, make a copy of the page and name the new page Market Analysis. This will become the primary data pane in Dashboard 2. Refer to Figure 8-59 (area 3).

Building the Dashboard 2 Navigation Button

The navigation button in Figure 8-59 (area 2) is a text table that includes a calculated value that is a string combined with a shape. The calculation used to make the text is displayed in Figure 8-63.

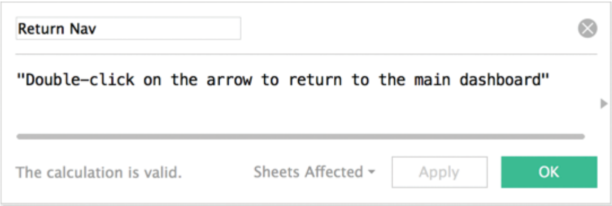


FIGURE 8-63 *Calculated value for navigation button*

As you can see, this calculation is nothing more than a text string wrapped in double quotes. Single quotes will also work—just be consistent. The field is named Return Nav, and that is what you will use to build the view. Create a new sheet and name the sheet Nav Button Step 1. Add the field to the view. Your sheet should look like Figure 8-64.

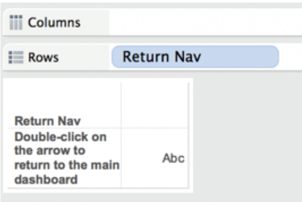


FIGURE 8-64 *Nav button, Step 1*

To finish this simple text table, duplicate the Nav Button Step 1 sheet and name the new one Nav Button. Then follow these steps:

1. Hide the field labels for the rows.
2. Use the Marks card to change the mark type from automatic to shape.
3. Select More Shapes and then the Arrows shape palette.
4. Pick the black arrow pointing to the upper right.
5. Use the Size button on the Marks card to make the mark fill the space.

6. Format the Return Nav font to 12-point, italics, Arial, and use a brown color.
7. Use Format ➢ Borders and change all of the borders to "none."

When you finish these steps, the Nav Button sheet should look like Figure 8-65.

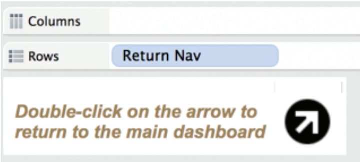


FIGURE 8-65 *Completed Nav Button sheet*

Now that the Nav Button sheet and Market Analysis sheet views are finished, Dashboard 2 can be assembled.

ASSEMBLE DASHBOARD 2

Create a new dashboard, name it Dashboard 2, and set the size to exactly 800 × 600 pixels, a size consistent with the Main Dashboard. Then select the Show Title option to display a title object and type in the dashboard title that you see in Figure 8-66 (Dashboard 2–Market analysis and comparison to a retail website).

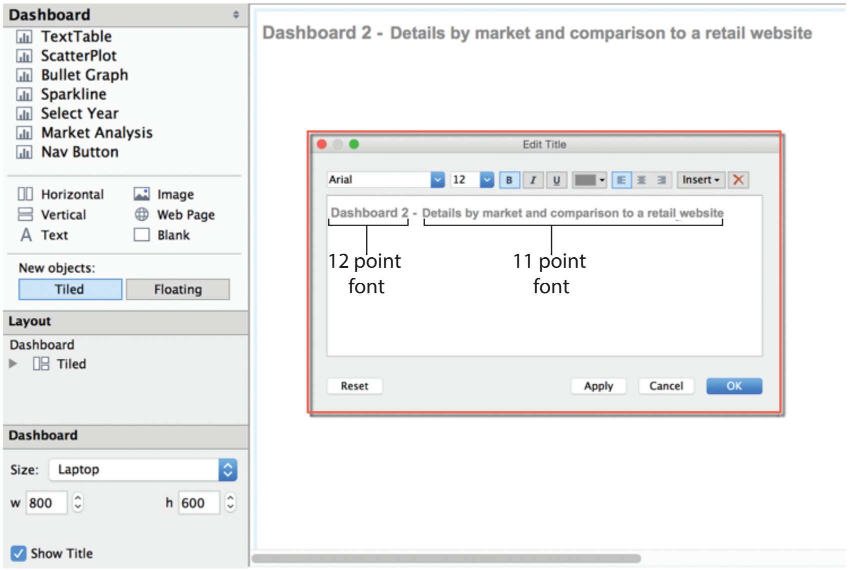


FIGURE 8-66 *Dashboard 2*

Superimposed on top of the new dashboard is the Edit Title dialog box. Note that the font is Arial 12-point for the word “Dashboard” but the rest of the text is Arial 11-point. All of the title text is bold. Use the following steps to add the remaining objects to the dashboard:

1. Drag the Market Analysis worksheet into Dashboard 2.
2. Delete the Color legend.
3. Edit the Market Analysis worksheet to Fit Entire View.
4. Add a text object that says “Hover over the Market Analysis text table to filter the web page below.”
5. Change the font of the text object added in step 4 to Arial, 12-point, italics, and bold, and center the text.
6. Place a Web Page object below the Text object you added in step 5. Leave the URL blank for now.
7. Add a Horizontal layout container above the title of the dashboard and then place the title inside the layout container.
8. Drag the lower edge of the layout container down to provide more vertical space.
9. Add the Nav Button worksheet to the right side of the layout container.
10. Delete the Nav Button title by pointing at it, right-clicking, and selecting the Hide Title menu option.
11. Set the Nav Button to fit the entire view and arrange it so that it fits in the space well.

When you finish these steps, your version of Dashboard 2 should look similar to Figure 8-67.

The blank area at the bottom of Figure 8-67 shows the Web Page object that is currently empty. When you define a URL action with a valid web URL, that space will contain a live website. The bones of Dashboard 2 are in place. Adding three additional actions to this workbook will provide navigation between the Main Dashboard and Dashboard 2 and also provide the web address needed to populate the Web Page object. Do the following steps to complete the design:

- Create a filter action using the Bullet Graph in the Main Dashboard to filter the Market Analysis text table in Dashboard 2.
- Add a filter action to return the user from Dashboard 2 to the Main Dashboard.
- Make a URL action in Dashboard 2 to filter the Web Page object in Dashboard 2.

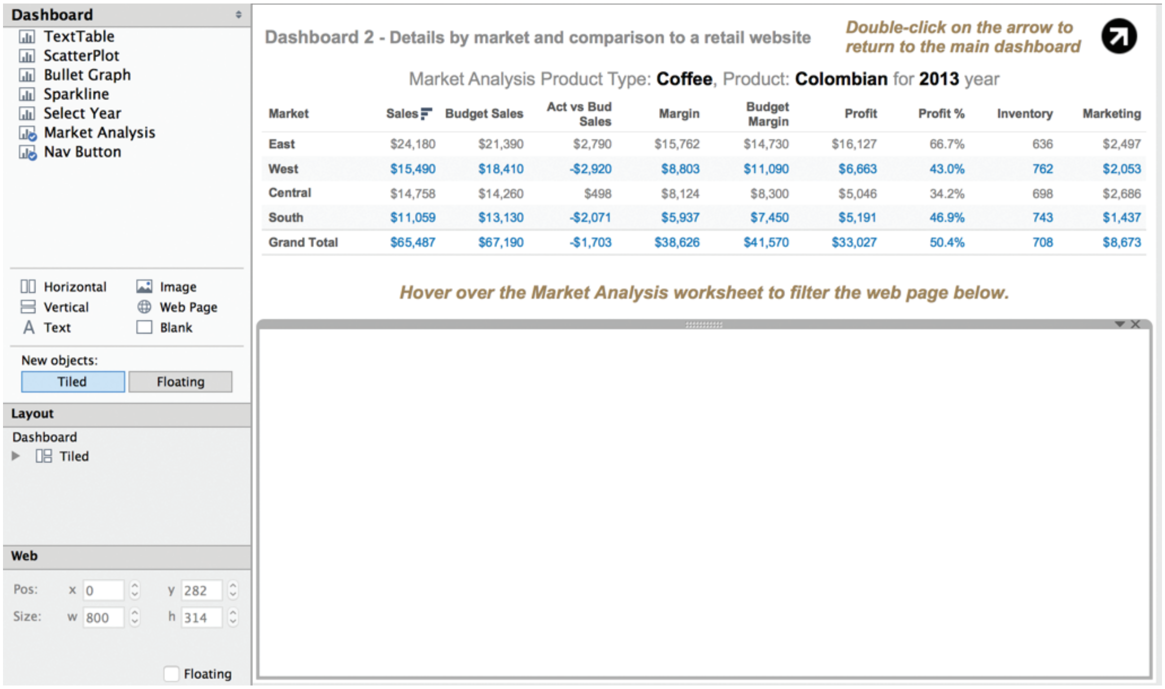


FIGURE 8-67 *Dashboard 2, Steps 1-11 completed*

Creating the Filter Action to Navigate from the Main Dashboard to Dashboard 2

You will now create a filter action that will be triggered from the Bullet Graph in the Main Sales Dashboard to the Market Analysis text table in Dashboard 2. The purpose of the action is to allow the audience to analyze sales of a selected product type and product—by market.

Understanding what information is passed in this filter action is important. The Sales versus Budget by Product bullet graph in the Main Dashboard includes the following dimension fields:

- Select Year
- Product Type
- Product

All of these fields need to be passed in the Filter action from the Main Dashboard to the Market Analysis pane in Dashboard 2. The other important consideration is how to invoke the Filter action. The Bullet Graph in the Main

Dashboard already includes a Highlight action that is triggered by Select (point and click). Therefore, you want the information consumer to stay in the Main Dashboard when they click a bar or row heading so that they see the related highlighting in the surrounding views.

Also, in my experience users can be startled if they "jump" to a completely new dashboard simply by clicking a mark or heading. For these reasons, you will trigger the Filter action to navigate the user to Dashboard 2 using the menu selection. Figure 8-68 shows the Edit Filter Action dialog box settings you should use to define the action.

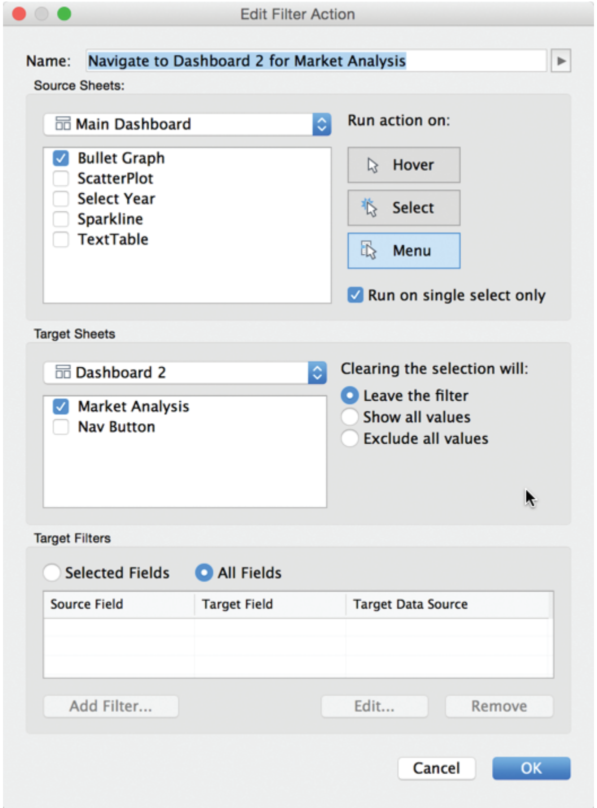


FIGURE 8-68 *Bullet graph menu filter action*

Look closely at Figure 8-68 and notice that the Source sheet area has the Main Dashboard and Bullet Graph selected. The Run action on is Menu. The Target Sheets area has Dashboard 2 and the Market Analysis text table selected. Clearing the selection will leave the filter on. This is important. If the filter is not left on, the text table in Dashboard 2 will be potentially unreadable. Finally, the Target Filters area defines that All Fields are to be used in the filter action. This means that the key dimensions (Select Year, Product Type, and Product) will all be passed to the Market Analysis in Dashboard 2.

When you create actions, always check to ensure that the result is correct. If you don't, you could potentially present incorrect information. Once you complete the filter action, test it by trying to run the filter. Figure 8-69 shows the tooltip that will appear when you point at a mark in the Bullet Graph.

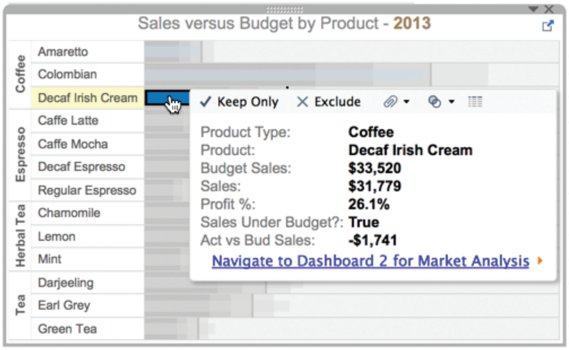
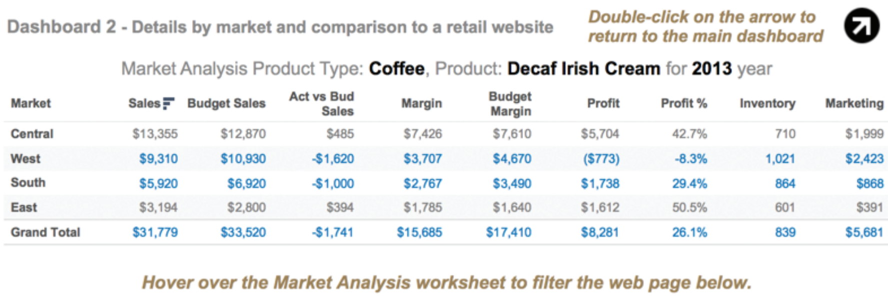


FIGURE 8-69 *Menu action*

The tooltip displays the filter action created in the last step as blue text. Clicking on the blue text will take you to Dashboard 2. You can also execute the action by pointing at the row header and clicking to expose the action.

Using either method to trigger the action, the Market Analysis text table should be filtered for Coffee, Decaf Irish Cream, and the year 2013. Be sure that the "Run on single select" option is checked. You can see the result in Figure 8-70.



Market	Sales	Budget Sales	Act vs Bud Sales	Margin	Budget Margin	Profit	Profit %	Inventory	Marketing
Central	$13,355	$12,870	$485	$7,426	$7,610	$5,704	42.7%	710	$1,999
West	$9,310	$10,930	-$1,620	$3,707	$4,670	($773)	-8.3%	1,021	$2,423
South	$5,920	$6,920	-$1,000	$2,767	$3,490	$1,738	29.4%	864	$868
East	$3,194	$2,800	$394	$1,785	$1,640	$1,612	50.5%	601	$391
Grand Total	$31,779	$33,520	-$1,741	$15,685	$17,410	$8,281	26.1%	839	$5,681

FIGURE 8-70 *The filtered result in Dashboard 2*

Try using the filter action on a few different markets in the Bullet Graph just to be sure that it is working properly. The title created earlier in Dashboard 2, which includes the dynamic title objects for Product Type, Product, and Select Year, provides visual confirmation that the filter action passed those dimensions correctly from the Main Dashboard to Dashboard 2.

When you are building these kinds of actions in the real world, you will probably have to iterate through this a few times to get everything working. If the filter action doesn't work as you expected, your source and target sheets do not contain all of the necessary matching fields. When you are new to Tableau, it may require a few attempts before you get everything to work the way you want.

Next, you'll create the URL action that will populate the Web Page Object in Dashboard 2.

Making the URL Action

If you aren't a coder, creating a URL action that passes dimensions from your data to a web page embedded in a dashboard might sound intimidating. Don't worry. This is not difficult. It is similar to the dynamic titles you have already seen. If you know how to search for something on a web page, you can build URL actions.

To demonstrate another way you can invoke an action, this example uses the Run Action On Hover option to trigger the action. This means that when users hover anywhere over the Market Analysis text table, the URL search will be executed. In an actual use case, it might be better to trigger the action using Select or Menu.

Open your browser and search Amazon's website for "coffee, decaf irish cream." Figure 8-71 shows my search.

Doing a search outside of Tableau first helps you understand how the website encodes the search string. Notice that I restricted the search on Amazon's website to include grocery and gourmet food options. I also separated each dimension (coffee, decaf irish cream) with a comma. As you see in Figure 8-71, the search variables worked.

Copy the URL from the web browser address, which you see at the very top of Figure 8-71. Paste this code into the URL field in the URL action dialog box in Figure 8-72.

Use the Test Link button to ensure that the link works properly. What you've accomplished at this stage is passing the static variables that were part of the original search made to the website.

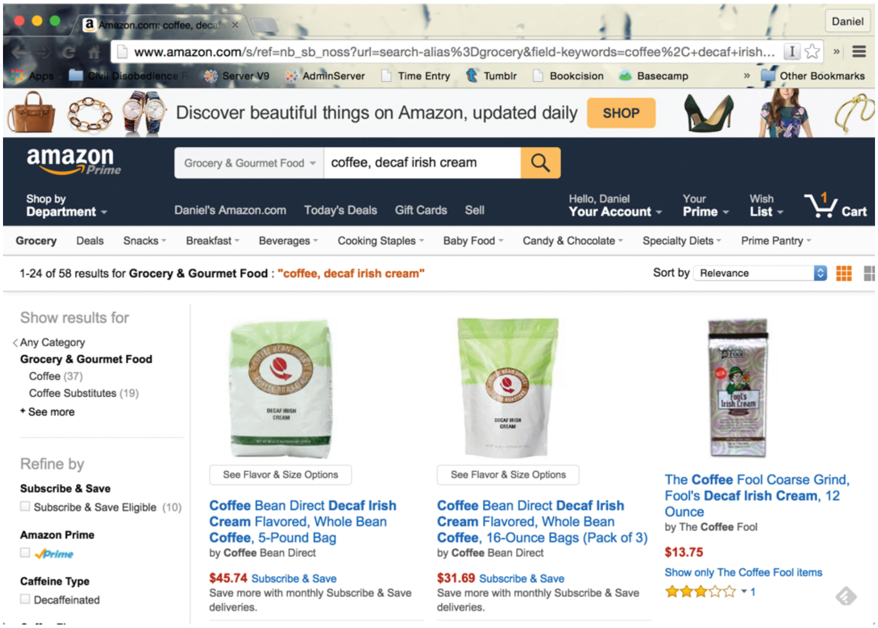


FIGURE 8-71 *Searching Amazon's website*

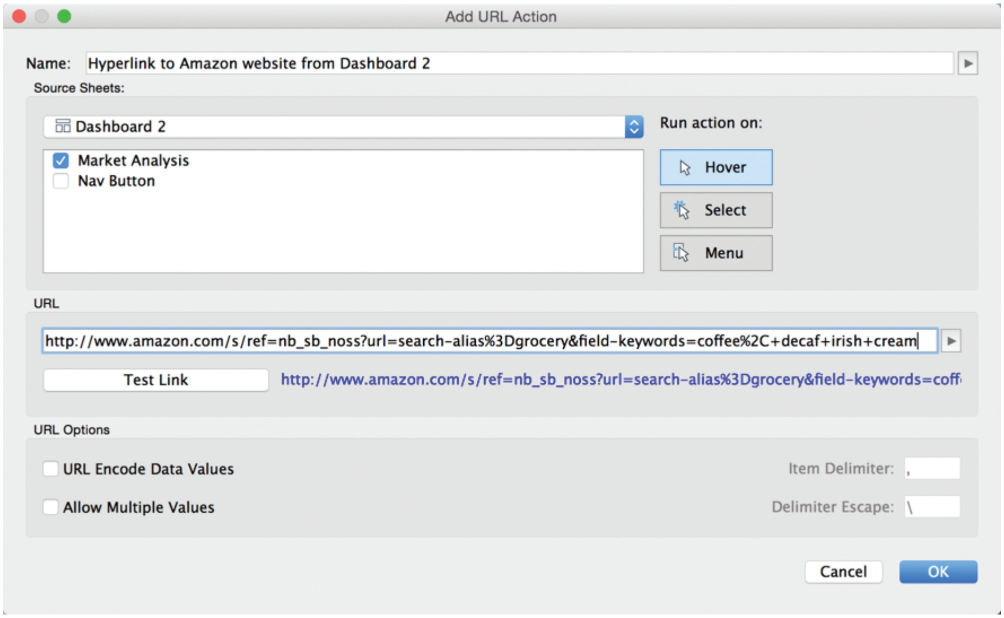


FIGURE 8-72 *URL action dialog box*

To enable Tableau to automatically change the search sent to Amazon based on selections you make in your dashboard, you must replace the static search keywords contained in the URL string with fields inserted by Tableau. Figure 8-73 shows you what parts of the string to replace.

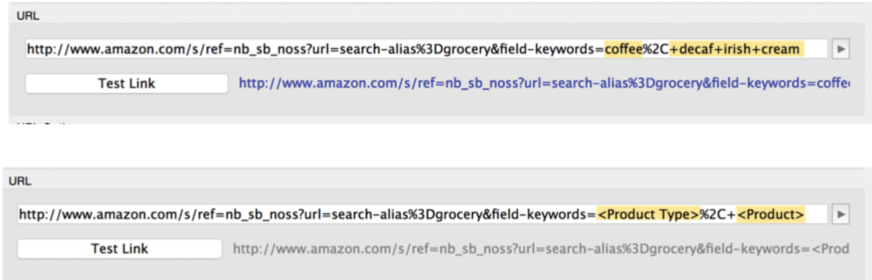


FIGURE 8-73 *Inserting URL variables*

The top half of Figure 8-73 shows the original search string. The bottom shows the string with the Tableau fields inserted using the insert arrow to place the desired fields. Use the Test Link button to verify the search is working correctly.

Go back to the bullet graph in the Main Dashboard and execute the menu action with a different product. Hovering your mouse pointer over the Market Analysis text table should trigger another search of the website, and the products displayed should reflect the new search fields inserted by the URL action. Figure 8-74 shows Dashboard 2 after selecting the Expresso, Café Mocha bar in the bullet graph while in the Main Dashboard.

Using embedded Web Page objects with URL actions provides a method for combining your data with information from the Web without having to be a programming expert. URL actions can be used in many different ways. I hope you can see the potential for enhancing your dashboards with information from the Internet. Next, you will add an action to provide a way for the user to navigate back to the Main Dashboard from Dashboard 2.

Creating a Home Button

The text table added in the upper-right area of Dashboard 2 will be the source sheet used for the filter that will take the user back to the Main Dashboard. Figure 8-75 shows the selections made to define the action.

When you double-click that button or text, you will return to the Main Dashboard. The last selection you made in the Bullet Graph will still be highlighted. Click empty space in the graph to clear that selection on the Main Dashboard.

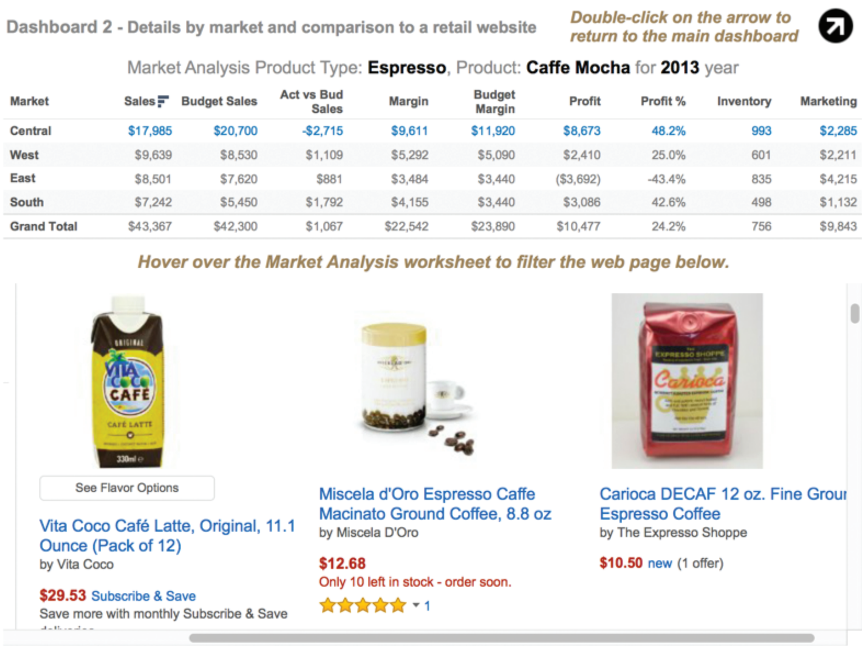


FIGURE 8-74 *Dashboard 2 with embedded website*

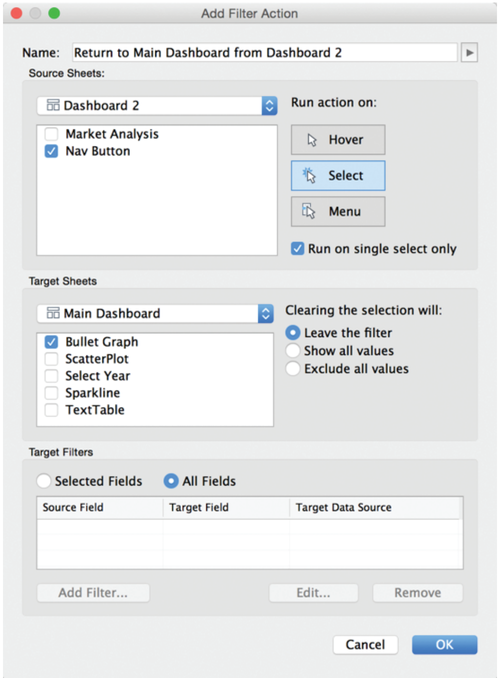


FIGURE 8-75 *Action to return to the Main Dashboard*

One additional change is required to the Main Dashboard to ensure that the Filter Action from the Main Dashboard works properly. The Product Type heading in the Bullet Graph needs to be hidden prevent the possibility of the filter action including more than one product. Why? Because if more than one product is included in the filter, the Market Analysis text table in Dashboard 2 will not display the numbers properly. Hide the Product Type heading by pointing at the heading, clicking the right mouse button, and unchecking the Show Heading option.

Congratulations! You just completed building an advanced dashboard in Tableau. Later you will add one final "disappearing" text table in the Main Dashboard that uses another option for clearing filter action selections.

The final design work on both dashboards relates to tooltips. After looking over a new design and reviewing the information presented, Tooltips are a great way to provide supplemental information and instructions for your audience.

ADDING DETAILS ON DEMAND WITH TOOLTIPS

The dashboard design is nearly complete. Getting outside feedback at this stage is helpful. Your audience may want to look at data in ways you didn't consider, which can lead to revisions to the layout, content, and filtering. Assuming that all of the design criteria have been fulfilled, finalizing the design normally requires adding textural content to provide relevant details on demand that enhance the content.

Tooltips appear when you hover over marks in worksheets and dashboards. They are an efficient way to convey detailed and context-specific information because they appear only on demand. Best of all, they offer a great option for adding a lot of details because they appear in the view only when the audience points at a mark. Tooltips contain the fields used in your views by default and any other field that is used as a filter or used on the Marks card. They can also include manually added notes.

Using the Tooltip Editor

You can see the tooltip associated with any workbook by going to the Worksheet ➢ Tooltip menu. Within a dashboard, click inside the data view of interest first, and then access the Worksheet ➢ Tooltip menu. Figure 8-76 shows the tooltip for the Bullet Graph in the Main Dashboard.

The left side of Figure 8-76 shows the Edit Tooltip dialog box for the bullet graph. On the right is the tooltip as it will appear when pointing at a mark within the view. The fields and selections in the live tooltip are the defaults.

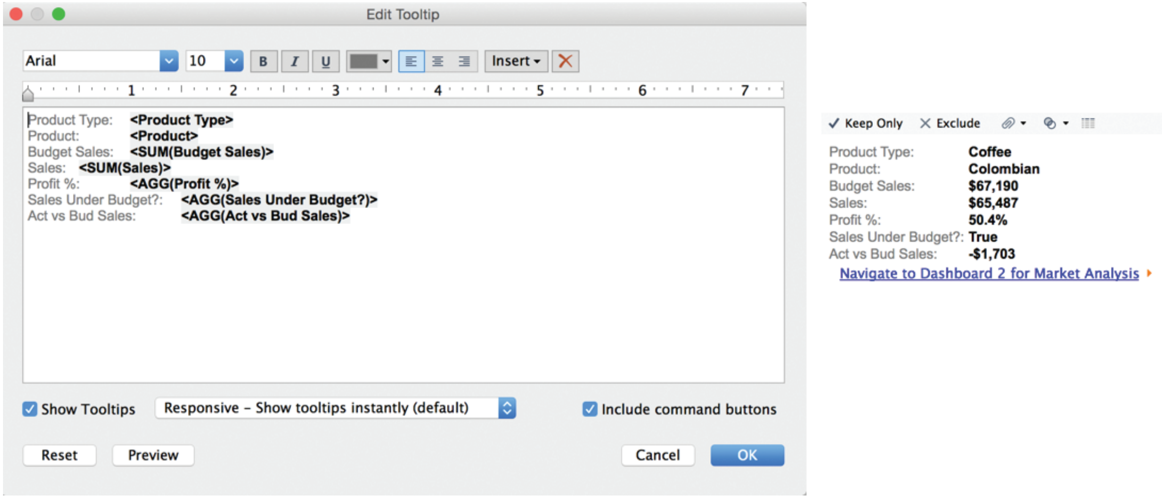


FIGURE 8-76 *Tooltip editor—bullet graph*

The editor is similar to the title editor. The top of the Edit Tooltip dialog box includes text font, size, and other style controls. The Insert button allows you to place dynamic fields into the tooltip. Any text that appears within angle brackets (< >) is a field value that will change based on the context of the selections made within the view.

The bottom of the Edit Tooltip dialog box contains options that give you control over what and how the tooltip displays.

- **Show Tooltips:** Uncheck if you don't want the tooltip to display.
- **Drop-down options:** Responsive (default) or on Hover.
- **Include command buttons:** Checked (default)

The Responsive option makes the tooltip appear instantly while the Hover option will have a small delay. The Include command buttons option should not be selected in a dashboard that you are publishing to a large number of users. There are some commands that you may not want to grant access because they enable a user to exclude marks, group marks, and create sets, and they may expose the underlying data used to create the view. You can see these controls in an actual tooltip on the right side of Figure 8-76. They appear at the top of the tooltip. When the Include command buttons option is not selected, these will not appear in the tooltip.

ENHANCING TOOLTIPS AND TITLES

The last steps in the design of any dashboard should be reserved for customizing the tooltips and putting the finishing touches on titles. I recommend that you play with the dashboard for a few minutes and think about the questions you have. This might also be a good time to invite a few users to look at the dashboard. They will provide additional insight that will inform the content and design of your tooltips. That feedback might also lead you to place additional small instructions to highlight options that may not be obvious.

The following figures show customized tooltips that you can add to the Main Dashboard. The left side of each figure shows the tooltip as it will appear in the dashboard. The related Edit Tooltip dialog box appears to the right. Four of the five data views in the Main Dashboard are presented. The Select Year view will be presented later. Figure 8-77 is the tooltip for the Summary by Product Type view.

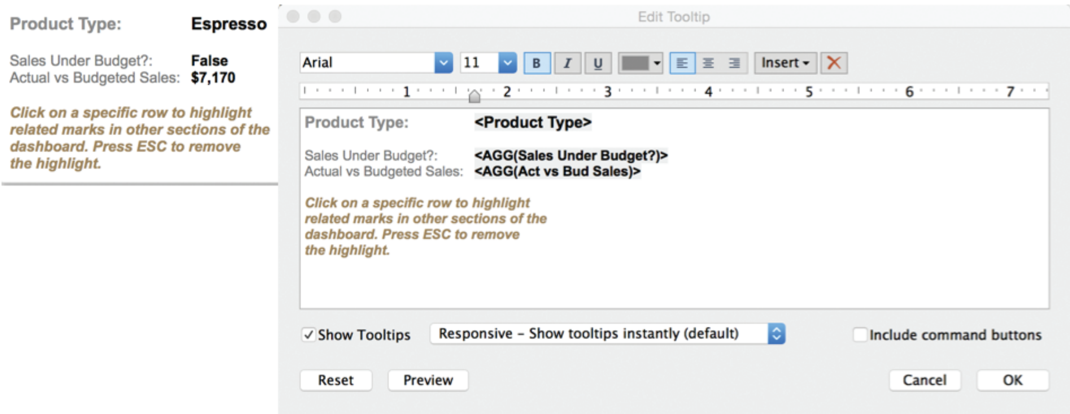


FIGURE 8-77 *Summary by Product Type tooltip*

Figure 8-77 includes an instruction in brown and italics that provides information on the highlight action that can be invoked from this and several other data views in the dashboard. Note the perfect alignment of the dynamic data text. The small grid alignment tool just above the text entry area is used to align each row. To align a specific row, click the row and then move the small pointer to the position you want the text to start.

Figure 8-78 shows the tooltip for the Sales versus Marketing Expense scatter plot.

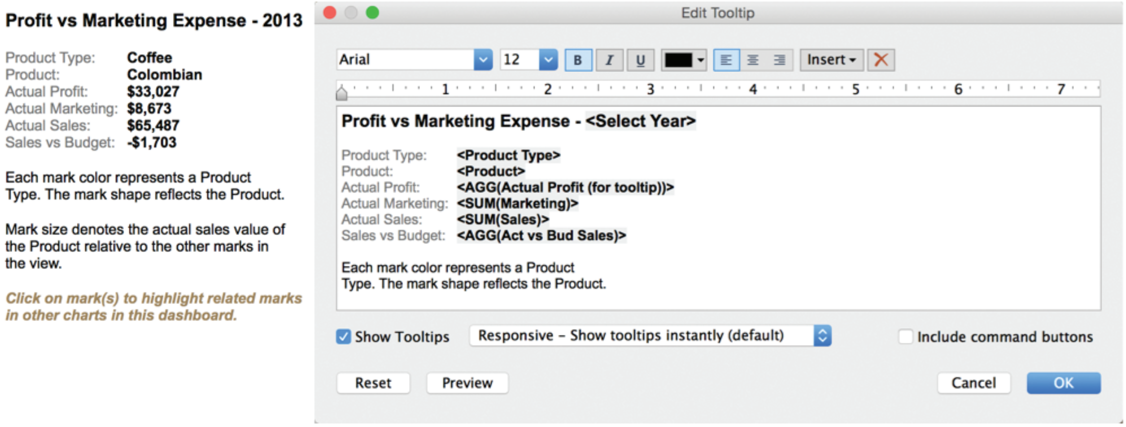


FIGURE 8-78 *Sales versus Marketing Expense tooltip*

Notice that Figure 8-78 uses different font sizes for the text at the bottom of the tooltip. The field `<Actual Profit (for tooltip)>` was copied from the original Actual Profit field just to provide for a different number format in the tooltip than appears in the axis of the view. The axis is in thousands while the value in the tooltip is to the dollar.

Figure 8-79 is associated with the Sales versus Budget by Product bullet graph.

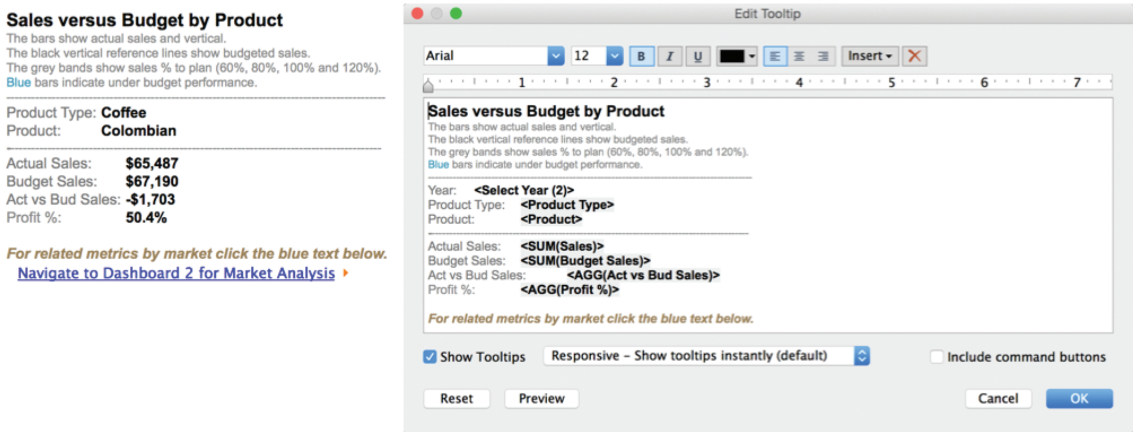


FIGURE 8-79 *Sales versus Budget by Product tooltip*

The tooltip in Figure 8-80 includes nuanced text formatting, lines to define sections (8-point font is used for the lines). A Menu action also appears at the bottom of the tooltip. Just above the blue text that triggers the action is brown text that provides instructions describing what will happen when the text is clicked.

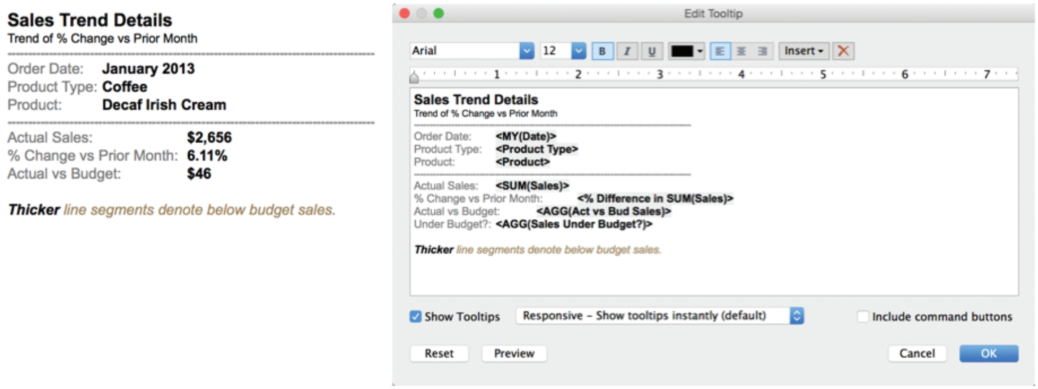


FIGURE 8-80 *Sales Trend tooltip*

The Sparkline tooltip includes a new wrinkle that I haven't previously discussed. After reviewing this dashboard, I thought that it would be nice to add the Boolean calculation in the Sparkline as well. The blue/gray color legend is used everywhere else to communicate that information. Because color is being used here to convey Product Type, I decided to use size to indicate sales performance versus budget. The text at the bottom of the tooltip provides that information. This represents a compromise, but this might be an acceptable compromise that the dashboard audience favors.

ADDING A READ ME DASHBOARD

Most people don't include Read Me worksheets in workbooks. If you are serving a large user base, the additional hour of time required to document your work in a Read Me dashboard could save you time in the long run. A well-documented workbook that provides an explanation of particularly complex calculated values, the data sources used, experts consulted, or other ancillary details will reduce your phone and e-mail traffic. If you have to manage a very large number of dashboards, a well-documented workbook can jog your memory if you need to revisit an old design or help in training new staff.

The Coffee Chain dataset used for creating the two sample dashboards in this chapter demonstrates that the size of your data isn't as important as the quality of information you can extract from it. The market analysis information contained in Dashboard 2 demonstrates how easy it is to use external information from websites with your data and make it interact with your proprietary data.

BONUS: ADDING A FLOATING DASHBOARD OBJECT

This example neglects to utilize one of the available action clearing options—Exclude all values. This option hides the object that is displayed when the action is cleared. In this way you can add data panes in your dashboard that float over other dashboard objects when the action is selected but disappear when the selection is removed. It can be useful if you want to guide your users through a series of questions by forcing a selection to expose more (filtered) data. It is also useful if you have limited space in your dashboard. I tend to avoid the use of floating objects in production dashboards because it is difficult to anticipate the effects that different screen resolutions have on the placement of the object. Because the example dashboard is fixed at a pixel resolution of 800 × 600, a floating object can be used with a predictable outcome.

Figure 8-81 zooms into the upper right of the Main Dashboard and shows a floating object that displays the percentage of sales by market when a year is clicked in the Select Year pane.

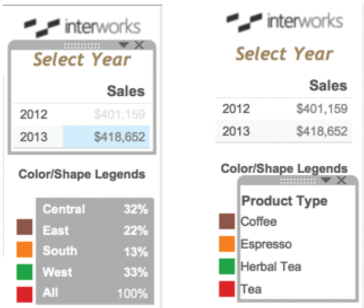


FIGURE 8-81 *Floating object showing sales mix*

The left image in Figure 8-81 shows the gray percentage sales mix table the way it appears to the information consumer when the year 2013 is selected. The image on the right shows the actual placement of the sales mix pane. Notice that the pane does not cover the color images from the Product Type color legend. Covering the colors would prevent users from clicking on the legend to highlight related marks in the other panes within the dashboard. While the dashboard includes a highlight action for that highlighting from within the other data panes in the dashboard, careful placement of the floating object preserves the user's ability to highlight from the color legend.

Pressing the Esc key or clicking the mark or row heading in the Select Year text table removes the table. Using this technique, you can fit additional information into the same space.

The Select Year pane also includes a tooltip that provides instructions about the filter that applies to the other charts in the Main Dashboard and details about the Floating Sales Mix % text table. Figure 8-82 shows the tooltip.

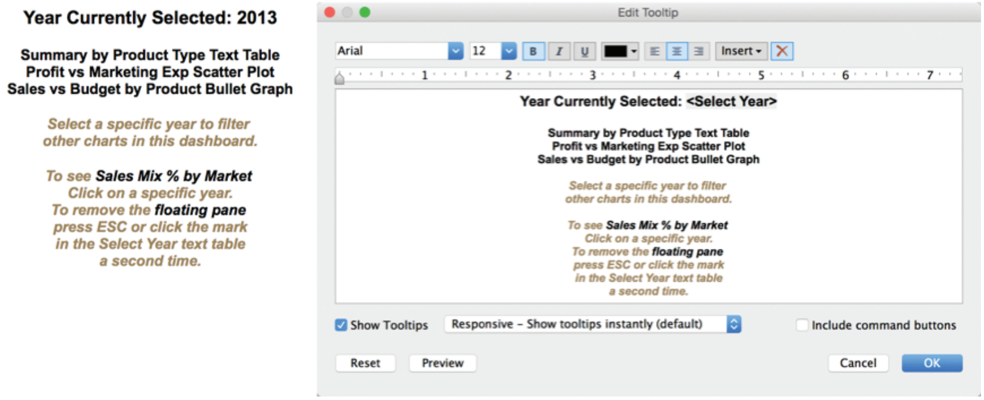


FIGURE 8-82 *Select Year tooltip*

The Select Year tooltip provides additional instructions regarding filtering by year for the Main Dashboard and details about the floating object. Figure 8-83 shows the tooltip for the Floating Sales Mix % pane.

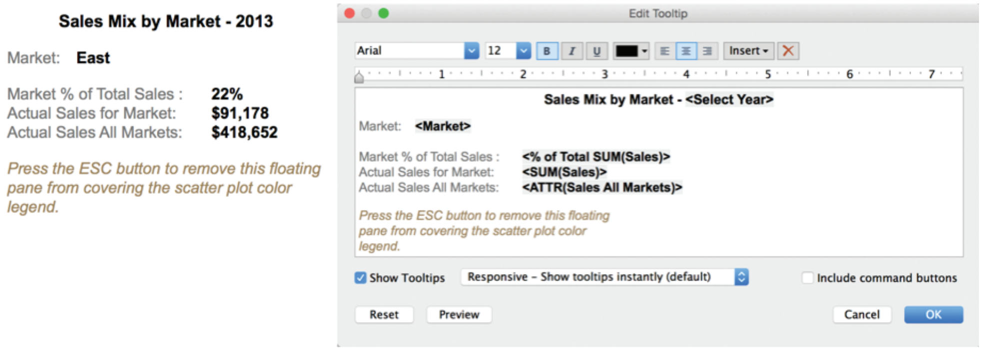


FIGURE 8-83 *Sales Mix by Market tooltip*

Pay particular attention to the Sales All Markets field you see in Figure 8-83. This is a Calculated Field that uses a Level of Detail function to display total sales for all markets. You see the calculation in Figure 8-84.

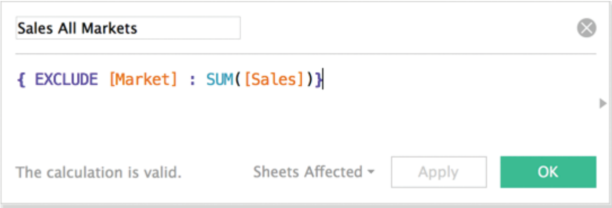


FIGURE 8-84 *Sales All Markets LOD calculation*

Before Tableau V9, the All Markets calculation would have been difficult because the table used in the view shows the data by individual market. Level of Detail Expressions make it easy to define calculations that provide results in a different Level of Detail from the view. Using the Exclude expression provides an easy way to display the total sales for all markets. As the user points at different markets in the Sales Mix by Market table, the total sales for the year being viewed is displayed along with the sales for all markets. Refer to Chapter 4 and to the function reference for Level of Detail calculations in Appendix E for more information on Level of Detail expressions.

Steps to Create a Disappearing Dashboard Object

To add the floating object that you see in Figure 8-81 to the workbook, follow these steps:

1. Create a text tab visualization that contains sales and a quick table calculation for percent of total sales. Style the text table to your taste (see Figure 8-85).
2. Add the worksheet to your dashboard while selecting the Floating New Object option (see Figure 8-86).
3. Define the action using the Select Year text table as the Source Sheet and the Sales Mix by Market text table as the Target Sheet. Be sure to select the Exclude all values option (see Figure 8-87).
4. Style the tooltip, as you saw earlier in Figure 8-83.

Figure 8-85 uses a Quick Table calculation to compute the Percent of Total, Tableau (Down). The image shows Select Year on the Filters shelf. This will be replaced by a filter action from the Main Dashboard. The Marks card includes the Sales All Markets LOD calculation, which is used in the related tooltip to display sales for the selected year for all of the markets.

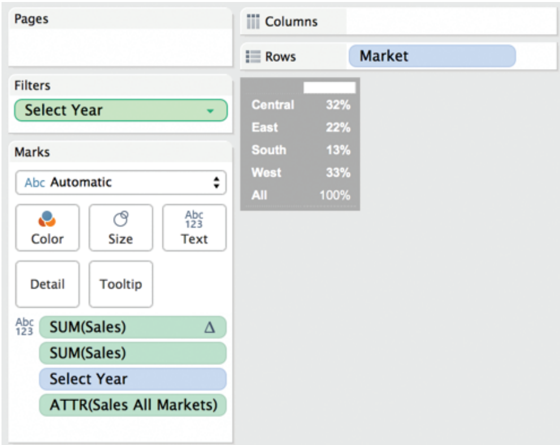


FIGURE 8-85 *Create the text table.*

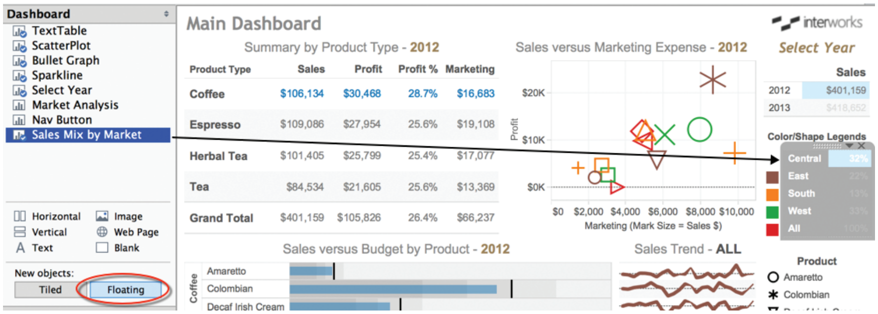


FIGURE 8-86 *Add the floating object.*

Figure 8-86 shows the process of positioning the Sales Mix by Market text table in the space over the Product Type color legend. Note that the New Objects Floating option is selected. Care should be taken to ensure that the color marks in the legend are not covered by the floating text table.

Figure 8-87 shows the Edit Filter Action dialog box for the Sales Mix by Market text table. Use the Clearing the selection will option, Exclude all values radio button.

After defining the action, test it by clicking either row in the Select Year text table to see exactly how the floating (normally hidden) Sales Mix worksheet appears. You will need to do some editing to hide the worksheet title. Press Escape (Esc) or click the same mark in the Select Year text table to see how the action behaves when each year is selected. Using floating actions that are hidden until a specific selection is made can be an effective way to display data in dashboards that have limited screen real estate.

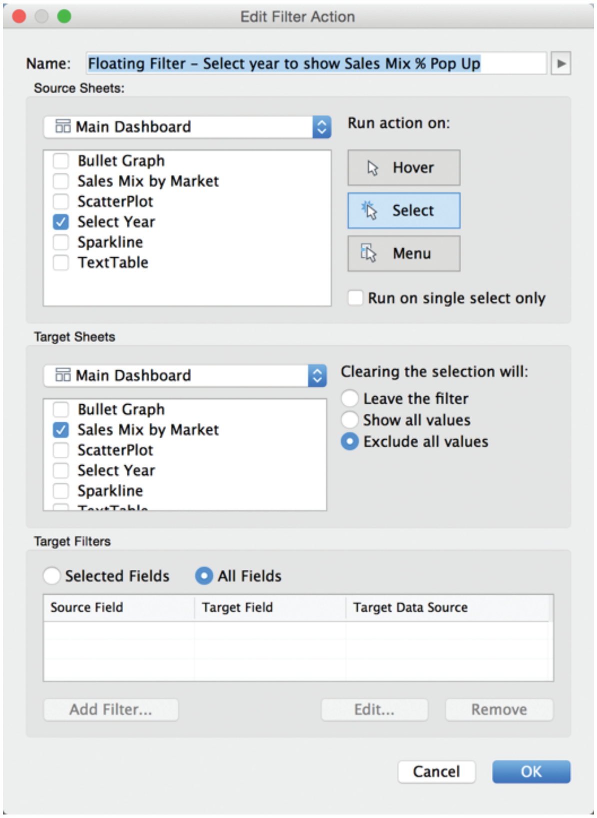


FIGURE 8-87 *Define the action.*

FINISHING THE TITLES IN THE MAIN DASHBOARD

To put the finishing touches on this dashboard, you need to add some small instructions to the dashboard (see Figure 8-88 for suggested additions).

As you can see in the figure, instructional text was added to the main dashboard title and to the title for the bullet graph. The font style (italics and brown color) is consistent with all of the instructional text in this workbook. If you consistently style instructional text for your dashboard titles and tooltips, your audience will be able to quickly identify these elements in your designs.

Next you learn how to use packaged workbooks to share your workbooks with others who do not have access to Tableau Desktop or Tableau Server licenses. This is facilitated through Tableau's free Tableau Reader desktop software.

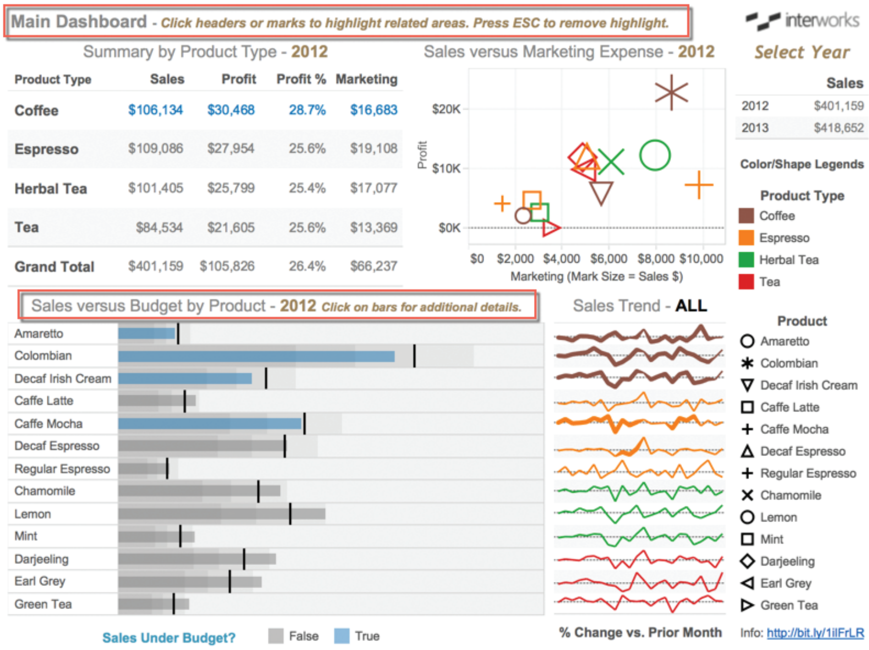


FIGURE 8-88 *The finished dashboard*

SHARING YOUR DASHBOARD WITH TABLEAU READER

The most secure way to distribute Tableau workbooks is via Tableau Server. But when you want to share a workbook with someone who doesn't have licensed access to Tableau Server or Tableau Desktop, Tableau Reader provides a free alternative.

Distributing content with Tableau Reader requires that you save the Tableau workbook file as a packaged workbook. Tableau Packaged Workbooks (`.twbx`) require local file sources, such as the following:

- Excel
- Access
- Text files (`.csv`, `.txt`, and so on)
- Tableau Data Extract files (`.tde`)

If the data source for the workbook you want to share with Tableau Reader comes from a server-based database (SQL Server, Teradata, Oracle, and so on), you must extract the source data first—saving extracted data as a Tableau Data Extract. Then save the workbook as a Tableau Packaged Workbook.

SECURITY CONSIDERATIONS FOR PUBLISHING VIA TABLEAU READER

Tableau Reader is intended to make your workbooks available to anyone—even those that do not have a Tableau licensed product. There are security considerations that you should be aware of when you distribute workbooks this way. Do not rely on filters to shield sensitive data that is included in the data sources used in the workbook. Tableau packaged workbooks are like Zip files. They can be unpackaged, which will expose the data source file.

If your data source includes sensitive information, you can use The Filters option when you create the data extract to exclude the sensitive data from the extract file. Figure 8-63 shows the Extract Data dialog box that is accessed by pointing at the data source (in the data window) and right-clicking.

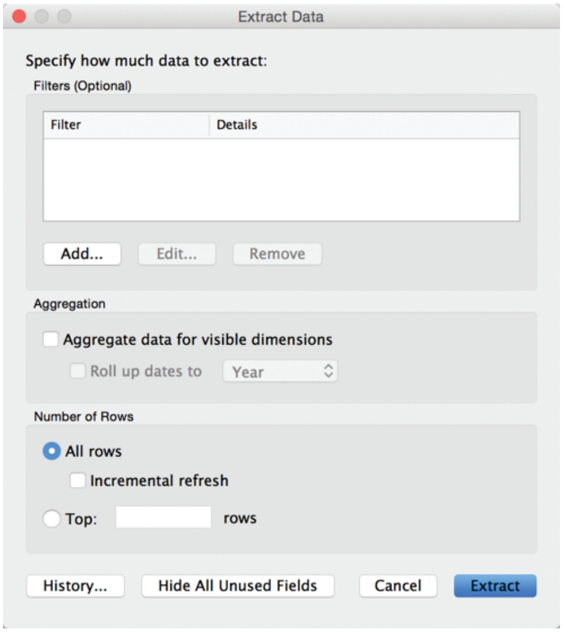


FIGURE 8-89 *The Extract Data dialog box*

One way to exclude information when creating the extract is to exclude data by filtering. You can also use aggregation to reduce the granularity of the data included in the extract. For example, selecting “Aggregate data for visible dimensions" aggregates the extract file so that it will include only data to support the visualizations in your workbook. In addition, any fields that you hide in the data window will not be included in the data extract.

Excluding sensitive information from the data extract file allows you to control the risk of data loss caused by unauthorized distribution of proprietary data.

USING THE TABLEAU PERFORMANCE RECORDER TO IMPROVE LOAD SPEED

Distributing content that loads fast and responds quickly to query requests is one of the most critical aspects of dashboard design. A slow-loading dashboard will not provide a good user experience.

Tableau provides a built-in tool called the Performance Recorder that provides detailed information about your workbook's performance characteristics. This tool analyzes Tableau's log files and builds a Tableau Workbook that analyzes the key performance attributes of your workbook.

To use the Performance Recorder, start Tableau. Go to the Help menu and select the Start Performance Recording option you see in Figure 8-90; then open the workbook that you want to analyze.

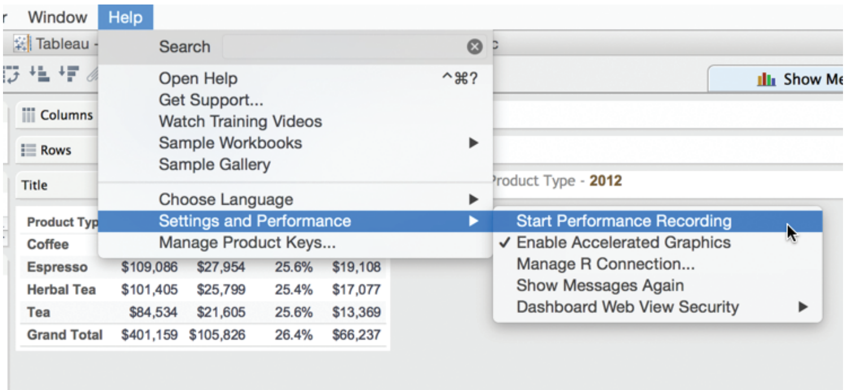


FIGURE 8-90 *Starting a performance recording*

When you have the workbook open, refresh all of the worksheets; use as many different actions, filters, and highlights as you can so that queries are generated and visualization are rendered. When you are finished, return to the Help menu and stop the Performance Recorder. Tableau will generate a dashboard that looks like Figure 8-91.

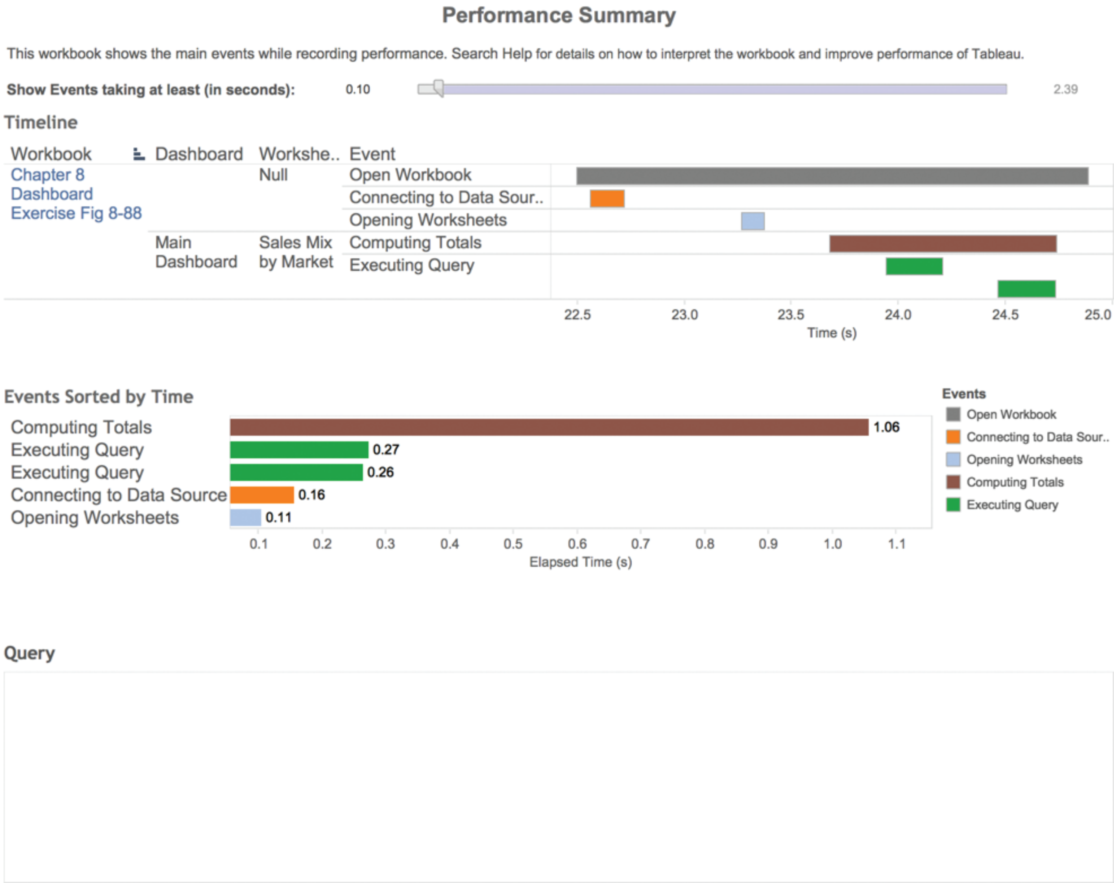


FIGURE 8-91 *Performance analysis workbook*

The dashboard generated by the Performance Recorder gives you information on the data connection, queries, and rendering speeds. If you have a workbook that loads slowly, the Performance Recorder gives you the ability to find speed leakages and test corrections by comparing updated performance recordings against your original record.

SHARING DASHBOARDS WITH TABLEAU ONLINE OR TABLEAU SERVER

Tableau offers a cloud-based option called Tableau Online. This service provides a low-cost alternative to sharing workbooks to licensed users of the service. Tableau Server is a self-managed solution that can be maintained inside or outside of your organization's firewall. Workbooks are published to Tableau Online or Tableau Server. People consuming the workbooks are granted access to them by a designated administrator that controls security.

The process for publishing workbooks to Tableau Server on Tableau Online is similar. Once the workbook has been published, authenticated users are able to access it using a web browser. See Chapters 11 and 12 for more details on Tableau Server and Tableau Online.

Over the past few years, tablet computers have been growing in popularity. Tableau has made producing workbooks for table consumption a seamless experience for the designer.

In the next chapter, you learn about designing workbooks for consumption in a browser and on tablet computers, such as iPad or Android devices.

NOTES

1. Robert McKee, *Story: Substance, Structure, Style and the Principles of Screenwriting* (New York: Regan, 1997).
2. Stephen Few, *Show Me the Numbers: Designing Tables and Graphs to Enlighten* (Burlingame, CA: Analytics, 2012), 79.
3. Edward Tufte, *The Visual Display of Quantitative Information* (Cheshire, CT: Graphic), 1983.

CHAPTER 9

Designing for Mobile

Just as the personal computer replaced the mainframe and the laptop replaced the personal computer, even more mobile devices will eventually replace the laptop. This trend toward smaller and more powerful devices means more dashboards and visualizations will be consumed on mobile devices.

Leading technology research firms Gartner and International Data Corporation have reported on an explosion of growth in mobile devices—cell phones, and more recently tablet devices. In November 2012, Gartner reported that "821 million smart devices will be purchased worldwide in 2012; sales to rise to 1.2 billion in 2013."[1]

Gartner also forecasts tablet purchases by businesses will more than triple from 13 million units in 2012 to over 53 million units in 2016. And they think tools such as Tableau will play an increasing role in the production of mobile applications: "By 2018 more than half of all B2E mobile apps will be create by enterprise business analysts using codeless tools."[2]

Clearly business use of tablets is expanding rapidly; analysts and analytical tools will be a large component of that adoption.

This trend is echoed in the Business Information (BI) world with increasing numbers of people using mobile devices (tablets and smartphones) to consume data. Mobile deployment has become a key component of most successful Tableau implementations. The next section describes the physics of mobile data consumption, security considerations, usage patterns, and the design best practices for building dashboards for mobile consumption.

THE PHYSICS OF MOBILE CONSUMPTION

Mobile consumption of Tableau dashboards is a function of the Tableau Server, Tableau Online, and Tableau Public environments. Because Tableau doesn't store data on your mobile device, a few prerequisites must be in place to enable mobile consumption:

- Installing Tableau's native iPad or Android application is not required (but may be desirable). Alternatively, a standard mobile browser is necessary.
- Provide a server name and Internet address for the connection to Tableau Server.
- Select a username.
- Select a password.

Mobile consumption is a default capability of Tableau Server; no additional configuration is required to enable mobile access. It should be noted that mobile device usage patterns can differ from non-mobile consumption. Mobile users typically use more sessions of shorter duration than desktop users. The interesting result of this usage pattern typically means increased session counts at the server level as users exploit the "just-in-time" nature of mobile access to data.

SECURITY CONSIDERATIONS FOR MOBILE CONSUMPTION

Unlike the Tableau Reader or Tableau Desktop tools, Tableau's native apps for iOS and Android are completely Server-based. This means neither mobile tool downloads an interactive data file or workbook file onto the Tablet's physical storage. Beginning with V9.1 it is possible to download static images of sheet views and dashboards to a tablet for offline viewing. All interactive data and reports are accessed entirely through the web connection to Tableau Server.

If a user were to lose a tablet, the only Tableau-related information residing there would be information about workbooks (publisher, date modified, and name) rather than the sensitive data contained within the workbooks being accessed. Of course, any machine with access to secure information should be password-protected.

Because mobile devices typically exist outside of the corporate network, options for network accessibility should be considered before deploying Tableau for mobile consumption if they haven't already been separately addressed.

Whether the Tableau Server has been positioned inside or outside of the DMZ (a secure area in a private network providing access to authorized users from the public Internet) dictates what special procedures are necessary to enable mobile access. If the Tableau Server is not positioned within the enterprise DMZ, solutions to provide mobile access typically include virtual private networks (VPNs) to secure the logon, or they utilize third-party Enterprise Mobility Management Suites to secure mobile access to data.

Once access to the Tableau Server machine is established, users can view dashboards and reports with the same authentication protocols used for desktop browsing. Tableau supports single sign-on (SSO) through both SAML- and Kerberos-based authentication for mobile devices in the event those systems already exist within the enterprise.

Specific mobile permissions are not configurable at the Tableau Server level without duplicating accounts or servers. If a separate set of mobile reports is desired, this process is typically mediated through a separate server or as a function of proxy-based relays.

OFFLINE ACCESS

Versions 9.1 and higher of the Tableau Mobile application allow for "snapshotting" of reports. These snapshots are high-quality image (PNG) files that are stored on the tablet for offline viewing. These files are secured using the mobile OS's native security application. Organizations unwilling to allow storage of these images on mobile devices can disable snapshotting through Tableau Server's Sites configuration menus. Figure 9-1 shows the Web Authoring selection activated.

Offline snapshots are enabled by default and will be created for any dashboard or view a user enters into the app's "favorites." Figure 9-2 shows a dashboard favorite that has been stored as a snapshot.

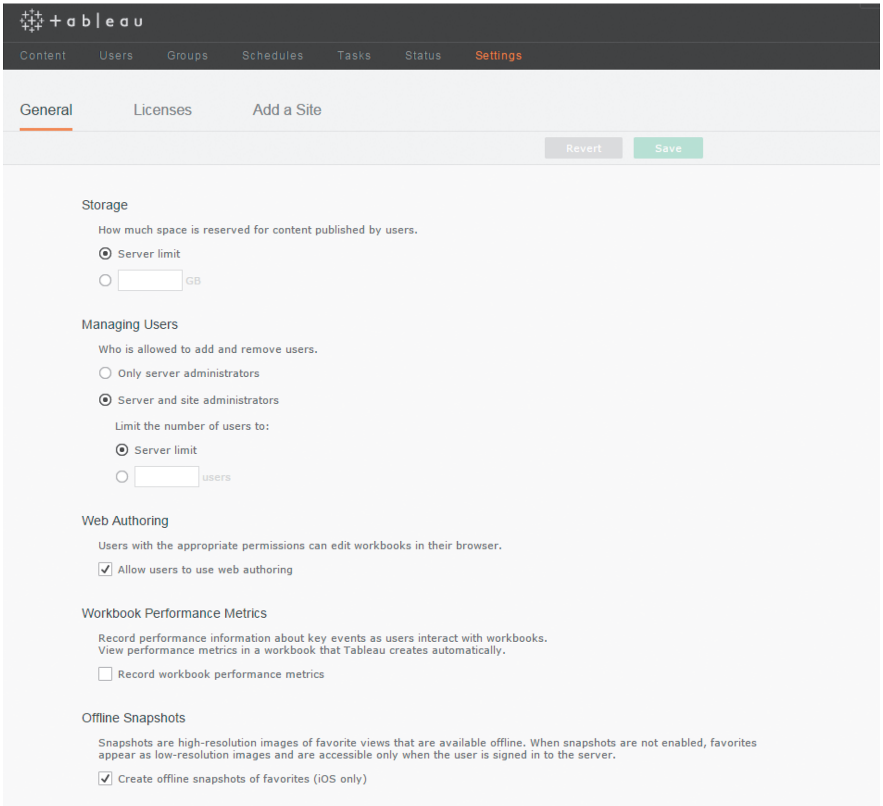


FIGURE 9-1 *Web authoring general configuration*

If you look at the top of Figure 9-2, you can see that you can toggle between a Snapshot or Live View tab if you are connected to the web.

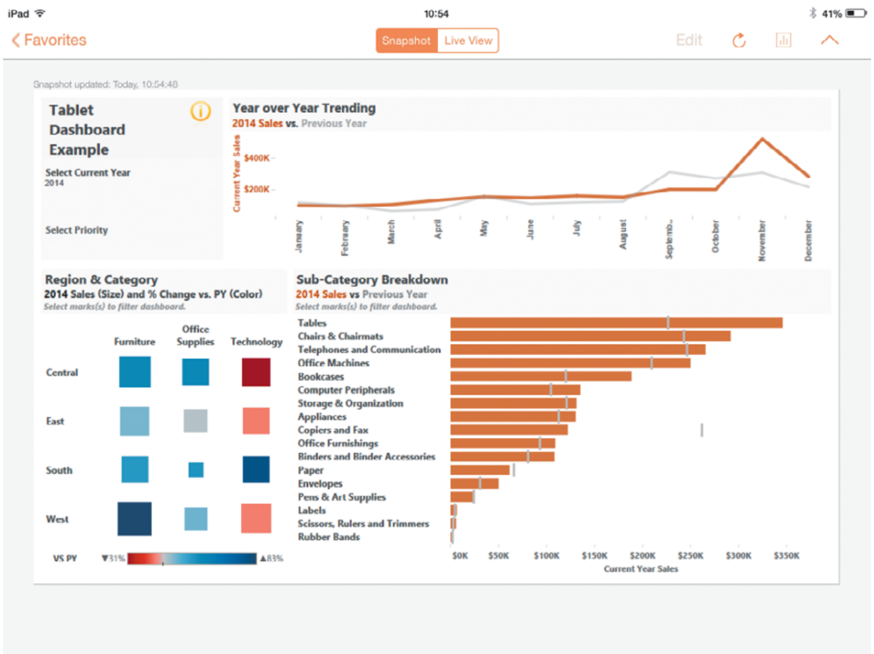


FIGURE 9-2 *Offline favorite snapshot*

TYPICAL MOBILE USAGE PATTERNS

Users accessing Tableau dashboards and reports from a mobile device typically have a different set of goals/intent for their experience versus those accessing data on the desktop. While this rule is not hard and fast, mobile users normally have a narrower scope and more defined use criteria.

JUST-IN-TIME USE

The Pew Internet Trust recently found that 86 percent of smartphone owners used their phone in the past month to make real-time queries to help them meet friends, solve problems, or settle arguments.[3]

The Pew report provided additional details regarding the activities mobile users engage in when using their portable devices. Do any of these items sound like needs your Tableau users might have?

- Solving unexpected problems that they or someone else encountered
- Deciding whether to visit a business
- Finding information to settle an argument regarding facts

- Looking up a score
- Getting up-to-the-minute information

All of these activities are supported by Tableau's mobile environment. If users are able to satisfy these Just-In-Time needs, they can seamlessly integrate that information into their daily activities.

MOBILE DESIGN IMPLICATIONS

Just-In-Time use has significant implications for dashboard and report design. The mobile market is not homogeneous. Do not assume all of your mobile users will have the same intent. But you can make some reliable assumptions about mobile information consumers and their desire for Just-In-Time information.

Mobile users want the most up-to-date information possible for asking questions and solving problems. Mobile users are more likely to be looking for a specific answer to a specific question, rather than embarking on a multi-hour session of complex analysis. This tendency should inform your dashboard designs for mobile consumption.

These requirements imply that mobile dashboards need to be more focused on specific areas and answering the kinds of questions that arise most frequently.

DESIGN BEST PRACTICES FOR MOBILE CONSUMPTION

Mobile device screens are obviously smaller than personal computer monitors, and your input method will be less precise than a mouse pointer. These differences were carefully considered when Tableau enabled their products for mobile usage. You also need to think about how your dashboard designs need to be adapted for this kind of consumption. When developing for a mobile environment, consider the following:

- Lack of a control button and no multi-select through Ctrl+click.
- The hover action is not available as an action trigger.
- The finger for pointing and clicking versus a mouse pointer.
- Zoom via a pinch/expand gesture in place of function keys or a mouse scroll wheel.

These differences have significant implications for your mobile dashboard designs. The emergence of mainstream touchscreen-based operating systems including iOS, Android, Windows, Smart Tables, and Smart Screens means many of the design criteria specific to mobile devices will soon be applicable to a much larger user segment.

DESIGN IMPLICATIONS RELATED TO SCREEN RESOLUTION

Consumer display resolution is a primary consideration for dashboard design, regardless of the consumption environment. Because there are fewer possible resolutions available for mobile devices, it is easier to create “table-friendly” designs. Fixed dashboard sizing—desirable for mobile designs—also provides performance advantages at the Tableau Server caching level. As with all dashboard designs, it is best to err on the side of caution and size your view for the lowest resolution anticipated. This ensures that information consumers won’t be forced to scroll to see all of the content. Tableau has predefined dashboard resolutions for mobile devices. You can also define any resolution required should your needs not fit industry-standard sizes.

BEST PRACTICES FOR MOBILE DESIGN

Designing for mobile devices is similar to designing for personal computers. Many of the best practices in designing for the PC apply. There are some additional allowances that have to be made because of the smaller screen size:

- Design for a specific orientation.
- Consider the limits of finger navigation.
- Reduce the number of worksheet objects displayed.

DESIGN FOR A SPECIFIC ORIENTATION

The best dashboard designs for mobile devices should be optimized for a specific design. Most of the tablet dashboards that I’ve seen are designed for landscape mode viewing; however, what is most important is that you commit to an orientation and build the dashboard for viewing that way. Tableau’s dashboard worksheet includes two predefined orientations for tablets—one for landscape mode (1020 × 625 pixels) and another for portrait mode (764 × 855 pixels). You can tweak those values to fit any resolution needed by using the Exactly mode when creating your design. This selection allows you to define specific pixel height and width if the mobile devices you’re designing for don’t conform to the default values.

CONSIDER THE LIMITS OF FINGER NAVIGATION

The primary interaction medium will be users’ fingers, which do not have the precision of a mouse. If a dashboard design feature is actuated through a filter or highlight action, ensure that the selection options are large enough for users to easily select without accidentally hitting a neighboring point. Nothing makes for a grumpier tablet user than one waiting for an unnecessary filter to render.

To avoid this trap, design dashboards with one of two alternative navigations. Choose heat maps, bar charts, highlight tables, or bullet charts to trigger actions. These provide discrete layout boundaries and preclude overlapping or closely spaced marks. Conversely, scatter plots have continuous axes that typically produce clusters of overlapping marks. This would not be a good choice for a filter action if precision is required. Figure 9-3 illustrates the point.

FIGURE 9-3 *Chart style and finger navigation*

The scatter plot on the left of Figure 9-3 includes many closely spaced or overlapping marks. This makes it a poor tablet-based action trigger because it is nearly impossible for the user to select a particular mark unless it is one of the outliers that are not included in the cluster.

The heat map on the right side of Figure 9-3 has marks that are regularly spaced. No mark is too small to provide a good click target. The heat map not only communicates data effectively but also provides for easy action triggering. To ensure that the smallest values were large enough, the mark size was edited—increasing the minimum mark size to avoid all possibility of individual marks being too small to click.

Tableau recently updated their application to support multi-select by tapping, holding, and then circling marks. Figure 9-4 shows six marks in the Region & Category heat map that have been selected.

FIGURE 9-4 *Multi-select by tap and hold action filter*

The selected marks in Figure 9-4 could be used to trigger an action to filter the other charts in the dashboard. Tableau has also tuned quick filters by changing their behavior, making them automatically expand into versions that are more easily selected using your finger on mobile devices. This doesn't require any special effort when you design the dashboard. Tableau detects the consumption environment and changes the design of the quick filter automatically.

These design differences provide a mobile-optimized interaction, though it does slow the experience slightly vis-à-vis consumption on a PC. For example, three selections are required to initiate a mobile quick filter—one click to activate the Quick Filter dialog box, another to pick a value, and a third to return to the dashboard.

REDUCE THE NUMBER OF WORKSHEETS BEING DISPLAYED

Because of the reduced screen size of mobile devices, it is best to use no more than three worksheet objects in a dashboard. One of those may need to be a very small text table limited to a single measure. Designs with too many worksheet objects are generally difficult to see.

A TABLET DASHBOARD EXAMPLE

The following example was created with the Superstore sample data. The dashboard has three primary data visualizations, one quick filter, one parameter control, and a filter action triggered by selections made in the heat map. You'll also see how a small text table with a shape can be used to hold more detailed instructions within a tooltip. Figure 9-5 shows a sales dashboard that is designed to be consumed using a tablet in landscape orientation.

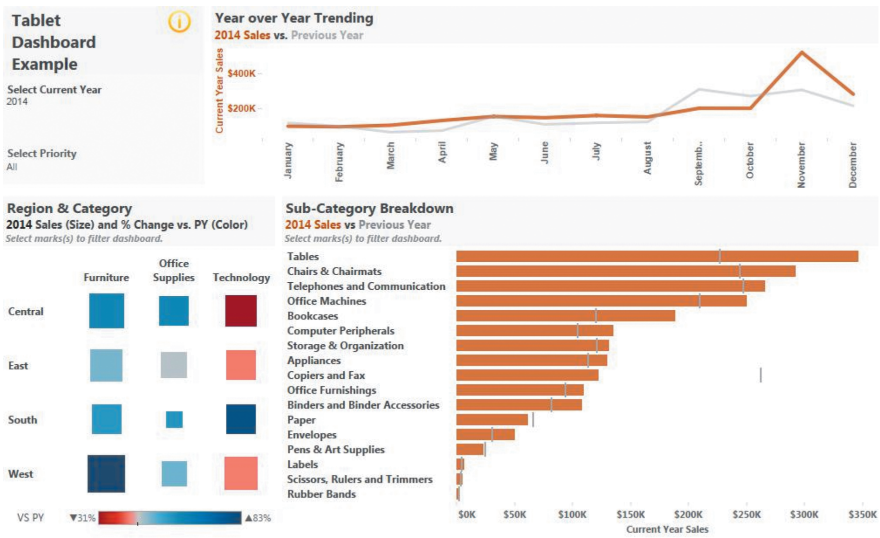


FIGURE 9-5 *Tablet dashboard in landscape mode*

The dashboard in Figure 9-5 utilizes best-practice techniques. Using three data visualizations, the top-line chart displays current- and previous-year sales using color to distinguish each year. Region and category are broken down in a heat map that displays current sales with a secondary measure of change from previous year sales denoted by color. The bullet chart breaks product categories down further into subcategories, comparing current and previous year's sales. Utilizing both a heat map and a bullet chart provides discrete separation that will allow the user to filter using an action executed by selecting marks from either visualization. A parameter control allows the user to change the "current" benchmark year, enabling self-service comparison across many possible years. A quick filter allows the user to filter the entire dashboard by order priority, a fourth dimension, while conserving space.

Figure 9-6 shows how the slider-type parameter control pops out while using the Select Year parameter—making it easier to select the appropriate year. After the selection is made, the filter will collapse to its former size.

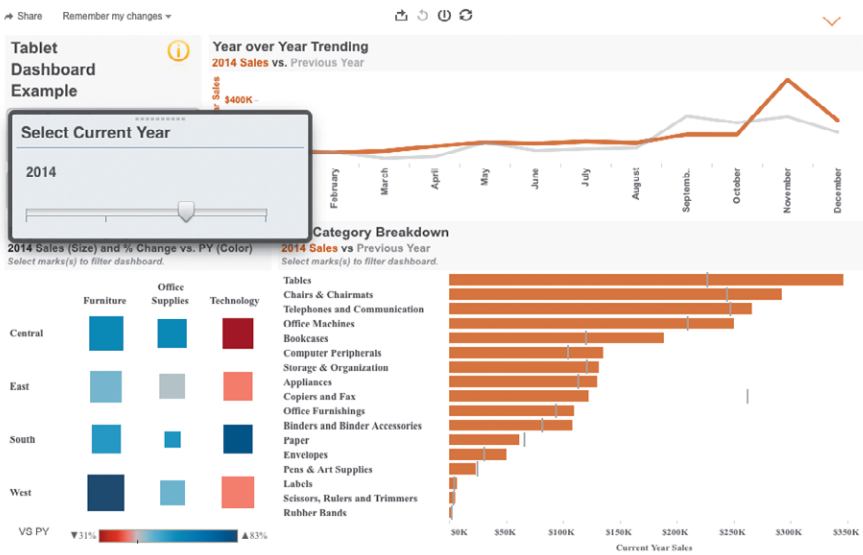


FIGURE 9-6 *Expanded year parameter control*

The order priority is changed using a quick filter that pops out to allow multiple selections, as you see in Figure 9-7.

A filter action placed in the heat map allows the user to filter the dashboard. As you can see in Figure 9-8, selecting a mark also causes a tooltip to display related details.

In Figure 9-9, a small text table using a question mark shape contains navigation instructions.

When the user points at the light bulb image, a tooltip is displayed that contains detailed instructions.

FIGURE 9-7 *Select Priority quick filter*

FIGURE 9-8 *Filter action and tooltip*

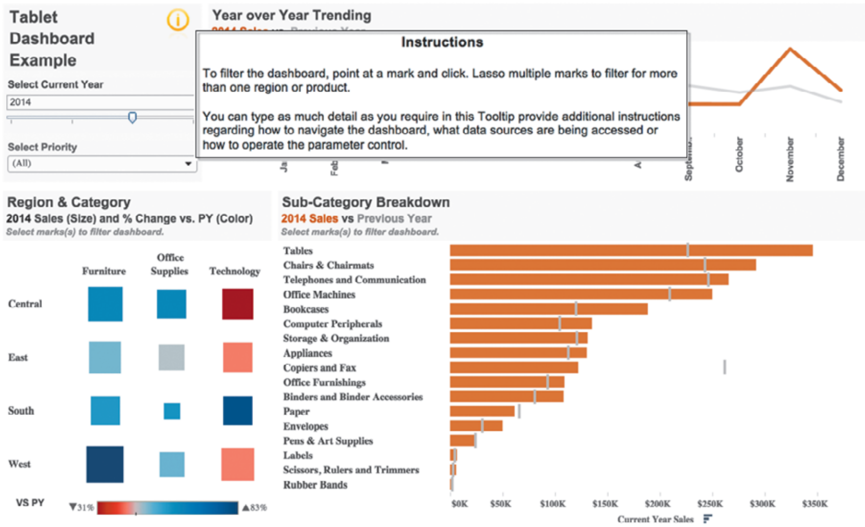


FIGURE 9-9 *Small instructions near the work*

For someone needing timely information regarding sales by product line, this dashboard is an easy-to-navigate, Just-In-Time environment. The dashboard provides details in the worksheet object titles that change based on the filter and parameter selections made. The portions of the titles in a bold orange color confirm those selections (see Figure 9-10).

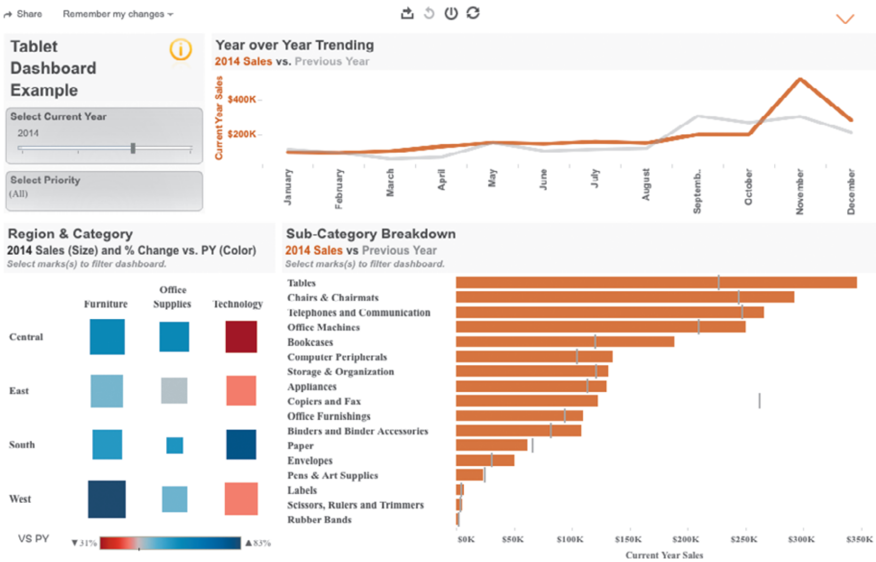


FIGURE 9-10 *Titles confirm filter and parameter selections*

This mobile dashboard is designed for quick response. It loads and filters quickly and is easy to read because it is designed for answering a specific set of questions related to sales, growth, and the product line.

MOBILE AUTHORING AND EDITING

Mobile authoring and editing are native to the Tableau Mobile application. These capabilities are also available through browsers on a mobile device. The mobile-specific editing capabilities closely mirror those of the standard web editor, though the user interface has been adapted to support touchscreen interaction and the lack of a mouse. The same Tableau Server permission settings that enable or disable web editing apply for mobile editing. Chapters 11 and 12 cover the setup options in detail. Figure 9-11 shows the mobile editing interface on a tablet.

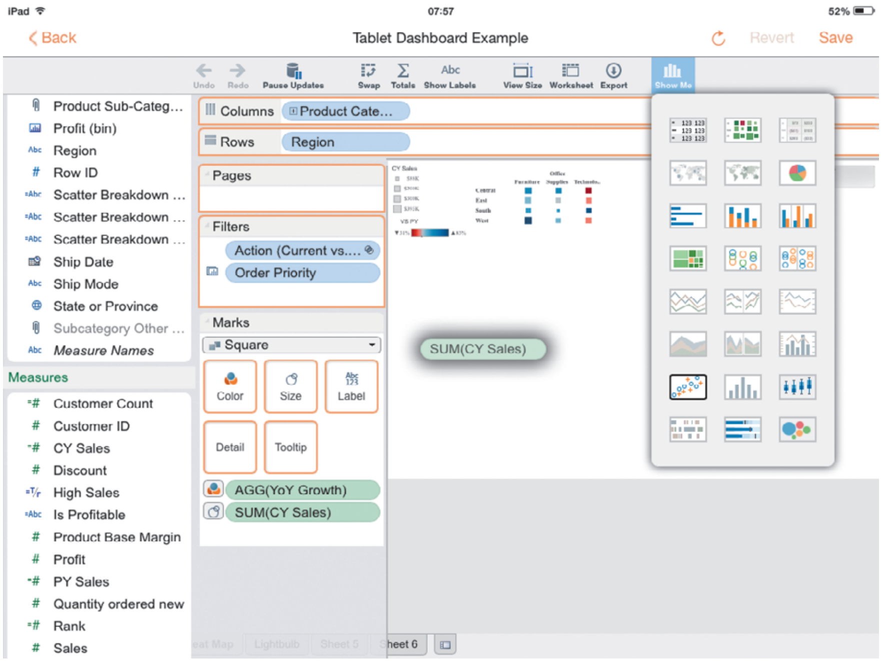


FIGURE 9-11 *Mobile editing interface*

Chapter 12 includes additional details on Tableau's web editing menu options in Server. These controls are identical to what you see displayed in Figure 9-11.

A NOTE ON PROJECT ELASTIC

During Tableau's 2014 user conference, Tableau's management introduced a new product under development. Project Elastic will be designed for ad hoc mobile exploration of self-contained mobile datasets. This will be a completely new standalone product. Initial versions will only allow for ad hoc exploration of local datasets, not live database connections or Tableau Server data sources.

I've seen the early alpha product demonstrations and recently had discussions with one of the lead developers involved. Later this year, I expect to see a working prototype for testing. No firm date has been announced for this product. In future releases of this book, I plan to cover Project Elastic in detail.

In the next chapter, you learn how to use Tableau's Story Point feature to share insights with others and give information consumers the ability to step through specifically annotated visualizations in a predetermined order.

NOTES

1. "Gartner Says 821 Million Smart Devices Will Be Purchased Worldwide in 2012; Sales to Rise to 1.2 Billion in 2013," Gartner, Inc., accessed July 27, 2013, `http://www.gartner.com/newsroom/id/2227215`.
2. "Gartner Say By 2018, More Than 50 Percent of Users Will Use a Tablet or Smartphone First for All Online Activities," Gartner, Inc., last modified December 8, 2014, accessed July 25, 2015, `http://www.gartner.com/newsroom/id/2939217`.
3. "Pew Internet: Mobile, Highlights of the Pew Internet Project's research related to mobile technology," by Joanna Brenner, last modified June 6, 2013, accessed July 27, 2013, `http://pewinternet.org/Commentary/2012/February/Pew-Internet-Mobile.aspx`.

CHAPTER 10

Conveying Your Findings with Stories

When Tableau first introduced Stories, I didn't fully appreciate their value. A few weeks later, I was asked to judge a data visualization contest at Facebook's campus in Menlo Park. The contestants were provided multiple datasets and were given 60 minutes to analyze one of them. Each team had 60 seconds to present their findings to the judges.

The winning team presented a cogent set of findings with a compelling storyline. Even more impressive—each person on the winning team contributed 20 seconds of verbal narrative supported by Tableau Story Point visuals simply by clicking through descriptive captions. That's when the real value of Stories became apparent to me.

Perhaps the story could have been told with a series of visualizations and dashboards, but each individual Story Point made it easier for the team to communicate quickl0,y and clearly annotating each Story Point highlighted important facts without the need for verbalizing the finding. It was an impressive presentation that didn't feel cramped by the 1-minute time limit.

TURNING ANALYSIS INTO INSIGHT

Stories are a type of sheet made up of dashboard or worksheet views that have been sequenced to support a guided analytic tour of your subject. These views are expressed as Story Points and introduced through navigation captions that include descriptive text provided by the author.

Creating a story is similar to building a dashboard. You drag the desired source material into the sheet. Each Story Point can contain one worksheet or dashboard. You provide context to the sheet by adding text to captions contained in a navigation bar across the top of the view. The captions can be rearranged, edited, or deleted. Further emphasis can be added within the view through annotation, descriptions, filtering, or any other method supported within worksheets or dashboards.

You have probably used PowerPoint slide decks to tell stories and convey a priority or to convince your colleagues to buy into an action plan. Many Tableau users have embedded static Tableau images into PowerPoint slide decks in the past. Stories allow you to skip that step entirely while supporting your findings with fully interactive views of your data.

Stories can be a fun way to explore your favorite sport or hobby. You can explore the U.S. Supreme Court or pivotal moments in American history. Have fun and get creative. Building Stories about topics that interest you away from work will advance your skills that you apply on the job.

To build effective Stories, your process should include the following:

- Deciding what type of story you want to tell
- Building evidence in Story Points to support your thesis
- Organizing your arguments in a logical sequence
- Adding descriptions and annotations that highlight the important evidence
- Providing supporting details

Similar to how you would write a position paper, create an outline before you start putting your workbook together. Know where you're going to start, know the journey you're taking your audience on, and plan your finish so that you conclusion is supported with facts.

BUILDING A STORY

Once you know what you need to convey, creating a Story Point deck is easy. Your worksheets and dashboards should include relevant filters and actions for filtering and highlighting the content just as they would if you were performing your own analysis of the data. Remember that after content is placed in the Story, you must navigate back to the source worksheet or dashboard to make design changes. Thinking through your design before this stage will reduce your workload when polishing your Story Point deck at the end. These are the key points to remember:

- The Story Point main title is static.
- Each Story Point tab can hold one worksheet or one dashboard.
- Set the Story Point size pixel width and height to fit your presentation environment.

To create a Story, click the New Story tab to the right of the Sheet tabs, as shown in Figure 10-1.

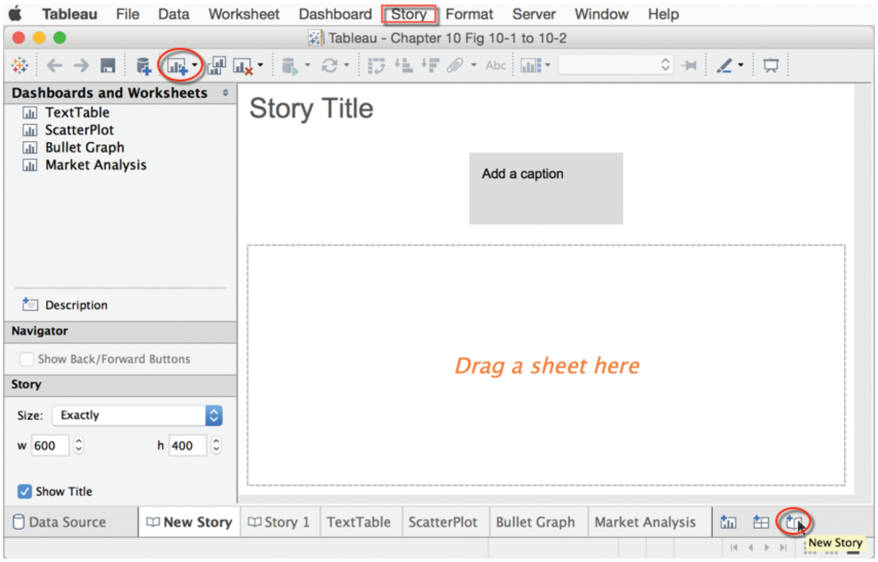


FIGURE 10-1 *Making a Story*

Alternatively, you can use the Story menu or Story icon to start.

THE STORY WORKSPACE

The left side of the Story Workspace in Figure 10-1 looks similar to a Dashboard workspace. Worksheets contained in the workbook are visible. Dashboards would also appear in that space if there were any in the workbook. The Description icon is used to add floating text boxes to the view. The Navigator check box lets you turn on (or off) arrows that enable scrolling of the Navigator bar. Figure 10-1 includes only a blank Caption. When more than one Caption is in view, these arrows will appear by default. The Story area allows you to define the width and height of your story. The Show Title check box appears at the bottom of the pane. Defining size and adding a title are consistent with how these features work in Dashboards.

The Story workspace allows you to define a couple of critical items before you begin to drag dashboards or views into the workspace:

1. Make the appropriate changes to the Story size you see near the bottom left of Figure 10-2.

2. Decide if you want a static story title at the top. In most cases, the dashboard or worksheet title will serve your needs. If you do want a Story Point title, make sure the Show Title option is selected.

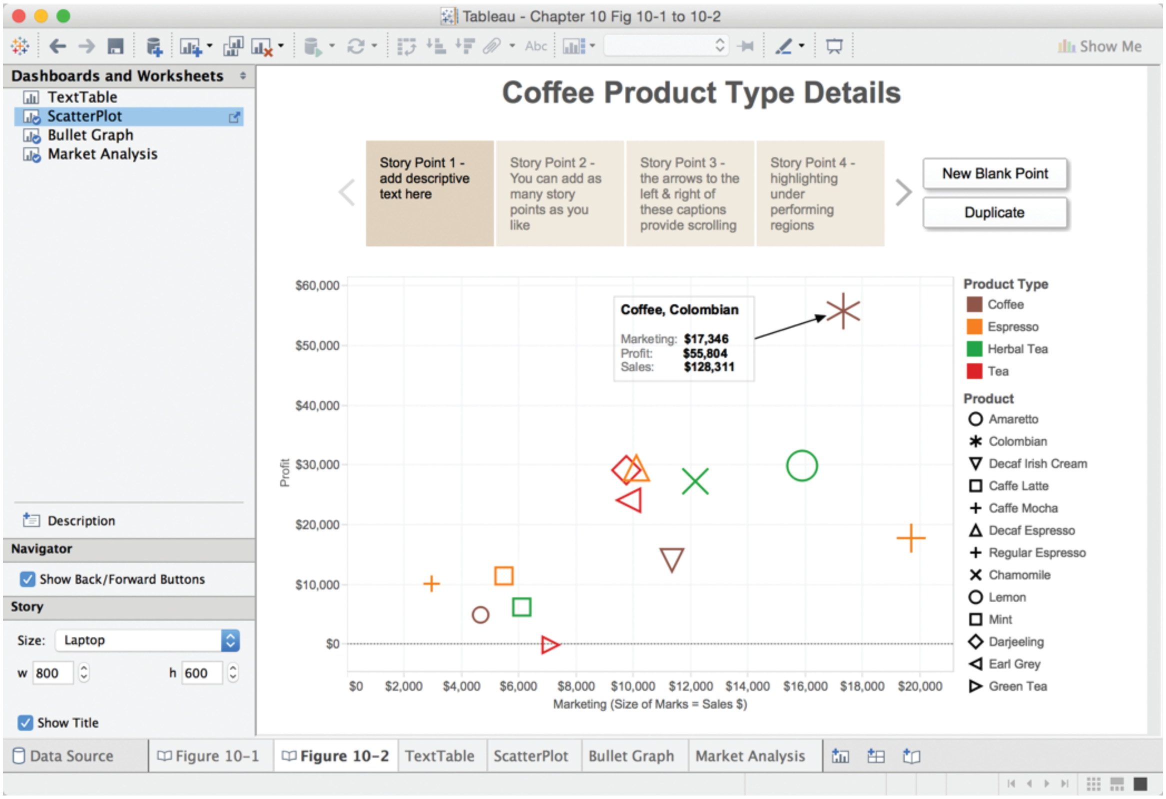


FIGURE 10-2 *The Story workspace*

The laptop-size pixel height and width are being used in the Story show in Figure 10-2. The Show Title option is selected, and the title has been customized by double-clicking within the title area, typing, and then formatting the text. Adding worksheets or dashboards into a Story is done by dragging them into the Story workspace. Notice that there are four Navigator buttons at the top of Figure 10-2. These buttons contain captions that have been edited to include customized text. Also notice that the Navigator button arrows are now visible because more than one button is in view and the Show Back/Forward Buttons option is checked in the Navigator area of the Dashboards and Worksheets pane.

The Navigator button that contains the caption text starting with Story Point 1 is currently active. Notice that the three other captions are colored with a lighter shade of brown. This color scheme is defined using the Format ➢ Story menu option.

You can add additional Story Points in three ways. To the right of the Navigator buttons, two of the following three options are displayed:

- New Blank Point
- Duplicate
- Save as New Point

Selecting New Blank Point creates a new Navigator button with a blank caption and an empty Story workspace that is ready to accept a new worksheet or dashboard. The Duplicate option will create an exact copy of whatever Story Point is active with a blank caption. Save as New Point only appears as an option if you have modified an existing Story Point that you want to save under a new Navigation button. Building Story Points is very easy to do if you are familiar with making worksheets and dashboards. If you want more details on the Story workspace tool options, go to the Help ➢ Open Help menu option and read the Stories section of Tableau's online manual.

A STORY EXAMPLE

The example that follows uses a Story created by InterWorks consultant Robert Rouse. The Story was embedded in a blog post that you can find at `https://www.interworks.com/blog/rrouse/2014/10/15/every-pitch-2014-mlb-season-visualized-tableau`.

The data source used for the workbook includes all of the pitches thrown by every pitcher in Major League Baseball for the 2014 season—that's 765,122 pitches. The type of pitch, speed, and placement of the pitch are also included.

After a worksheet or dashboard is added to a Story Point, you can edit the caption text at the top of the view. In Figure 10-3 you see the first Story Point caption has been edited to say "All the pitches of the 2014 season."

Robert's Story includes seven different Story Points. The Navigation buttons have been formatted to match the color and style of the dashboards used to create each Story Point.

FIGURE 10-3 *MLB pitches 2014*

FORMATTING STORY POINTS

The Start View Dashboard that is the source of the Story Point displayed in Figure 10-3 includes a lot of custom formatting. The text boxes at the top and the bottom were added in the source Dashboard used in the Story Point.

In Figure 10-4, the Eephus pitch is highlighted. The Eephus pitch isn't a common pitch type, and even avid baseball fans might not be familiar with the term. This was added to the view in the Story workspace by dragging the Description object from the Dashboards and Charts pane and positioning it at the top of the view.

A second Description was added in the bar chart area that includes dynamic data elements for average pitch speed and the percentage of total pitches thrown. This is highlighted in the lower-left area of Figure 10-4. You can see that the Eephus pitch was only 0.05 percent of the total pitches thrown in 2014, and the average speed of the pitch was 66 miles per hour.

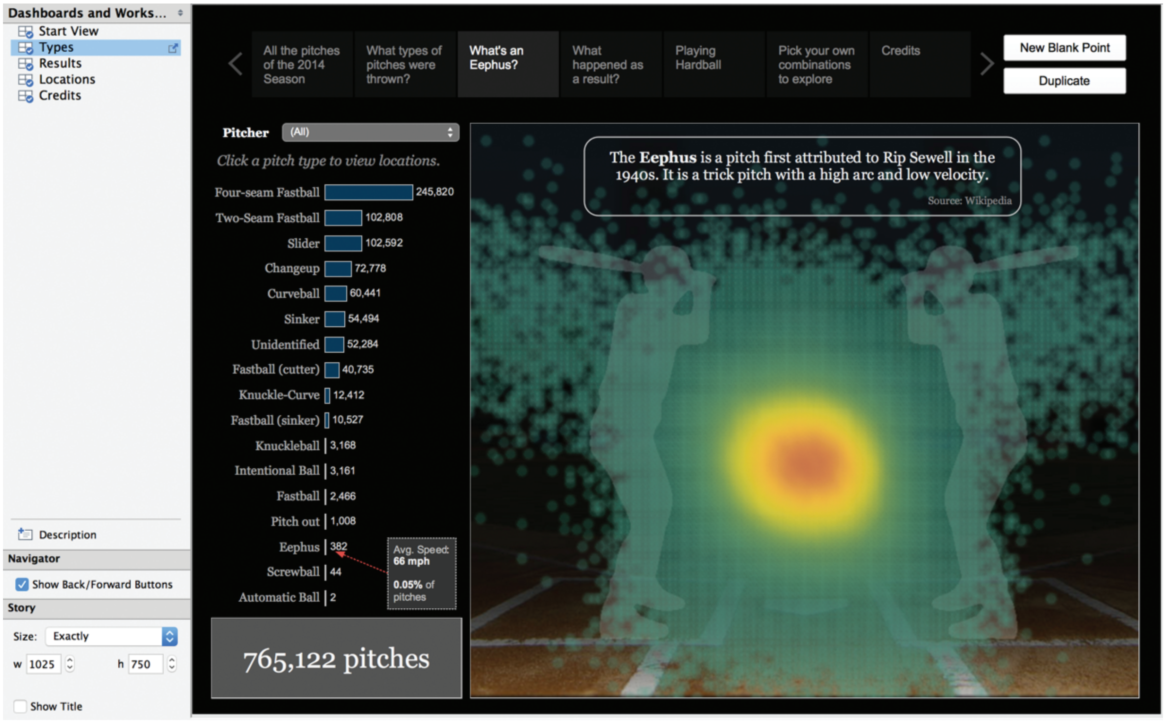


FIGURE 10-4 *Adding Descriptions to a Story Point*

To open the Format Story pane, select Format ➢ Story. The Story Point formatting for Figure 10-4 is shown in Figure 10-5.

As you see, the Format Story pane includes options for controlling shading, font, alignment, and borders. By combining design elements from your source worksheets and dashboards with additional (specific) design Descriptions, Captions, and formatting in your Story Point, you can create stylized Story Points with helpful descriptions and dynamic data to highlight important information. You can download the workbook containing Robert's Story Point deck in the Chapter 10 folder in the book's companion website.

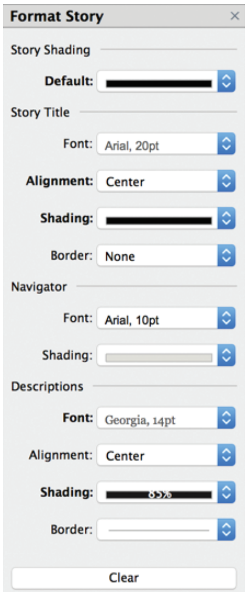


FIGURE 10-5 *Format Story pane*

SHARING YOUR STORY POINT DECK

Once you've completed your Story, you share it in the same way you publish any workbook view—by publishing to a secure Tableau Server location or to Tableau Public. The Story Point deck used in this chapter was published to Tableau Public and embedded in the blog post referenced earlier in the chapter. It appeared on the InterWorks blog November 13, 2014.

You can also share a Story Point by exporting the view as a packaged workbook and sharing the workbook with other people via Tableau Desktop or Tableau Reader. Keep in mind that if the dashboards and worksheets used in your Story Point change, the data used in the Story Point visualizations change as well, but the Story Point titles do not unless they are part of the underlying worksheet or dashboard. If you decide to publish Story Point decks to Tableau Server, make sure that your Description objects and annotations include appropriate information so that your audience can properly interpret the information. For this reason, Story Points are most appropriate for presentations that you are conducting live or for consumption in Tableau Server or Tableau Public using static datasets.

Finally, if you have to lead a discussion that relies on analysis you've created in Tableau, a Story can be an excellent tool for guiding your audience through the discussion. Using Presentation mode, you'll have a fully interactive slide deck that will allow you to respond to questions you may not have anticipated in the original presentation within your Story. Because Stories are connected to the source data you can filter, highlight an annotated existing Story Points to reveal new information. After your presentation, you can publish the workbook to Tableau Server for other people to review.

This completes Part I of the Tableau Desktop content. In Part II, you learn about how to install, manage, and automate Tableau Server.

PART II

SERVER

In this part

CHAPTER 11

Installing Tableau Server

The more you have in your cup, the more likely people are to want a drink.

SETH GODIN[1]

In the first ten chapters, you learned how to use Tableau to connect to data, analyze data, visualize data, build dashboards, and share stories. This chapter and the next two are about installing, managing, and automating Tableau's Server. Tableau provides three different tools for sharing information—Tableau Public, Tableau Online, and Tableau Server.

Tableau Public is a free cloud-hosted service aimed at bloggers, students, or data visualization enthusiasts who want to share their work publicly. It is not designed for enterprise environments that require data security. In fact, anything published on Tableau Public is freely available for anyone to download. This tool is generally not used in enterprise environments that need to control access to the information.

Tableau Online is another cloud-based information sharing environment (managed by Tableau Software) that provides data security without the need for installing any software or managing hardware. Your data is stored in a secure environment managed by Tableau Software. To start using Tableau Online, just sign up for the service and assign access to your users based on your security needs.

Tableau Server is for customers who need to control where and how the data is stored and managed. It can be installed on hardware behind your firewall or on cloud services that you contract with directly. You can license it by named user or based on the server hardware that it is installed on.

WHAT'S NEW IN VERSION 9?

Tableau V9 provides significant performance and usability improvements when compared to earlier editions of Tableau Server. The visual interface for both the server user and server administrator has been redesigned to provide more information at a glance. Server provides improved activity analysis via analysis dashboards that include information on view counts, user activity, actions by user, background task details, statistics for space usage, load times, and more.

Server's performance has been improved. Tableau Software published a Tableau Server 9.0 Scalability analysis.[2] This paper documents significant improvements in concurrent user scalability, query speeds, and reduced time-outs resulting in errors. These improved results do require increased investments in RAM, so you should consider investing in more RAM before you upgrade from V8 to V9.

InterWorks consultant Glen Robinson did comparison tests of Tableau Server 8.3 and 9.0 with Tableau Server deployed in the cloud on Amazon Web Services using two different CPU configurations.[3] His testing showed improved response times just under 3X using an 8-core CPU and 3.5X using a 16-core CPU. While results were different from those reported by Tableau Software, the performance improvements are still significant. Your results will be different, but you should expect to see better query response times even with increased user concurrency rates in Tableau Server 9 versus Server 8.3.

Tableau Server 9 also provides the administrator with more control over clustering in multi-node and high-availability environments. You can now set a preferred server for the Active Repository, which may be important if your server hardware is different for each of the servers in your cluster. For example, you can specify that the active repository run on the most powerful server in your cluster. See Glen's blog post referenced in the end notes to this chapter for more details. Tableau continues to provide more ways to secure your data through Secure Socket Layers, SAML, and Kerberos.

Tableau Server 9.1 also gives the administrator the option to schedule synchronization of all Active Directory groups. Synchronization can be customized hourly, daily, monthly, or on demand. This feature can be disabled by the server administrator. At the time of this writing, Tableau Server 9.2 is in beta release. Permissions locking features are being added to provide more control over Project view, interaction, and editing.

This chapter focuses on options you need to consider when installing or upgrading Tableau Server. Chapter 12 focuses on managing Tableau Server, and Chapter 13 discusses automating Tableau Server through the command-line tools and the enhanced SDI and API toolsets provided through Tableau Server Version 9.1.

REASONS TO DEPLOY TABLEAU SERVER

Most companies begin using Tableau by purchasing a few licenses of Tableau Desktop. While learning how to use the software, you will share your workbooks with other Tableau Desktop users in your company. You may also save your analysis in packaged workbooks so that you can share your workbooks with other team members using Tableau's free Reader product. Or you may have a few licenses of Tableau Online for providing access to mobile team members via the cloud.

As Tableau's value to your organization is proven, the number of people wanting Tableau visualizations and dashboards will grow. You have to scale to a larger number of analysts or information consumers. Some people may want to make their own (slightly different) versions of the reports you publish but don't require the facility that Tableau Desktop offers. Tableau Server allows you to share your work securely with a much larger number of people who will access your workbooks and dashboards via a web browser or tablet app. Tableau Server provides desirable features including:

- Data governance (security)
- Efficiency (sharing workbooks, data connections, and data extracts)
- Flexibility (consumption and editing options)

Server's architecture provides the flexibility to scale from a single box to large multi-server deployments. Tableau Server supports several different security protocols including Secure Socket Layers (SSL), Security Assertion Markup Language (SAML), and network authentication via Kerberos. Windows Active Directory can be utilized for user authentication. Tools are provided for setup and maintenance of access rights, scheduling, and notification. Downloading and installing Tableau Server normally can be done in less than two hours.

Tableau's growing partner network provides additional add-on tools for merging workbooks, style management, data source auditing, best practice analysis, and performance-tuning. Automation tools are available that allow you to create workflows for managing and monitoring growing enterprise deployments. Software development kits (SDKs) are available that allow you to create your own custom batch automation, determine data lineage, create dynamic parameters by binding data, and more. Given Tableau's rapid growth over the past several years, it's likely that this add-on market will continue to grow to address a wide variety of customer needs that are outside of Tableau's standard capabilities.

DATA GOVERNANCE

Securing proprietary or confidential data is not only a business need, but it can also be a legal requirement. Information managed by healthcare providers, insurance companies, and government entities is controlled by law. Businesses have a legal obligation to ensure private employee and customer data is kept confidential and secure.

Businesses must be concerned about the accuracy and consistency of the data being consumed by staff without being overly controlling. Tableau Server balances these needs well by supporting data governance best practices. It allows information technology staff to maintain control over data sources (providing a single version of the truth) while simultaneously providing information consumers with the ability to adapt reports to their own purposes—without the need for additional technical staff or needing to resort to creating new (unauthorized) data sources.

EFFICIENCY

Sharing reports is easy via Tableau's free desktop report consumption tool—Tableau Reader. However, this approach doesn't scale well and provides limited means for securing the underlying data. Updating desktop reports is easy but can be time-consuming if you have dozens of weekly reports to deliver. Tableau Server provides a secure environment for report consumption and can automatically update reports and inform users of new report availability via Server's subscription service. Administrators can monitor report consumption, server utilization, and performance.

Tableau Online provides similar benefits at a lower price point but requires that you publish Tableau reports outside of your firewall.

When the data source doesn't include all of the information desired, domain experts on your team can create starter workbooks that address aggregation needs, dimension grouping, and other particulars that are desirable to share with everyone using Tableau. Server facilitates this sharing by allowing users to publish the metadata via data source files through the server—saving everyone time and ensuring report consistency. When those data source files are modified, changes are automatically propagated to everyone using the published data source.

Personnel consuming reports don't need to install any software to view reports because everything is viewed via a web browser. Internet Explorer, Microsoft Edge, Firefox, Chrome, and Safari are all supported.

FLEXIBILITY

If you use data extracts, updates can be scheduled to run automatically at almost any time interval desired. Do not underestimate the level of demand that Tableau generates. Your deployment may quickly go from a few users to hundreds and then thousands. The number of reports will increase as well. Server provides users with an easy-to-navigate environment that allows them to ask questions and quickly get answers. It also provides administrators tools for managing and updating reports without the need for daily manual intervention.

Administrators can assign rights for publishing, consuming, and modifying reports. Interactive reports can be embedded into existing websites, and Tableau can pass through the security layer without requiring the user to re-enter login information. Authorized consumers can securely view and edit reports via the web on their desktop, laptop, and iOS or Android Tablet devices.

Tableau Server is a robust environment that provides technology managers with the tools to secure and maintain the environment while also providing information consumers with fast access to the information they need.

LICENSING OPTIONS FOR TABLEAU SERVER AND TABLEAU ONLINE

Tableau Server can be licensed in two different ways:

- Per-named-user basis
- Server core license

Core licensing provides unlimited access to any number of users. Pricing is based on the number of processor cores contained on the server(s) on which you deploy the software. Named-user licensing starts with a minimum of ten users. Core licensing requires an eight-core minimum. Although many factors can affect performance in a server deployment (hardware, network traffic, dashboard design), an eight-core configuration can support in the low hundreds of concurrent users.

Tableau Online is sold via a named-user license that requires a one-year commitment. You can start with a single license and add more as your needs grow.

DETERMINING YOUR HARDWARE AND SOFTWARE NEEDS

Tableau Server is a scalable system that is capable of meeting the demands of the most intense enterprise environments. Proper planning is an important first step before you settle on the appropriate hardware configuration and licensing options. At a minimum, you should consider the following details when planning your deployment:

- User count
- User concurrency rate
- Workbook complexity
- User locations
- Database locations
- Database size
- Extract usage—number and size

User count is easy to estimate because it represents the number of licensed users of Tableau Server that are able to make requests to the server. User concurrency rates represent the percentage of the licensed users that will be making requests at any single moment. For example, a deployment anticipating 1,000 licensed users with an expected concurrency rate of 10 percent implies that approximately 100 users will be active in the system at any moment. This is more difficult to estimate but tends to range between 2 and 10 percent of total licensed users. If you have an existing analytics system or web portal that is actively used for report distribution, do not assume that you will see similar usage levels in Tableau Server. In our experience, this is not a reliable. Well-designed, interactive Tableau dashboards increase server traffic when compared to legacy systems because Tableau is more popular with users.

Tableau workbook size and complexity can vary widely. For this reason, before you plan your Server environment, it is advisable to identify a core group of report designers, train them, and have them build some initial reports that can serve as a basis for planning. This typically doesn't require more than a month to accomplish and doesn't need to involve many staff. Not all requests made to Tableau Server are equivalent. Server will spend more resources to render dashboards with complex designs and large volumes of data than dashboards with simple designs and low record counts. Poorly designed dashboards are the most common cause of poor performance in Tableau Server.

If you have users in many locations or have database services deployed across multiple geographies, you may need to have a correspondingly larger number of Tableau Servers to support local demands if a central service isn't able to provide the desired responsiveness.

You must also consider the amount of data you have as well as the type of database sources you are using. Massive data or heavy demand, along with a database that wasn't designed for intense analytical loads, can create the need for shifting some of the analytical burden from the database to Tableau Server. This is accomplished by publishing Tableau Data Extract (`.tde`) files to Tableau Server.

NEW FEATURE: PERSISTENT QUERY CACHE

Beginning with Version 9.0, the query cache has been moved into its own process. This allows for more efficiency by sharing the cache between processes to increase the number of cache hits. This query cache is also now persistent, meaning that results in cache are maintained across restarts.

DETERMINING WHAT KIND OF SERVER LICENSE TO PURCHASE

If you don't require that your data and reporting be on your own network—behind your firewall—Tableau Online provides a convenient option. Tableau Online is a cloud-based version of Server. Tableau Software manages the hardware and is responsible for maintaining network performance. It is a good option if you are comfortable with Software-as-a-Service (SaaS) models and you don't have any legal restrictions preventing you from storing information this way. Your administrator of a Tableau Online is responsible for controlling access by setting permissions for publishing and viewing the data.

If your organization is unable to reside your data in the cloud outside of your firewall, Tableau Server's named-user licensing and core-server licensing allow you to directly control every aspect of Tableau Server's setup and configuration—inside or outside of your company's firewall. For most large enterprise customers, Tableau Server offers the most flexibility.

Tableau Server's named-user licensing is exactly what it sounds like—one license purchased per user, meaning that a license must be purchased for each individual user of the system. If there are ten distinct employees who need access to Tableau Server, then all ten of them must have a named-user license.

A question that many people ask is whether Tableau can be deployed on any kind of multiplexing device so that individual users can share the named-user license. The answer is no. Licenses are transferable, but this is not a practical way to split a single named-user license among an active user base. Named-user licenses are also referred to as Interactor licenses.

Core licensing allows customers to license Tableau Server by the server processor core—avoiding the purchase of licenses for specific named users. Core licensing provides greater flexibility, allowing for as many users as a server can support from a resource perspective. These licenses are typically sold in eight core multiples. Pricing for core licensing reflects the fact that a single core can support many users. It also provides the option of enabling a special guest account to enable unrestricted access to reports assigned by the administrator. The guest account must be enabled by the administrator.

The number of users you anticipate accessing the system typically determines which licensing model you choose. Smaller entities with low user counts typically find that named-user licensing provides a better value. Tableau Online will also appeal to this segment if externally hosted security is permitted. Large organizations with user counts in the hundreds typically find core licensing more cost effective.

In some cases, mixed licensing models might be desirable because hardware limitations imposed by the core licensing model can be alleviated through the selective use of named-user licensing and/or Tableau Online.

TABLEAU SERVER'S ARCHITECTURE

Tableau Server comprises many processes operating together. These may run concurrently, but typically all processes won't be running all of the time. These include:

- API Server (`wgserver.exe`)
- Application Server (`wgserver.exe`)
- Cache Server (`redis-server.exe`)
- Cluster Controller (`culutercontroller.exe`)
- Coordinator Service (`zookeeper.exe`)
- File Store (`filestore.exe`)
- Search & Browse (`searchserver.exe`)
- VizQL Server (`vizqlserver.exe`)
- Data Engine (`tdeserver.exe`, `tdeserver64.exe`)
- Backgrounder (`backgrounder.exe`)
- Data Server (`dataserver.exe`)
- Repository (`postgres.exe`)

The API Server handles REST API calls. The Application Server handles requests to the web application such as searching, browsing, logging in, generating static images, and managing subscriptions. The Cache Server handles the query cache. The Cluster Controller is responsible for monitoring Tableau Server's components, identifying failures, and executing failovers. The File Store replicates extracts across data engine nodes. Search & Browse is responsible for fast search, filtering, retrieval, and displaying content metadata.

The VizQL server handles the task of loading and rendering requested views. The Data Engine receives queries made to Tableau data extracts present on the server. These queries come from the VizQL processes. To service these queries, the Data Engine loads the Tableau Data Extracts into memory and returns the requested recordset. The Backgrounder runs maintenance tasks and data extract refreshes. The Data Server handles requests to Tableau Data Sources. These requests can come from the Tableau Server or from Tableau Desktop users. The Repository is the Postgres database Tableau Server uses to store settings, metadata, usage statistics, and workbooks.

SIZING THE SERVER HARDWARE

Tableau Server runs well within a variety of hardware configurations. It can be deployed for small organizations on a relatively inexpensive single system. It can also be deployed for large organizations with thousands of users on clusters containing many powerful machines. You get what you pay for in terms of performance from hardware expenditures. Our own test results and those reported by Tableau indicate performance gains in Tableau Server 9 are enhanced significantly with 16-core CPUs—more so than with prior versions.

The current minimum recommended hardware configuration for Tableau Server is a single machine with 32GB of memory and 8-CPU cores. Specific recommendations regarding the size and configuration of your deployment are affected by many factors, including the complexity and size of the dashboards, the data sources, the timing and frequency of usage, the network, the hardware configuration running the software, and whether or not you have the need for high availability redundancy. For these reasons, specific benchmarks are not provided. Consult with Tableau Software's technical staff or a qualified Tableau Software Partner to obtain specific recommendations.

As it is likely that the cost of hardware (particularly in a large-scale deployment) is going to be the least expensive part of your project cost, it would be prudent to oversize your hardware. If you skimp on hardware, you increase the changes for grumpy users, and you risk greater help-desk call volume.

A SCALE-UP SCENARIO

To scale Tableau Server up on a single system, choose a platform that can provide more physical CPU cores and more system memory. At this time, major hardware manufacturers are shipping servers that support up to 32 physical CPU cores and far more memory than Tableau Server will require. The ratio of CPU cores to system memory (1 CPU to 4GB memory) is a good general guideline to follow. Plan for more memory when use of very large Tableau data extracts is expected. The data engine will hold data extracts in memory if possible. This improves query performance.

Disk performance is a secondary consideration when planning for Tableau Server in most cases. The major exception is situations in which there is heavy use of the data engine with extracts that will not fit into memory. In this case, the data engine is forced to go to disk frequently—making faster I/O potentially worthwhile. Otherwise, even with heavy use of the data engine, Tableau Server does not benefit a great deal from more exotic I/O setups such as arrays of Solid State Drives (SSDs).

An example of a scale-up configuration for Tableau Server is a single machine with 24-CPU cores and 96GB of memory. Based on the current Tableau Server scalability tests, it's expected that this server could handle somewhere between 108 and 378 concurrent requests depending on workbook complexity.

A SCALE-OUT SCENARIO

To scale Tableau Server out, multiple servers will need to be provisioned, and the server processes will be split across them. In this case, the servers are not required to be configured identically. It is a common pattern to tailor each machine in a cluster to the processes running on it. Deploying Tableau Server on multiple servers is discussed in greater later in this chapter in the section on "Deploying Tableau Server in High Availability Environments."

An example of a scale-out configuration for Tableau Server is a cluster consisting of three machines each configured with 8-CPU cores and 32GB of memory. This configuration will provide slightly lower performance than the sample scale-up configuration because of the server communication overhead introduced by the cluster.

Regardless of whether you plan to scale up or scale out, if you decide to purchase under the core-license model, you need to determine the number of cores that you'll be required to purchase. Do this by counting the number of physical cores across all of the machines that will be running Tableau Server processes, excluding servers that are running unlicensed services only.

ENVIRONMENTAL FACTORS THAT CAN AFFECT PERFORMANCE

There are many environmental factors that can affect performance of Tableau Server. Typically, the most significant factors relate to network performance, the browser, and resource contention.

NETWORK PERFORMANCE

Users will be connecting to Tableau Server either through an internal network or via the public Internet. Slow network links between users, and Tableau Server can cause erratic dashboard behavior. Spotty Internet connections are a common cause of long dashboard load times. If you do experience slow connection speeds, the best solution is to increase the available bandwidth of the connection.

BROWSER

The user experience of Tableau Server is dependent on JavaScript. As such, some browsers can cause Tableau Server to feel unresponsive or sluggish because of their sub-par JavaScript performance. Older browsers are major offenders in this case. Chrome, Firefox, Safari, Edge, and modern versions of Internet Explorer all have superior JavaScript performance. If it takes a few clicks to get a quick filter drop-down selection to apply, you might be running into a browser performance issue.

RESOURCE CONTENTION

Tableau Server will not perform well in environments with other resource-hungry applications and services running on the same machine. Resource contention can cause slowness in each component process of Tableau Server. To get the most out of your Tableau Server license expenditure, ensure that Tableau Server is the only application running on the machine(s).

CONFIGURING TABLEAU SERVER FOR THE FIRST TIME

When installing Tableau Server, there are many configuration options to evaluate. These setup options are system-wide. Some are permanent and not easily changed after initial setup. For example, the user authentication method you choose is permanent, so you should carefully consider the option you select for user authentication before you begin installation of Tableau Server.

Before you attempt to install Tableau Server for the first time, go online and search for "Tableau Server Administration Guide 9." You can get a PDF of the entire administrator manual (around 600 pages) or access it via the Web. Review this guide first. After installing Tableau Server, you can also access the administrator manual and additional documentation from within Tableau Server.

This section will detail the steps required for a first-time installation of Tableau Server but will also include more advanced features related to Alerts and Subscriptions and different security options you can apply. I will not detail (but will outline) the steps you should take for upgrading Tableau Server.

It's not uncommon for new Tableau Server users to install Tableau Server on a laptop as a localhost for initial testing. To do this, your computer should have

- CPU (with at least two cores)
- 4GB RAM
- 15GB free disk space

This configuration will support the 32-bit version of Tableau Server. When you are ready to deploy Tableau Server in an actual server environment on premise or in the cloud, use the 64-bit version. This requires

- CPU (with at least 4 cores)
- 8GB RAM
- 15GB free disk space

After downloading and installing the Tableau Server zip file, you'll be presented with the typical Windows software installation screens that ask for you to verify the install location, region, and language options and then the activation screen where you can activate your license or start a 14-day trial. You can also choose to activate the software offline. See Tableau's online manual for details on offline activation. Now you are ready to start configuring Server.

GENERAL SETUP MENU TAB

When you install server for the first time, the configure menu presents four possible different menu tabs—General, Data Connections, Alerts and Subscriptions, and possibly a security setting tab. Figure 11-1 shows the General tab configuration menus.

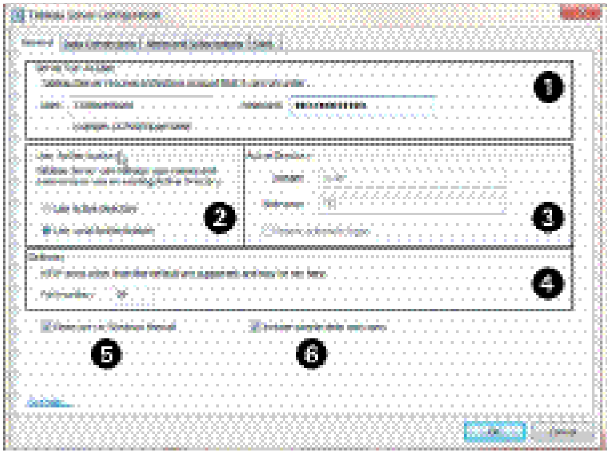


FIGURE 11-1 *Server configuration—General tab*

Server Run As User (area 1) refers to the Windows username that the Tableau Server service (tabsvc) will run under. By default, this is configured as the Network Service account. This can be changed to either a local machine account or a domain account. If you choose a domain account, specify the domain with the username. One reason to use a domain account is to provide access to data sources that require Window NT authentication without prompting users for credentials. In Figure 11-1, the account specified is `TSI\mcedward`, which matches the `DOMAIN\username` used in a Tableau On Demand training video on Tableau's website.

GENERAL: RUN AS USER, USER AUTHENTICATION, AND ACTIVE DIRECTORY

Area 2 of Figure 11-1 displays the options for authenticating users:

- Local Authentication
- Active Directory Authentication (AD)

It is important that you carefully choose the authentication method because this cannot be changed once the server is installed. This is permanent. Changing it later isn't easy. Tableau does provide procedures on its website if you are forced to change your user authentication later. Avoid that heartache by carefully choosing this before you install the software.

Local Authentication means that you will create a username and password setup inside Tableau. This is not the authentication method most administrators select. Using Active Directory authentication requires that users who are added to the Tableau Server must already exist within your Active Directory. Because most organizations already have Active Directory in place to provide security for network access, selecting Active Directory authentication allows you to reuse your existing security structure. Figure 11-1 (area 2) shows the User Authentication menu in under the General settings tab. In the figure, Local Authentication is selected. This is fine if you are doing a localhost installation, but if you are installing on a network server, you should select Use Active Directory.

Be sure to enter the domain name and nickname in Figure 11-1 (area 3) when choosing to authenticate with Active Directory. This domain name must be a fully qualified domain name. Using the AD method allows an additional option—Enable Automatic Log-on. This option enables users to automatically log into Tableau Server with the currently logged in Windows account credentials via the Microsoft Security Support Provider Interface (SSPI).

If you are installing Server 9.1 or greater, Tableau Server can be enabled to synchronize Active Directory Groups that have been imported into Tableau Server. You can also schedule synchronizations for daily, weekly, or monthly intervals at specific times. Before V9.1 this was possible only through Tableau Server's command-line tool tabadmin. Tableau's command-line tools are covered in Chapter 13.

GENERAL: GATEWAY PORT NUMBER

By default, Tableau Server accepts requests on port 80. Figure 11-1 (area 4). If you have a firewall or proxy in front of the Tableau Server host, you may need to modify this point number. If you aren't the system administrator, contact that person to get the specific port number for your network.

GENERAL: OPEN PORT IN WINDOWS FIREWALL

Select the Open port in Windows firewall check box (Figure 11-1, area 5) to open the specified port number shown in area 4 as well as port number 443 if SSL is enabled.

GENERAL: INCLUDE SAMPLE DATA AND USERS

If you decide to include sample data and users (Figure 11-1, area 6), Tableau will install a sample project with workbooks. This is a good way to check that your installation is working properly and is recommended. You can delete the files once you've confirmed that everything is working properly.

DATA CONNECTION TAB

The data connection cache options are defined in the Data Connections tab you see in Figure 11-2.

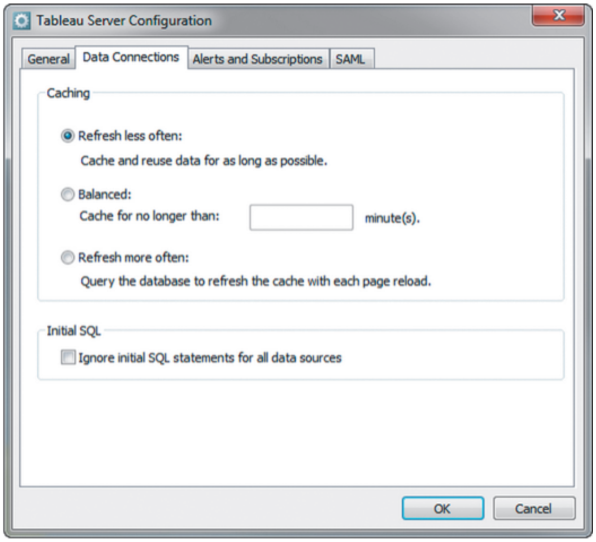


FIGURE 11-2 *Data Connections tab*

The caching options within Tableau Server dictate how often cached data will be reused and how frequently data will be queried from the data source. The options include

- Refresh Less Often
- Balanced
- Refresh More Often

Caching option selections can significantly affect performance. Reading from the cache is much quicker than querying the data source directly. In most cases, leaving this option set to Refresh Less Often will provide the best performance. The main reason to change to Balanced or Refresh More Often is to prevent old data from being reported from when you have a rapidly changing data source. You can change this setting later if your environment's needs change.

The Initial SQL section at the bottom of Figure 11-2 is important if your users will be connecting to a Teradata data source to build views. Tableau provides the option to define a SQL command that will run one time when the workbook is loaded into the browser. Unless you are accessing Teradata, this option should not be selected. For performance and security reasons, Teradata administrators may find it desirable to turn this option off.

ALERTS AND SUBSCRIPTIONS

This menu tab is where you set up email alerts for system administrators and for end users interested in receiving e-mail notification when workbooks are updated. Figure 11-3 shows the Alerts and Subscriptions menu tab.

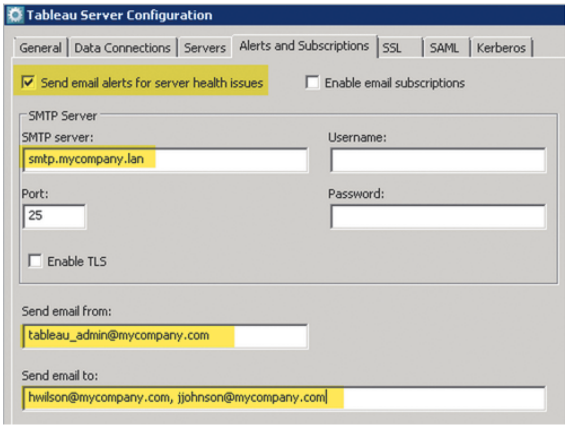


FIGURE 11-3 *Alerts and Subscriptions tab*

To enable e-mail alerts for systems administrators that provide notifications related to server health issues, click the check box highlighted in the upper left of Figure 11-3. This means that the administrator in the bottom section will receive an e-mail from the defined address `tableau_admin@mycompany.com` to the addresses `hwilson@mycompany.com` and `jjohnson@mycompany.com`. You must also enter a valid SMTP server address. If your SMTP account requires it (this is an optional setting), you may also have to enter a valid username and password for the SMTP server. The default port value is 25. You should change this only if you know you are using another port number.

Subscriptions allows Tableau Server users to receive e-mail notifications when a workbook is updated. To enable e-mail subscriptions, select the option highlighted at the top right of Figure 11-4.

Completing the SMTP Server information, send from e-mail account, and the Tableau Server URL settings that are highlighted in Figure 11-4 will give your server users the ability to create e-mail notifications that will come to their e-mail Inbox when the selected views are updated. Figure 11-5 shows the top of the server site page where your users can create subscriptions.

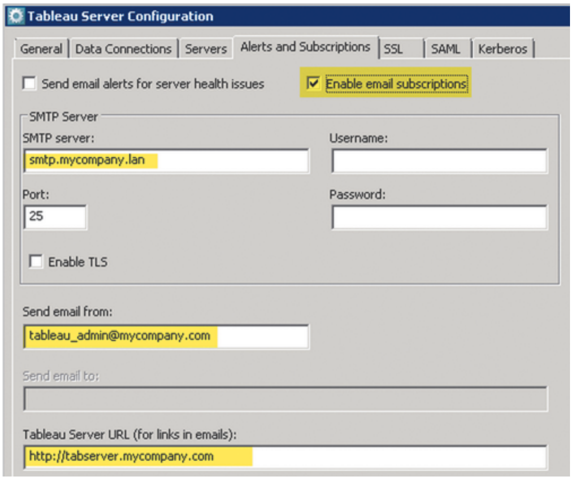


FIGURE 11-4 *Enable e-mail subscriptions*

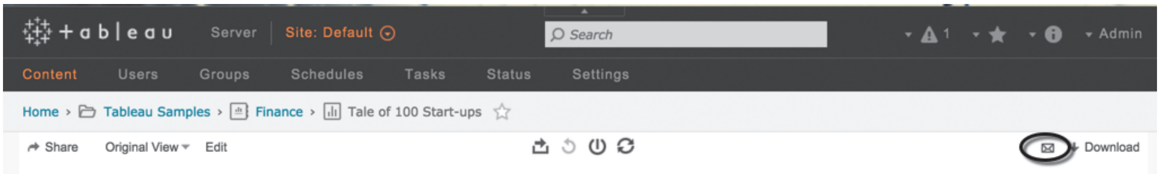


FIGURE 11-5 *Subscription icon*

After enabling subscriptions, a small mail icon will appear in the upper-right side of server web pages. This icon enables users to define subscriptions for the view. To subscribe, the user clicks the icon circled in Figure 11-5 to expose options for defining a title for the e-mail (it will default to the sheet name), the delivery schedule (server includes standard schedule or you can customize schedules), and whether to include the view currently being displayed or all the sheets in the workbook. When the subscription is delivered, it includes an image of the sheet. The user can click the image to open the view in Tableau Server.

SERVER PROCESSES

When Tableau Server is installed, various server processes are installed and configured automatically. Beginning with Tableau Version 9, new processes have been added that provide additional capability. You are now able to reconfigure these processes in multi-server clusters to assign specific processes to each machine. See the online Tableau Server Administrator manual for details. Tableau V9's new processes include the following:

- **API server:** Handles REST API calls
- **Application server:** Supports browsing and searching the web application
- **Backgrounder:** Executes tasks, including extract refreshes, tabcmd tasks, and "Run Now" tasks
- **Cache server:** Query case to improve load speeds
- **Cluster controller:** Monitors components, detecting failures and executing failover in clustered environments
- **Coordination service:** In distributed installations, ensures a quorum exists for automating decisions during failover
- **Data engine:** Stores data extracts and answers queries
- **Data server:** Manages connections to Tableau Server data sources
- **File store:** Automatically replicates extracts across data sources
- **Repository:** Stores user metadata
- **Search & Browse:** Searches, filters, retrieves, and displays content metadata on the server
- **VizQL:** Loads and retrieves views; computes and executes queries

If you are deploying Tableau on a single server, processes related to multi-server clusters will not be present as they relate to coordination of services within distributed environments. Earlier versions of Tableau Server had fewer processes. Added capability has increased the number of Tableau Server's processes. Tableau Server 9 is a more capable enterprise tool with improved scalability and security. Greater capability comes at the cost of more overhead and greater complexity in the background. For more information on Tableau Server, see Tableau's online Administrator Guide.

SECURITY OPTIONS

A few years ago Tableau Server security was an easy topic to write about because there were limited options. Fully covering this topic today would require hundreds of pages. I will strike a balance in this this section between what a curious Tableau Desktop user might be interested in and what an experienced, technical network administrator needs to know about how security protocols operate within Tableau Server. Tableau Server security is based on the following items:

- **User identity:** Handled via authentication
- **What users can do:** Authorization of access to data
- **Securing communications:** Network security protocols
- **Securing data:** Vendor access methods and protocols

Tableau Server authentication is accomplished globally through the User Authentication method selected under the General Settings tab shown in Figure 11-1. To prevent unauthorized intrusion to communications required between server users and the physical server(s), Tableau utilizes Secure Sockets Layer (SSL) encryption. This is referred to as External SSL. To encrypt communication between Tableau Server's Postgres data repository and other server components in your deployment, SSL is also used. This is called Internal SSL. To avoid the inconvenience of multiple sign-ons within a secure connection, Tableau supports a variety of different methods including SAML (Security Assertion Markup Language) and Kerberos (an authentication protocol developed by MIT and used by Microsoft Windows as the default authentication method).

After identifying users and securing communications, Tableau Server utilizes a hierarchy of site roles to define permissions for exactly what users can do. Site roles include

- **Server administrator:** With access to do anything
- **Site administrator:** With access to do anything on a particular Server site
- **Publisher:** Someone permitted to publish content to Server
- **Interactor:** Someone permitted to interact with views on Server
- **Viewer:** Someone who can merely look at views on Server

The exact details of what each level is allowed to do are defined by permissions. Permissions can be assigned to individuals, workbooks, data sources, projects (a collection of workbooks), groups (a collection of people), or all users. The combination of all of these different systems provides robust security. Tableau V9 greatly enhanced the interface for viewing and setting permissions by making the interface more visual.

EXTERNAL SECURE SOCKETS LAYER

This section assumes that you have already obtained an SSL certificate for your network. To enter the details for your network, follow these steps:

1. Open the Tableau Server Configuration Utility from your Windows Server Start menu by selecting Start ➢ All Programs ➢ Tableau Server 9.0 ➢ Configure Tableau Server.
2. In the Configuration Tableau Server dialog box, select the SSL tab.
3. Select Use SSL for Server Communication and provide the location for each of the certificate files you see in Figure 11-6.

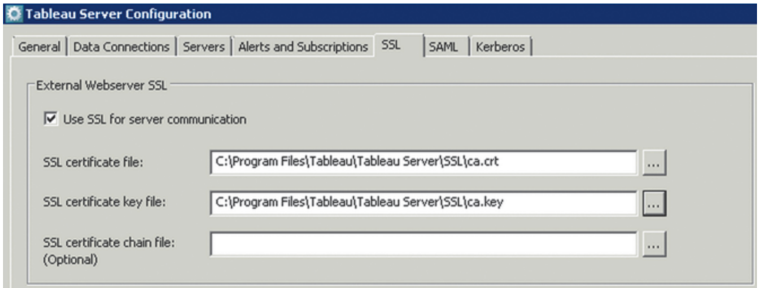


FIGURE 11-6 *External SSL setup menu*

- **SSL Certificate File:** Must be a valid PEM-encoded x509 certificate with the extension `.crt`.
- **SSL Certificate Key File:** Must be a valid RSA or DSA key that has an embedded passphrase and is not password protected with the file extension `.key`.
- **SSL Certificate Chain File (Optional):** Some certificate providers issue two certificates for Apache. The second certificate is a chain file that is a concatenation of all the certificates forming the chain for the server certificate. All of these certificates must be x509 PEM encoded, and the file must have a `.crt` extension (not `.pem`).

After entering the details, click OK. The changes will be applied when Tableau Server is restarted. Tableau Server currently uses SSL only over port 443. If you have to configure a multi-node cluster and your primary server is the only node that is running the gateway process, you can follow the steps defined earlier. If your setup has multiple gateways, you'll have to configure your load balancer for SSL and Tableau. If you need to change these settings after installation, select the Configure Tableau folder in All Programs. For more information about this more advanced configuration, refer to Tableau's online Administrator Guide.

You can also apply SSL internally to security communications between Tableau Server's Postgres data repository and other server components. This feature is disabled by default. If you want to enable it, go to Tableau's online manual and search for "SSL for Direct Connections" where you'll find detailed setup instructions.

SAML—SECURITY ASSERTION MARKUP LANGUAGE

Security Assertion Markup Language (SAML) is an XML-based open standard developed by the Security Services Technical Committee of the Organization for the Advancement of Structured Information Standards (OASIS). Is that enough acronyms for you? SAML separates development of security systems from applications. Open standards in the era of cross-platform mobile communications are desirable. By using a third-party Identity Provider (IDP) to handle the authentication component for Tableau Server, you can enable Single Sign-on (SSO) in Tableau Server. Using SAML, you can securely pass authentication information (such as a username or e-mail address) to Tableau once a login is successfully completed. Offloading the authentication function to an IDP provides a more seamless user experience while maintaining secure and centralized identity management.

Before you begin to configure SAML on Tableau Server, you must place the certificate files in a folder named SAML, such as:

```
C:\Program Files\Tableau\Tableau Server\SAML
```

If you decide to configure SAML during your first installation of Tableau Server, you will go to the SAML tab to complete the necessary details. If you wait until after your initial setup, go to the Tableau Server Configuration Utility on your Windows server and access Start ➢ All Programs ➢ Tableau Server, and then click the SAML tab and provide the addresses for the items you see in Figure 11-7.

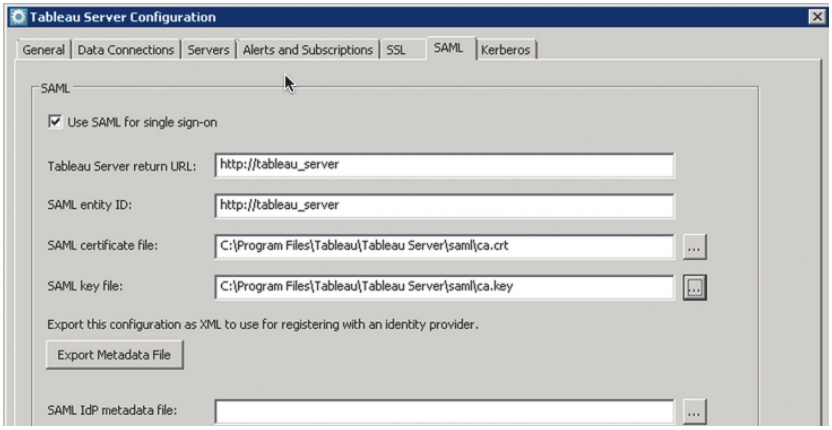


FIGURE 11-7 *SAML setup menu*

- **Tableau Server return URL:** The URL to which the Identify Provider (IDP) redirects the user after a successful login.
- **SAML entity ID:** Allows your IDP to identify the Tableau Server application. Use your Tableau Server URL to avoid confusion.
- **SAML certificate file:** A certificate file identifying the Tableau Server application.
- **SAML key file:** A private key for Tableau Server to decrypt messages from the IDP.

Once this information is completed, you export the metadata to configure your IDP. Do this by clicking the Export Metadata File button in the menu. Your IDP may output an additional metadata file, which you then add to the SAML IDP metadata file field at the bottom of the SAML menu. This completes the SAML configuration in Tableau Server. Unauthenticated users accessing Tableau Server will be redirected to your IDP's web page and, upon successful login, will be authenticated and redirected into Tableau Server.

If you have a multi-node cluster, refer to Tableau's online Administrator Guide and search for "Configure a Server Cluster for SAML."

KERBEROS—A TICKET-BASED SECURITY PROTOCOL

Kerberos is a network authentication protocol developed by the Massachusetts Institute of Technology (MIT) and adopted by Microsoft for Windows authentication in 2000. It works by using tickets to allow clients and servers to communicate securely. Once again, Tableau Server's online manual provides all of the details to set up Kerberos. This process must be done by your administrator. The steps are as follows:

1. Open a command prompt and change the directory to the location of Tableau Server's bin directory. The default location is `C:\Program\Files\Tableau\Tableau Server\9.2\bin`.
2. Type this command to stop Tableau Server: `tabadmin stop`.
3. Open the Tableau Server Configuration Utility from Start ➢ All Programs ➢ Tableau Server 9.2 ➢ Configure Tableau Server.
4. Click the Kerberos tab.
5. Click Export Kerberos Configuration Script. The script generated configures your Active Directory domain to use Kerberos with Tableau Server.
6. Have your Active Directory domain administrator run the configuration script to create Service Principal Names (SPNs) and the `.keytab` file.

The domain administrator must review the script to verify that it contains the correct values. Run the script at the command prompt on any computer in the domain by typing the script name. The script will create a file (`kerberos.keytab`) in a `\keytabs` folder in the location where the script was run.

7. Save a copy of the `.keytab` file created by the script to the Tableau Server computer. In step 3, enter the path to the `.keytab` file, or click the Browse button to navigate to the file. The key tab file will be copied to all the gateway nodes in your Tableau Server installation when you click OK in the Configuration utility. Do not rename the `.keytab` file. It must be saved using the `kerberos.keytab` name.

Click the Test Configuration button in the menu to confirm that your environment is working properly. Then click OK to save your Kerberos configuration and restart Tableau Server. Tableau's online manual provides a quick-start guide for this procedure and more technical details related to Kerberos. If you run into any problems, search the online manual for "Kerberos."

Now that you've enabled a secure authentication protocol for Tableau Server, in the next section we'll discuss how you manage what you permit users to do when they access Tableau Server.

MANAGING OWNERSHIP THROUGH HIERARCHY

Tableau Server has a robust system for managing access. To fully grasp it, you must understand the hierarchy of objects that contain reports and data within Tableau's environment. These objects include

- Workbooks and views
- Users
- Projects
- Groups
- Sites
- Permissions

Search the Tableau Online manual for "Manage Ownership" for details regarding who can change or be given ownership for each of these objects.

WORKBOOKS AND VIEWS

The Workbook object represents the Tableau workbook file published from Tableau Desktop. It contains dashboards and worksheets, which in terms of

Tableau Server are all known as Views. Permissions can be applied to specific Views within a Workbook or at the whole Workbook level. Workbooks and Views can belong to Projects and must be published to a Site.

USER

The User object represents a named-user who has access to the Tableau Server. Users must be granted a licensing level of Interactor or Viewer to log in to the server. It's possible to leave a user account on the server in an effectively disabled state by setting its licensing level to unlicensed. This can be useful for audit purposes. Users can be granted access to Views, Workbooks, Projects, and Sites. They can also be placed into groups. Also note that unlicensed Tableau Server users (who have been given publishing rights) can publish workbooks to Server even though they cannot view the published results on Server.

PROJECT

The Project is an object used to organize and manage access to Workbooks and data sources. Workbooks are placed into Projects within a Site. This can be used as an organizational tool by placing Workbooks with similar content into a single project. They can be used as an access restriction tool by granting access to a Project to a user or group and then publishing Workbooks into that Project.

GROUP

The Group is an object used to organize users in Sites on the Tableau Server. Users can be placed into Groups, and these Groups can, in turn, be given permissions to objects on the server. Groups can be created locally on the Tableau Server, or, if Active Directory authentication is in use, they can be imported from an Active Directory Group. Groups make managing user permissions within Tableau Server much easier.

SITE

The Site is the top level of the security hierarchy. Sites are completely separate Tableau Server instances from the user perspective. Users cannot log in to, or view, any information about Sites to which they do not have access. The base Tableau Server Site is known as the Default Site. Users that belong to more than one Site must choose which Site they want to see when they log in.

PERMISSIONS

Sites define separate work environments in Tableau Server. Permissions define what users or groups are permitted to do within a site. Tableau Server comes with several standard permission roles that can be assigned to Users or Groups.

- **Server administrator:** Can access, interact with, publish, and manage all objects on the server
- **Site administrator:** Can access, interact with, publish, and damage all objects within a site
- **Publisher:** Can access, interact with, and publish objects (workbooks)
- **Interactor:** Can access and interact with objects (workbooks)
- **Viewer: C**an access workbooks and publish objects (workbooks)
- **Unlicensed:** Can publish only

In addition to roles, there are *permission rules* that can be applied at the User or Group level. These permission rules grant users or groups specific capabilities such as the ability to view, interact with, or edit workbooks and data sources. Tableau Server comes with standard rules, but if these don't work, you can edit them to meet your needs. You can add customized permissions to Groups or Users.

The visual interface in Tableau Server V9 has been improved numerous ways. Permissions can now be reviewed quickly with the improved visual interface you see in Figure 11-8.

Tableau provides default permissions for All Users that can be edited by server or site administrators. As you see in Figure 11-8, all users have been granted View capabilities along with Interact capabilities (excluding Web Edit). One Save has been enabled within the Edit section. Specific permissions can be added to groups or users. Go to the online Tableau Server Administrator Guide and search for "Manage Permissions" to explore more details.

Using Groups and Projects to manage access is much easier than assigning user permission to workbooks or users individually. Depending on the sensitivity of data contained in workbook, some organizations choose to make heavier use of individual Sites rather than Projects. It is important to understand that moving content between projects is easy, but moving content between Sites requires republishing the content.

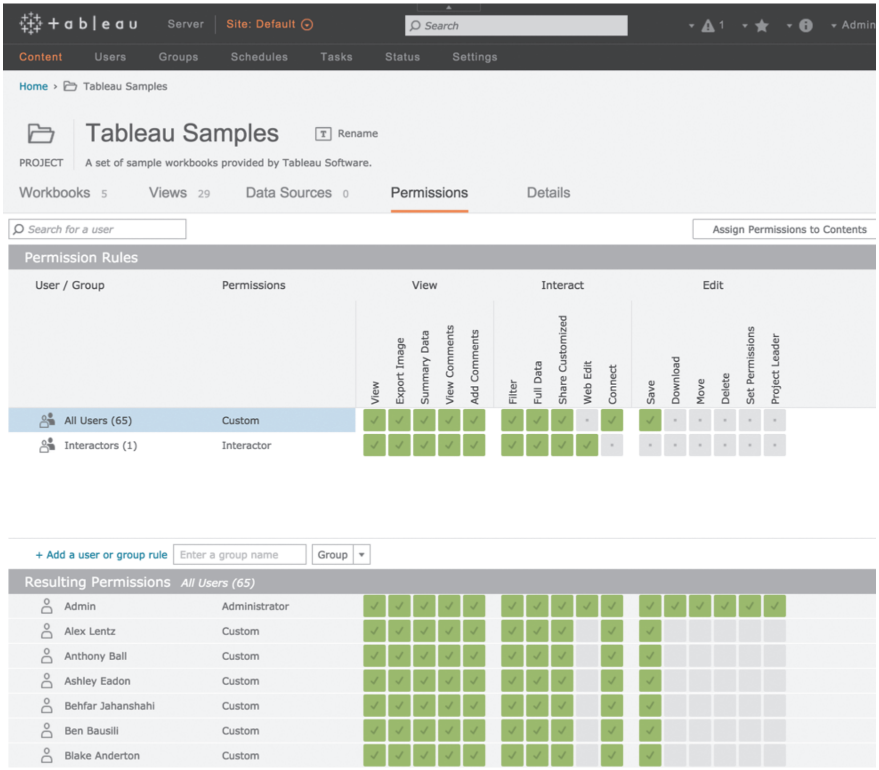


FIGURE 11-8 *The Permissions window*

If you have particularly sensitive data for departments such as human resources or sales, it may make sense to create separate sites for those work groups to ensure that the data is secure. Another common use for Sites is to create an alternate site for development and testing on the server. Tableau's online manual contains nearly 40 pages of details related to permissions. Read the manual before you develop a permission strategy.

PERMISSIONS FOR WEB EDIT, SAVE, AND DOWNLOAD

The last few releases of Tableau Server have provided new capabilities for Tableau Server users. In some cases, light users of Tableau Desktop may no longer require a Desktop license. Server users (with the appropriate permissions enabled) can now:

- Edit workbook views (and create new fields via calculations)
- Save views
- Download workbooks

Refer to Figure 11-8 and notice that there are three categories of permissions that can be enabled: View, Interact, and Edit. Web editing is included in the Interact category, while Save and Download are part of the Edit category. Enabling these three permissions gives users the ability to create new views in the browser or on a tablet. Save those views for their instance of the file and download the source workbook to their desktop.

Tableau made significant improvements to web editing in Version 9. Ad hoc calculations were enabled for Tableau Server for the first time. I expect Tableau to continue to add more web editing capabilities in future releases.

PROVIDING DATA SECURITY WITH USER FILTERS

While it has been possible for years to provide row-level security for data published to Tableau Server, with Version 8 forward it is much easier to achieve. Tableau desktop users can publish workbooks to Tableau Server that filter views based on usernames by creating filters in views, filters embedded in data sources, or via a hybrid method that utilizes the data source to apply filters.

Apply a User Filter in a View

There are two ways to apply a user filter directly in Tableau Desktop that can be used in a workbook or data source published to Server V8 or later. First, let's look at the steps to create a data filter in a view.

To create your own version of the examples to follow, you need to have a live connection to Tableau Server that has at least four users. Or you can download and install Tableau Server on your personal computer, create an administrative user account, and add at least four users to the system. Then open Tableau Desktop and create a view that looks like Figure 11-9.

The filled map color-encodes the four regions in the data set: east, south, central, and west. Follow these steps to create the user filter:

1. Go to the main menu and select Server ➢ Create User Filter.
2. Sign in to Tableau Server.
3. Select a dimension to apply the filter.
4. Drag the user filter from the Sets pane to the Filters shelf.
5. Test the filter with the User Emulator in the bottom right of the worksheet.

Figure 11-9 shows the Server ➢ Create User Filter ➢ Region selection for creating the filter set. If you aren't already logged in to Tableau Server, you will be prompted to sign in to Server to create the user filter. Figure 11-10 shows the User Filter dialog box.

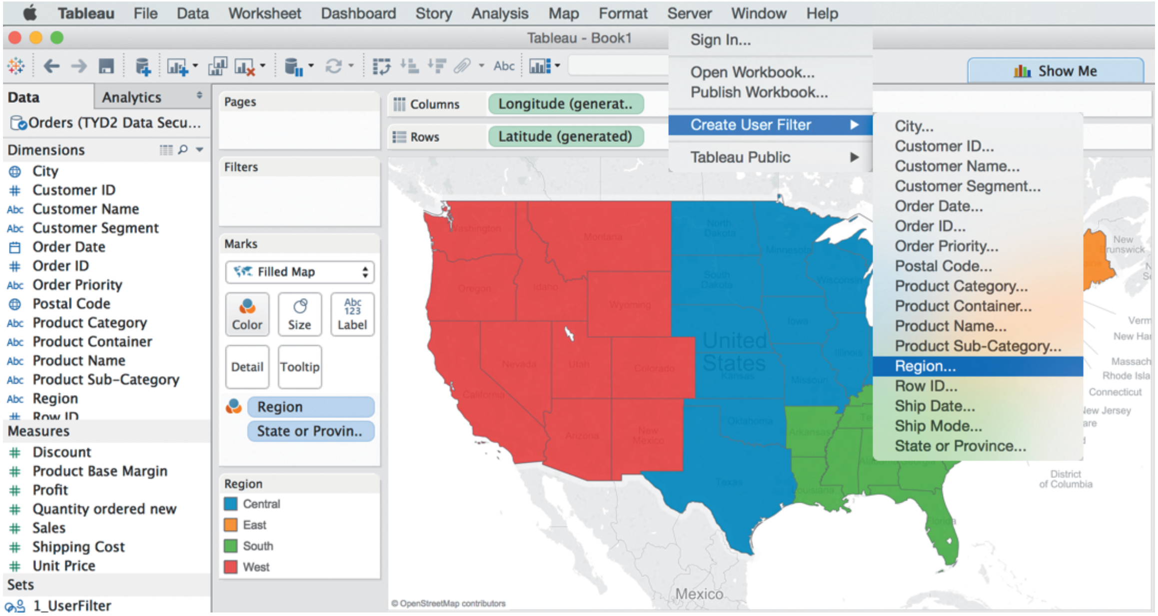


FIGURE 11-9 *Region map*

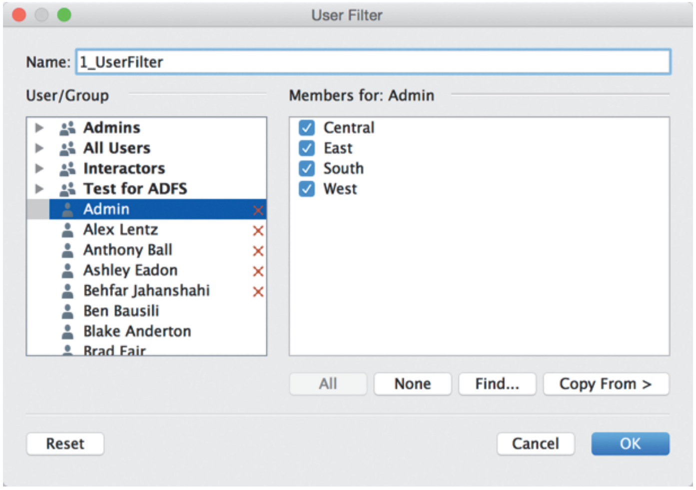


FIGURE 11-10 *Create the user filter.*

Add the following filter selections:

- **Admin:** Apply to all regions.
- **Alex Lentz:** Central region.
- **Ashley Eadon:** South region.
- **Anthony Ball:** East region.
- **Behfar Jahanshahi:** West region.

The filter name used in the example is 1_UserFilter. You can use your own usernames to apply to each region. The red X you see next to the four names in Figure 11-10 indicate that those names have been applied in the filter.

Next, drag the 1_UserFilter from the Sets pane to the Filters shelf you see in Figure 11-11.

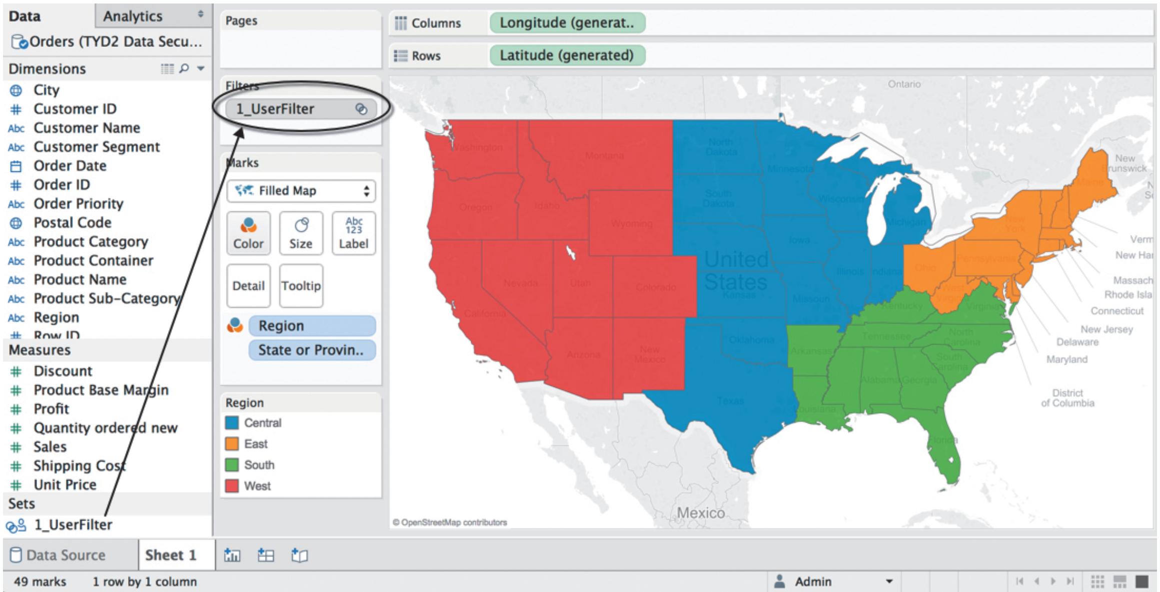


FIGURE 11-11 *Add the user filter.*

At the bottom right of Figures 11-11 and 11-12 you see the user emulator. The emulator shows you what the view will look like for that user when it is published to Tableau Server. Admin user was given permission to see all of the regions. Figure 11-12 shows how the map will be filtered for each user.

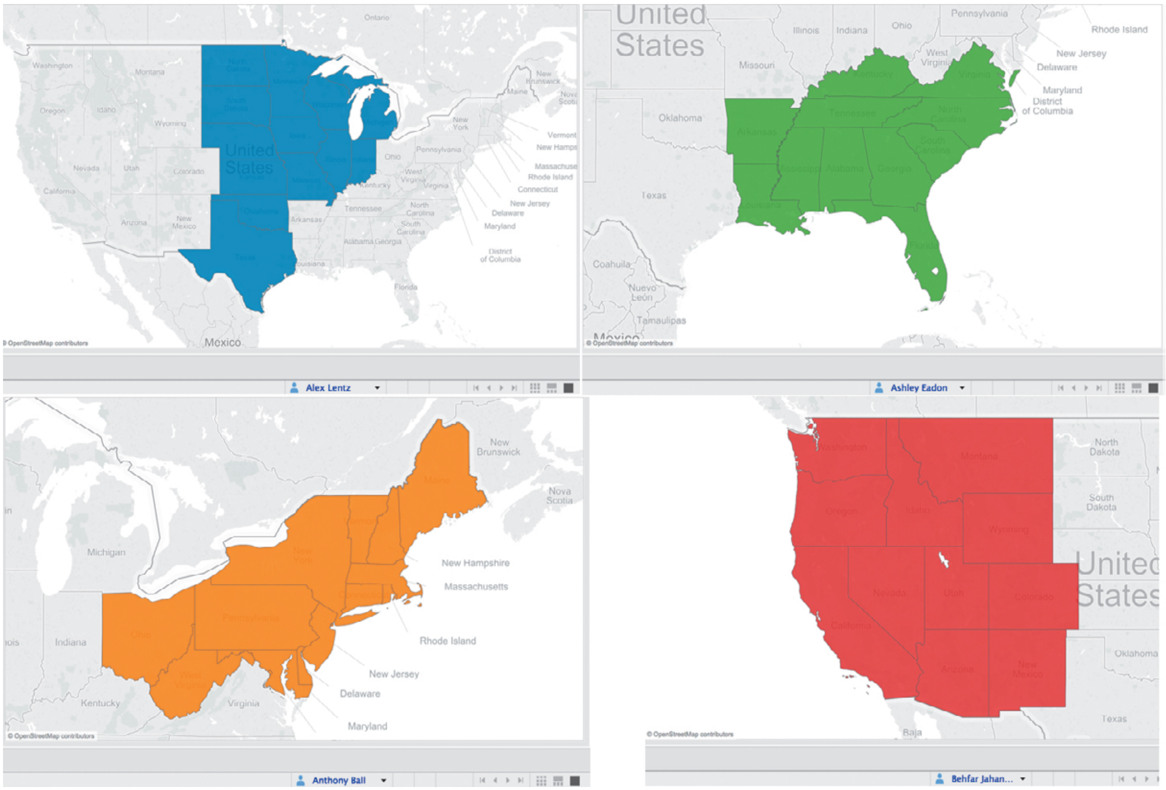


FIGURE 11-12 *User filter emulator*

You can see that each user is filtered for a particular region. As long as the users on Tableau Server do not have edit or download permissions, they cannot alter the user filter. You can also apply this user filter directly to the data source.

APPLYING A USER FILTER TO A DATA SOURCE

Remove the 1_UserFilter set from the Filters shelf. Now you'll apply that same filter directly to the data source. The procedure is as follows:

1. Right-click the data source.
2. Select ➢ Edit Data Source Filters ➢ Add.
3. Pick the 1_Userfilter set.

Figure 11-13 shows the dialog boxes open with the 1_UserFilter set highlighted.

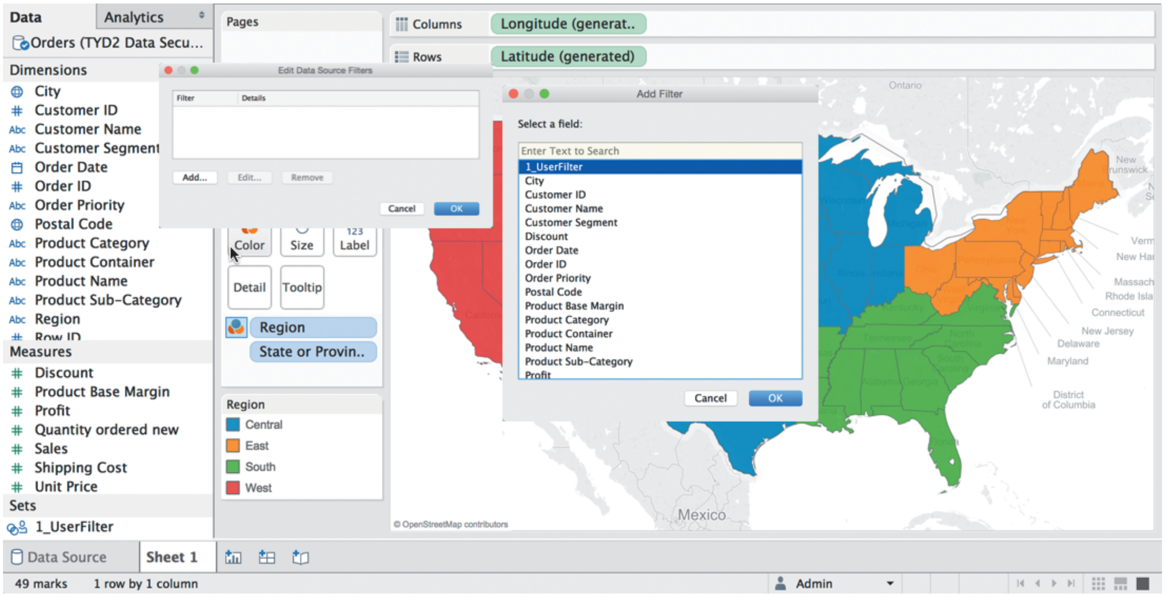


FIGURE 11-13 *Applying the data source filter*

Select OK to apply the user filter to the data source. The resulting filter action should be identical to the results you see in the first example shown in Figure 11-12. The Admin user can still see all of the regions. The last example uses a hybrid approach to applying the user filter.

CREATING A HYBRID FILTER FROM THE DATA SOURCE

If you have data in your data source that you can use to enable user filters, it can be utilized to achieve similar results to the Tableau-only data security methods but with the added benefit of little to no maintenance in Tableau. In the next example, an additional tab in the TYD2 Data Security Example Ch11.xlsx data source will be used to filter the data. This tab in the Excel data source includes the managers by region. Figure 11-14 shows a small section of the Orders table that includes sales information and the Managers table that defines the regional management responsibilities. These two tables will be joined to create the hybrid user filter.

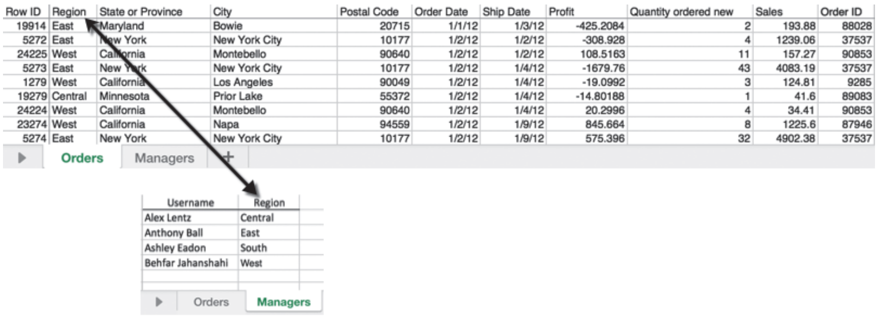


FIGURE 11-14 *Orders and managers tables*

A left outer join was used to combine the data from both tables in Tableau, as you see in Figure 11-15.

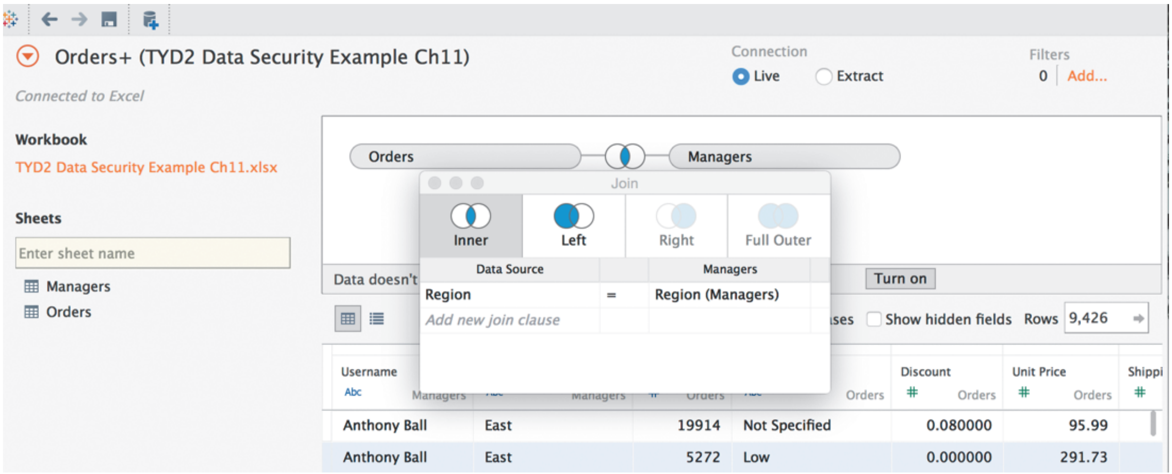


FIGURE 11-15 *Joining the manager data*

The manager names used in the table exactly match their usernames in Tableau Server. If you are building your own example, make sure that the names you add to the manager table exactly match the users you have added in your instance of Tableau Server. To create the hybrid filter, you'll create a calculation that utilizes a user function. Figure 11-16 shows the calculation.

The `FULLNAME()` function validates the [Manager Name] field from the data source against the users in Tableau Server. This formula results in a Boolean (True/False) result. The filter is enabled in the view by applying the new field to the Filters shelf or by adding it to the data source. The end result will look exactly like the Figure 11-12. The beauty of this approach is obvious if you have thousands of users in a database that you are maintaining. Any update to the database would pass through to the Tableau workbook or to the data source.

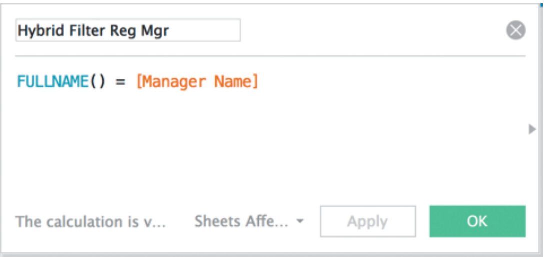


FIGURE 11-16 *User function calculation*

If you want to view a video that has similar examples to those presented here, Tableau Software has an excellent training video illustrating these techniques at `www.tableau.com/learn/tutorials/on-demand/data-security-user-filters`.

WHAT IS THE DATA SERVER?

As a Tableau Server administrator, you should learn what the Data Server is and how it can help you manage workloads more efficiently. The Data Server provides a full range of publishing options that provide access, flexibility, and control.

Packaged Data Sources (`.tdsx` files) are shortcuts that do not contain any actual data but have all of the information needed to connect to a data source and any metadata created in the source Tableau workbook including:

- Default properties
- Calculated fields
- Groups
- Name aliases
- Renamed field
- Other metadata

Publishing data sources to Tableau Server leverages the knowledge of domain experts within your company for a large number of other less knowledgeable people to use. Publishing workbooks with the data embedded is also possible.

These files and connections can be refreshed via scheduled updates or on demand and are then automatically propagated to all authorized users. The type of data source (live connection or embedded) is identified using icons.

Data Server gives end users safe access to clean data while giving the server administrator a single point of control. This governance model provides convenience, flexibility and control.

WHEN AND HOW TO DEPLOY SERVER ON MULTIPLE PHYSICAL BOXES

Earlier in this chapter, you read about considerations for sizing hardware for Tableau Server—specifically the concepts of scaling up and scaling out. Scaling up refers to using more powerful single-server hardware. Scaling out refers to bringing in more machines to help carry the workload. Clustering, distributed environments, and scaling out all refer to the same concept: running Tableau Server on more than one machine to spread the workload.

The decision to scale out Tableau Server in a cluster is normally made when a single server cannot support the expected workload and when adding additional machines represents a lower expected cost than scaling up to a substantially more powerful single machine. Tableau's multiple processes can be assigned to different machines in the cluster to achieve efficient division of the workload.

For instance, an environment that makes use of large data extracts could devote an entire machine in the cluster for running data extract engine processes. This machine could include a larger amount of system memory and fast I/O to support the need to quickly load and query many data extracts. In addition, another machine with very fast CPU cores could be dedicated to VizQL processes if high numbers of concurrent view requests are anticipated. Clustering Tableau Server can also provide high availability capabilities by creating redundant core processes on multiple machines. High availability configurations will be in the next section of this chapter.

In Tableau Server clustered environments, the first machine you install Tableau Server is known as the Primary Server, or the Gateway. All other machines are known as Workers. The Gateway handles all of the requests to the Tableau Server and communicates with the workers to satisfy those requests. To set up a distributed cluster environment, follow these steps:

1. Install Tableau Server on the primary machine. (Note the IP address of this machine.)

2. Stop the Tableau Server service on the primary machine.
3. Install the Tableau Server worker software on all of the Worker machines.
4. Return to the primary (Gateway) server and open the configuration utility.
5. Select the Servers tab and click the Add button.
6. Type the IP address of one of the Worker machines in the dialog box.
7. Specify the number of, and each type of, processes to deploy on the Worker.
8. Click OK.
9. Repeat the same steps for each Worker machine.

Once all of the Workers are added to the cluster, save the changes within the configuration utility and restart the Tableau Server service on the primary machine. For more information about clustered Tableau Server deployments, see the "Distributed Environments" section of the Tableau Server Administrator Guide.

DEPLOYING TABLEAU SERVER IN HIGH AVAILABILITY ENVIRONMENTS

Strategies to guarantee constant availability are broadly referred to as high availability. These strategies necessitate that core components of Tableau Server be redundant to minimize the chance of unplanned downtime. Realizing this goal requires deployment in a distributed environment and running redundant critical processes on separate servers.

Achieving significant redundancy can be realized using a three-server cluster, but to achieve a fully redundant configuration, at least four servers are necessary.

THREE-NODE CLUSTER

In this configuration, the Primary Server or Gateway (the server on which you first install Tableau Server) hosts the following processes:

- Search and browse
- Licensing
- Cluster controller
- Coordination service
- Gateway processes

The two Worker servers are used to host identical configurations that should include:

- Cluster controller
- Coordination service
- Gateway
- VizQL Server
- Application server
- API server
- Backgrounder
- Cache server
- Data server
- Data engine
- File store

Each worker also contains the Active Repository. A load balancer is recommended to direct traffic to active nodes in the event of a failure. The loss of a Worker machine can occur without making the cluster inaccessible. However, because there is only a single Gateway machine, should that server go offline, the cluster will be inaccessible to users. To have complete fault tolerance, a four-node cluster is required.

FOUR-NODE CLUSTER

In a four-node cluster, a backup Primary Server is added to make that critical node redundant. However, the Primary backup server must be promoted to active status manually. There is currently no automatic fail-over for Primary (Gateway) machines.

The high availability setup process is similar to the basic cluster configuration. The following are the steps to set up a high availability configuration:

1. Install Tableau Server on the primary machine (note the IP address of this machine).
2. Stop the Tableau Server service on the primary machine.
3. Run the Tableau Server Worker installer on the other machines included in the cluster (the primary server IP is needed for this step).

4. Open the configuration utility.
5. Select the Servers tab and click the Add button.
6. In the Add Tableau Server dialog box, type the IP address of the first of the Workers.
7. Specify the number of each type of process.
8. Ensure both the extract storage and repository storage are included on this host's settings. Click OK.
9. Start the Tableau Server service on the Primary Server machine.
10. View the server status and observe that the instances of the extract engine and repository on the new Worker appear to be down. This will be resolved once the primary server has transmitted all data for these processes to the new worker machine.
11. After the worker extract engine and repository processes switch from Service Down to Service Standing By, stop the Tableau Server service on the primary machine again.
12. Open the configuration utility on the primary server.
13. Clear the extract storage in the configuration utility on the Host and the repository storage on This Host check boxes for the primary server. Remove all other processes to configure this machine as a Gateway only. Click OK.
14. Click the Add button on the Servers tab.
15. In the Add Tableau Server dialog box, type the IP address of the second Worker and specify the number of each type of process. Be sure to check both the Extract Storage and Repository Storage on this host's settings. Click OK.
16. As an optional step, you can configure e-mail alerts about the cluster status from the Email Alerts tab in the configuration utility.
17. Close the configuration utility and restart the Tableau Server service.
18. Once the service comes back up, check the status of the cluster from the Tableau Server maintenance page. You should see the IP address of the primary server listed with only the Gateway service. You should also see the two Worker server IP addresses listed with the remaining Tableau Server processes. One Worker will have an active data engine and repository, and the other Worker will have standby copies of these processes.

The three-node configuration presented earlier may be augmented with a redundant Gateway server to increase reliability. For more information about making the Gateway redundant and the manual fail-over process, see the "Configuring a Highly Available Gateway" section of the Tableau Server Administrator Guide.

LEVERAGING EXISTING SECURITY WITH TRUSTED AUTHENTICATION

Tableau Server is frequently deployed in landscapes containing legacy systems that already have security protocols to prevent unauthorized access. These systems may include internal website portals, content management systems, or existing reporting interfaces. Is it possible to embed an interactive Tableau visualization into a site that already contains a legacy security protocol? The answer is yes. This is commonly referred to as single sign-on. The Tableau Server system for enabling this is called Trusted Authentication.

When using Trusted Authentication, it is assumed that the web server containing the embedded views will handle the user authentication. The person attempting to access the embedded view must be a valid user on both the web page and Tableau Server. The web page server passes the username of the person who has logged in to the Tableau Server. So, the usernames must match or be programmatically transformed to match.

Tableau Server must also be configured to acknowledge the web page server as a trusted server. This is configured using the Tableau Server Administration (tabadmin) tool. See Chapter 13 for more details on Tableau Server's command-line tools.

The web page server must also be able to perform a POST request and transform the response into a URL. This means that static web pages that are not supported by a scripting language will not be able to support these requirements.

If the web page server uses Security Support Provider Interface (SSPI), configuring Trusted Authentication is unnecessary as long the users are valid members in Active Directory. In that case, Tableau Server authenticates the user via Active Directory as long as the users are also licensed to access Tableau Server. The flowchart in Figure 11-17 illustrates how security data travels between each component.

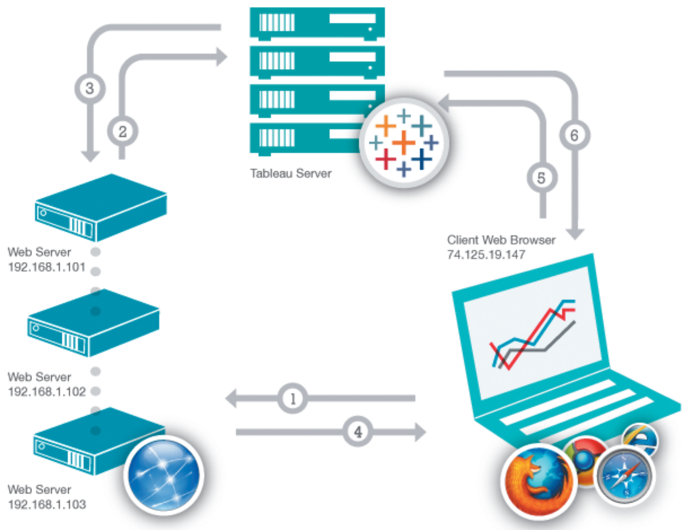


FIGURE 11-17 *Trusted Authentication*

If all of the requirements are met, Trusted Authentication works in the following way:

1. **A user visits the web page:** When a user visits the web page with the embedded Tableau Server view, it sends a GET request to your web server for the HTML for that page.
2. **Web server POSTS to Tableau Server:** The web server sends a POST request to Tableau Server. That POST request must have a username parameter. The username value must be the username for a licensed Tableau Server user. If the server is running multiple sites and the view is on a site other than the Default site, the POST request must include a target site parameter.
3. **Tableau Server creates a ticket:** Tableau Server checks the IP address of the web server that sends the POST request. If it is set up as a trusted host, then Tableau Server creates a ticket in the format of a unique nine-digit string. Tableau Server responds to the POST request with that ticket. If there is an error and the ticket cannot be created, Tableau Server responds with a value of –1.
4. **Web server passes the URL to the browser:** The web server constructs a temporary URL for the view using either the view's URL or its object tag (if the view is embedded) and inserts it into the HTML for the page. The ticket will include a temporary address that will look similar to this URL

address: `http://tabserver/trusted/<ticket>/views/requested-viewname)`. The web server passes the HTML for the page back to the client's web browser.

5. **Browser requests view from Tableau Server:** The client web browser sends a request to Tableau Server using a GET request that includes the URL with the ticket.
6. **Tableau Server redeems the ticket:** Tableau Server sees that the web browser requested a URL with a ticket in it and redeems the ticket. Tickets must be redeemed within three minutes after they are issued. Once the ticket is redeemed, Tableau Server logs the user in, removes the ticket from the URL, and sends back the final URL for the embedded view.

The Tableau Server installation manual provides examples of the code required for the web server to handle the POST to Tableau Server, converting the ticket into a URL and embedding the view in many languages. These examples are included as a part of the Tableau Server installation. Navigate to this Drive location to view them: `C:\Program Files\Tableau\Tableau Server\9.2\extras\embedding`.

For additional tips on using Trusted Ticket Authentication, see the section "Using Trusted Ticket Authentication as an Alternative Single Sign-on Method" in Chapter 12.

DEPLOYING TABLEAU SERVER IN MULTI-NATIONAL ENVIRONMENTS

Tableau Desktop and Server support a wide range of locales and languages. This makes it easy to deploy in organizations with diverse nationalities. Language settings refer to the translation of text in the user interface elements within Tableau. Locale refers to the format of numbers and dates.

Default language and locale options can be configured at the server level by users with system administrator permission. Set up these options in Tableau Server at the Server ➢ Settings ➢ Language and locale menu (see Figure 11-18).

Users can also configure their individual language and locale settings from the User Account page. However, users must do this from their view of the User Account page. Administrators cannot set language and locale options for a specified user. When a user changes these settings, this overrides the default language and locale settings designated by the administrator.

If the user does not have a language and locale specified on his user account page, those settings can also be taken from the user's web browser—if the browser is using a language that Tableau supports. If the language is one

Tableau does not support, English will be used. Also, keep in mind that the author of a workbook in Tableau Desktop can specify language and locale in the workbook. Settings specified in the workbook take precedence over all other language and locale settings.

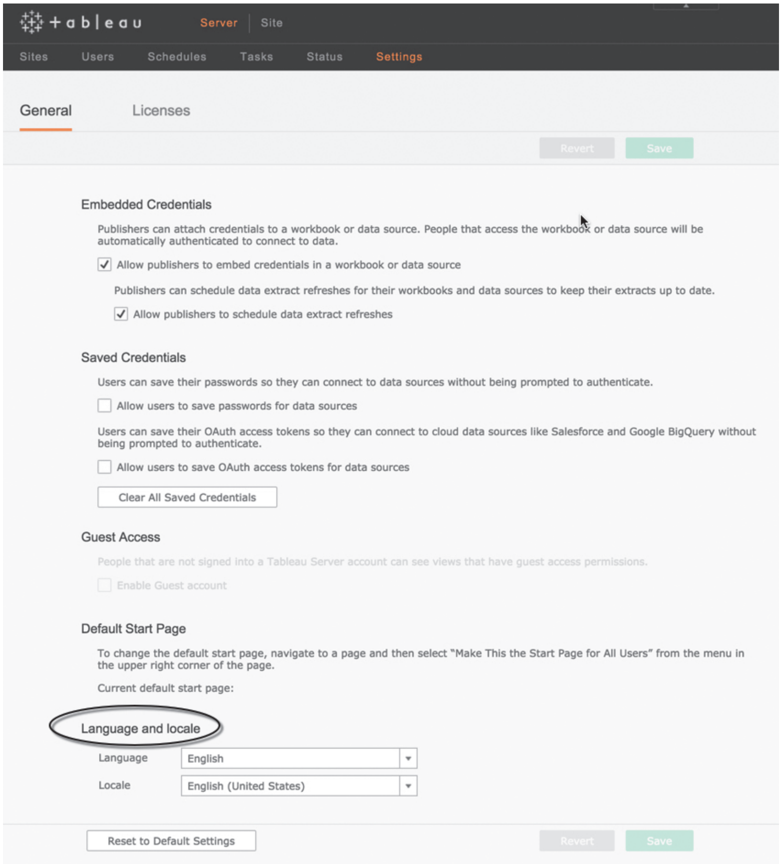


FIGURE 11-18 *Language and locale setting*

The order of precedence—from highest to lowest priority—is designated as follows:

1. The Tableau workbook
2. The user preferences page
3. The locale specified by a user's browser
4. The Tableau Server maintenance page
5. The computer on which Tableau Server is installed

Keep in mind that language options do not translate any report content—only Tableau user interface elements.

TABLEAU SERVER PERFORMANCE RECORDER

At the end of Chapter 8, you learned how to use Tableau's Performance Recorder to improve workbook performance in Tableau Desktop. There is also a separate Performance Recorder that allows you to record and view information about Tableau Server performance at the workbook level.

Prior to Tableau Version 8, this data had to be collected and analyzed manually from log files or via a third-party application that was created by InterWorks. The Performance Recorder creates a Tableau workbook of your Tableau workbook's performance. Information about the following events is captured and displayed visually:

- Query execution
- Geocoding
- Connections to data sources
- Layout computations
- Extract generation
- Data blending
- Server rendering

Performance Recorder is disabled on Tableau Server by default. To activate performance recording on the server, navigate to the Server ➢ Sites ➢ Settings page and check the Workbook Performance Metrics option. Figure 11-19 shows applicable portion of the page.

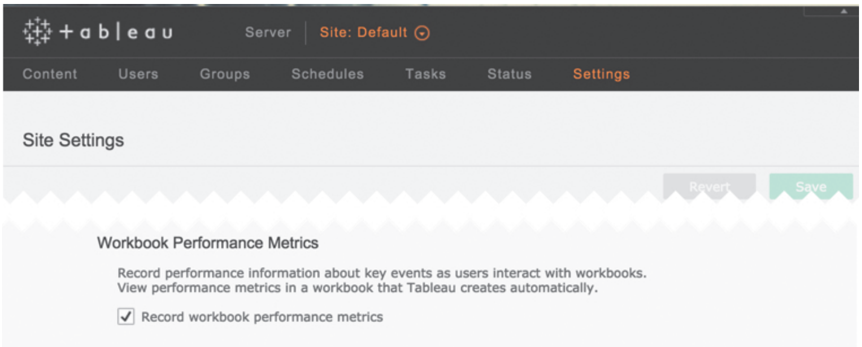


FIGURE 11-19 *Enabling performance recording*

There are several other site setting options on that page. Figure 11-19 shows only the workbook performance metrics check box that is near the bottom.

When you are ready to use the Performance Recorder, you must append the code `?:record_performance=yes&` to the end of the page URL just before the session ID, as you see highlighted in the URL script at the top of Figure 11-20.

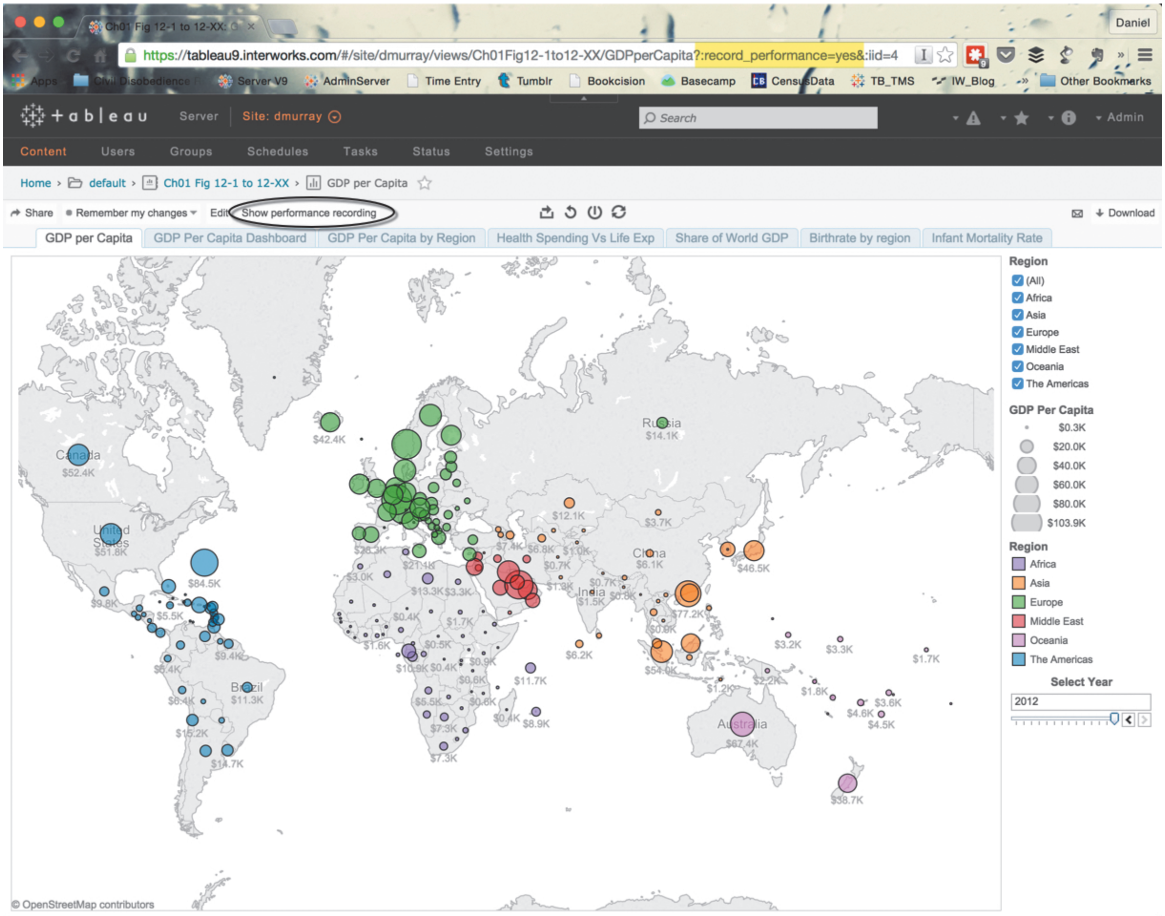


FIGURE 11-20 *Properly inserted recording script*

If everything is working correctly, the Show Performance Recording menu will appear in the view's status bar—circled in Figure 11-20. Clicking this link will open a view that is generated from the recorded performance data. Note that the performance recording view does not automatically update. To see the most current data, close and open the view again.

Once it is activated, the Performance Recorder will continue capturing data about interactions with the view until the user navigates away or removes the string from the URL. Figure 11-21 displays an example of the information available in the Performance Summary display.

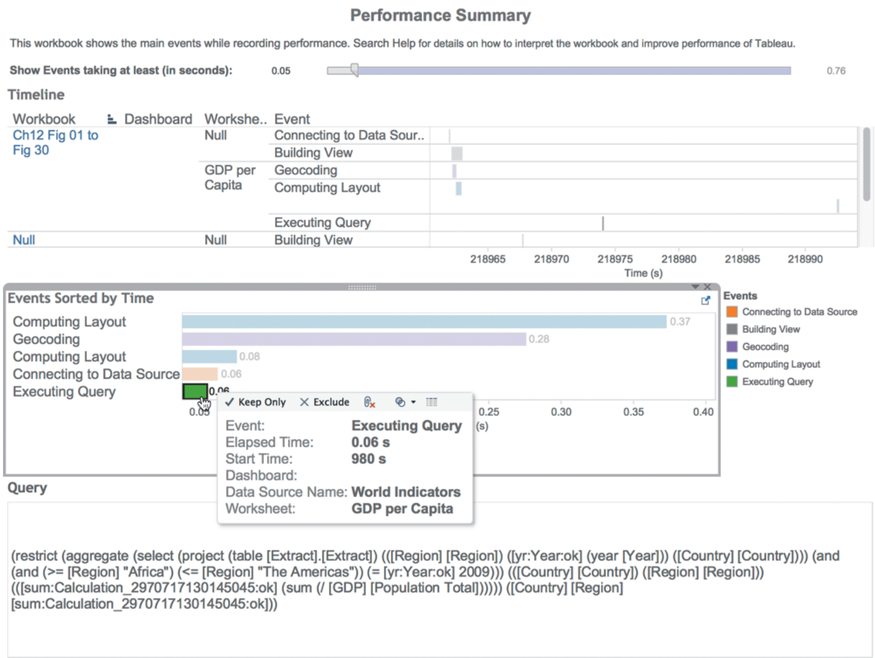


FIGURE 11-21 *A Performance Summary workbook*

The dashboard Performance Recorder contains three panes:

- **Timeline:** A Gantt chart displaying event start time and duration
- **Events Sorted by Time:** A bar chart showing event duration by type
- **Query:** Appears when clicking an Executing Query event in the bar chart

SHOW EVENTS FILTER

This filter allows you to cull the details that appear below to show only the items requiring the specified minimum cost in time.

TIMELINE GANTT CHART

The Timeline Gantt Chart displays by workbook, dashboard, or worksheet when each event occurred. Event start time is indicated by the bar's horizontal position, and the duration of each event is indicated by the individual bar length. The event type is color encoded.

EVENTS SORTED BY TIME

This section of the workbook shows the duration of recorded events in descending order. This is useful for observing the execution time of each event that occurred during the performance recording. This will help you identify any lengthy events that may be the cause of performance problems.

QUERY TEXT

Optionally, the workbook also displays the query text for any specific green-colored Executing Query event that you want to examine in detail. This is a handy feature that allows you to review any query text that may be of interest without having to leave the Tableau Performance Summary dashboard.

PERFORMANCE-TUNING TACTICS

The Performance Summary report generated by the Performance Recorder informs you about the specific events that may be contributing to slow performance. Once you understand the events most affecting performance, try the following tactics to address the performance problem.

QUERY EXECUTION

Query Execution represents the time that it takes for the data source to execute a query and retrieve the data requested by the worksheet. If the data source is a database, it is very helpful to see the queries issued by Tableau in order to identify inefficiencies. Common issues include poor indexing strategies, fragmented indexes, database contention, insufficient database resources, and inefficient SQL queries. If the data source is the Tableau data engine, there are fewer troubleshooting options.

GEOCODING

Geocoding represents the time Tableau needs to locate geographical dimensions. If this event type is consuming too much time, consider geocoding your records in the source dataset and passing a pre-calculated latitude and longitude to Tableau rather than having Tableau generate the geocodes when rendering the map view.

CONNECTING TO THE DATA SOURCE

Connecting to the data source conveys the time required for Tableau to connect to the data source. This event is typically not a large percentage of total worksheet time. In rare cases, there can be a network or data source issues that extend connection times. To rule out these issues, examine the network topology between the Tableau Server and the data source server.

LAYOUT COMPUTATIONS

The time needed for Tableau Server to compute the visual layout of the worksheet is the *Layout Computation event*. This can be influenced by server resource contention as well as worksheet complexity. The more marks that are visualized within the workbook, the more time that workbook will require to load and refresh. It may be necessary to restrict the number of marks simultaneously displayed through techniques such as actions, filters, and aggregation. Large text tables can be particularly costly and are not a good visual analytic technique. If all these tactics fail to result in noticeable improvement, it may be necessary to provide additional resources to the server.

GENERATING EXTRACT

The amount of time that the data engine spends generating an extract is called the *Generating Extract event*. The size of the data source (the numbers of rows and columns) along with the time Tableau spends compressing and sorting the data are the major factors affecting the time required to generate extract files. Starting with the release of Tableau Server V9, persistent query caching reduces the impact of costly queries after the cache is warmed.

If your extract file is still taking too long to refresh in your environment, it may be possible to speed up the process by removing unnecessary columns from the extract. This will reduce the time required for generating, sorting, and compressing the remaining columns. Should the problem persist, you may want to ensure that all fields have the appropriate data type assigned to them in the underlying database. Improperly defined field types in the source database can affect the performance during the extract creation, as well as any subsequent queries needed to be performed against the extract file.

If extract generation speeds are still not good enough, try running more data engine processes or placing them on their own Worker instance.

BLENDING DATA

The amount of time that Tableau Server spends performing data blends is the *Blending Data event*. This event can take a long time when working with large amounts of data from the blended data sources. Filtering before the blend at the data source level can be effective. If possible, consider moving data into a single data source so that joins can be used instead of blending.

SERVER RENDERING

The amount of time that Tableau Server spends rendering the computed layouts into a format to send to the client browser is the *Server Rendering event*. The time it takes to complete this event can be impacted by the load on the VizQL processes as well as the complexity of the layouts. Refer to Tableau Server's online manual and the Interpret a Performance Recording section for additional information.

Whether specifically mentioned or not, most of these events can be quickened by restricting the amount of data visualized through filtering or aggregation. This can also be achieved by using faster hardware or adding more resources on Tableau Server. As far as workbook performances goes, if it doesn't perform well in Tableau Desktop, it won't perform well in Tableau Server either. For this reason you should use the Performance Recorder on the desktop to troubleshoot performance issues there before publishing an under-performing workbook to the server.

MANAGING TABLEAU SERVER IN THE CLOUD

Increasingly, organizations are choosing to move away from hosting on-premise servers by migrating to cloud-based solutions. Flexibility and decreased initial costs are two reasons for pushing software into the cloud.

WHAT DOES IT MEAN TO BE IN THE CLOUD?

Before I discuss cloud-based Tableau Server hosting options, it might be helpful to define what I mean by "cloud-based." The expression "in the cloud" has become a catchall term in recent years for any service that isn't hosted by an on-premise server. That definition doesn't quite capture the scaling implications of the cloud, though. Cloud solutions are typically hosted and rapidly scalable systems. As mentioned at the beginning of this chapter, Tableau Software has two server versions that operate only in the cloud.

TABLEAU'S CLOUD-BASED VERSIONS OF SERVER

Tableau Public is a Tableau Server implementation hosted by Tableau Software that is free to use but comes with some caveats. Chief among these is that all workbooks and data hosted on Tableau Public are just that, public. This is probably a deal-breaker for most organizations. However, if your organization wants to make data available to the public anyway, this is a great (free) solution. Other caveats with Tableau Public are

- Data sources are limited to 10,000,000 rows per data source.
- Only file-based data sources can be used.
- Data limited to 10GB per account.

Tableau Online (fee-based) provides an added measure of control and security beyond Tableau Public. It is a cloud-based version of Tableau Server that is licensed on a per-named-user basis with no minimum requirement on the number of licenses. The software is installed and maintained by Tableau Software in a secure hosting facility. It is easy to use Tableau Online. Once you have signed up, you can start publishing workbooks for other licensed Tableau Online users to view.

There are a few differences between Tableau Online and Tableau Server including the following:

- Workbooks published to Tableau online must use Tableau Data Extract, which must be refreshed regularly. Live connections to Amazon Redshift are supported as well.
- No guest access. Everyone using Tableau Online must be licensed to use the service.
- Tableau Software creates and maintains your site.
- No minimum user requirement.

Tableau Software continues to make additional features available via Tableau Online. Custom branding is now supported (June 2015); SAML authentication (May 2015) and Tableau Online are now able to auto sync with other cloud services such as Salesforce and Google Analytics. And you can sync with data stored behind your firewall.

At the beginning of this chapter, I introduced Tableau's three different server products: Tableau Server, Tableau Pubic, and Tableau Online. Currently, the majority of Tableau Server customers want to host Tableau Server on premise, behind their company firewall. But an increasing number of organizations are choosing to host Tableau Server in the cloud.

PUTTING TABLEAU SERVER IN THE CLOUD

Although Tableau Server is most frequently hosted within company networks, it can also be hosted in the cloud or by utilizing Amazon EC2 instances and most other services that provide cloud-based Windows Server platforms. Amazon EC2 is not currently a platform supported by Tableau Software, but it does work. There are a few items to consider if you want to deploy Tableau Server using a cloud service provider. You are still fully responsible for the installation and maintenance of Tableau Server deployed this way—unless you also want to farm out this work on a contract basis to consultants.

Tableau Server needs to be accessible to your users, so make sure ports are opened in any firewalls and that the server will accept traffic from your users' network addresses. Active Directory integration can be tricky with these platforms, so consider local authentication if you encounter issues.

When deploying Tableau Server in multi-node configurations, ensure that the IP addresses of the nodes are static so that node communication won't be impaired through system restarts. Also ensure that firewall rules are in place to allow nodes to communicate with one another. The most common issues with running Tableau Server in a cloud environment are networking related. Once the Tableau Server is installed and accessible, administering it is similar to administering a locally installed host.

MONITORING ACTIVITY ON TABLEAU SERVER

As your server deployment grows, you can monitor usage activity to ensure the best experience for your users. Tableau Server includes a Server Status page that you see in Figure 11-22.

The Server Status page is divided into four different sections: Process Status, Analysis, Log Files, and a button for rebuilding the Search & Browse process index.

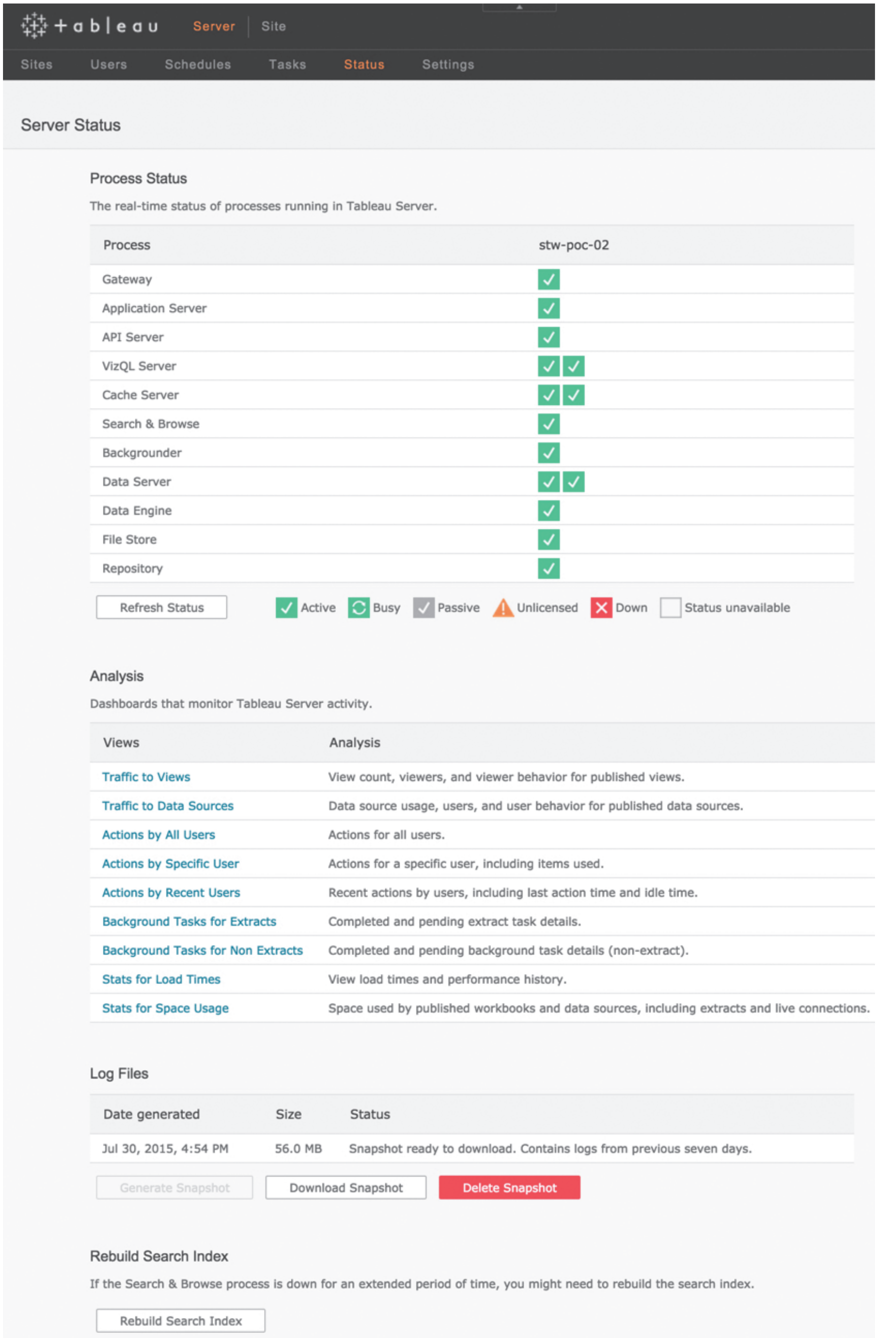


FIGURE 11-22 *Server Status page*

STATUS SECTION

The Process Status section displays the current status for every process available on each machine deployed. The example in Figure 11-22 is for a single server. If you have a multi-cluster setup, you will see each machine's IP address in separate columns along with the processes assigned to the Primary and Worker Servers.

ANALYSIS SECTION

The Analysis section provides links to embedded Tableau workbooks that provide visual analysis of important metrics related to your server activity. This was improved dramatically with the release of V9. There are nine different canned reports available including:

- Traffic to views
- Traffic to data sources
- Actions by all users
- Actions by a specific user
- Actions by recent users
- Background tasks for extracts
- Background tasks for non-extracts
- Stats for load times
- Stats for space usage

Figure 11-23 shows the Traffic to Views page.

Opening any one of the views provides access to the other views within the workbook via tabbed access across the top of the workbook web page as well.

LOG FILES SECTION

Beginning with Tableau Server 8.2, access to generate and review log files became much easier. To generate a snapshot of logs, navigate to Server ➢ Status and select the Generate Snapshot button. Once the generation of the snapshot is completed, it is available to review using the Download Snapshot button shown in the Log Files section in Figure 11-22.

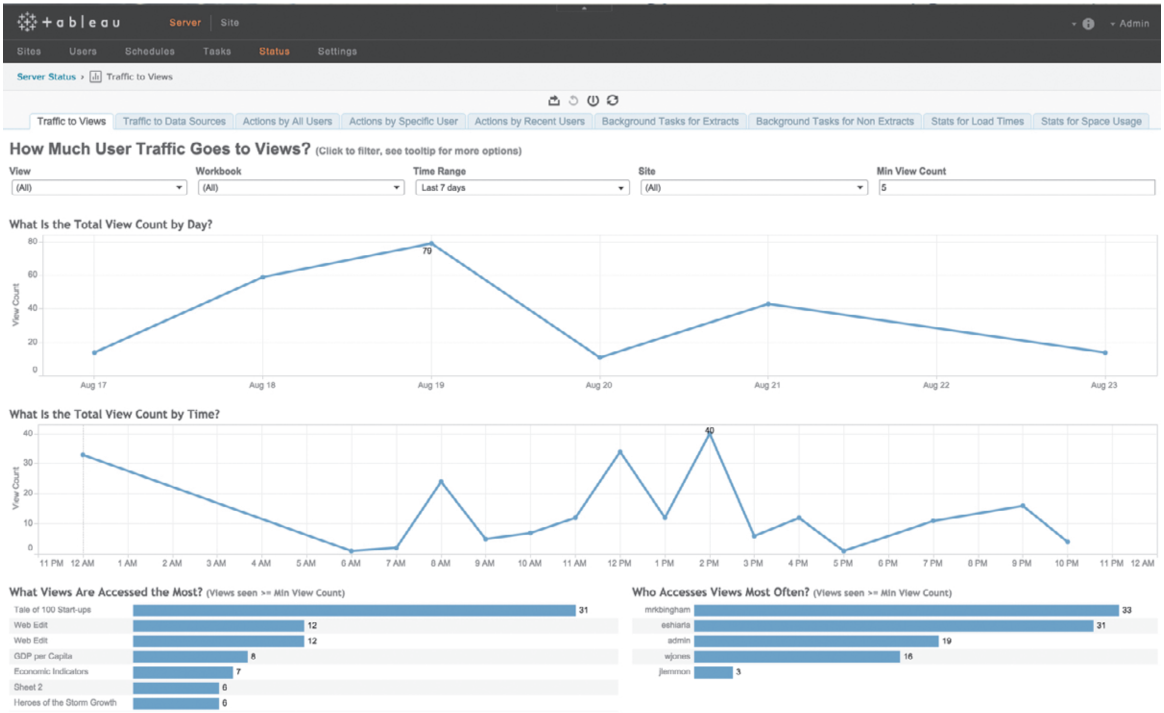


FIGURE 11-23 *Traffic to Views page*

REBUILT SEARCH INDEX SECTION

A problem could occur in your server setup that causes the server's search index to become corrupted. If this happens, users may not get the correct results when searching for workbooks or data sources. Run the Rebuild Search Index if searches are not retrieving the expected results.

EDITING SERVER SETTINGS AND MONITORING LICENSING

Separate pages are provided that give access to additional server settings and for monitoring Tableau Server licenses.

SERVER SETTINGS GENERAL PAGE

General server settings are accessed through the Server ➢ Settings ➢ General page (see Figure 11-24).

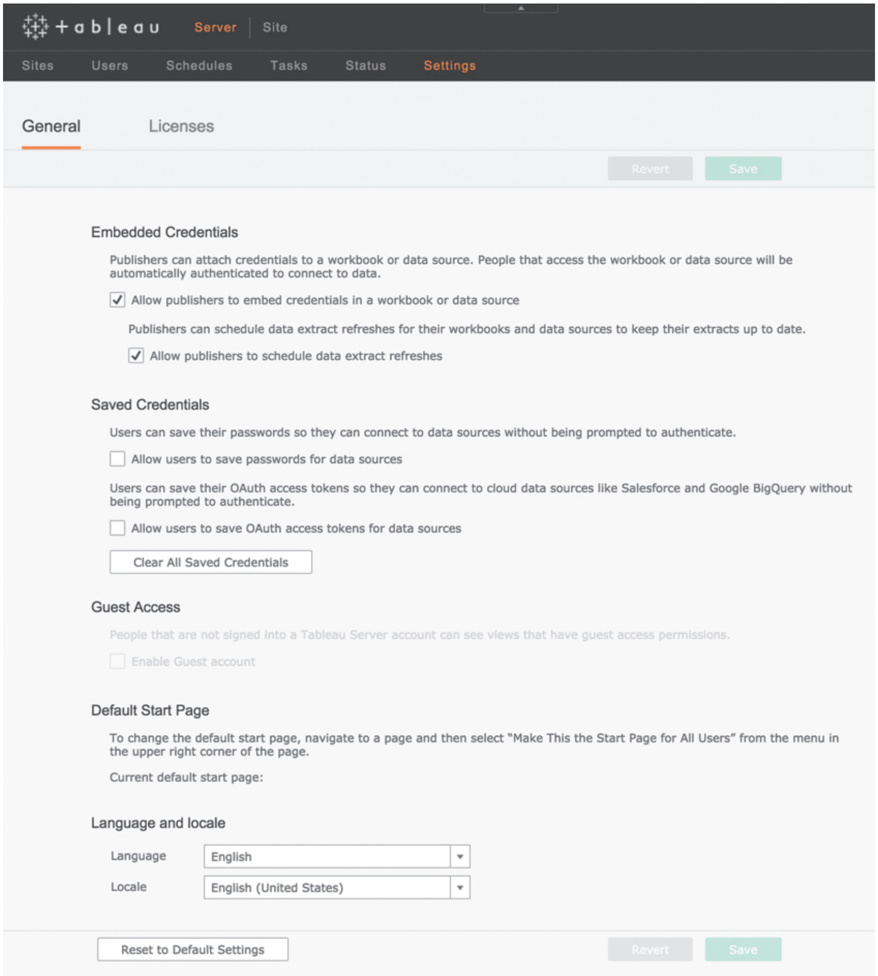


FIGURE 11-24 *General server settings*

From the General page, you can control whether publishers can embed or save credentials and whether or not to allow guest access to Tableau Server. You can also define a default start page or set the default language and locale (covered previously).

SERVER SETTING LICENSE PAGE

Your server license key, seats, and maintenance expiration date are displayed here as well as the current seat licenses in use, licenses available for use, and the number of unlicensed users. Tableau Software has made significant strides improving the tools available for the security, reporting, and administration of the site.

PARTNER ADD-ON TOOLKITS

Tableau's growing partner ecosystem now provides add-on tools for Tableau Desktop and Tableau Server that may include additional capabilities that will help you manage Tableau Server. See Appendix A for summary information on Tableau Software's product line and add-on products provided by Tableau Partners.

NOTES

1. Seth Godin, *Linchpin: Are You Indispensable?* (Penguin Group, 2010), 154.
2. Neelesh Kamkolkar, "Tableau Server 9.0 Scalability: Powering Self Service Analytics at Scale," 2015, `http://www.tableau.com/learn/whitepapers/tableau-server-90-scalability-powering-self-service-analytics-scale`.
3. InterWorks Europe, "Comparing Performance on Tableau 8.3 Server vs. Tableau 9.0 Server," blog entry by Glen Robinson, February 12, 2015, `http://interworks.co.uk/blog/comparing-performance-tableau-8-3-server-vs-tableau-9-0-server/`.

CHAPTER 12

Managing Tableau Server

All good-to-great companies began the process of finding a path to greatness by confronting the brutal facts.

—JAMES COLLINS[1]

Tableau Server facilitates information sharing and team collaboration by making interactive dashboards and views accessible to authorized individuals via any of the popular web browsers available today. Reports can be directly consumed via iOS (Apple) or Android devices. Beginning with Tableau Server Version 8, authorized staff can edit existing reports or create new analysis using Tableau Server.

Users can also share metadata including joins, groupings, sets, name aliases, and other customized data by publishing Tableau Data Source files to Server. You will learn how to take advantage of these features and more in this chapter.

MANAGING PUBLISHED DASHBOARDS IN TABLEAU SERVER

After Tableau Server is installed, those creating reports and analysis must be provided publishing rights. Staff who will be consuming reports must be granted access rights. Once you've created a workbook containing at least one worksheet, you can publish that information to Tableau Server. Workbooks containing many different worksheets and dashboards can be published in full, or by selecting any combination of worksheets and dashboards, you can publish a subset of what is contained in your workbook. Figure 12-1 shows the Tableau Desktop menus used for publishing to Tableau Server.

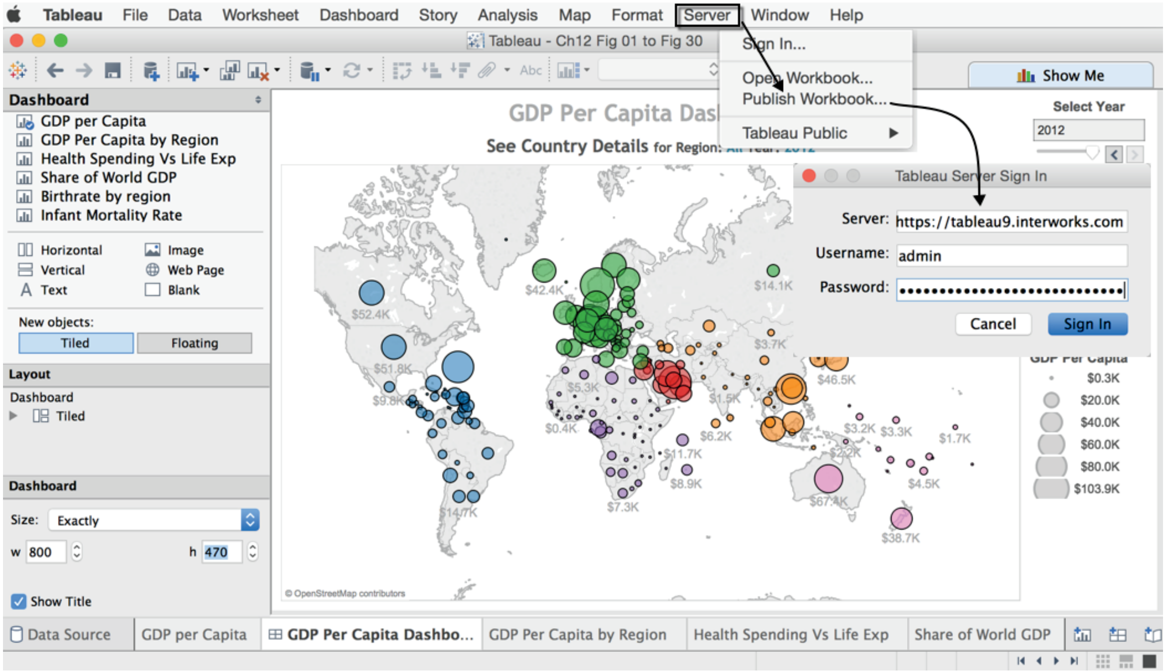


FIGURE 12-1 *Publishing from Tableau Desktop to Server*

Publishing a workbook requires three steps:

1. Open the workbook you wish to publish.
2. Select the Server menu and click Publish Workbook.
3. Enter the server URL, your username, and your password, and click OK.

The Publish Workbook to Tableau Server dialog box will appear, as you see in Figure 12–2.

Using this menu, you define when, how, and what details will be published to the server. If your workbook's data source is a Tableau Data Extract (`.tde`) file, you can also schedule regular data update using the Scheduling and Authentication button at the bottom left of Figure 12-2.

You can tell Tableau to organize published workbooks in a variety of ways:

- **Project:** Folders for grouping workbooks
- **Name:** Naming workbooks
- **Description:** Providing descriptive details workbooks
- **Tags:** User-defined tagging of workbooks

- **Permissions:** Controlling what users are permitted to do and who can access the workbook
- **View:** Hiding or sharing specific views

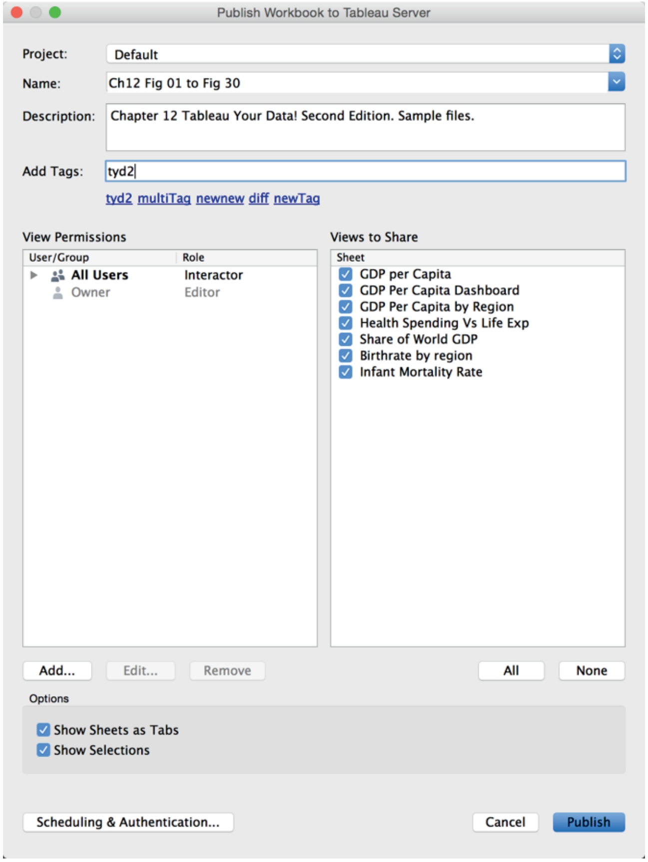


FIGURE 12-2 *Publishing dialog box*

Combining these different folders and views gives you the capability to provide access to your work to the appropriate level for individuals, teams, work groups, and projects. The specific purpose of each of these is explained in more detail in the material that follows.

PROJECT

Projects are folders for organizing your reports and controlling access to those reports. Server comes with one Default project folder. Those with administrative rights can create additional projects. Figure 12–2 shows a workbook called Ch12 Fig 01 to Fig 30 being published to project Default.

NAME

You can accept the name assigned to the workbook when it was created in Tableau Desktop or choose to define a new name that will appear in Tableau Server when the workbook is published. You define new workbook names using the Name field shown in Figure 12-2.

TAGS

Tagging published workbooks is optional but provides another way to search for reports. These can be helpful if you publish a large number of reports. Enter each tag separated by a comma or space. If the tag you are entering contains a space, surround the tag with quotation marks (for example, "Production Benchmarks"). In my example, the Add Tags field includes a tag called `tyd2`.

VIEWS TO SHARE

The Views to Share option allows you to select specific sheets or dashboards that you wish to include in what you publish to Tableau Server. Notice that all of the worksheets in Figure 12-2 are being published. Any sheets that are not selected are hidden on Tableau Server, but these are still available within the workbook if it is downloaded from Tableau Server.

OPTIONS

Appearing at the bottom of the publishing dialog box are more optional selections that control the appearance of what is published.

Selecting the Shows Sheets as Tabs option will generate tabs when the report is published to Tableau Server—facilitating navigation between worksheets and dashboards in the published workbook. The Show Selections option allows selections you've made on a worksheet or dashboard to persist when the workbook is published to Server and will be displayed to users consuming the workbook.

If the data source that you are using for the report being published comes from an external database or file, you will also see a check box for the inclusion of external files; selecting that option generates a copy of the source file on Tableau Server. Custom image files used in any view will also be saved. If you have a live database or an extract file being utilized by the workbook, you should also see a button in the lower left for Scheduling and Authentication.

Selecting that button allows you to set the refresh schedule for data extract sources or to change how a live database connection is authenticated on Tableau Server. Details regarding scheduling updates and authentication are covered in-depth later in this chapter.

Select the Publish button shown at the bottom of Figure 12-2 to initiate the upload to Tableau Server. Upon completion, a pop-up will appear displaying the newly published workbook.

If your Tableau Server instance is configured for multiple sites, you will also see a Select Site dialog box to define on which server site the workbook will be published. Tableau's default is a single site. Multiple sites are partitions of the same physical server.

EDIT

Those authorized to do so can optionally add, edit, or remove permissions for all users, groups, or individual users by selecting the Edit button and using the View Permissions tool, as shown in Figure 12-3. This menu allows you to see and edit permission types for different roles (viewer, interactor, editor, or custom).

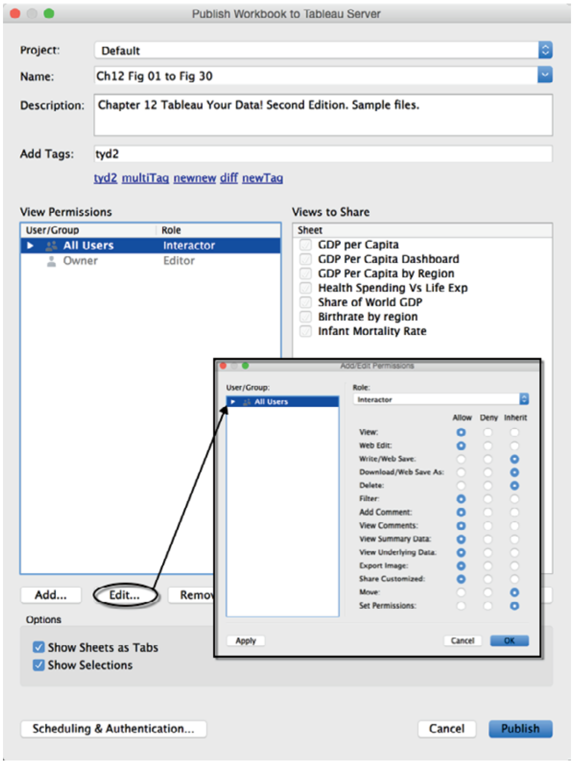


FIGURE 12-3 *Adding and editing roles and permissions*

The menu window exposed in Figure 12-3 shows the role definition details. Clicking the arrow to the left of All Users will expose different user types. Clicking the blue drop-down for the Role field displays other role security and access options. In this case, you see the defined permissions for the Interactor role.

NAVIGATING TABLEAU SERVER

When logging into Tableau Server, the user interface will vary depending on the user's administrative rights. As you would expect, administrators have access to views for reviewing metadata related to different published workbooks, the number of views, data sources, who owns the material, and when the material was created. Normal users (interactors) have search tools that facilitate access to content. Figure 12-4 shows an administrative Site view of Content.

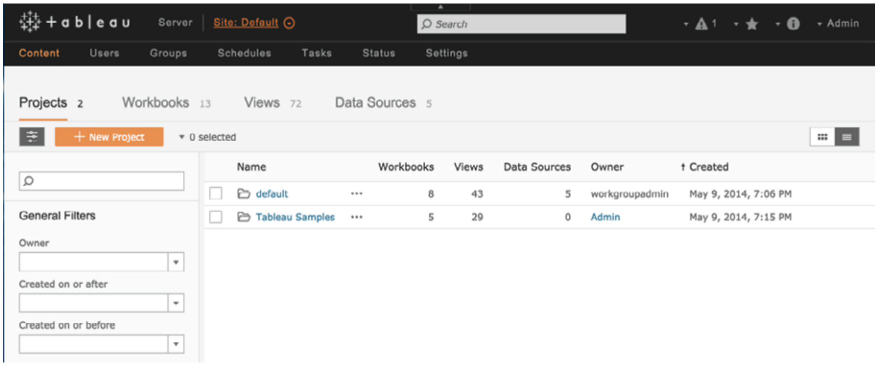


FIGURE 12-4 *Site Content view*

Administrators can create new projects, filter for owners, and filter before or after dates of interest. Across the top of the view, you see counts of projects, workbooks, views, and data sources. Figure 12-5 shows an interactor (non-administrative) view of the site.

The non-administrative view provides access to data that the user has been authorized to view but doesn't include access to functions reserved for users with administrative responsibilities. You can see the difference comparing the top menus shown in Figures 12-4 and 12-5. A normal user has access to published data and data sources. The administrator gets access to Content, Users, Groups, Schedules, Tasks, Status, and Settings.

Figure 12-6 shows you one way you can navigate from a global workbook view into the Tableau Samples content.

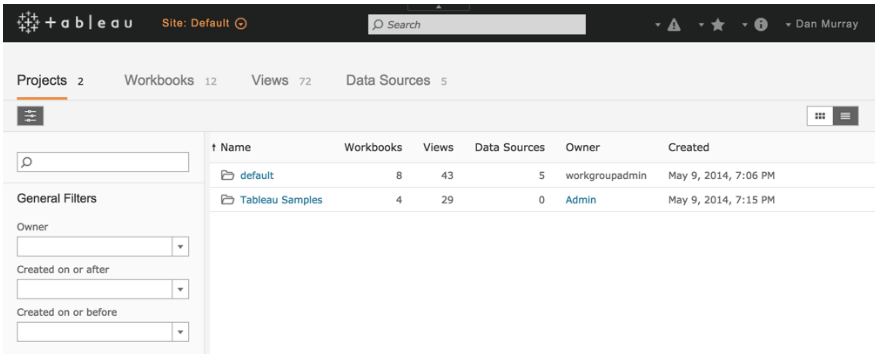


FIGURE 12-5 *Non-administrator search view*

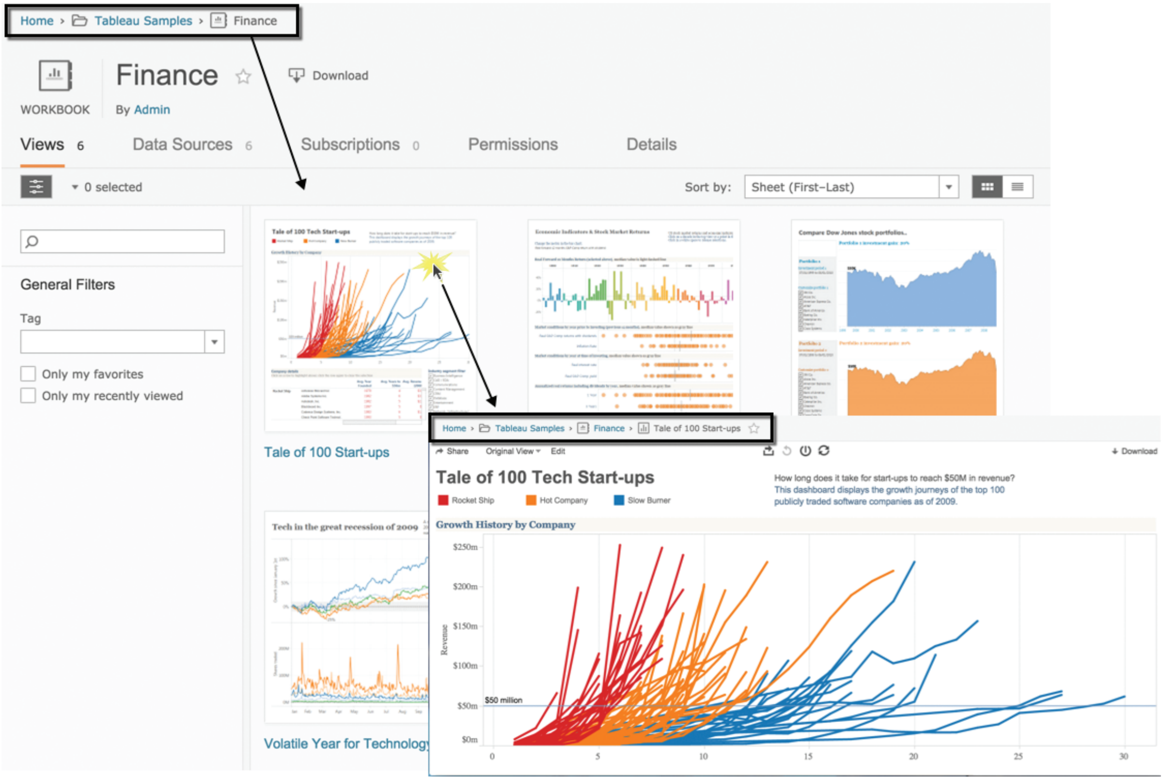


FIGURE 12-6 *Navigation breadcrumbs*

Notice the navigation breadcrumb trail that Tableau Server provides. Figure 12-6 shows this in the upper-left side of the browser window. The initial view from Home ➢ Tableau Samples ➢ Finance is more detailed when the Tale of 100 Tech

Start-ups workbook is selected. If you are looking for content but are unable to find it using the default navigation, there are other ways you can search by filtering. Figure 12-7 shows a filtering for “sales” workbooks.

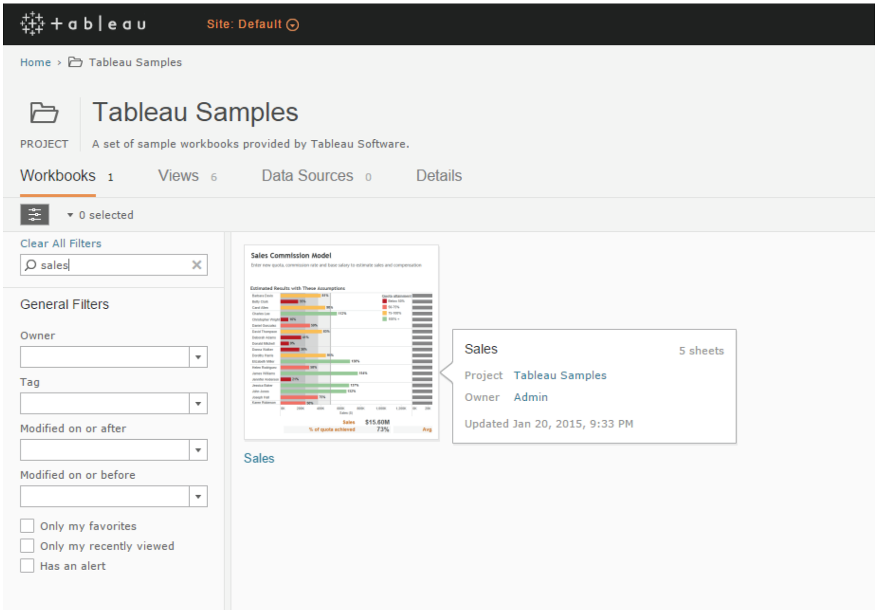


FIGURE 12-7 *Searching by filter*

If you have a lot of content on Tableau Server, searching by using filters can get you quickly to relevant files, as you see in Figure 12-7. Filtering for “sales” while in a project folder restricts the filter search to the content of that folder.

Searching for “sales” using the global search at the top-right of the browser window will return any content in Tableau Server that contains the word “sales” in the title, caption, or other parts of the workbooks. Figure 12-8 shows the result of a global search.

If you have a lot of content on Tableau Server, using the search functions saves time, but having an organizational plan for your workbooks will help you control access and provide your users with easy ways to navigate to the content that they are interested in.

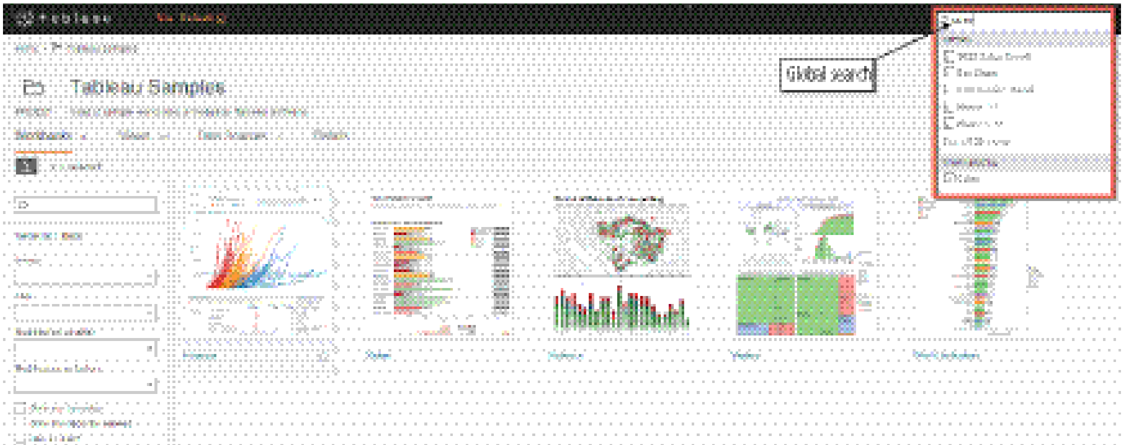


FIGURE 12-8 *Using global search*

ORGANIZING REPORTS FOR CONSUMPTION

Publishing reports to the web is the first step in effectively sharing information on Tableau Server. As your user base grows and reports proliferate, finding reports you're interested in requires organization. User security, group security, and site security were discussed in Chapter 11. Tableau Server provides additional ways to organize reports though projects and tags. Projects are virtual folders in which you publish workbooks and data sources. Tableau also provides built-in support for adding security to each project—allowing you to more easily manage security across multiple workbooks and data sources. Users can also tag particular workbooks or data source files with keywords. This provides you with a user-defined search term that is helpful for locating files when there are a lot of published workbooks. Defining a sensible framework for projects and tag recommendations might be helpful to your user base and provide some consistency across your enterprise. These could be defined in advance, but you may allow users to define additional tags that meet their specific needs as well.

For example, you may define projects by business unit or function—leveraging tags and adding context to each search. In a university setting, three different departments might be consuming reports:

- Admissions
- Financial Aid
- Career Services

The Admissions office might be concerned with tracking the number of students applying each year and whether they were accepted and enrolled. Financial Aid would like to track the amount of aid offered and accepted. Career Services might be interested in monitoring the progress of students that have graduated and are seeking employment. Setting up projects for each office could be advantageous because it will facilitate security while organizing the reports logically for the staff of each department.

Adding tags to each workbook could provide additional context regarding the details. Examples in a university might include the following:

- **Admissions:** Undergraduate admissions, "accepted vs. denied," enrolled, declined, graduate
- **Financial Aid:** Aid, grants, loans, scholarships, transfer scholarships, undergraduate, graduate
- **Career Services:** Offers, accepted offers, max salaries, median salaries, undergraduate, graduate

Look closely at the admission tag example included in the list before "accepted vs. denied." When tags include spaces, they must be wrapped in quotations. Notice the same tags are being used in different projects and workbooks. This allows a user to search for similar analysis performed across different departments.

For example, if the dean of a college wanted to quickly find all the reports available analyzing undergraduate students, the dean could search using the Undergraduate tag and quickly access reports related to admissions, financial aid, and career services.

ADDING TAGS TO WORKBOOKS

Users can add tags to any workbook they have the right to access. Figure 12-9 shows a tag being added to one workbook contained within a project.

On the left side of the window in Figure 12-9 you can see the view has been filtered for the Tableau Samples project. Selecting a workbook and clicking the tag menu option exposes existing tags that can be applied to the selected workbook. If none of the existing tags works, clicking the menu in the upper-right side of the workbook image opens a dialog box that can be used for entering a new tag. In Figure 12-9, you can see tag the tag option in the menu. Selecting that will expose another dialog box for entering the desired tag. The Change Tags dialog box includes an edit to add a tag for "sales."

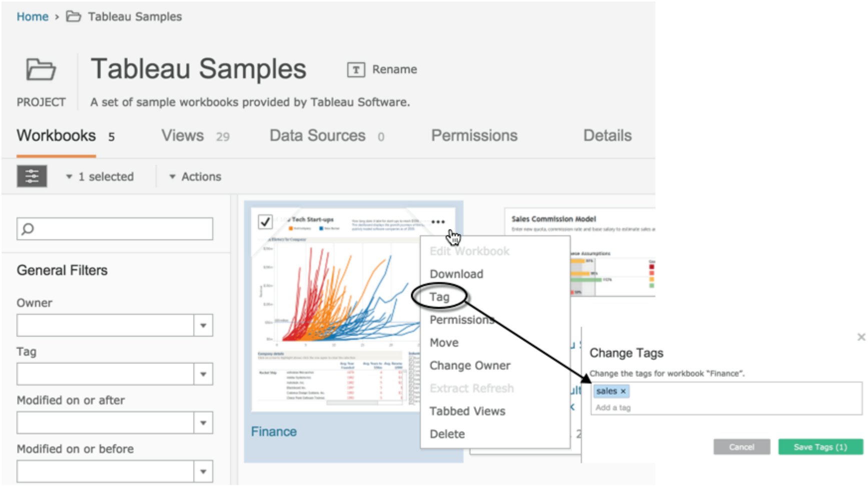


FIGURE 12-9 *Adding a tag to a workbook*

Tags can also be added directly when publishing a workbook. This option is located in the menu that is presented when you publish a workbook.

CREATING A FAVORITE

Favorites are workbook views or dashboards that you use often and want to save for quick access. They are accessible via the star drop-down menu in the upper-right corner of the browser window or from the click box menu on the left side of the screen in the General Filters area. Figure 12-10 shows the favorite menu. It shows three different tagged workbooks including Sales, Science, and Variety. Favorites are indicated with a gold star in the view.

Figure 12-10 shows five workbooks. Three have been defined as favorites. Any workbook or worksheet can be made into a favorite from the thumbnail or list views by clicking the star below the view of the workbook. Figure 12-11 shows a list view of the workbook that has been filtered to display only favorites.

To make any workbook a favorite, select the star associated with it. This will toggle the star—coloring it yellow—and will add the item to the favorites menu in the upper right, as shown in Figure 12-10. You can also add a favorite from the thumbnail view. The most convenient way to access favorites is via the star drop-down menu that you see in the upper right of Figure 12-10. You can also filter for favorites by using the General Filters menu you see on the left side of Figure 12-11. The check mark in the Only my favorites option indicates that the view is displaying only favorited workbooks. If that view had not been filtered for favorites, you would see all of the workbooks, including ones that have not been favorited. Items not favorited are indicated with a clear star.

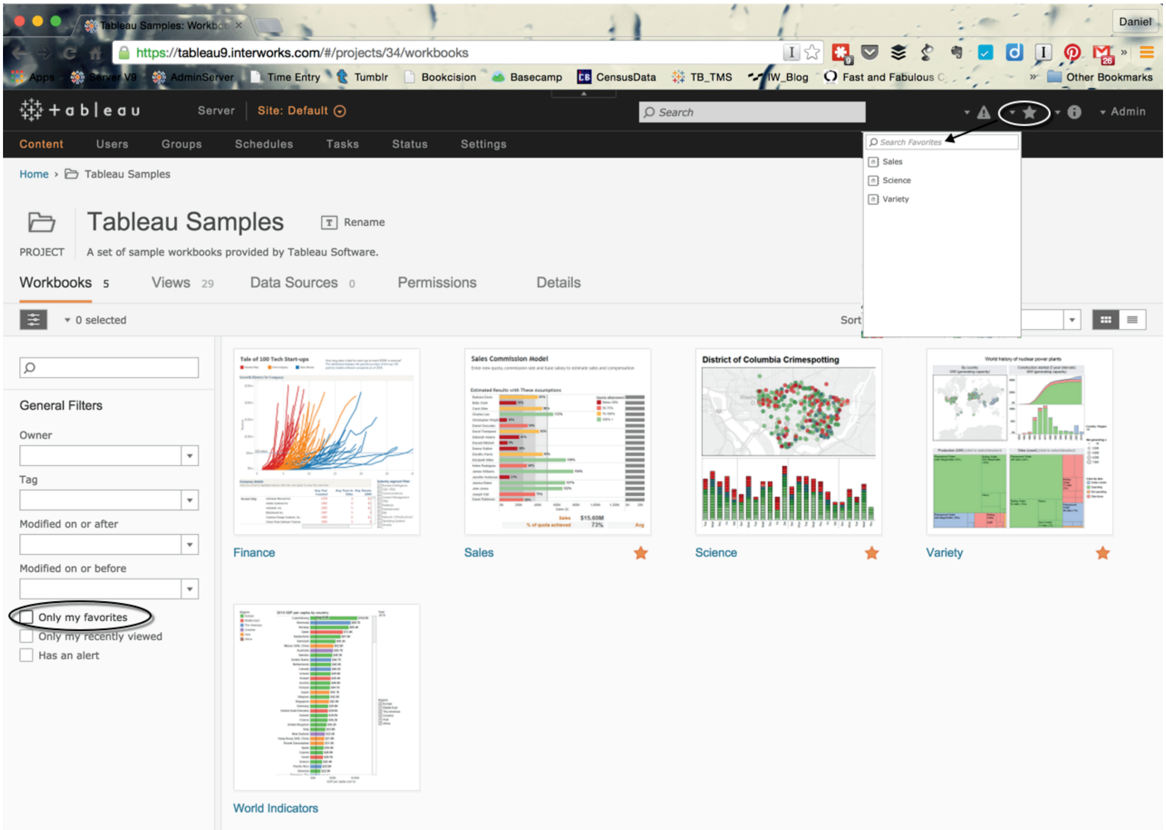


FIGURE 12-10 *Favorites*

By combining user and group security with projects, favorites, and tags, you can control access to sensitive information and allow users to set up their own means for facilitating easy access to the information that is most important to meet their particular needs.

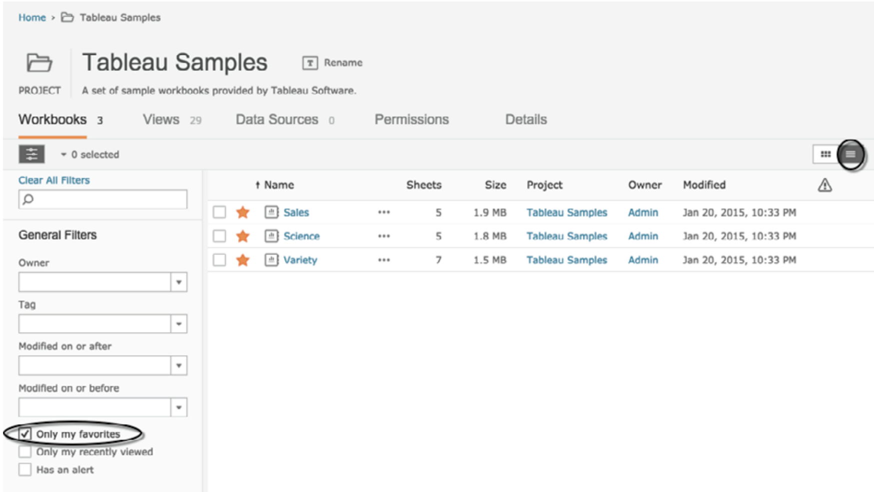


FIGURE 12-11 *List view filtered for favorites*

OPTIONS FOR SECURING REPORTS

Managing the security of data and reports is an important consideration. With the exception of a core-licensed server (with guest accounts enabled) all users must log in to Tableau Server before they have access to view any information. Applying permissions at the project level, you can efficiently manage access to a large number of workbooks and data sources while still providing the flexibility to alter security for a single group or user at any time. Securing reports is done using a combination of application layer and data layer controls.

- **The Application Layer:** Tableau Server credentials
- **The Data Layer:** Database security

THE APPLICATION LAYER

In Chapter 11, you learned about details related to managing security for users along with external security protocols that are supported by Tableau Server including External and Internal Secure Sockets Layer (SSL), Security Assertion Markup Language (SAML), and Symmetric key cryptography supported via Kerberos and smart cards. Please refer to Chapter 11 for more details on these external protocols.

Tableau Server provides application layer security through user credentials. Users can be managed in one of three ways:

- Local authentication

- Microsoft Active Director
- Trusted Ticket Authentication

Once a user has been authenticated to access the Tableau Server environment, you specify which projects, workbooks, and data sources that user is permitted to see. This is called object-level security. Tableau supports the assignment of object-level permissions for any user group or user by utilizing any of the following objects:

- Project
- Workbook
- Data source

In a top-down approach, permissions can be assigned at the project level—which may be inherited by any workbook or data source published to that project. Permissions assigned to a user group will automatically propagate to all users within the group unless a user has explicit permissions overriding the group settings. The publisher has ultimate control over whether to accept the default permissions or define customized permissions. Tableau Server comes with three standard permission levels already defined. These are called Roles and include Viewer, Interactor, and Editor. Figure 12-12 shows the interactor role permissions.

The permissions menu is accessed when your workbook is published by clicking the Add button below the View Permissions area, as shown in Figure 12–12. Other roles can be viewed by selecting the Role menu drop-down arrow. Custom roles can be defined by selecting a user or group and then choosing the Custom role option from the Role menu. You can also use the Edit button next to modify an existing role permission. This allows you to set customized permissions when assigning your custom role to a specific user or group. Tableau's manual provides additional step-by-step instructions for defining permissions. Access the appropriate section of the manual from the Help menu in Tableau and search for "Setting Permissions."

DEFINING CUSTOM ROLES

Customizing roles is done by defining the permissions for a new role type that you define or editing an existing role. Understanding the permissions that you allow is important. Depending on the selections made, you may grant the ability for people to republish reports, change filters, redesign the workbook views, build new views, export data, download the workbook, share custom views, or even set new permissions. For a detailed description of each capability, use Tableau's Help menu and search for "Permissions."

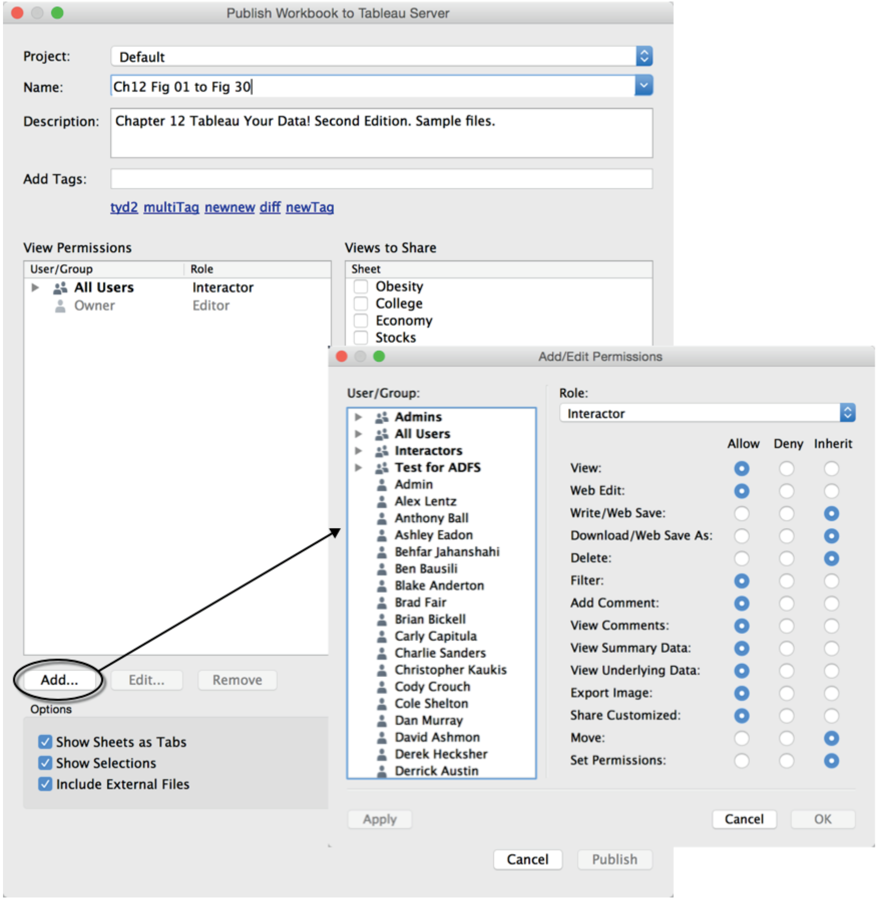


FIGURE 12-12 *Add/edit user permission*

Care should be taken when granting permission to prevent the unauthorized dissemination of data. The lists that follow categorize permissions by risk level. High-risk items provide the ability for the user to override permissions or disseminate data. Medium-risk items convey the ability to alter or export views. Low-risk permissions pertain to viewing and commenting capabilities.

High-Risk Permissions

- Write/Web Save
- Download/Web Save As
- Move
- Set Permissions
- Connect

Medium-Risk Permissions

- Web Edit
- View Summary Data
- View Underlying Data
- Export Image

Low-Risk Permissions

- View
- Delete
- Filter
- Add Comment
- View Comments

These risk assessments are meant to be guidelines only. If your data is highly sensitive, care should be taken to mask confidential information at the data source level to ensure that confidential information is not inappropriately exposed.

A PERMISSION-SETTING EXAMPLE

Permissions can be defined so that it is possible to reuse a single workbook for groups with different access rights. For example, you may choose to group users by office. This was described in the university example mentioned earlier in this chapter (admissions, financial aid, and career services). Permissions for related projects could be set so that each office gains access only to the workbooks specifically related to their individual (office) groups.

At the same time, the university president's office could access the workbook, but with different permission settings that permit access to all of the projects and all of the related data details.

As a result, financial aid users won't see the admissions or career services reports. Instead, they will only see and have access to their financial aid reports. Yet the university president's group will be able to view reports related to all three groups. Using this model, administrators can efficiently manage security for large and diverse entities.

Analyzing Existing Permissions

Because of the large number of possible permission scenarios driven by group permissions, object permissions, and roles, it is sometimes challenging to determine how to define permissions in large deployments. But, if you edit the permissions of any content type, the Permissions screen provides you with a visual and interactive visualization of the relevant permissions and how they apply to individual users. Tableau Server V9.0 improved the visualizations used for this activity considerably. Figure 12-13 shows the permissions screen for the workbook named Finance.

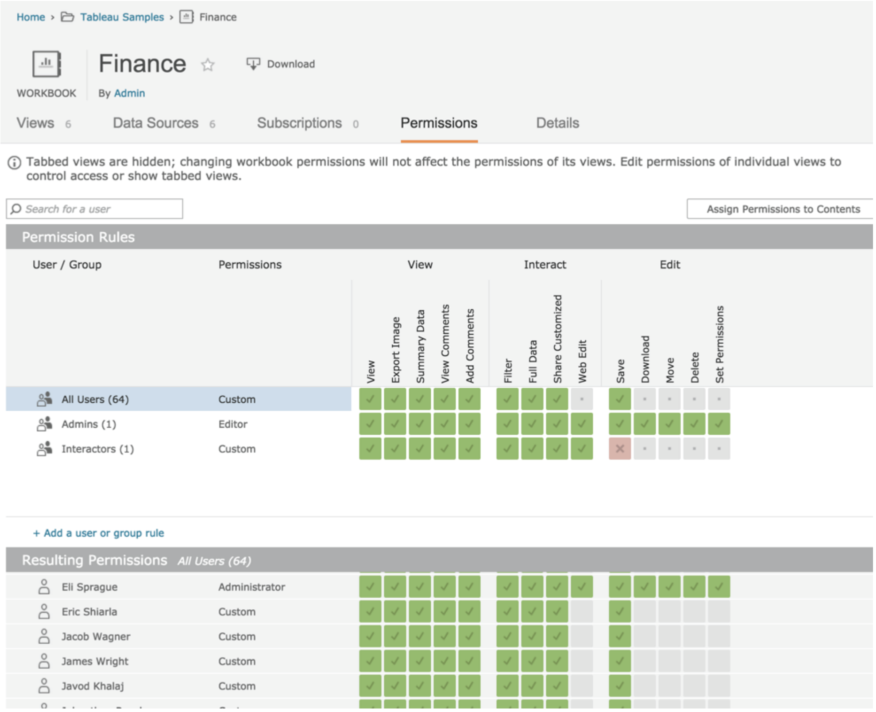


FIGURE 12-13 *Permissions rules view*

You can see there are three rules defined (All Users, Admins, and Interactors). All Users are able to save content, but the Interactors group has been denied this capability as denoted by the red box under Edit ➢ Save that you see in Figure 12-13. In the Resulting Permission area below, you can see that most users are given the ability to save this workbook. One user included in the Interactors group (Mat Hughes) has been denied that ability. If you hover your mouse over the grayed box on Mat Hughes, as shown in Figure 12-14, Tableau displays the rule applied to deny access.

The group rule applied for Interactors (of which Mat is the only member) is the source of the permission denial. The highlighting in the User/Group area helps you see the specific role in play. This visual display and highlighting are helpful, especially when you have a large number of roles, groups, and users to manage.

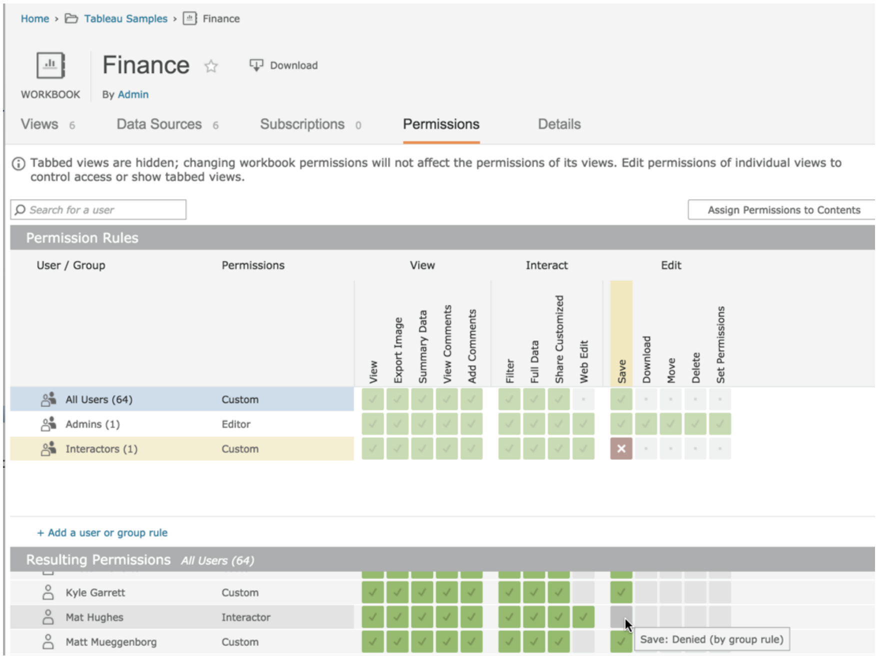


FIGURE 12-14 *Permissions denial details*

The Data Layer

When employing a live database connection in Tableau Desktop, you must provide credentials to authenticate the database server. This data-level security is persisted on Tableau Server as well. When publishing a workbook or data source, you must choose what type of authentication you'll associate with your live connection.

It's important to understand the difference between application layer and data layer security. When a user logs into Tableau Server, the user is authenticated at the application layer but not the data layer. When accessing any report that utilizes a live connection, the user must also be authenticated by that data source. How the user is authenticated is predicated upon what settings you select when publishing the workbook or data source. Your choices boil down to four options:

- Prompting the user to enter credentials
- Using embedded credentials
- Using a Server Run-As account
- Using SQL Server Impersonation (available for SQL Server only)

Embedded Credentials

Tableau also offers administrators the option to permit users to save their data source credentials across multiple visits and browsers. This is enabled through the Embedded Credentials settings option in the Server ➢ Setting ➢ General screen in Tableau Server. Figure 12-15 shows the General settings screen with the appropriate selection checked. You also have the option to embed the connection username and password for the database from within Tableau Desktop.

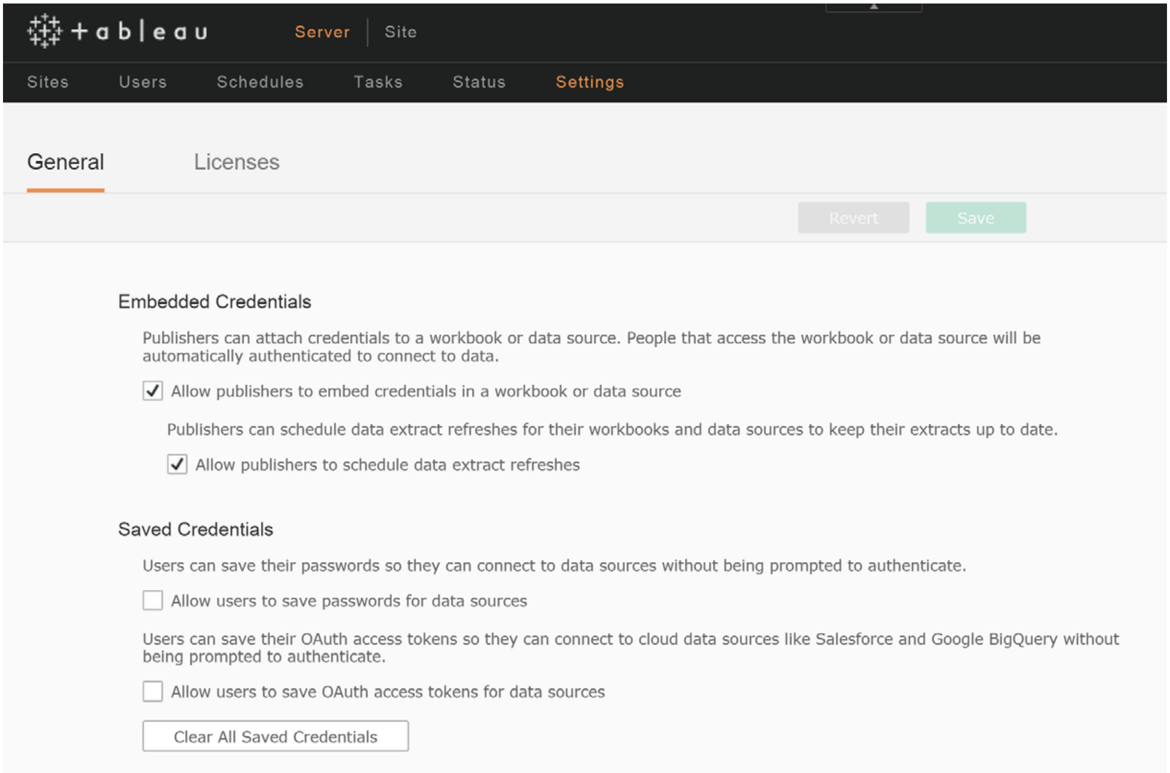


FIGURE 12-15 *Enabling embedded credentials*

By using this option, all users who utilize the connection will have the same level of access as the publisher of the workbook. This is a convenient feature for users—relieving them from having to log in a second time. However, enabling embedded credentials removes the ability to manage data-level access on a per-user basis.

Server Run As Account Via Windows Active Directory

Tableau Server runs in Windows Server environments. Therefore, Tableau Server installations utilize an Active Directory service account to run. A beneficial consequence of this fact is that Windows Active Directory (AD) can be used to eliminate redundant logins for Tableau Server users.

When a report is viewed on Tableau Server using a data connection employing this method, the Server Run As Account will be used to authenticate against the database. Your database administrator will need to ensure the Server Run As Account has the proper access to connect to and query the tables and views used in your connection. Use Tableau Server's online manual and search for Run As User to view the setup details for this feature.

SQL Server Impersonation

An option available only when connecting to a SQL Server database, impersonation is another way of eliminating the need for users to log in twice while still preserving the ability to manage data level access on a per-user basis. This also allows the SQL Server database administrator to control security policy from the database and propagate those policies to Tableau Server.

To use SQL Server Impersonation, each Tableau Server user will need individual accounts on SQL Server with credentials matching those on Tableau Server. For instance, if you have chosen to use Active Directory to manage your Tableau Server users, you must grant the same Active Directory accounts access to SQL Server. The user either will need to be the Server Run As account or have their credentials embedded in the workbook during the publishing step by selecting the Impersonate Via Embedded Password option in the authentication menu.

When a user views a workbook that has implemented SQL Server Impersonation, they are authenticated using the Server Run As account or via embedded SQL Server credentials. This account then impersonates the user connecting and accesses the database with their defined permissions. Search Tableau Server's online manual for SQL Server Impersonation for more details regarding setup and configuration.

Tableau Server provides a variety of ways to manage security. There are external security protocols that Tableau supports as well, including Security Assertion Markup Language (SAML) and Symmetric Key Cryptography (Kerberos). To find out more about these protocols, go to the Help menu to access Tableau's online manual.

The next section covers how Tableau Server provides more flexibility and efficiency through the Data Server.

IMPROVE EFFICIENCY WITH THE DATA SERVER

The Tableau Data Server provides a way to manage data sources that have been published to Tableau Server. These published sources can include direct connections to a database or Tableau Data Extract files. Authorized staff can set permissions associated with the connections and also set refresh schedules for data extract files. The metadata associated these published sources becomes available to any workbook that uses the data source. Metadata includes:

- Custom calculated fields
- Ad hoc groupings
- Ad hoc hierarchies
- Field name aliases
- Custom fonts and colors

The Data Server is efficient because it provides a flexible way to spread heavy workloads by enabling Tableau Server to absorb some of the demand normally handled by the primary database server.

While using data extract files is not a requirement, data extract files frequently perform better than the host database. The Data Server also saves time—enabling the work of a single individual to be shared by many. Data sources published to Tableau Server can be accessed by authorized Tableau Desktop users to create new analysis.

Next, you'll learn how to publish a data source to Tableau Server and then use the Data Server to centrally host and share files, schedule automatic updates, and leverage incremental extract refreshing for near real-time data.

PUBLISHING A DATA SOURCE

Publishing a data source file to the data server is done from Tableau Desktop by opening the workbook containing the data source you wish to make available for others to use. From the workbook, access the menu for publishing the data source by right-clicking in the Data window containing the data source in the upper-left section of the worksheet, as shown in Figure 12-16.

After right-clicking and selecting Publish to Server, a server login dialog box appears. You will be required to enter the server URL, your username, and password to access the server. If you have a multiple site deployment, you'll also need to enter the site you want to publish to as well. Once the server login is completed, a dialog box will appear, as you see in Figure 12-17.

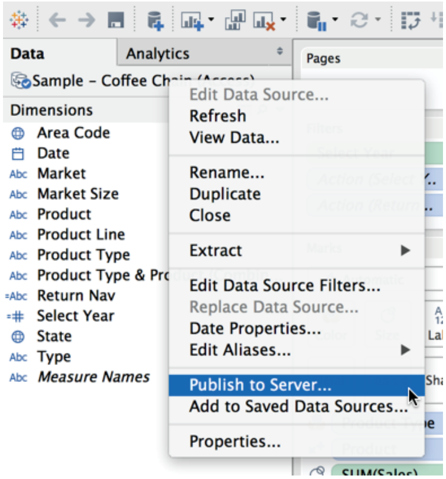


FIGURE 12-16 *Publishing to the data server*

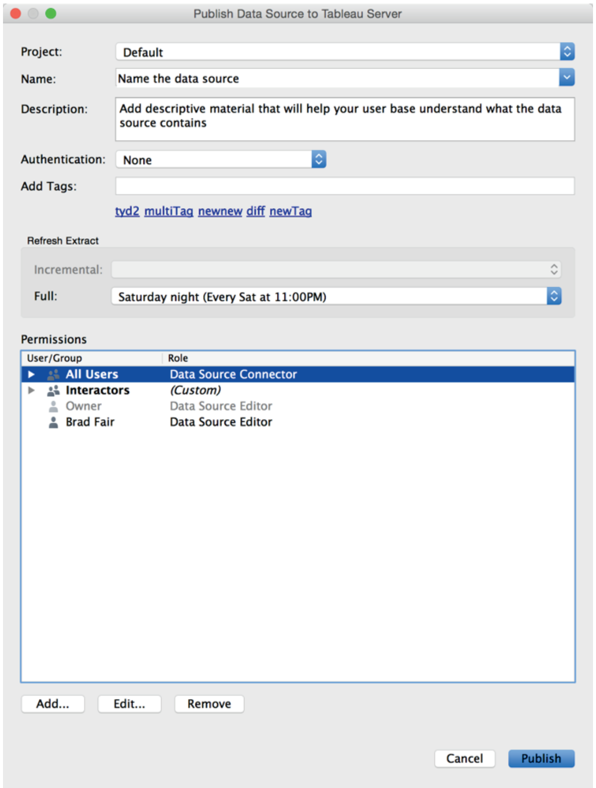


FIGURE 12-17 *Dialog box for publishing a data source*

Define the parameters for publishing the data source by selecting the project, a data source name, the authentication method, tags, how and when you want server to refresh the extract, and finally what permissions you wish to assign to the extract. Most of these topics have already been covered in Chapter 11 or in earlier sections of this chapter. In the next two sections, you will learn more about the options for updating data source files and how to use incremental updates.

Manual vs. Automatic Updates

One potential benefit of using an extract—a portable copy of your original data set—can also be a drawback. The extract may not reflect the latest changes occurring in the data source until the extract is refreshed. Tableau provides two different methods for updating extract files—manual and automatic updates.

Manual Updates Using Tableau Desktop

Manually updating data extracts can be done via the data menu or by right-clicking the data menu. Follow these steps to refresh the data source file:

1. Start Tableau Desktop if it is not already running.
2. Open the workbook containing the extract you wish to refresh.
3. Select the data menu and refresh all extracts (or add data from file to append new data).
4. A dialog box will appear displaying the extracts that are available to update.
5. Click the Refresh button to update the extract files.

If your workbook contains multiple extract files, they will all be updated using this method. You can also update individual extract files in the workbook by pointing at the data source in the data window, right-clicking, and then selecting Extract and then Refresh.

This manual process is one way you can append data from a separate source file or database—assuming the separate source contains the same fields as the original data source. To do so, follow the same steps as described, but in the last step choose Add Data From File instead of Refresh. In Chapter 13, you'll see how Tableau Server's command-line tools can be used to automate manual processes like this.

Automatic Updates Using Tableau Server

If you have many different data sources and workbooks using data source files published to Server, manually updating large numbers of files would be impractical. Tableau Server comes with a predefined update schedule and allows you to create your own custom update schedules.

To schedule updates, you'll need to first publish your extract to Tableau Server directly using Data Server or indirectly by publishing the workbook that uses an extract as its data source. During the publishing process, you have the option to select a refresh schedule to have Tableau Server automatically update the extract.

Tableau Server includes predefined schedules or your server administrator can define custom schedules set to recur at a monthly, weekly, daily, or hourly time interval. Schedules can also be defined to allow jobs to run concurrently or sequentially, with an option to change the priority of the schedule relative to others that may occur at the same time. Figure 12-17, presented earlier, shows the schedule option in the Refresh Extract section. The drop-down box next to Full contains the available options. You can see in the example that the extract will be refreshed every Saturday at 11:00 p.m.

Defining a Custom Refresh Schedule

For those users granted administrative rights, you can create custom refresh schedules from the Tableau Server admin menu. You can see these schedules in Figure 12-18.

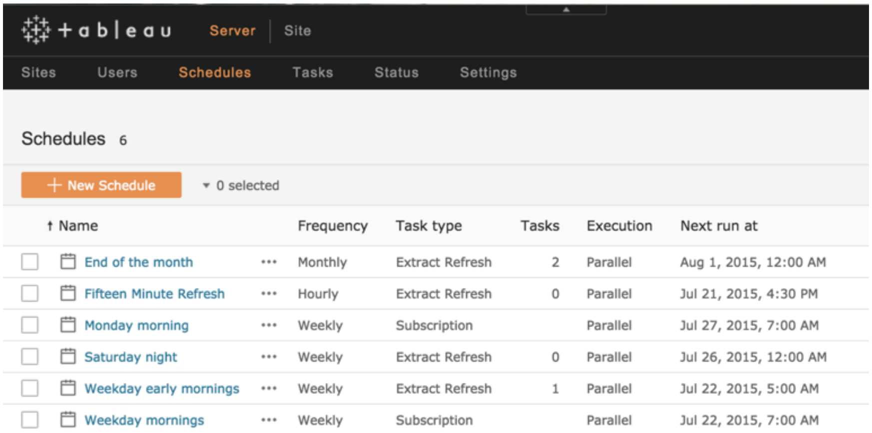


FIGURE 12-18 *Admin schedule menu*

The admin/schedules menu provides a list of what is available and summary information regarding the available schedules—their type, scope, the number of times run, how they run, as well as the next scheduled runtime. To define a new custom schedule, you must select the New menu option you can see above the check boxes in Figure 12-18. Selecting that exposes the custom schedule dialog box you see in Figure 12-19.

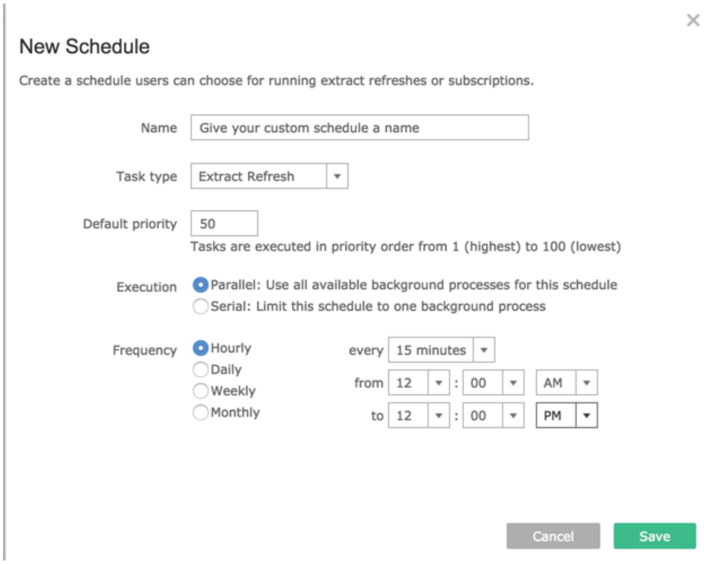


FIGURE 12-19 *Creating a custom schedule*

Give the schedule a clearly descriptive name and fill in the highlighted blanks. Then click the Create Schedule button. This makes the schedule available for use. As you can see, there is plenty of flexibility for controlling when data extracts are refreshed. In Figure 12-19, you see the minimum refresh cycle that Tableau provides is every 15 minutes.

Incremental Updates

What if you have a large or very active data source? These files can take time to update. You can reduce the time required for extracting data by using incremental updates. Typically, when refreshing extracts, the current rows are truncated and completely replaced by a new copy of the dataset. In contrast, incremental refreshes allow you to specify a date, a date-time, or an integer value field contained in your data, to specifically identify new records in the data source.

When an incremental refresh is used, Tableau will check for the maximum value of the field in your extract and compare that value to each row in the original data source—importing only the rows with a later or higher value. This approach will reduce the time required to update your extract. The larger the source file, the more significant will be your potential time savings.

You define this option in Tableau Desktop when you build the extract definition by selecting the incremental refresh option and then selecting the field that you want to use to identify the new data. The field options you see in Figure 12-20 include the dates, key records, and other fields.

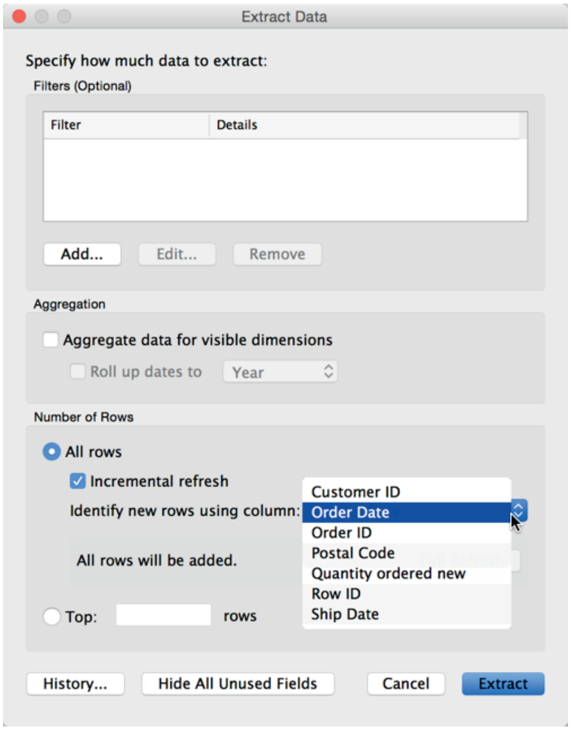


FIGURE 12-20 *Enabling incremental updates*

If you choose to use incremental refreshes, you are not excluding the option for a full refresh. On the contrary, you are only allowing for the additional choice of an incremental refresh when performing either a manual or automatic update. It is advisable to run full refreshes of the data periodically because the incremental refresh may not capture all of the changes in the source data set if your data administrator allows users to modify old (previously refreshed) data.

CONSUMING INFORMATION IN TABLEAU SERVER

As your Tableau deployment matures, you may have hundreds of reports and data sources being published, updated, and consumed. Facilitating access to information and encouraging collaboration is of primary importance and the principal value business information systems provide. Tableau Server provides tools for finding information, commenting on reports, sharing discoveries, or customizing views, Tableau Server even allows information consumers to create completely new visualizations from within Tableau Server.

FINDING INFORMATION

Tableau's security structure provides an initial level of categorization, but Tableau Server also allows information consumers to customize access further through tagging, marking favorites, and even altering existing workbooks without the need for a desktop license.

Tagging

You learned about tagging from the publishing perspective earlier in this chapter. Any user with security to access output published on Tableau Server can tag projects, workbooks, views, and data sources. While users can add and remove their own tags, administrators and publishers can see all of the tags applied to workbooks. Tags are applied from the content menus for projects, workbooks, views, or data sources. To apply the tag, select the Tag menu option at the top and either select an existing tag or enter a new tag. Figure 12-21 shows a tag being applied from within the content thumbnail view for workbooks.

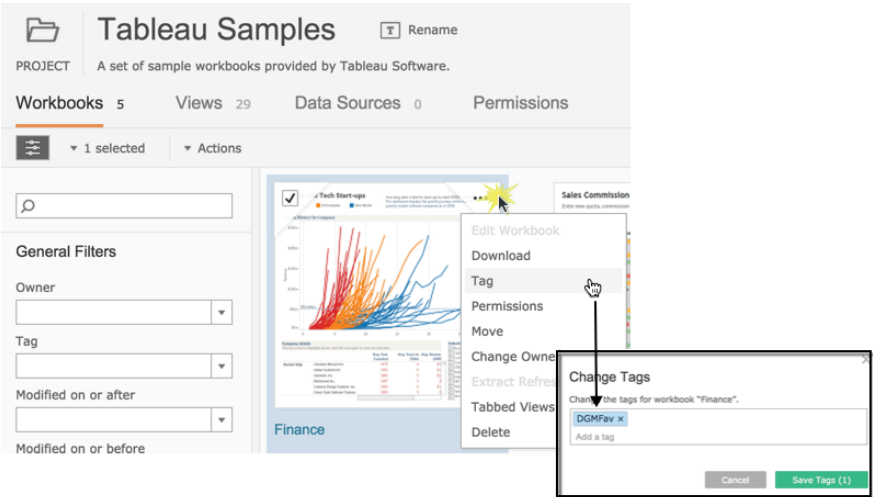


FIGURE 12-21 *Tagging workbooks*

The thumbnail workbook view checked in Figure 12-21 will have the DGMfav tag applied to it. This tag provides a means for filtering based on that tag. Note that the new tag won't appear in the Tag menu until the view is refreshed in your browser.

Removing Tags

If you want to delete a tag associated with a project, workbook, view, or data source, the process is similar to adding or changing a tag. In Figure 12-22, you see a list view of the same project folder and workbook as the last example. This time it's being presented using a list view.

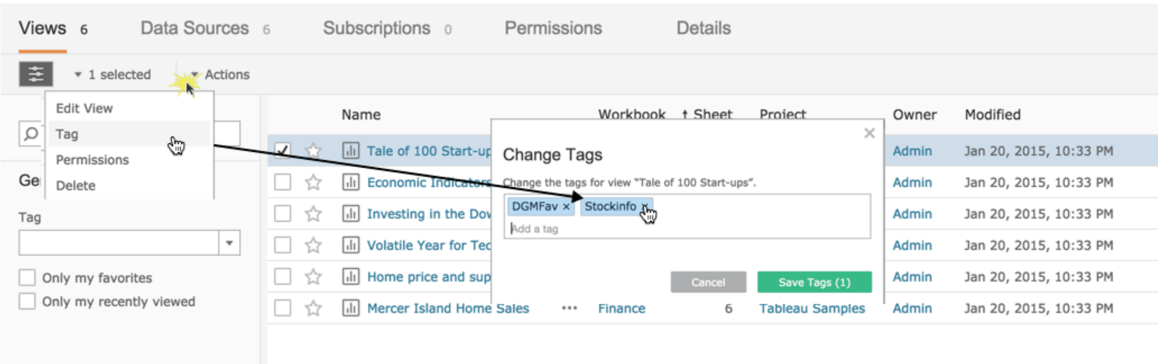


FIGURE 12-22 *Deleting a tag*

To delete a tag from the list view, select Actions ➢ Tag ➢ Change Tags, and then delete the tag. In Figure 12-22, you see the Stockinfo tag being deleted. Of course you could add more tags from this pop-up if desired. Tags are a great way for users to define search keys that specifically meet their needs without having to involve technical resources.

Favorites

Next to every workbook or view listing in Tableau Server is a star icon that allows your users to create a personal favorites list. If the icon is colored yellow, that item is a favorite. Grayed items are not. Clicking the star will add it to the favorites list. Clicking a second time toggles it off the list. The Favorites list is a bookmarking mechanism that provides fast access to your most frequently used items. Refer to Figures 12-10 and 12-11, presented earlier in the "Creating a Favorite" section of this chapter to review the details regarding how favorites are added.

Sharing Comments and Views

Comments can be added to any server view if the user has the proper permission and assuming the Post Comment save option has not been disabled through an embedded view. Comments are found at the bottom of the view in Server, giving users the opportunity to share ideas and ask questions. Figure 12-23 shows a comment being added to a workbook view.

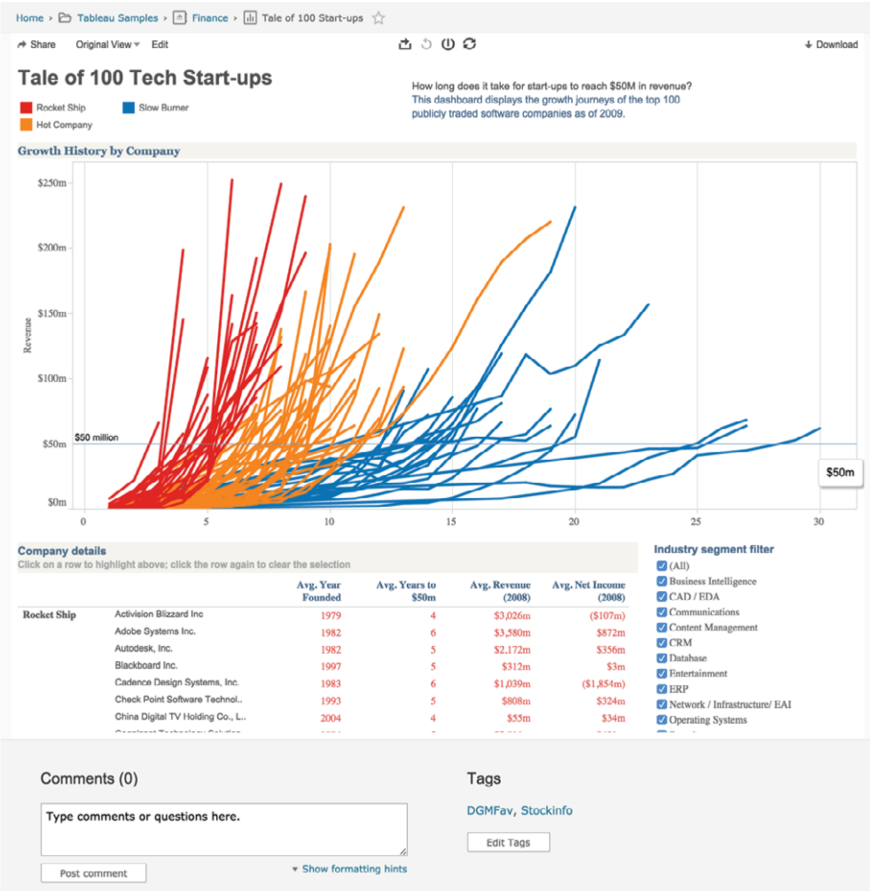


FIGURE 12-23 *Adding a comment or question*

Views can be shared with anyone via the Share menu found at the top of the page, as you see in Figure 12-24.

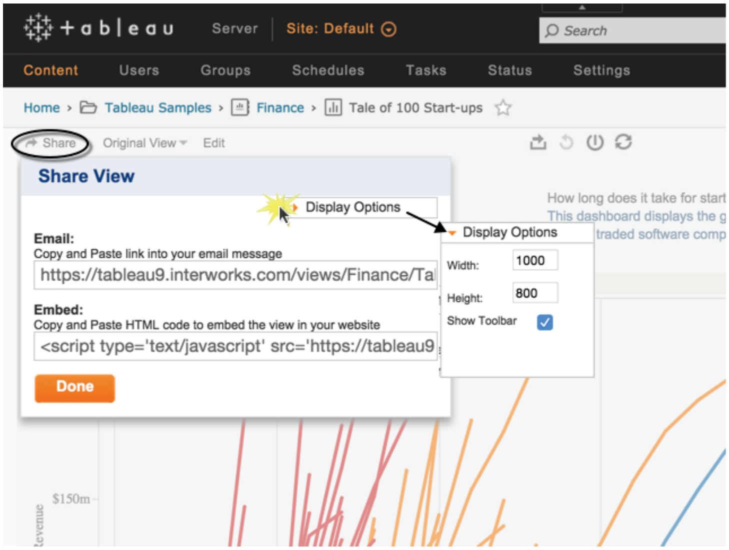


FIGURE 12-24 *Share or embed a view.*

To share the view in Figure 12-24, copy the Email link. If you want to embed the view in a blog or website, copy the Embed link and paste it into your blog or website. Both of the options allow you to optionally set the pixel height and width of the view and to define whether or not the toolbar or tabs are displayed.

Customizing Views

Users can make mark selections, apply filters or highlights, and then save those settings in a customized view. Figure 12-25 shows a customized view of a workbook filtered to show a particular industry segment.

Saving any customized view requires three steps:

1. Click the Remember my changes link.
2. Provide a name for the customized view.
3. Click the Remember button to save the view.

Your customized view is saved, and you are redirected to a unique URL that is generated for the view. The Remember My Changes link is also changed to the name of your customized view, as you see in Figure 12-25. If you click the link, you'll see a list of all the customized views you have saved along with a link to the original view published. To rename or delete any of your customized views, click the Manage Custom Views link at the bottom of the listing.

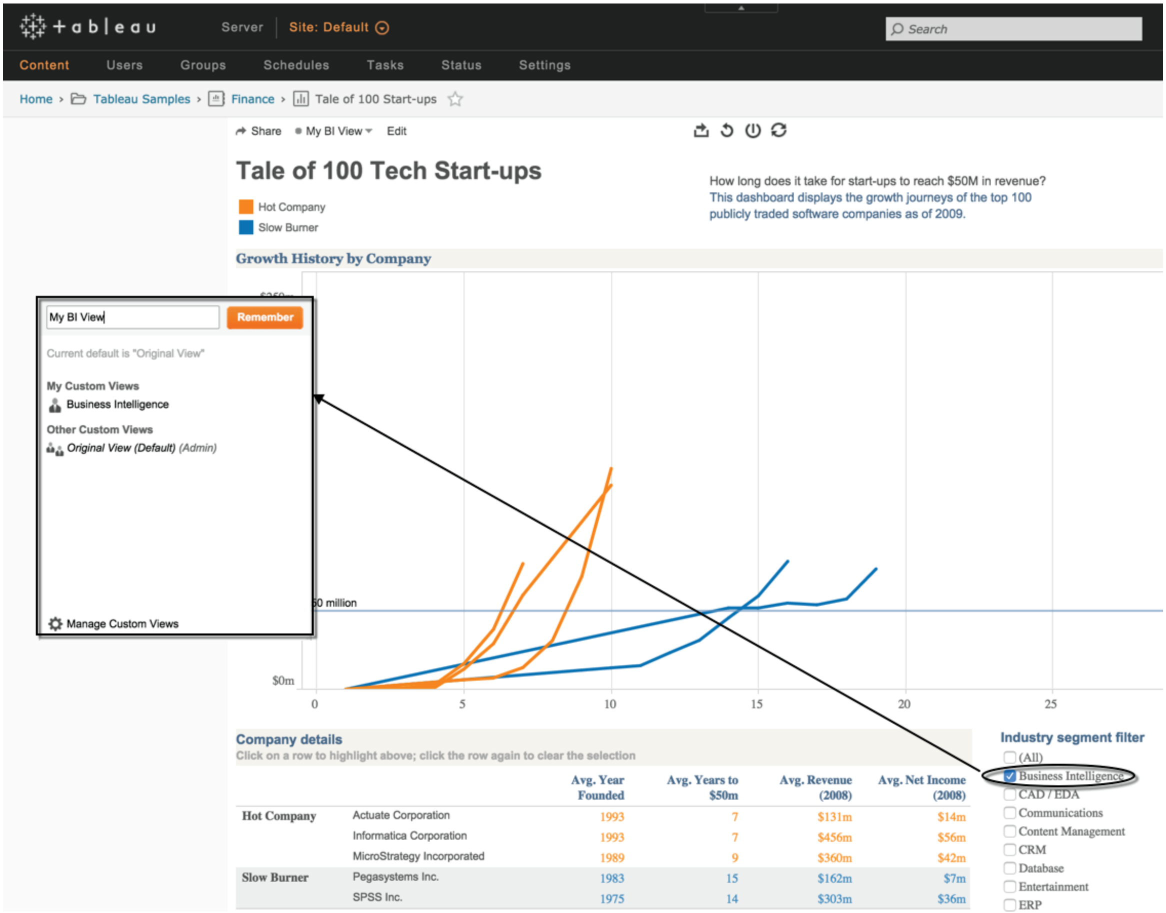


FIGURE 12-25 *Saving a custom view*

Other users accessing the view on their own will still be presented the original view as first published unless they use the unique URL for your customized view. Customized views provide a good way for users to save frequently used filter combinations without the need to rely on the publisher.

A significant new feature that arrived with the release of Tableau Version 8 is web authoring. This goes beyond saving customized views by allowing users that lack a Tableau Desktop license to alter and create totally new visualizations within Tableau Server.

AUTHORING AND EDITING REPORTS VIA SERVER

Customized views work well for making small changes to existing files or parameters, but more robust editing is sometimes necessary. Tableau Server's in-browser editing functionality provides a simplified version of the Desktop experience. It allows users to edit existing workbooks, create new visualizations, and then save the work to Tableau Server. This capability is great for middle-tier users who don't typically need all of the features that Tableau Desktop offers but want the ability to probe data in ways that were not anticipated by the workbook creator. Web-Table authoring (see Chapter 9) gives these users the ability to self-serve information and analysis from any device capable of accessing Tableau Server via a web session without the need to install any additional software.

WHAT IS REQUIRED TO AUTHOR REPORTS ON THE WEB?

In Chapter 9, you learned about mobile access to Tableau Server reports as well as mobile report authoring. You observed that a personal computer's web-based interaction is very similar to the tablet-based interaction. This is also true when considering the web/tablet authoring functionality. This functionality, like all web-based interfaces to Tableau, is exclusively a function of the Tableau Server environment and cannot be done utilizing only the Desktop or Reader products. To author on the web you need the following:

- A site with web authoring enabled
- A live web connection
- A Tableau Server Interactor license
- A standard web browser with a live web connection
- A pre-existing workbook that must be published to Tableau Server
- Appropriate permissions to be able to Web Edit

If you want to author via a tablet, you must download Tableau's iPad application from Apple, or the Android application from Google Play if you plan to use an Android tablet. While these elements are essential, they are not sufficient. The server permission for Web Edit must be allowed, as you see in Figure 12-26. This permission is also required to author via a desktop or laptop computer via the Internet.

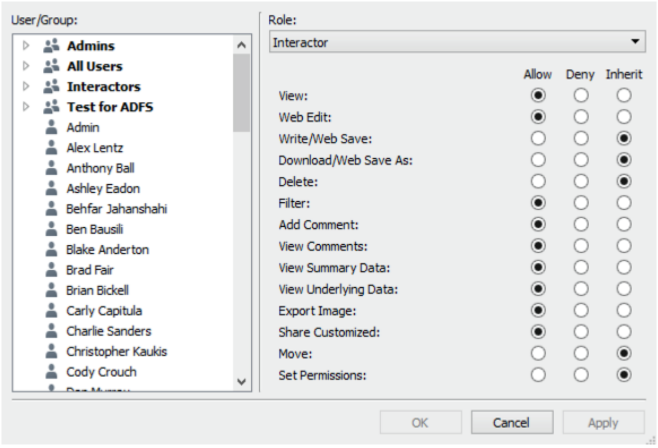


FIGURE 12-26 *Permitting web editing*

As with all permissions on Tableau Server, web editing can be configured at multiple levels—user, group, workbook, project, or site. Tableau Server users can also access saved data sources via the Content tab and create a new workbook using that data source. One Desktop license is required to publish the original report template or data extract file to the server, but any number of Tableau Server Interactor licensees can edit the report or create new reports from published data sources. With the release of V9, Tableau has continued to add capabilities to web authoring—most notably, the ability to create formulas has been added along with the Show Me facility. Tableau does not impose additional licensing fees to access the Web/Tablet authoring tool.

SERVER DESIGN AND USAGE CONSIDERATIONS RELATED TO WEB AND TABLET AUTHORING

Tableau's web-tablet authoring system is largely a client-side functionality provided through an HTML5 layer. This means that the Web/Tablet authoring system will have limited impact on the majority of Tableau Server processes.

The Tableau Server administrator should be aware that users editing views via this method will generate activity on the server's VizQL process. And, if the workbook being edited is based on a data server extract-driven data source, those processes will also experience increased loads. This impact is identical to the effective impact of adding additional Tableau Desktop interactions—presuming a server-mediated data connection.

Should web/tablet authoring result in a high number of workbook saves or creates via the server Save As dialog box, Tableau Server will experience additional demand placed on its repository and storage systems in a manner similar to the load that would be expected via Tableau Desktop utilization.

These additional loads are a good thing. They mean that your user base is engaged and actively using the system.

DIFFERENCES BETWEEN DESKTOP AND WEB OR TABLET AUTHORING

The experienced Tableau Desktop user will immediately notice that the web-tablet editing interface closely mirrors the familiar desktop environment. Editing through the web or tablet is very similar to the Desktop tool, though it is simplified and limited in a few ways. This section will detail the functional differences between the two authoring experiences. While limitations in the web-tablet authoring environment are highlighted, you should not interpret this section as a negative critique. Web authoring is a significant innovation that will provide benefit to the majority of your user base.

The goal is to highlight the differences so that you are aware of what can be done via the web versus what must be done using the Desktop application. The Web-Tablet authoring environment is designed to provide a simplified version of the Desktop experience. It is not intended to replace the Desktop application.

Drop Areas for Rows and Columns but No Show Me! Drop Area

Many of the standard desktop options and layouts are available within the web-tablet authoring interface. The left side re-creates the data window, including any data source(s), measures, and dimensions. The Column and Row shelves, along with the Pages, Filters, and Marks cards, also exist in their standard positions. Users can create visualizations in the same drag/drop manner that is fundamental to Tableau's Desktop authoring experience. One difference is that all fields must be dragged to the shelves and that the in-view drop areas for rows and columns do not exist. And, while Show Me is available, it can't be moved around the screen. It is available only from the menu at the top of the view. Figure 12-27 shows the web editing interface.

You can see that the Data and Analytics windows, along with dimensions and measures, are available on the left just like Desktop. Pages, Filters, and the Marks cards are all there as well, along with the Columns and Rows shelves. Anything that would appear on the Data window in Desktop should also appear in the Data window within the web editing environment. The Analytics window does not include all of the tools available in Desktop. Figure 12-28 shows a side-by-side comparison.

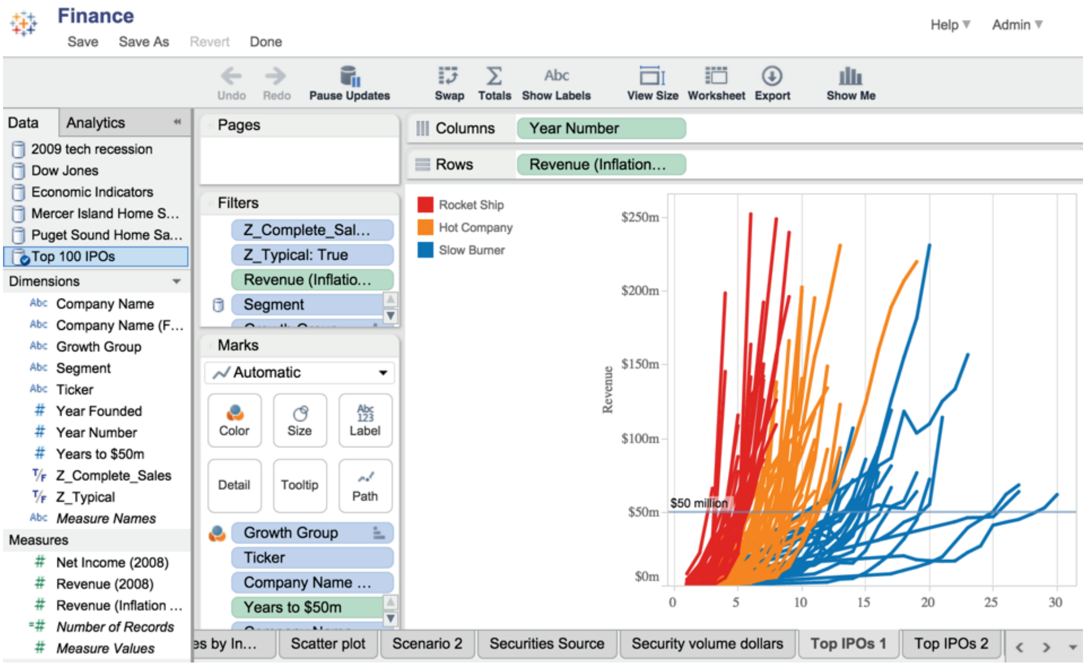


FIGURE 12-27 *The web-tablet authoring view*

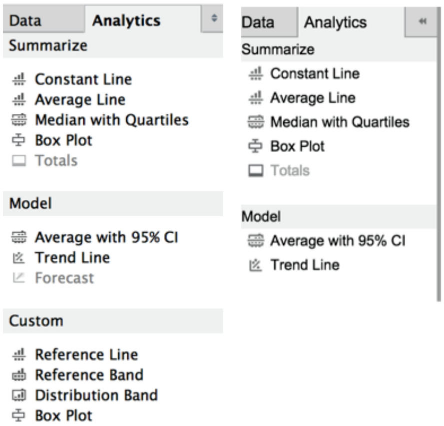


FIGURE 12-28 *Analytics pane in Desktop and Web*

You can see that the web editing environment does not include forecast modeling and all of the Custom reference line and band options are not available in the web environment.

The menu across the top of Figure 12-27 does not include many of the desktop menu and icon bar options available in Tableau Desktop. But what is available provides a good set of tools for creating new views and analysis using published workbooks and data sources.

No Dashboard Support

There is no dashboard display in the server authoring environment. In fact, dashboard editing is not interpreted in the web-tablet editing experience. Any workbook that contains dashboards will display those views broken into their component parts (even if hidden) versus being displayed as the combined entity you see in the desktop application.

Data Source Manipulations Are Not Supported

All data sources needed for analysis must be included at the time of publishing from Tableau Desktop. The web authoring system does not allow for any manipulation of the metadata layer. You can't add new data sources, remove unused data sources, create calculated fields or parameters, change default field properties, or edit relationships between data sources. In general, the web authoring environment doesn't support metadata management. These capabilities exist exclusively in the desktop tool.

It is worth noting that as of Tableau Server V9.0, web users are now able to create calculated fields during web authoring.

Context Menu Functionality

Some, but not all, functionality accessed through right-clicking in Tableau Desktop is supported in web authoring either via right-clicking objects or clicking their menu-based controls. Field-specific controls are shown in Figure 12-29.

The dimension-specific control in Figure 12-29 was exposed by selecting the small drop-down arrow in the segment dimension pill. Similar measure-specific controls can be exposed by clicking the drop-down arrow in the measures pill shown in Figure 12-30.

Note that you can also invoke Quick Table Calculations from the same menu.

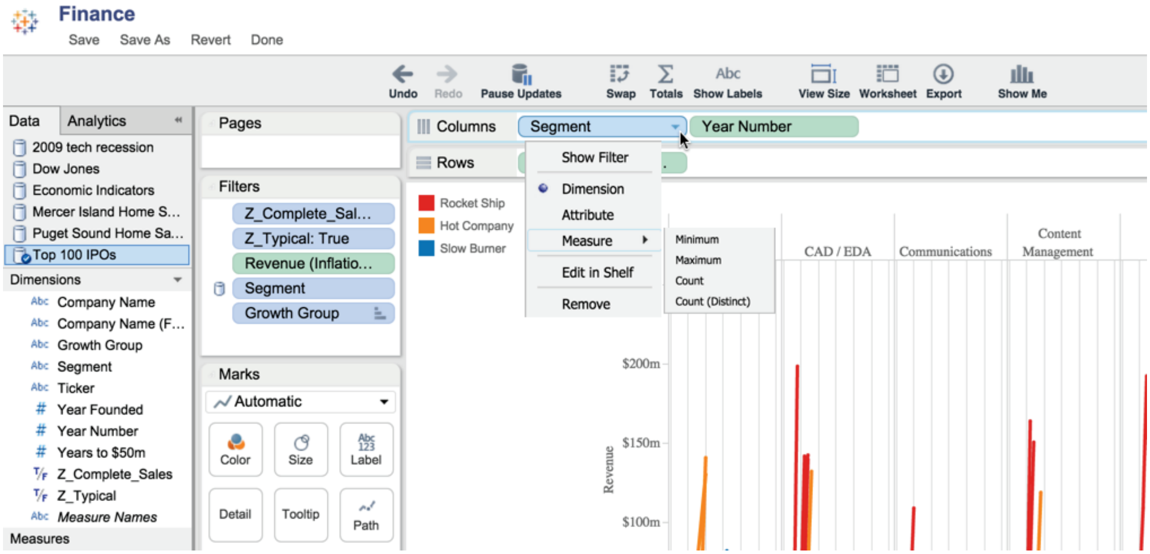


FIGURE 12-29 *Field-specific controls*

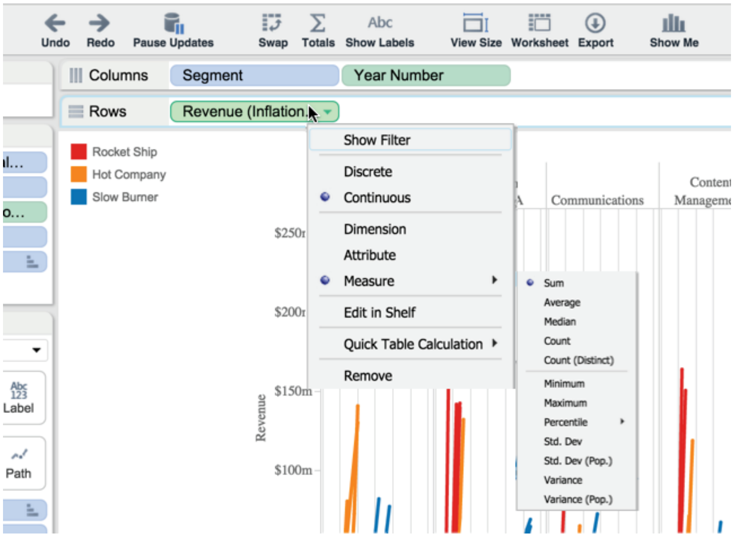


FIGURE 12-30 *Measure-specific controls*

Quick Filters Only: No Complex Filtering

Desktop users accustomed to creating "complex" filters like "Top 10" or utilizing specific conditions, will notice that these filters will persist in the web authoring session. However, web editors can't add new versions of these complex filters. Web editors are able to add quick filters to views, and the full suite of quick filter types is available. In addition, complex filter behavior can be accomplished in web authoring mode by creating calculated fields that accomplish the desired filtering.

Cell Sizing Is Exclusively Menu-Based

Cell sizing is controlled exclusively through the cell size menus; users can't drag elements of the visualization to resize those items, nor can they drag to resize the sheet as a whole. And web editors cannot drag to control the "fit" of the view within the design space. Figure 12-31 displays the web tools that are available for fitting and sizing the web view.

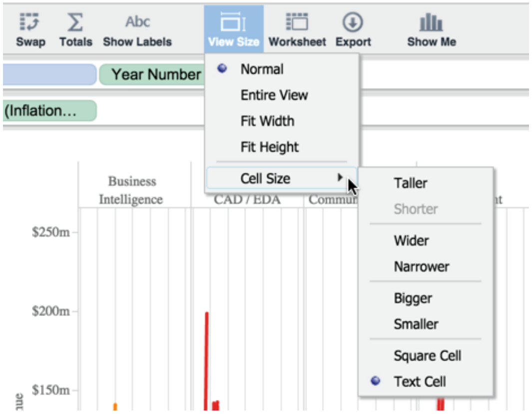


FIGURE 12-31 *Web fit and cell-sizing controls*

Even though Desktop dragging to resize elements isn't available in the web environment, the View Size ➢ Cell Size menu provides this facility.

Sorting Is Only Available Through Quick Sorts

Unlike Tableau Desktop, authors cannot set sorts based on specific fields, default sorts, or pre-sort information in a robust manner. Sorting is exclusively allowed through the in-visualization quick sorts that are omnipresent on headers in all Tableau visualizations.

Limited Control of Color, Size, Text, and Tooltips

Tableau Desktop allows nearly infinite control of color palettes, size ranges, shapes, and tooltip content. The web-tablet environment provides none of these fine-grain controls.

Adding Calculated Values in Tableau Server

The capability to create calculated values in Tableau Server was added in Version 9. Similar to Tableau Desktop, ad hoc calculations can be added by typing formulas into the Rows and Columns shelves or on the Marks card. Figure 12-32 shows an ad hoc calculation being added to the Columns shelf.

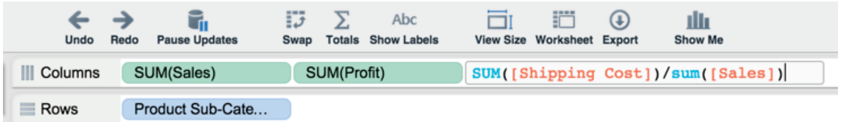


FIGURE 12-32 *Adding an ad hoc calculation*

Complete the calculation value just as you would on Desktop by hitting the enter key on your keyboard. You can also add a new field to the view by dragging the pill for the new calculated value to the Data shelf. And you can give that field a name by clicking the new field on the Data shelf and editing the name in the calculation dialog box you see in Figure 12-33.

You can also create a new calculation in Tableau Server by using the drop-down arrow you see at the top right of the data window in Figure 12-34.

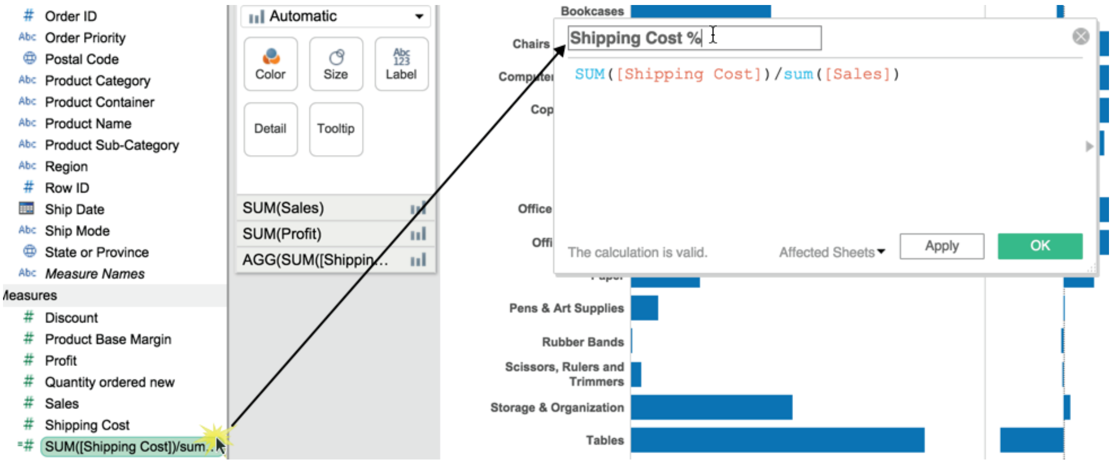


FIGURE 12-33 *Naming the new calculated value*

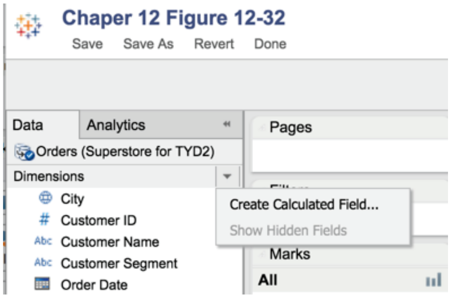


FIGURE 12-34 *Adding a calculation from the data window*

Creating new fields in Tableau Server through calculated values is a useful capability. It enables users without access to Tableau Desktop to do deeper analysis of published data sources and workbooks. You can't do everything that you can do with calculations on Server (number formatting is possible only on Desktop for now). But this new capability makes Server a more full-featured environment for creating new views.

SAVING AND EXPORTING VIA THE WEB-TABLET ENVIRONMENT

Tableau Server provides a number of ways to export or save visual images, data, and workbook views.

EXPORT

All of the exporting features available to Desktop users are available on Server as well. Images, data, crosstabs, and PDF export formats are supported. You access these options from the Export menu shown in Figure 12-35.

These same options are available within Tableau's tablet applications as well.

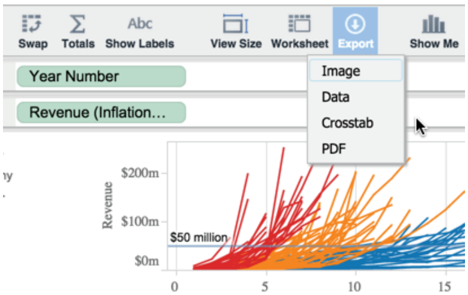


FIGURE 12-35 *Server export functions*

SAVE AND SAVE AS

Recall from the beginning of this chapter in Figure 12-3, the web/edit permission must be Allow for users to enter Server web editing mode. Similarly, the ability to overwrite an existing workbook is also permission-based. Tableau Server does not save a copy of the original workbook by default. Saving on Server is equivalent to republishing on Tableau Desktop. The Save As dialog box provides the ability to republish the workbook under a new name or into another project.

It is possible to set permissions these functions separately. In this way, you can give web authors Save As permissions without the Save facility so that they can save a version of the file for their personal consumption on Tableau Server. This prevents web users from accidentally over-writing the original content.

RECOMMENDATIONS FOR IMPLEMENTING WEB/TABLET AUTHORING

Tableau Server has not been designed to entirely replace the Desktop tool. Enterprises should view web-tablet authoring as a supplementary tool that enables a previously under-served cohort group access to Tableau's ad hoc analysis and reporting capabilities.

Paired with relevant data source access and training, Server facilitates self-service business analysis in a controlled environment—providing users with the ability to ask questions that were not anticipated within the original design of the report. The following are key points to remember when designing reports that will be used as starting points for web-tablet authors:

- Give worksheets logical name(s) that will not be obscured by the standard Desktop practice of hiding sheets that have been added to dashboards.
- Design template workbooks and template data sources that can be readily approached by non-technical users who may not have data analysis experts or experience with Tableau Desktop.
- Provide transparent information about data sources including refresh dates, sources, assumptions, and contact information for the original publisher.
- Create a specific Sandbox project/area where new users can save work and gain confidence.

SHARING CONNECTIONS, DATA MODELS, AND DATA EXTRACTS

Tableau's data server lowers access barriers while providing data governance controls. Database administrators can define data connections once, publish them to the data server, and manage access by applying Tableau's object level. Consequently, data analysts using Tableau require no knowledge of the underlying tables, joins, or related criteria driving the connections.

OFFERING A COMMON DATA LIBRARY

It's common for organizations to manage and use a variety of data sources. Transactional data may live in one database while historical data is maintained in a completely different database. Business users may maintain their own spreadsheets of forecasts and budgets. While Tableau can easily connect to each of these disparate data sources, Data Server offers the capability to host these connections in a central place. This reduces the potential for data misuse because Tableau users can simply connect to Tableau Server and the data they need access to, regardless of source system. Data Server simply acts as a proxy to databases while also serving as a host for items such as Excel and Access files, or even data extracts.

SHARING DATA MODELS

Earlier in the chapter, in the section related to the Tableau Data Server, you learned how data sources can be published and used by many different users. The related metadata that comes with the shared data sources allows data administrators to manage inconsistencies through the following:

- Consistent field name aliasing
- Consistent field grouping
- Consistent application of field hierarchies

This permits the organization to tap into the best subject matter experts to create and validate calculations and publish the resulting data models for your entire organization to benefit from. These capabilities can reduce variations in how business rules are interpreted and applied, while giving analysts the ability to do their own ad hoc analysis by adding their own customizations on a per-workbook basis.

Inheritance of Updates

Once a data source has been published to Data Server, workbooks using the connection automatically inherit any future updates to the data source. This simplifies the process of data source maintenance while reducing the risk of outdated business rules persisting in production when an underlying change has been made.

EMBEDDING TABLEAU REPORTS SECURELY ON THE WEB

If your organization is accustomed to consuming information via a specific web portal, Tableau provides a variety of ways to embed your reports as interactive dashboards or static images—all while persisting the same licensing and security framework available to you on server.

WHEN TO EMBED A DASHBOARD

When does it make sense to embed a dashboard rather than simply having your users access it directly from Tableau Server? If your user base is already familiar with a particular web portal, it makes sense to use that website as a repository for interactive Tableau visualizations and dashboards. In addition, there may be advantages to using existing security options that may already be defined on the web portal.

Going to Your Users

It can be frustrating for less technical users if they are asked to remember many different logins and websites to access information they need to do their job. Many people don't enjoy the accelerating change fostered by advances in technology. Learning new tools reduces efficiency during the learning cycle.

By embedding Tableau into an existing web portal that your users are familiar with, you provide them with the benefits Tableau has to offer while minimizing the change they have to absorb.

WHEN YOUR REPORTS ARE A PIECE OF A LARGER SAAS OFFERING

If you offer a service to your clients, you want to control the branding on the products they are using. Instead of providing them access to Tableau Server directly, you can embed the content (workbooks, dashboards, visualization) in a web portal that carries your unique branding. Planned properly, this access method provides seamless integration. Your customers don't need to know the underlying technology is Tableau Server. Embedding workbooks into a website reporting portal provides a way for you to present a single cohesive product to your end users.

PROVIDING A MORE ROBUST ENVIRONMENT

What happens when, over the years, multiple reporting environments have been created as the result of different initiatives spread over many teams? One part of the company may rely on Business Object reports that were created a decade ago. Other teams may depend on SSRS reports to make day-to-day decisions. Tableau can be a vehicle for consolidating disparate tools into a single, cohesive system.

What is worse than bad business intelligence? Good business intelligence that nobody can find. By creating a single seamless environment for your users, they no longer have to track down and find the reports that may be out there. The probability of redundancy is reduced.

While getting to this happy goal may require considerable effort, Tableau's ability to connect to many different data sources reduces the time and complexity required to bring your data into a single integrated system.

HOW TO EMBED A DASHBOARD

Embedding dashboards usually boils down to one of these methods:

- Using Tableau's JavaScript code
- Using the dashboard's URL in an iFrame or Image tag
- Writing your own code using Tableau's JavaScript API

No matter which method you choose, you can control your embedded view through the use of Passed Parameters. We'll explore all three methods next while also diving into the details of the parameters you can use.

Note all the embedded solutions in this section require users to log in via the embedded view as they normally would when accessing Tableau Server directly. You'll learn more about providing a single sign-on experience for your users later in this chapter in the section "Tips and Tricks for Embedding Dashboards."

Using Tableau's JavaScript Code

The easiest way to embed a dashboard in another web page is to use the JavaScript code provided by Tableau in its Share button. With the options for setting the width and height of your dashboard and the ability to turn on/off the toolbar and tabs, this provides a mechanism for quickly embedding your dashboards into another web page. The following is a quick example of the resulting code:

```
<script type="text/javascript"
src="https://yourtableauserver.com/javascripts/
api/viz_v1.js"></script><div class="tableauPlaceholder" style=
"width:979px; height:662px;"><object class
="tableauViz" width="979" height="662"
style="display:none;"><param name="host_url"
value="https%3A%2F%2Fyourtableauserver.com%2F" /><param name=
"site_root" value="&#47;t&#47;YourSite"
/><param name="name"
value="YourWorkbook&#47;YourView" /><param name="tabs"
value="yes" /><param name="toolbar" value="yes" /></object></div>
```

Notice the use of `<param>` tags to pass specific values to Tableau Server. Through the use of these tags, it is possible to pass additional parameters such as an initial filter. For example, the following entry will initially filter the embedded view by restricting the Region dimension to West only:

```
<param name="filter" value="Region=West"/>
```

The name and site root parameters are the only ones required when embedding a view.

Using an iFrame or Image Tag

Another option is to use the URL for a dashboard or a view in an iFrame or image tag. Additional parameters can still be passed but must be included at the end of the URL. The `embed` parameter is required, but all others are optional. An example of an embedded view using an iFrame follows this paragraph. The dashboard is once again filtered to the West region only, while also restricting to the date June 1, 2012.

```
<iframe src="https://yourtableauserver.com
/t/views/MyWorkbook/MyDashboard?:
embed=yes&Region=West&Date=
2012-06-01" width="800" height="600"></iframe>
```

Note the required `embed` parameter is set first. A value of `Yes` hides Tableau's default navigation options and comments section below the view while also moving the toolbar and share options below the view.

Writing Your Own JavaScript API

You can also write your own code by leveraging Tableau's JavaScript API. This is often the preferred choice by web developers looking to embed Tableau views into their existing web applications as it offers a deeper level of control.

Tableau provides developers with the ability to interact with embedded views in real time. By listening for events generated by Tableau views, developers can capture actions performed by a user and respond to them in rich, interactive ways. For instance, developers can respond to a user selecting marks on an embedded view and trigger a response in their web application. Developers can also interactively set filters and select marks within an embedded view in real time—no longer limited to simply setting initial values prior to a view loading. The best part is that each of these API functions is enacted as they would be if the user had performed the action in the view itself, meaning no page refreshes occur. The result is a completely seamless experience between your application and the embedded Tableau reports.

See Chapter 13 for more details on using Tableau Server's APIs.

FURTHER CONTROL USING PASSED PARAMETERS

Whether you choose to use an iFrame or JavaScript, you can pass additional parameters to the view. Search Tableau's website to find a complete list of supported parameters.

TIPS AND TRICKS FOR EMBEDDING DASHBOARDS

Tableau has streamlined the embedding process over the last two years. The tips and tricks that follow provide some ideas.

Filter Formats for Dimensions, Measures, and Dates/Times

When passing dimension filter parameters, simply list each value in a comma-separated list. To filter on multiple dimensions, separate each with an ampersand. The general form is:

```
Field=Value1,Value2,Value3&Field2=Value1,Value2
```

You can filter on measures in the same manner by passing explicit values. However, Tableau Server does not support filtering by a range of values or using greater than or less than logic.

To filter on a date or date/time field, use the following form:

```
DateField=yyyy-mm-dd hh:mm:ss
```

When filtering a date/time field, the time component is considered optional.

Know Your Character Limits

Theoretically, there is no limit to the number of parameter values you may pass to an embedded view. However, you may ultimately run into a URL length restriction imposed by the end user's browser.

While HTTP protocol does not impose a cap on URL length and many modern browsers can handle URLs with up to 80,000 characters, Internet Explorer 8 and 9 have a maximum character limit of only 2,083 characters.

As a result, you should strive to keep your URLs under this limit to ensure compatibility. Keep in mind the complete URL—not just the parameters and corresponding values—are included in this length.

USING TRUSTED TICKET AUTHENTICATION AS AN ALTERNATIVE SINGLE SIGN-ON METHOD

When an embedded view is accessed, the same authentication mode enabled on Tableau Server is used to verify the user's identity. For instance, if your server is configured to use local authentication, your users will be required to log in via a form provided by the embedded view. This can be cumbersome if the users have already authenticated in the web application. To work around this, Tableau provides a couple of options for single sign-on authentication, a process where your user is authenticated by your web application and is not required to further authenticate themselves by any embedded Tableau views.

If your server is configured to use Active Directory and SSPI, you can enable SSPI on your web server as a single sign-on solution given the user is in your Active Directory and is a licensed Tableau Server user. In all other situations, you will need to use Trusted Ticket Authentication as an alternative single sign-on method.

When using Trusted Ticket Authentication, the web server assumes all responsibility for authenticating users. Before embedding the view, the web server passes two POST parameters to Tableau Server:

- Username (must match a licensed Tableau server user)
- `Client_ip`

The web server will receive a response in the form of a `unique_id`, which is used in the embedded view's URL, as shown in this form:

```
https://yourtableauserver.com/
trusted/unique_id/t/views/MyWorkbook/MyDashboard?:embed=yes
```

If you are using JavaScript, the ticket parameter can be used:

```
<param name="ticket" value="unique_id"/>
```

Once a `unique_id` has been issued, it must be redeemed within 15 seconds from a machine matching the `client__ip` specified or it is considered no longer valid. When Tableau Server receives the request, the user is logged in as they would be if using forms authentication and the trusted ticket URL is resolved to that of a standard request.

Before a web server can make a Trusted Ticket Authentication request, it must first be "white listed" on Tableau Server. This can be accomplished using the following TabAdmin command, where `xxx.xxx.xxx.xxx` represents IP addresses for any trusted web servers:

```
tabadmin set wgserver.trusted_hosts "xxx.xxx.xxx.xxx, xxx.xxx.xxx.xxx"
```

USING SUBSCRIPTIONS TO DELIVER REPORTS VIA E-MAIL

Do you have dashboards or views that you want to look at every day? Tableau Server can send you notification e-mails with links to the desired contents using a defined schedule. Tableau enables this through subscriptions. Subscriptions are not enabled in Tableau Server by default. The feature must be enabled by an administrator. See Chapter 11 for details. In this chapter, you learn how to subscribe to views.

CREATING SUBSCRIPTION SCHEDULES

Once subscriptions are activated, you will be able to subscribe to your favorite views and receive regular e-mails. Figure 12-36 shows a list of schedules for Tableau Server.

The schedule list includes one custom schedule (Dan's Hourly Update) that I added to Tableau Server. Notice that Tableau provides a useful list of schedules for different time increments. Note that these schedules also control Tableau Data Extract runs. Provided you have the access rights, you can view subscription run history. Figure 12-37 shows what a subscription history looks like.

Two subscriptions runs were completed: one containing a single view and another containing every worksheet within the source workbook.

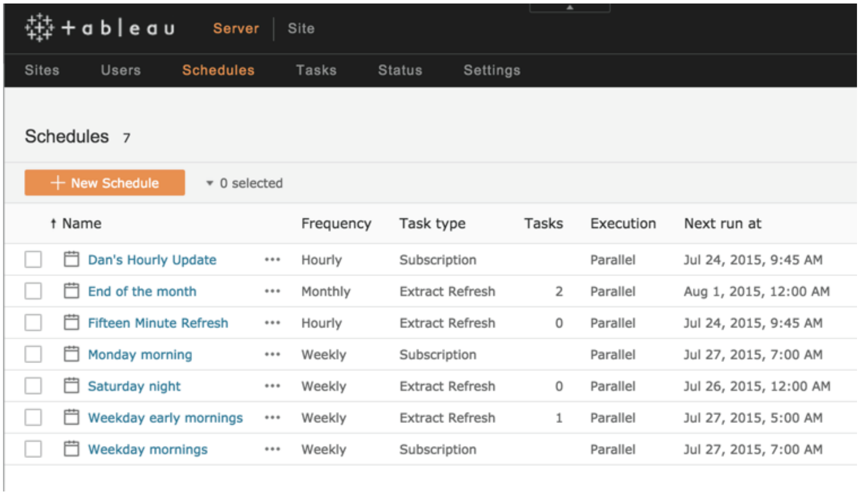


FIGURE 12-36 *Schedules*

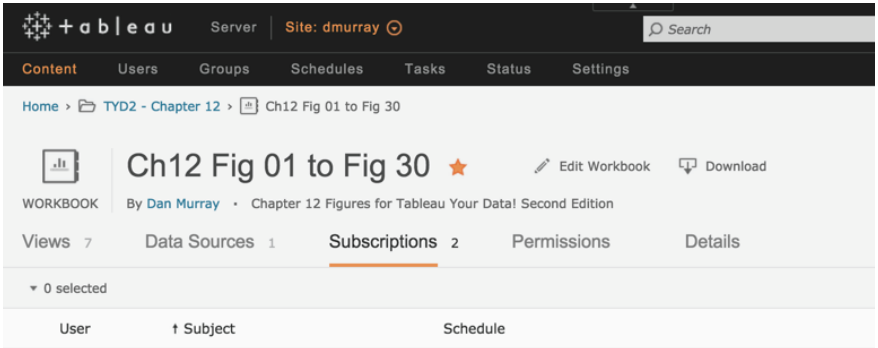


FIGURE 12-37 *Subscription run history*

Subscribing to a View

Creating a subscription requires a few steps:

1. Navigate to the view you wish to subscribe to.
2. Click the mail icon in the upper left of your browser window.
3. Complete the Subscribe dialog box.

Figure 12-38 shows the Subscribe dialog box.

FIGURE 12-38 *Creating a subscription*

My e-mail address was added automatically by Server. This is defined by the administrator. Enter the subject as you want it to appear in your e-mail header, select the appropriate schedule from the drop-down list, and then decide if you want to receive a single view or all of the views available in the workbook. At the scheduled time, Tableau Server generates an e-mail containing an image of the view. Figure 12-39 shows the e-mail message generated by the subscription.

The resulting e-mail contains a dashboard view. Clicking that e-mail view will open the view in Tableau Server presuming you have access to the Internet. You can even access the subscription from a cell phone or tablet computer. Notice the "Manage my subscription settings" link at the bottom of Figure 12-39. This is another optional setting that your administrator must enable. Adding this to the subscription details provides access to the Settings screen where the Subscriptions checkbox is located. This enables the user to change the subscription settings.

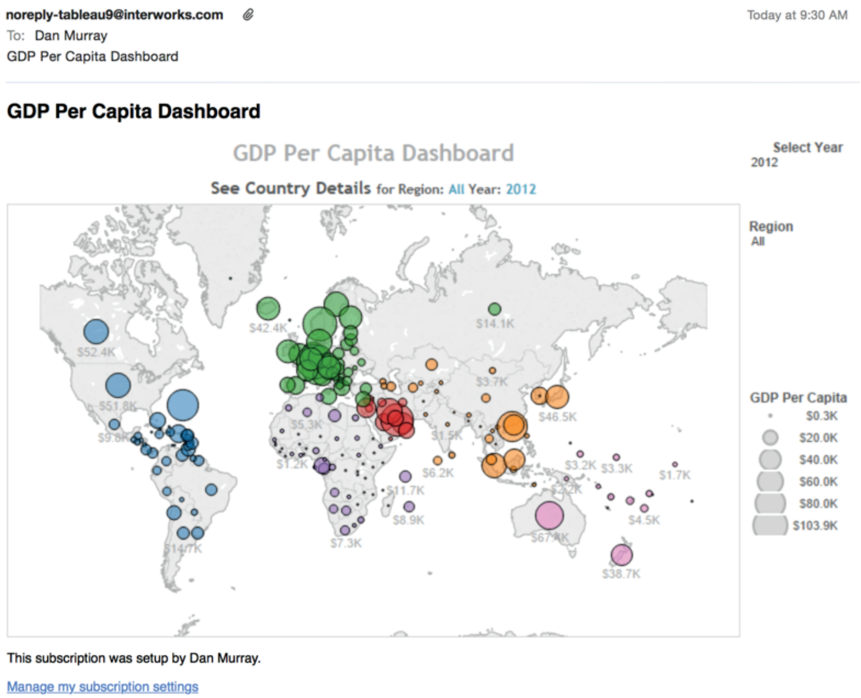


FIGURE 12-39 *A subscription e-mail*

Subscriptions can be a convenient way to receive workbook views that are frequently used. In the next chapter, you learn about more sophisticated ways you can automate Tableau Server, such as through APIs and Tableau's command-line tools.

NOTE

1. James C. Collins, *Good to Great: Why Some Companies Make the Leap—and Others Don't* (New York: HarperBusiness, 2001), 88.

CHAPTER 13

Automating Tableau Server

As your Tableau Server deployment expands, the number of users and amount of data you have to manage will grow. Tableau provides three command-line tools and a set of APIs (Application Program Interfaces) that will help you automate routine tasks. Most of the functions these tools provide are available directly within Tableau Server's user interface.

Using Windows Notepad (or your favorite text editor), you can automate tabcmd to run via a batch file. Then, by using Windows Task Scheduler, you can trigger the batch file to run at a specific time or based on a specific triggering event. Publishers can use similar techniques to automate extract refreshes without opening Tableau Desktop or leveraging Tableau Server by using the Data Extract Command-Line Utility. Of course, many popular scripting or programming tools can call Tableau's command-line functions to automate tasks. How you use these tools is limited only by your desire and creativity.

If you are a system administrator and accustomed to writing script and using the Windows Command Processor and Windows Task Scheduler, you will not have difficulty incorporating any of Tableau's command-line tools or APIs into your existing toolset. Many people don't use these utilities because their full functionality is not clearly understood, or they have not seen specific use case examples. Tableau Software provides some good introductory videos on their website. You can find those by searching for "On Demand Training" and looking in the Server section for the tabcmd and tabadmin videos.

TABLEAU SERVER'S APIS

Tableau added a set of APIs in V8 that provide administrators with additional tools for automating routine and complex tasks. These APIs include the following:

- Tableau Server REST API
- Data Extract API

With the Tableau Server REST API, administrators are provided another mechanism for managing Tableau Server resources programmatically. Unlike the tabcmd utility, the REST API is a web service that does not require an installer or need to be called from the command line.

The Data Extract API offers users the ability to create extract files from a variety of data sources using their language of choice—Python, C/C++, or Java. Prior to the availability of the Data Extract API, users were required to use Tableau Desktop or the Tableau Data Extract Utility to connect to and extract their data sources.

WHAT DO TABCMD AND TABADMIN DO?

Tableau's two command-line tools are tabcmd and tabadmin. Tabcmd provides functions for performing workflow tasks such as publishing workbooks, adding users, or exporting workbooks as image or data files. Tabadmin is designed for server administration—configuring server options, activating users, resetting passwords, and other tasks associated with managing the deployment and usage of Server within the enterprise.

A person with publishing rights might want to use tabcmd to automate repetitive tasks associated with updating and publishing data sources. A server administrator can leverage tabadmin to set up a new site, grant or revoke user rights, back up data, alter default session time-out settings (get input from Tableau Support or a qualified Tableau Partner before changing these settings), or reset user passwords. Think of tabcmd as a tool for helping those who publish and share. tabadmin is an automation tool for staff with administration responsibilities—helping them control access, tweak settings, or observe system status.

INSTALLING THE COMMAND-LINE TOOLS

When Tableau Server is installed, tabcmd and tabadmin are automatically installed in Tableau Server's `bin` folder. Depending on the operating system being used (Window 32-bit or 64-bit), the program will be installed in one of these locations:

- **32-bit:** `C:\Program Files\Tableau\Tableau Server\9.1\bin`
- **64-bit:** `C:\Program Files (x86)\Tableau\Tableau Server\9.1\bin`

If you are using a different version of Tableau Server, the portion of the address that says 9.1 should be replaced with the specific version number that you are using. If you are running a distributed environment—with multiple worker machines—and you want to utilize tabcmd on one or more of the worker boxes, you must install tabcmd on those other machines. Tableau provides an installer program for doing that. Those programs are

- **32-bit:** `C:\Program Files\Tableau\Tableau Server\9.1\extras\Tabcmdinstaller.exe`
- **64-bit:** `C:\Program Files\Tableau\Tableau Server\9.1\extras\Tabcmdinstaller.exe`

Copy the Tabcmdinstaller.exe program to the computer that you want to install it on and double-click the file to run the program. The program provides prompts as it installs. Tableau Software recommends installing the tabcmd program on the root drive (`C:\tabcmd`).

Because the setup program doesn't automatically add the `bin` folder containing tabcmd or tabadmin to the Windows PATH system variable, you have to manually navigate to the `bin` folder subdirectory to use the programs. This can be avoided if you modify your computer's PATH system variable to include the path to the `bin` folder. Doing this allows you to run the executable commands without needing to manually enter the directory location of the `bin` folder. To start using tabcmd, open the Windows Command Prompt. Figure 13-1 shows you how to do that in a Windows 7 environment.

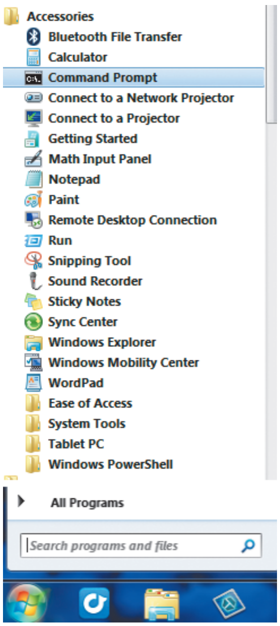


FIGURE 13-1 *Opening Windows Command Prompt*

If you are using a different version of Windows, find the accessories folder by searching your computer's hard disk. Once you've entered the accessories folder, click Command Prompt to open the Command Prompt window. In order for you to have access to the tabcmd program files, you must first navigate to one of the `bin` folders listed in the first section. If you are using a 64-bit version of Windows, type in the following command and press Enter:

```
cd "C:\Program Files(x86)\Tableau\Tableau Server\9.1\bin"
```

This will change the active directory to the `bin` folder that holds the tabcmd program. Assuming that your Tableau Server address is `http://mytableauserver.com`, and that your Tableau Server uses port 80, start a tabcmd session by typing the following into the Command Prompt window:

```
tabcmd login -s http://mytableauserver.com -u USER -p PASSWORD
```

The end of the string immediately following the ".com" is case sensitive.

After entering the tabcmd login command and the `-s` site URL, substitute the URL location of your Tableau Server installation. Then enter your username and password after the `-u` and `-p` global option variables.

The instance of Tableau Server used in this example is a local installation on a laptop. The username is Admin, and the password is Admin. The command-line entry to log into this server appears in Figure 13-2.

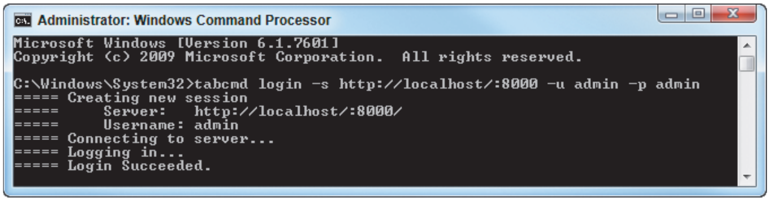


FIGURE 13-2 *Tabcmd login example*

Notice the portion of the script that includes the server address also includes an additional element (:8000). This defines the TCP/IP port for the local server instance and is required because the port assigned to the local server isn't the default value Tableau Server normally uses. You can find more details regarding the default port settings in the Tableau Server online manual by searching for "TCP/IP Ports." After completing this step, you can now issue other commands to Tableau Server.

SETTING THE WINDOWS PATH

If you want to avoid having to manually change your current directory to the Tableau Server `bin` folder every time you want to run an executable file, add the `bin` folder to your Windows PATH system variable. Edit PATH by going to the Windows Control Panel, clicking System and then Advanced System Settings, and selecting the Environmental Variables button to expose the dialog box you see in Figure 13-3.

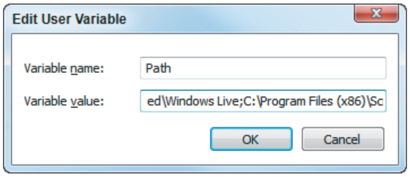


FIGURE 13-3 *Editing the PATH system variable through Control Panel*

If this seems intimidating, free utilities are available on the web that make this process easier and provide a larger editing window. Figure 13-4 shows a free utility called Eveditor in which the PATH system variable has been edited to include `bin` folder.

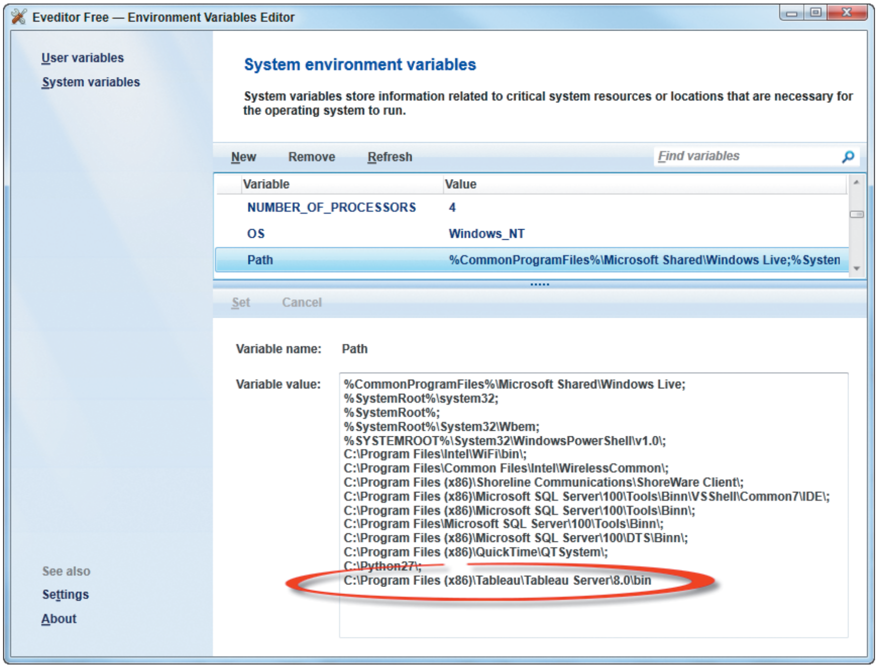


FIGURE 13-4 *Editing PATH using a free editing tool*

Adding the path for the Tableau Server `bin` folder eliminates the need to manually type in the path every time you want to start tabcmd or tabadmin in a batch file. Later, you'll see how to dynamically set PATH commands inside executable batch files—enabling dynamic setting of the file path so that tabcmd can always find the script it needs to execute.

Keep in mind that third-party tools (such as Eveditor - Windows Environment Variables Editor) are not supported by Tableau Software. You may be successful with Eveditor or other tools that you enjoy using, or you may experience problems. This is outside of Tableau Software's control.

WHAT KIND OF TASKS CAN TABCMD DO?

The tabcmd utility provides the capability to automate routine tasks concerning workflow management activities related to the following:

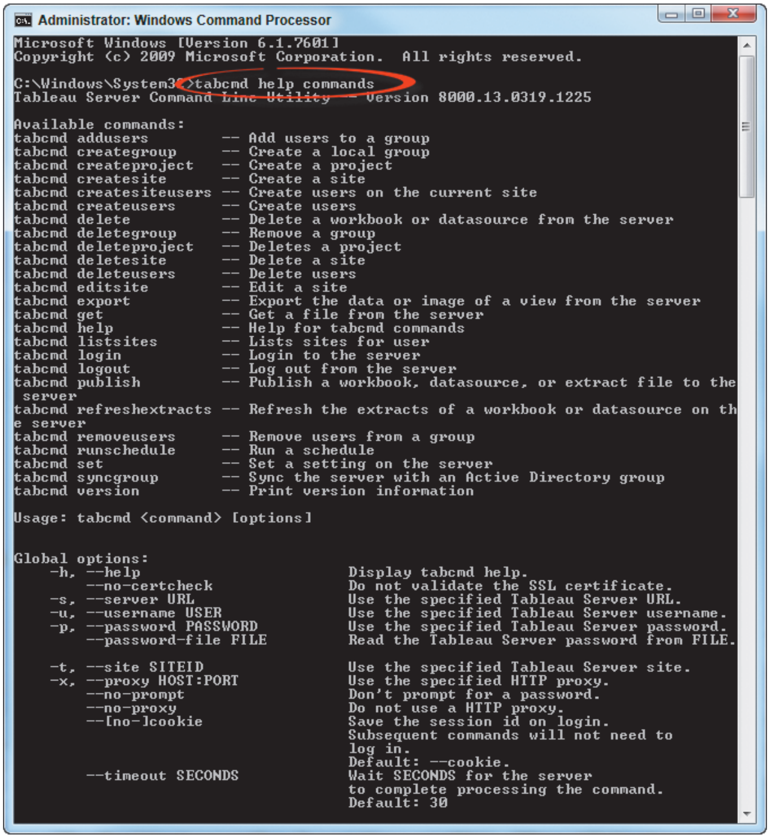
- Users, groups, projects, and sites
- Data management, publishing, updating
- Session management
- Security, site listings
- Server version information

The level of access and control is dependent on the type of administration rights assigned to the person using tabcmd. System administrators can manage data connections, groups, projects, and workbooks. They can add users to groups and projects. But they are not able to alter user licensing levels. Systems administrators have full rights—including the assignment of licensing levels for users and managing the server itself. System administrators can assign some administrative roles to site administrators. That role determines how much control will be given to a site administrator. Site administrators can manage groups, projects, workbooks, and data connections. If the system administrator permits it, they can also add or remove site users.

The tabcmd utility currently provides 27 functions with an additional 12 global option settings.

You can access a complete function reference in Tableau Server's online manual in the tabcmd Commands section, located at `http://onlinehelp.tableausoftware.com/current/server/en-us/tabcmd.htm`.

Tabcmd also has a built-in help function for listing the available commands by entering **Tabcmd Help Commands**. Figure 13-5 shows the help command display.



```
Administrator: Windows Command Processor
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\System32>tabcmd help commands
Tableau Server Command Line Utility -- Version 8000.13.0319.1225

Available commands:
tabcmd addusers         -- Add users to a group
tabcmd creategroup      -- Create a local group
tabcmd createproject    -- Create a project
tabcmd createsite       -- Create a site
tabcmd createsiteusers  -- Create users on the current site
tabcmd createusers      -- Create users
tabcmd delete           -- Delete a workbook or datasource from the server
tabcmd deletegroup      -- Remove a group
tabcmd deleteproject    -- Deletes a project
tabcmd deletesite       -- Delete a site
tabcmd deleteusers      -- Delete users
tabcmd editsite         -- Edit a site
tabcmd export           -- Export the data or image of a view from the server
tabcmd get              -- Get a file from the server
tabcmd help             -- Help for tabcmd commands
tabcmd listsites        -- Lists sites for user
tabcmd login            -- Login to the server
tabcmd logout           -- Log out from the server
tabcmd publish          -- Publish a workbook, datasource, or extract file to the
 server
tabcmd refreshextracts  -- Refresh the extracts of a workbook or datasource on th
e server
tabcmd removeusers      -- Remove users from a group
tabcmd runschedule      -- Run a schedule
tabcmd set              -- Set a setting on the server
tabcmd syncgroup        -- Sync the server with an Active Directory group
tabcmd version          -- Print version information

Usage: tabcmd <command> [options]


Global options:
    -h, --help                    Display tabcmd help.
        --no-certcheck            Do not validate the SSL certificate.
    -s, --server URL              Use the specified Tableau Server URL.
    -u, --username USER           Use the specified Tableau Server username.
    -p, --password PASSWORD       Use the specified Tableau Server password.
        --password-file FILE      Read the Tableau Server password from FILE.

    -t, --site SITEID             Use the specified Tableau Server site.
    -x, --proxy HOST:PORT         Use the specified HTTP proxy.
        --no-prompt               Don't prompt for a password.
        --no-proxy                Do not use a HTTP proxy.
        --[no-]cookie             Save the session id on login.
                                  Subsequent commands will not need to
                                  log in.
                                  Default: --cookie.
        --timeout SECONDS         Wait SECONDS for the server
                                  to complete processing the command.
                                  Default: 30
```

FIGURE 13-5 *The tabcmd help function display*

Entering **Tabcmd Help** and then a specific command name causes more complete options for that single command to be displayed.

LEARNING TO LEVERAGE TABCMD

In the following examples, you'll see progressively more advanced ways to use tabcmd including:

- Manually creating and running a tabcmd script
- Creating a Windows batch (`.bat`) file to run a saved script
- Using Windows Task Scheduler to automatically run a saved script

MANUALLY ENTERING AND RUNNING A SCRIPT IN TABCMD

The most basic way of using tabcmd is to manually enter commands that can also be accessed from the Tableau Server manual. This is also a good way to test tabcmd before you attempt to create a script that automatically runs tabcmd.

A common task required of a content administrator is to create groups on Server and assign users to those groups. Figure 13-6 displays the script used to create a new group called Executives.

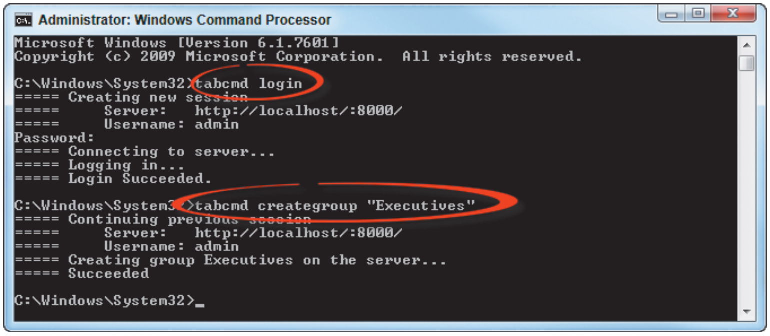


FIGURE 13-6 *Adding a new group to server*

The first command in Figure 11-6, tabcmd login, initiates a new session and prompts the user to enter a password. It is also possible to append the password to the end of the login command by adding `-p` or `--password` followed by your actual password. The script Tabcmd Creategroup “Executives” triggers the addition of the new group to server. At the bottom of the script, you can see that tabcmd provides a status while processing and then confirms that the operation succeeded.

The next step is to assign users to the group. By creating a list of valid usernames (`egroupadd.csv`) and saving it in the Tableau Server `bin` folder, tabcmd can assign the specified users to the executive group. Figure 13-7 shows a list of server users on the left (Allen, Bill, Cal, Dave, Eric). On the right you see the executed script.

This is the script used to add the users: `tabcmd addusers "Executives" --users "egroupadd.csv"`.

These activities can be done directly in the Tableau Server GUI environment, but tabcmd may be a more efficient way to make group assignments if they change frequently or you have a large number of users to assign.

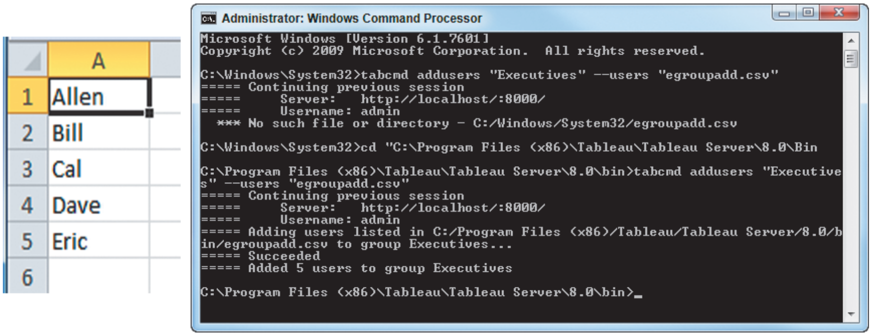


FIGURE 13-7 *Adding users to the new group*

RUNNING TABCMD SCRIPTS VIA BATCH FILES

If you find yourself using the same script repeatedly, you can use a text editor to create and save the script for reuse later. Windows includes a text editor program called Notepad that can be used to enter and save a tabcmd script. Notepad is normally located in the Windows accessories folder. Another Windows application—Task Scheduler—can be used to launch the script saved using Notepad. There are many other programming tools you can use for this purpose, but these are part of the Windows toolset.

THE STEPS REQUIRED TO CREATE BATCH PROCESSING SCRIPTS

Regardless of whether you prefer to use Windows Notepad or some other text editing software, the basic steps to create a batch process are the same:

1. Create the tabcmd script in Notepad or another text editor.
2. Save the script as a `.bat` executable file.
3. Double-click the batch file to execute the script.

In this scenario, the script is still run manually, but you no longer have to type all of the instructions every time you want to make changes, export data, or update files. These may be activities you repeat periodically—often enough to warrant saving a script but not so often that you need to fully automate processing.

In the next example, you'll see how to create a script in a text editor, save the script as a batch file, and then run the script using a CSV source file that provides the usernames and permissions needed to update Tableau Server.

Assume you have five new users to add and will be provisioning Interactor licenses for all of them. Figure 13-8 shows the CSV file with the names of the users.

	A	B
1	Username	Password
2	Brenden	b1
3	Phil	p1
4	Joe	j1
5	Paul	p1
6	Darren	d5

FIGURE 13-8 *CSV file containing new user list*

Creating a robust script that will work flexibly is the goal. Doing that requires a little knowledge of Windows commands and tabcmd. Figure 13-9 shows one way to accomplish adding the users.

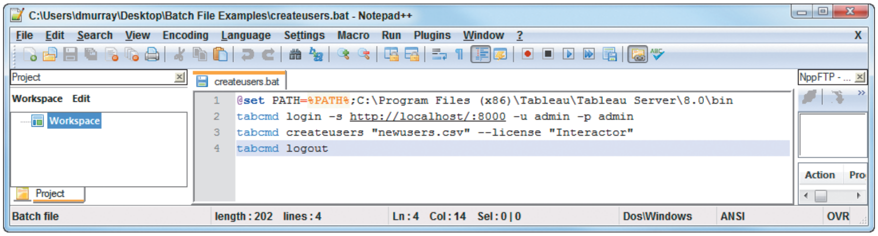


FIGURE 13-9 *Creating and saving script in a .bat file*

For the batch file to run properly, place it in the same directory as the CSV file that contains the users who need to be added to Tableau Server. The first line of the code in Figure 13-9, `-@set PATH=%PATH%`, defines the path for the file to search if any elements are not located there. These Windows commands allow you to define the path for the batch session only. This is a better practice than blending data files in with Windows system files (not a good practice). It also has the virtue of persisting only while the batch file is being executed—rendering the earlier example of permanently editing the PATH system variable unnecessary.

The rest of the script in Figure 13-9 includes tabcmd commands that are located in the `bin` folder specified by the `set path` command. In fact, you can define many different paths using this method for files that you want to keep separated.

The bullet list that follows may be easier to read than Figure 13-9. Alter the specific code where applicable to match your system's setup and the name of the CSV file that you created to load new users:

- **Line 2:** Log in to Tableau Server.
- **Line 3:** Create the users from the `newusers.csv` file.
- **Line 4:** Log out of Tableau Server.

When the program starts, you'll see each command run, and when it is finished, the command window will close automatically. Figure 13-10 shows a screenshot of the Windows Command Processor window—running the script.

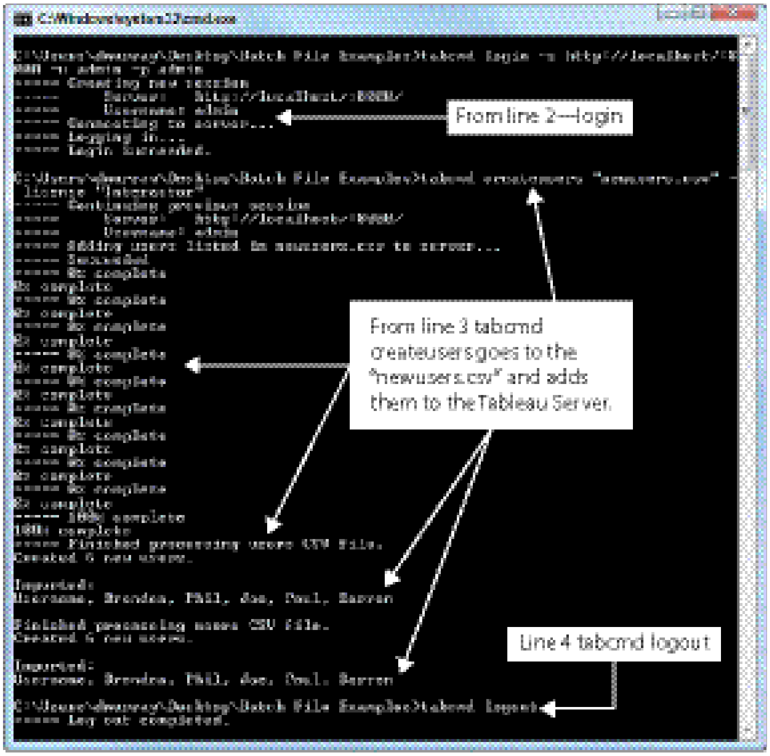


FIGURE 13-10 *The executed script*

As the program finishes, the screen automatically clears. If you want to keep it in view, add a fifth line to the script using the `Pause` command. With just a few lines of code, you can update many records this way. If you aren't using Active Directory to secure Tableau Server, this method provides a quick way to mass load hundreds of users from a file.

USING WINDOWS SCHEDULER TO FULLY AUTOMATE SCRIPTS

By adding the batch file from the previous example to Windows Task Scheduler, the file can be updated based on a trigger event or a specific time schedule. For example, if you have an actively expanding user base, it might be necessary to create new users in the system every day. The system administrator could add new users to the `Createuser.csv` file and schedule updates every day at a specific time. Figure 11-12 shows the Task Scheduler application. A new task was created to add users to Tableau Server (daily). The following steps were used to define the schedule:

1. **General tab:** Name and describe the task and set security options.
2. **Triggers tab:** Define what causes the action (daily at 7:00 a.m.).
3. **Actions tab:** Select the batch file to run (point to the `Createuser .bat` file).
4. **Conditions tab:** Set desired limitations for the run to occur.
5. **Settings tab:** Specify additional settings affecting the task behavior.

This will cause the file to be updated on a regular basis without the need for the batch file to be manually selected. Figure 13-11 shows the task scheduled for automatic update of new user additions.

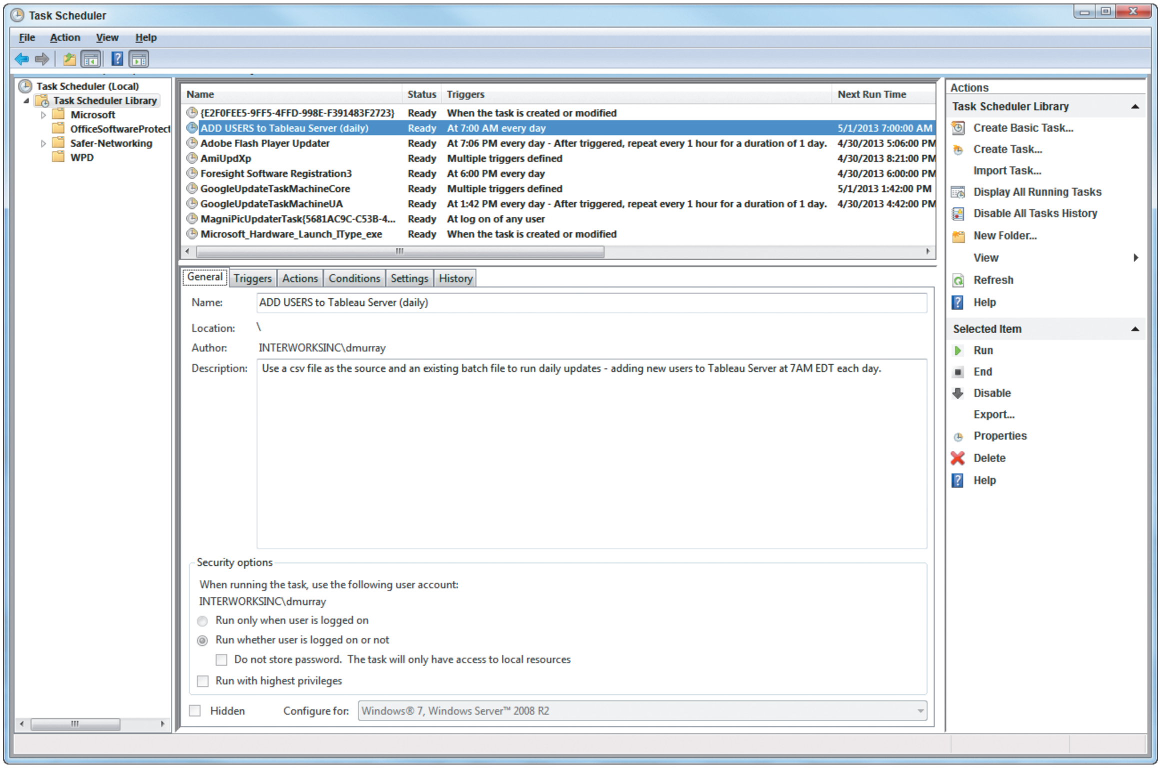


FIGURE 13-11 *Task Scheduler window*

Even if you're on vacation, updates can continue if you delegate the task of adding the usernames and license level to `Newusers.csv`.

COMMON USE CASES FOR TABCMD

There are many different ways to utilize tabcmd to automate repetitive or intensive production issues. If you find yourself doing repetitive tasks consistently, you should consider using tabcmd to automate the process to Save time, improve accuracy, and enhance the way you can share and update files.

The examples presented next are intended as a sampling of the ways you could use tabcmd. You will undoubtedly think of many more ways to automate processes that repeatedly require your attention.

Retrieving Forecast Data from Workbooks

Tableau's forecasting ability can be used to create initial projections based on historical patterns. The tabcmd Export function can be used to publish forecasted data points from a workbook view. Exporting data in CSV format can then be used to update a source database or a spreadsheet. This first-pass view of the forecast can then be tweaked and returned to a database and stored.

Even more commonly, historical data can be published as well. Even though it may be easier to analyze data using Tableau, some users may lack license access. You may wish to share exported PDF, PNG, or CSV files with vendors that don't have access to Tableau. Alternatively, you might publish packaged workbooks specifically for partners and allow them to access specific groups on your server.

Manage Data Governance via tabcmd

You may want to create a quality control directory that you publish raw files to for review and then, after auditing and approval, use the `Publish` command to move the preliminary file into a production group or project. This is an interesting alternative to heavy-handed quality control. Instead of focusing on the end report, IT can focus on ensuring the quality of the data extract file and provide information to consumers with a vetted preliminary view that can be modified to suit specific needs.

Using tabadmin for Administrative Task Automation

The tabadmin toolset is intended for use by the designed server administrators responsible for configuring and maintaining Tableau Server's data and metadata. Tabadmin has its own set of commands that are exclusively used for these purposes. You can find the complete list of Tabadmin commands at `http://onlinehelp.tableau.com/current/server/en-us/tabadmin_cmd.htm`.

Typically, a very limited number of technical staff members are tasked with the responsibility of developing, maintaining, and monitoring system performance. The tasks performed using Tabadmin include the following:

- Tabadmin help
- Conducting system backups and restores
- Displaying information on system status
- Cleaning service log files
- Resetting the password for the Tableau Server account
- Enabling or disabling access to Tableau Server's Postgres database
- Creating zipped log files
- Stopping Tableau Server

Starting Tableau Server

The `tabadmin` command tool may also be useful for additional tasks. The recommendations in the list that follows should be attempted only under the direction of Tableau Software Support or a qualified Tableau Server Partner.

- Altering default timeout provisions for queries
- Changing default timeout limits for idle users
- Creating a server log file
- Configuring Tableau Server processes
- Printing Tableau Server license information
- Printing information on active users
- Setting primary and secondary gateway hosts
- Executing system changes via the `configure` command

The Tabadmin command-line utility is the primary tool maintaining the safety and performance of the server. Tableau Software provides extensive documentation on its website. Refer to Chapter 11 for more details on setup and configuration. Refer to Tableau's online help, the Tableau Server Administrator Guide, at `http://onlinehelp.tableau.com/current/server/en-us/help.htm#ports.htm`, for details on default TCP/IP port settings.

Tableau's online manual is an excellent resource for the latest information related to Tableau Server.

AUTOMATING EXTRACTS WITH THE EXTRACT API

Tableau Server provides an easy-to-use extract refresh scheduler that meets the needs of most publishers and server administrators. There are times when a solution outside of Tableau Desktop and Tableau Server may be required. Two common scenarios are

- Tableau does not provide a native connection to the data store you'd like to use.
- Tableau Server cannot access the data you'd like to refresh.

The first scenario requires use of Tableau's Data Extract API, while the second scenario can be handled by any publisher with Tableau Desktop installed on their machine.

DATA EXTRACT API

The Data Extract API provides developers with a tool for creating programs to access and manipulate data from any source and convert that data into a Tableau data extract file. The Data Extract API is especially useful in situations where the data source is not natively supported by Tableau. While it can also be used to automate extract creation and refreshes outside of Tableau Server, the Data Extract Command-Line Utility (`http://onlinehelp.tableau.com/current/pro/online/mac/en-us/extracting_TDE.html`) is able to handle those tasks without the need to write a program.

Using the Data Extract API does not require the purchase of Tableau Desktop or Tableau Server and is available in 32/64-bit versions for Python, C/C++, and Java on both Windows and Linux. Tableau does not offer a Mac solution at the time of this writing.

To get started, visit Tableau's Get the Data Extract API page (`http://www.tableau.com/data-extract-api`). After accepting the license agreement, you'll be given the option to download a package for your language and environment of choice. Each package includes documentation and samples that demonstrate usages of the API given the language and environment you've chosen. For this section, you'll use Python because it is free, is easy to use, and has a large support community.

Download and install the latest version of Python 2.7.x from the Python website at `https://www.python.org/downloads`. Choose either the 32-bit or 64-bit version based on your hardware's capability. For Windows users, Python provides an MSI package that will automate installation for you. By default, Python will install to the root of your C: drive with the version number embedded in the name: `C:\Python27`. Rather than using this path every time you use Python, it is more efficient to add it to the PATH environment variable under Control Panel ➢ System ➢ Advanced system settings ➢ Environment Variables.

Once Python has been installed, it is time to install Tableau's Data Extract module. Navigate to the downloaded file and extract its contents. Within the extracted folder there will be a file named `setup.py`, which you can use to install the module. Open a Windows command prompt, navigate to the extracted folder, and type the following command: **`python setup.py install`**.

A number of log entries will be displayed as Python installs the module. Once it has finished, you'll be taken back to the command prompt. To test whether the module was successfully installed, begin by opening the Python Interpreter by typing **python**.

You'll be presented with a command-line console interface where you can enter commands after the `>>>` indicators. Test the module installation by typing **import dataextract**. If no errors are returned, the module has been installed correctly. Exit the Python Interpreter by typing **exit()**.

Within the extracted folder is a samples directory. In that folder Tableau provides a sample Python script named csv2tde.py. You can use this script to turn any CSV file you have into a Tableau data extract files after you enable some simple updates to the corresponding `schema.ini` file. Running this script requires a single command. Begin by opening the `schema.ini` file in your favorite text editor. Its contents are

```
[myfilename.csv]
ColNameHeader=True
col1=col1name col2datatype
col2=col2name col2datatype
...
```

Within the `schema.ini` file, you provide the name of the CSV file and some information about the columns it contains. For each column, provide its name and match it to one of the following data types:

- Bit
- Byte
- Short
- Long
- Integer
- Single
- Double

- Date
- DateTime
- Text
- Memo

Once finished, place your CSV file, `schema.ini`, and the csv2tde.py Python script in the same directory. Then, within your open command prompt, run the script by typing the following command: `python csv2tde.py myfilename.csv`.

As the script runs, you'll see a notification informing you the extract is being created and the total elapsed time to complete the program run.

While in most scenarios you won't be using the Extract API to generate extracts from CSV files, this script does a good job of showing the common steps to creating any Extract API program:

1. Import the API module.
2. Create an `extract` object.
3. Specify the extract table definition (column names and data types).
4. Create a table in the extract using the definition.
5. Add rows consisting of data columns to the extract table by iterating through your data.
6. Once complete, close the extract file using the `close()` function.

Let's look at one way you might utilize the Data Extract API with Python to pull data from a website and append the information to a file.

USING THE EXTRACT API WITH PYTHON

In this example, Python with the Data Extract API is used to create an extract file and then append new data to an existing Tableau data extract (`.tde`) file. The sample data pulls the average February temperature for the 48-contiguous U.S. states from a government website. The extract file created will have three columns:

- Date
- Average Temp
- Anomaly

The lines in the example script that are preceded with a # sign are notes in the script that are ignored by Python. The first three lines in the program import the extract and other necessary modules. Important Python keywords used in this program include

- `def`: Used to create a new user-defined function
- `try`: Specifies exception handlers
- `finally`: Executed at the end of a `try` statement to clean up resources
- `if`: Used to determine which statements will be executed
- `return`: Executes a function and returns a value
- `print`: Print to console
- `with`: Ensures the declared resource (extract file in this case) is closed even in the event of an exception

A small heading is printed before starting the real work. Two calls are made, each to `download_noaa_data` with slightly different requests. The `download_noaa_data` function is a utility function that performs the high-level actions, including pulling the data from the given URL and calling other functions to populate the extract. It is important to note that the extract is declared using the `with` statement to ensure that it is properly closed. It calls `create_table_if_needed` to ensure the Extract table is present. Then the script opens the table and retrieves the table definition. The weather data is parsed into the correct format next. Four lines of header text are skipped, and then each remaining line is split into individual components. These components are converted to Python types and passed into `create_row`. Exiting the `with` statement closes the file, and then the `urlopen` returned object is closed and the program exits.

The `create_table_if_needed` function checks for an existing Extract table and creates the table if one is not already present. It makes the `TableDefinition` with the columns and adds the table to the extract file using the `addTable` call. This part of the script uses the Type class constants to specify the types in the `addColumn` calls.

`Create_row` makes a Row object and sets the column values for the columns from the functions arguments. It then calls the `insert` method to insert the new row of data into the table.

Here is an example of how the Python script should look:

```
# Tableau Data Extract API requires Python 2.6 or higher and is not
compatible with Python 3.
# http://onlinehelp.tableau.com/current/pro/online/en-us/extracting_TDE_
API.html for more information.

import datetime
import urllib2
```

```
import dataextract as tde

# Given data for a row, create and add to extract.
def create_row(tde_table, tde_table_def, date, avg_temp, anomaly):
    tde_row = tde.Row(tde_table_def)
    tde_row.setDate(0, date.year, date.month, date.day)
    tde_row.setDouble(1, avg_temp)
    tde_row.setDouble(2, anomaly)
    tde_table.insert(tde_row)

def create_table_if_needed(tde_file):
    if not tde_file.hasTable('Extract'):
        tde_table_def = tde.TableDefinition()
        tde_table_def.addColumn('Date', tde.Type.DATE)
        tde_table_def.addColumn('AvgTemp', tde.Type.DOUBLE)
        tde_table_def.addColumn('Anomaly', tde.Type.DOUBLE)

        tde_file.addTable('Extract', tde_table_def)

def download_noaa_data(url):
    data = urllib2.urlopen(url)

    try:
        # Open extract, Extract table and table definition.
        with tde.Extract('our extract.tde') as tde_file:
            create_table_if_needed(tde_file)
            tde_table = tde_file.openTable('Extract')
            tde_table_def = tde_table.getTableDefinition()

            for line in data.readlines()[4:]:  # Skip header, add each
line to extract
                print line
                raw_date, raw_avg, raw_anomaly = line.split(',')
                date = datetime.datetime.strptime(raw_date, '%Y%m').
date()
                avg = float(raw_avg)
                anomaly = float(raw_anomaly)
                create_row(tde_table, tde_table_def, date, avg, anomaly)

    finally:
        data.close()

print 'Downloading United States 48-contiguous average temps from NOAA'
print 'Downloading from start to 2010...'
download_noaa_data('http://www.ncdc.noaa.gov/cag/time-series/us/110/00/
tavg/1/02/1895-2010.csv?base_prd=true&firstbaseyear=1901&lastbasey
ear=2000')

# Now append a second batch to same extract.
print 'Downloading after 2010...'
```

```
download_noaa_data('http://www.ncdc.noaa.gov/cag/time-series/us/110/00/
tavg/1/02/2011-2015.csv?base_prd=true&firstbaseyear=1901&lastbasey
ear=2000')
```

If you want to learn more about the Extract API, Tableau provides a series of videos on the API download page at `www.tableau.com/products/api-download`.

DATA EXTRACT COMMAND-LINE UTILITY

In cases where Tableau Server cannot directly access the data you'd like to refresh, the Data Extract Command-Line Utility can be used to automate refresh tasks. This is a common scenario when hosting your workbooks with Tableau Online, Tableau's cloud-based hosting solution, rather than on premises.

The utility is available on any machine where Tableau Desktop is installed under Tableau's Program Files path.

- **32-bit:** `C:\Program Files\Tableau\Tableau 9.1\bin`
- **64-bit:** `C:\Program Files (x86)\Tableau\Tableau 9.1\bin`

Tableau's online documentation (`http://onlinehelp.tableau.com/current/pro/online/mac/en-us/extracting_TDE.html`) provides a full list of command and parameter options available. Available commands include

- `refreshextract`
- `addfiletoextract`

Each command accepts a set of parameters, including a useful `--help` parameter, which provides additional information for many commands. Other parameters enable you to define where the data is located; the data source, username, and password (if required); where the extract should be published; and so on.

In the example that follows, you'll see how to add data to an existing extract on Tableau Server, assuming the data is currently residing on your machine in a CSV file.

Begin by opening a command prompt and navigating to the Tableau Program Files path listed earlier. Next, type the following command:

```
      tableau addfiletoextract -s http://mytableauserver.com  -t
"MYSITENAME" -u MYUSERNAME -p MYPASSWORD --project "MYPROJECTNAME"
--datasource MYDATASOURCE --original-file "C:\PATHTOMYFILE\FILENAME.CSV"
```

You can also refresh your data source using data residing in a database. Tableau automatically uses the same database and server specified in the published data source, but you'll need to supply the `--source-username` and `--source-password` parameters if not using Windows Authentication to connect to the database.

REST API

Tableau's REST (Representational State Transfer) API (Application Program Interface) enables you to manage Tableau Server programmatically. The REST API has been part of Tableau since 8.2, but it had to be turned on via `tabadmin set api.server.enabled true` before it was available. In Tableau Server 9, the REST API is turned on by default but can be disabled with `tabadmin set api.server.enabled false`.

For those unfamiliar with REST, it is an architecture that provides guidelines and best practices as put forth by the W3C (World Wide Web Consortium). All that means is that Tableau's software engineers worked with a well-defined and vetted set of rules when building this service into Tableau Server.

For you, it means that you will communicate with Tableau Server using common HTTP methods, such as POST, GET, PUT, and DELETE. These requests must also be properly formatted as XML, as defined by Tableau (you'll get to these details in just a moment).

Please note that the REST API can also be used with Tableau Online (Tableau's Cloud offering), but it is limited to a subset of REST API methods. Check Tableau's online help for details at `http://onlinehelp.tableau.com/current/api/rest_api/en-us/help.htm#REST/rest_api_ref.htm#methods_not_available_in_tableau_online`.

INITIAL TRANSACTIONS

The REST API can be used most effectively by the system administrator who has full access to all of the REST API methods. Site administrators can also use the REST API but will not have access to all of the REST methods. Tableau's REST API documentation indicates what and who can access these methods. At a high level, this is what an initial transaction looks like:

- A user sends an XML-formatted HTTP request to the Tableau Server REST endpoint (the designated URI that accepts REST requests).
- The REST service returns with its own XML formatted HTTP response, which tells you if your request was successful. If you were requesting some specific set of information, it will return that as well.
- You can now parse out the information returned to use in your application.

Let's take this high-level overview and see how you would use it to authenticate access to the service:

1. The authentication endpoint is located at `https://www.yourtableauserver.com/api/2.0/auth/signin`.
2. You POST XML formatted data to the endpoint in step 1:

```
<tsRequest>
 <credentials name="username"  password="password" >
   <site contentUrl="site-name" />
 </credentials>
</TsRequest>
```

3. The code variables in the following list must be specified:
   - `username`: The user you are authenticating as.
   - `password`: The password for that user.
   - `site-name`: This is actually what Tableau normally calls the site ID. It is case sensitive. You must authenticate to a Tableau site. If it is the default site, you leave the site name empty.
4. Tableau's REST service will respond with an XML-formatted response of its own:

```
<tsResponse
   xmlns="http://tableausoftware.com/api"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://tableausoftware.com/api http://
tableausoftware.com/api/ts-api-2.0.xsd">
 <credentials token="0ec4eb5fe07bb01b786b77f41bba87db">
   <site id="1f6b0e92-bf05-4c3f-bb43-02be4ed9e36f"
contentUrl="my-site"/>
   <user id="93bd1906-d31b-4a12-86cf-e16fd0fa3efd"/>
 </credentials>
</tsResponse>
```

This response includes some information that you will need to use to communicate and interact further with the REST API.

- `token`: The token is an authentication passcode that you will need to use when making any other REST requests. It tells Tableau Server that you've already been authenticated.
- `site id`: This is *not* the Tableau site ID as you normally know it. It is a unique identifier for the site that you are connecting to, which you will need to store for future/additional requests.

- `site contentURL`: This is what Tableau documentation typically calls the site ID.
- `user id`: Much like the site ID, this is a unique identifier for the user who is being acted upon. You will need to store this if you want to use it for future/additional requests.

In whatever language being utilized (Python, Ruby, JavaScript, or something else), you will need to extract the relevant values out of the XML response in order to use them for future requests.

If you are unfamiliar with programming and are not sure how to get started, try a free Chrome (Google Browser) extension called Postman. Postman is an easy way to get familiar with REST APIs, and it requires no programming experience. You simply install the extension, and then with the interface it provides, you define your endpoint, how you would like to send or request the data (POST, GET, PUT, DELETE, and so on), and the body of the message you'll be sending. In this case, you POST the raw XML data, as you see in Figure 13-12.

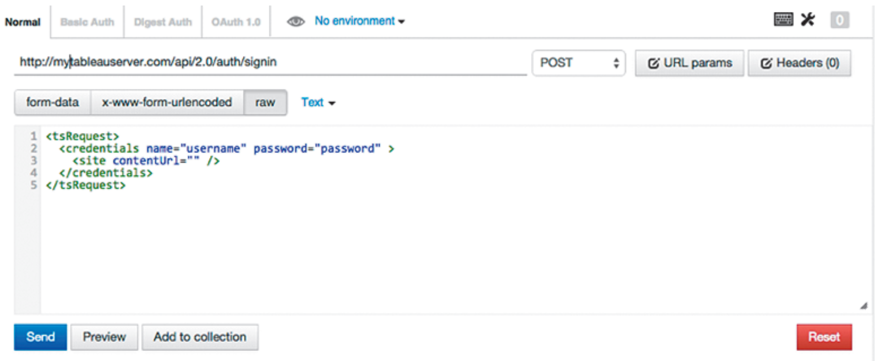


FIGURE 13-12 *Postman*

Once you have entered in the information, you can click Send, and Tableau Server should reply with an XML response. If you have incorrectly entered in information, the REST API may respond with an error message. Carefully look over what the error message says and fix any mistakes.

A common error is to type in **default** for the `contentUrl` when, in fact, it should be left empty if you want to connect to the default site. If you need to authenticate to a different site, then enter that site in the Tableau site ID. If everything worked properly, you should receive a response that looks similar to Figure 13-13.

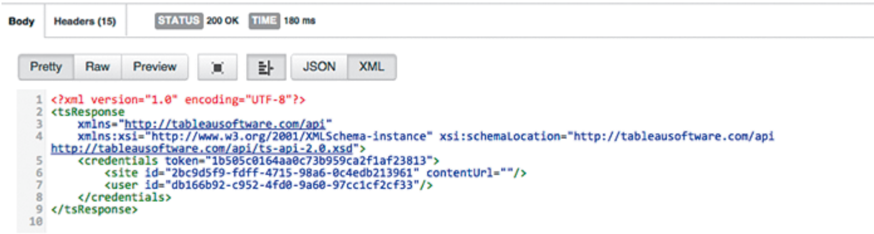


FIGURE 13-13 *Response*

Notice that the response `contentUrl` is empty. This is expected if you are connecting to Tableau's default site (which in this example you are). If it is any other site, the site `contentUrl` will have a value.

Now you will have a token, a unique identifier site ID, and a unique identifier user ID that you can use to further interact with Tableau Server. The REST API allows you to interact with sites, projects, workbooks, data sources, users, groups, permissions, jobs, favorites, and publishing.

A full listing of functionality can be found at `http://onlinehelp.tableau.com/current/api/rest_api/en-us/help.htm#REST/rest_api_ref.htm#API_Listing`.

Let's take a look at an example using Postman. It's not immediately obvious how you should use the token retrieved. It would be fair to assume that there might be an additional XML tag that must be added to send the data back, but in this example there is not.

The token gets sent to Tableau via the Header. In Postman, there is a Header button, which will let you input a new header variable. The token header variable name is `X-Tableau-Auth`, and the value is the string of alphanumeric digits that were returned in the credentials token attribute when you authenticated. In this example the token provided is

```
1b505c0164aa0c73b959ca2f1af23813
```

Figure 13-14 shows the appearance in context.

To add a new project to our site, you can POST to a new endpoint:

```
/api/2.0/sites/site-id/projects
```

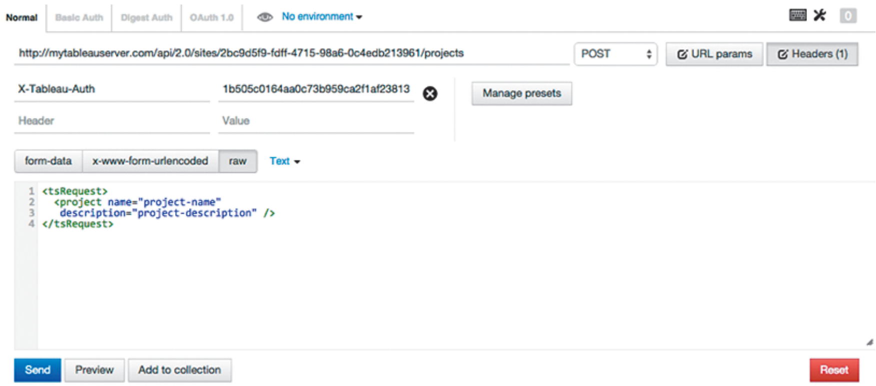


FIGURE 13-14 *Response*

You know that this is the endpoint because it is explicitly defined in the documentation as the URI to use when adding a project. The `site-id` section of the URI must be replaced with the unique identifier site ID, in this case:

`2bc9d5f9-fdff-4715-98a6-0c4edb213961.`

The request body structure is also defined in the documentation, so you know from there how to form this XML.

```
<tsRequest>
<project name="project-name"
description="project-description" />
</tsRequest>
```

You will replace `project-name` with the name of your project and the `project-description` with your description.

Once you've made your changes, click Send. This creates a new authenticated request to Tableau Server. If everything went well, you should see a 200 OK result. This indicates that the project was created successfully on your site. You can continue to use the authentication token for this session to pass any more requests to the server.

The preceding sample shows how to create something on the server, but you can also retrieve information in a similar fashion. For instance, if you want a list of users in a specific site, you can use the same principles to send a request to the appropriate endpoint: `/api/2.0/sites/site-id/users/` with the authentication token header. There is no body XML data because you're simply requesting info.

Tableau Server will respond with an XML response listing the users in the specified site. It is up to the application engineer to parse this data and use as needed.

Tableau's REST API is a great way to manage the server or retrieve server data remotely. The recent updates to the REST API have greatly increased its capabilities, making it a formidable tool to include in your Tableau arsenal.

Tableau Software continues to improve Tableau Desktop and Tableau Server. In the appendixes that follow, you'll find details on Tableau's product ecosystem, supported database connectors, keyboard shortcuts for Windows and the Mac, and recommended hardware configurations for the desktop and server products.

If you are new to Tableau and are not familiar with SQL syntax, the function reference should be a helpful resource that you refer to often. It provides an extensive listing of Tableau functions with code samples and explanations that will give you an alphabetically sorted reference for entering the correct syntax for each function.

Finally, Appendix F provides an overview of the companion website where you will find all of the example files used in this book.

PART III

CASE STUDIES

In this part

CHAPTER 14

Ensuring a Successful Tableau Deployment

Why should you engage outside consultants? Is starting an internal user group worth the time and effort? In this section, I outline the data landscape and how you can effectively use consultants.

DEPLOYING TABLEAU—LESSONS LEARNED

Whether you work in business, education, or government, your workplace is a complex machine with many moving parts, evolving goals, and internal politics. Successful deployments do not happen in a vacuum, nor do they follow a single formula for success. Leadership wants safety and innovation. Every data project the InterWorks team has been engaged in over the past 15 years has balanced these needs.

Unlike most technology solutions, Tableau's ease-of-use and flexibility create grassroots demand. If you work in IT and you are planning a major Tableau deployment, it's likely that many people outside of your group have been using Tableau Desktop before you learned about Tableau Software. It is also possible that your internal customers are demanding that you deploy Tableau's products (Desktop and Server) so that more people can create and consume dashboards and workbooks. This makes your job easier. You don't have to sell your users because they are selling you. The question is: How do you deploy Tableau Desktop and Server quickly, safely, and effectively?

EFFECTIVE USE OF CONSULTANTS

I spent many years purchasing technology, deploying solutions, and running businesses before I became a consultant. After trying to deploy technology without the help of consultants, I realized that consultants are a tool for getting to the right solution faster and with more safety. Bosses don't like surprises, especially ones that cause delay and unanticipated expense. Careful consideration of your present data needs, team knowledge, and available capacity will drive the degree you employ outside resources, but you would be wise to utilize experienced and knowledgeable resources to augment your internal team.

Effective use of consultants for a Tableau deployment doesn't normally mean farming out all the work. In addition to Tableau product knowledge and deployment experience, what other qualities should you be looking for?

YOUR TEAM'S CURRENT KNOWLEDGE

InterWorks has provided consulting services for many of the most technically savvy companies in Silicon Valley and across many different industries worldwide. Technically adept companies are a significant part of our practice. Why? Based on my latest count, at least 250 data products are available today. It is impossible for a single entity to possess expert skill in every tool. These companies engage consulting help because they want predicable outcomes as quickly as possible. Consultants are used to augment their already capable teams due to existing project workloads and the desire to learn how the Tableau product stack works in real-world situations. Managers want their initial deployment to be done right.

There are many more entities (businesses, schools, government) that have limited IT resources. They have many of the same needs but don't possess the budget to hire permanent staff. Consultants can be engaged to provide advice, training, project management, and even ongoing management.

What is your team's present workload? How fast do you need to deploy? How many people will you need to train? What does your data landscape look like?

THE DATA LANDSCAPE

You have an existing data landscape. It may be the result of a carefully planned effort, but it is more likely that it evolved over time as your team grew and your needs changed. Your data landscape is a combination of the following:

- **Data foundation:** The software and hardware in place
- **Data quality:** The degree of your data's accuracy
- **Data governance:** Security based on proprietary and legal requirements
- **Data training:** The desired versus current knowledge of team

- **Data availability:** The amount, timing, and quality of data you want to make available to your internal and external customers
- **Data discovery:** Your team's ability to use data to create insight
- **Data culture:** Your company's desire to base decisions on data versus gut feel

Your level of readiness in these areas will determine the type and number of outside resources you need to ensure a successful outcome. Major Tableau deployments are less about Tableau's software and more about assessing your data foundation and the skill and knowledge of your team.

Experience has taught me that Tableau is always a useful addition to any company's data landscape. Look for partners that have proven track records that can also provide relevant references. Industry-specific experience is less important than technical know-how, business acumen, and the ability to articulate technical topics in business terms.

After you have deployed Tableau, it's wise to find an internal resource (a Tableau zealot) that is willing to run an internal user group. I also recommend getting involved with a Tableau User Group within your city. One of the best ways to broaden your knowledge of Tableau is to see how people in other industries are using Tableau.

THE TABLEAU USER GROUP AT CIGNA

Donna Costello is responsible for the Business Intelligence Center of Excellence (BI COE) at Cigna. She invited me to speak at one of Cigna's internal user group meetings. I was so impressed with the Cigna team and the fun way they engaged their Tableau user base that I asked Donna to share how they manage their internal user group.

TAKING CARE OF VIZNESS

Cigna is a global health service company that offers health, life, accident, dental, and disability insurance. Cigna uses Tableau as a tool to enable rapid analytics of big data. From the outset, Cigna's Tableau users faced three major challenges common to many companies:

- Many of the users were new to Tableau.
- The users are geographically spread across the country.
- The users are separated across a diverse range of business segments.

In light of this, Tanya McNamar, Andrew Duncan, and Caitlin "CJ" Barry from Cigna's Business Intelligence Solutions (BIS) Team created a monthly Tableau User Group (TUG). They designed the Cigna TUG to bring together twenty of Cigna's Tableau users and provide them with the opportunity to learn and share visual analytics solutions by discussing best practices, hearing peer presentations, watching live demos, and communicating training needs. The Cigna TUG's value lies in the combination of visual inspiration, skill development, and sense of community it brings to the business. Along with creating much of the early content, Andrew coined the TUG's catchy mantra "Taking Care of VIZness." Cigna applies these practices to initiate, sustain, and improve their use of Tableau and the organization of their user group.

RESOURCING

The TUG is a partnership between all Tableau users in the company, the business, and IT, but there must be a dedicated TUG champion. BIS decided early to hand off ownership of the TUG to Donna Costello from their BI Center of Excellence (BI COE), who has maintained focus on the growth of user skills and the needs of the user base. The BI COE carefully organizes all TUG documents, from recordings to presentations to workbooks, in one intranet repository by date and general title of each session. The BI COE champion also tracks licensed users and regularly updates the attendee list to ensure new users will have support as they get started with Tableau.

CADENCE

In an effort to have an organized and exciting session, the BI COE executes two planning sessions prior to the execution of every TUG, usually more than 30 days in advance. During planning, the BI COE meets with volunteers to discuss topics the volunteers would like to showcase. While there is often a long list of topics to select from, the BI COE focuses on two topics that the volunteers bring to the table. During this initial conversation, the TUG champion focuses on selecting the material and making sure the different showcases connect or flow properly at a high level. Exact details and perfect presentation are less important at this point. One month out, the BI COE emails a communication to the TUG audience with an agenda. One week out, the BI COE meets again with the volunteers to do a dry-run of the presentation. On the day prior to the TUG, the BI COE sends materials to the attendees in order to allow them the opportunity to consider any questions ahead of time.

Cigna conducts its TUG on the same day and time every month: the third Thursday of every month 1:00–2:00 p.m. At the conclusion of the TUG, the BI

COE posts the recorded session and materials to the BI COE intranet site and sends an email to users with the posted location.

FORMAT

The BI COE uses a standard suite of mediums and capabilities during its TUG, including WebEx, PowerPoint, Tableau, and Internet websites. The BI COE records each TUG using WebEx. These recordings are valuable for users who want to go back and re-watch techniques they saw or catch sessions that they missed. Each session starts off with a screen share of the PowerPoint slide deck, which primarily houses an agenda. The agenda regularly consists of the following:

- Welcome, presenter introductions, and agenda review (by the TUG champion)
- One or two dashboard showcases, techniques, tips, best practices, or user stories (usually 15–20 minutes for each volunteer or guest speaker)
- Enterprise updates such as performance or upgrades (by the Server Administrator or IT)
- Question and answer (from the audience)
- Group announcements (by the TUG champion)
- New resources (by the TUG champion)

TOPICS

The BI COE has found no shortage of topics to showcase. We have discovered that as time goes on, the topic options become more plentiful. Here are a few combinations of sessions conducted by the Cigna TUG:

- What is the Tableau User Group? The Tableau overview and the business segment dashboard demo.
- Business segment Tableau dashboard overview.
- Business segment Tableau case study and dynamic hierarchies in Tableau.
- Business segment dashboard showcase, training on row-level indicators.
- Dashboard sharing techniques, an overview of the recent Tableau upgrade.
- Visualizing JIRA data, blends, and joins in Tableau.
- Business segment dashboard showcase—Mapping techniques.
- Dashboard action techniques.

- Mapping and dual axis visualization training.
- Discovery tool demonstration and control chart creation.
- Guest Speaker—Tableau Zen Master Dan Murray, Tableau Adoption & Deployment.
- Tableau upgrade features.
- Business segment dashboard showcase, Tableau Public use cases, building a calendar and dashboard best practices with Story Points.
- Guest Speaker—Tableau Zen Master Joe Mako.
- Guest Speaker—Tableau Zen Master Mark Jackson.

EFFECTIVENESS AND ATTENDANCE

Every quarter, the BI COE sends a survey to the attendees following a TUG session. Studies show that Monday morning is the optimal time to send a survey. In order to measure effectiveness, it is important to spend a significant amount time early on thinking through the questions you want to ask and keeping these consistent from quarter to quarter. The BI COE evaluates effectiveness based on the following survey questions:

1. What department do you work in?
2. How would you rate your overall knowledge or skill of Tableau?
3. Evaluate the following statements:
    - I will be able to apply the knowledge learned.
    - The content has been organized and is easy to follow.
    - The materials distributed afterwards were pertinent and useful.
    - Participation and interaction were encouraged.
    - Adequate time was provided for questions and discussion.
4. Please rate your overall satisfaction with the monthly Tableau User Group meeting.
5. Please rate your overall satisfaction with our Tableau User Group SharePoint Site.
6. Check three topics that interest you the most:
    a. Connecting to Data
    b. Advanced Tips and Tricks
    c. Table Calculations

   d. Structuring Your Data
   e. Working with Filters & Parameters
   f. Basic Maps and Geo-Coding
   g. Working with Multiple Data Sources
   h. Groupings, Hierarchies and Sets
   i. Visualization Basics
   j. Working with Trend and Reference Lines
   k. Creating Custom Calculated Fields
   l. Working with Different Chart Types
7. How important is Tableau to your department?
8. What projects have you worked on using Tableau?
9. Do you have further feedback that will help us better serve you?

TRACKING PARTICIPATION

The BI COE tracks month-to-month attendance trends by populating a Tableau dashboard with a post-TUG report from WebEx. The survey and attendance report are important pieces of data required to measure the TUG's effectiveness and continue the skill growth of Cigna's Tableau users.

SUCCESS

Using these practices, Cigna's TUG grew from 20 members to more than 70 in a few months. They are building on this foundation with the goal of reaching 100 members in 2015. Cigna plans to introduce Tableau challenges as part of the Cigna Tableau User Group.

PART IV

APPENDIXES

In this part

- **APPENDIX A:** Tableau's Product Ecosystem
- **APPENDIX B:** Supported Data Source Connections
- **APPENDIX C:** Keyboard Shortcuts
- **APPENDIX D:** Recommended Hardware Configurations
- **APPENDIX E:** Understanding Tableau Functions
- **APPENDIX F:** Companion Website
- **GLOSSARY**

APPENDIX A

Tableau's Product Ecosystem

Tableau's product ecosystem has evolved over the past ten years but remains simple. Desktop is for discovery, design, and interactive consumption. Server is for scaling your deployment to an audience for consumption. However, Tableau has started to blur the lines between Desktop and Server by adding design capabilities into Server. Tableau Online is a cloud-based deployment of Server. Tableau Mobile provides optimized apps for consuming Server reports on iOS and Android devices. Tableau Reader is a free desktop application that enables information consumers to interact with Tableau packaged workbooks. And Tableau Public is a free cloud publishing platform that anyone can use to analyze data and publish interactive dashboards.

TABLEAU DESKTOP

Tableau Desktop comes in two flavors: Professional and Personal editions. The Personal edition only provides connections to files that exist on your desktop, and you can only share files that have been saved as packaged workbooks with people who are using Tableau Reader.

Professional provides the full suite of connectivity options and provides publishing options for Tableau Server, Tableau Online, or Tableau Public. See Appendix B for the current list of connection options.

TABLEAU SERVER

If you produce a large number of workbooks that have to be updated regularly or you have a large number of people consuming your work, Tableau Server will save you time. It allows your audience to view and interact with your work via a web browser. Tableau Server enhances data security and scalability as well. And Server provides a means for embedded live, interactive dashboards and visualizations in an existing web portal. Server will also automatically refresh data extracts that have been published to Tableau Server.

Server is licensed in two ways: named user or the processor cores in your server. Named-user licensing makes sense in smaller deployments when fewer than 150 people need access to Tableau Reports. In larger deployments with dynamic access requirements, core licensing is more cost-effective and reduces administrative time because the license is defined by the number of cores in your database server's processor.

Tableau Server provides enhanced security and permits users to customize their access to reports within boundaries defined by the server administrator. Tableau Server's interface provides users with tools for finding, organizing, and commenting on reports. Server enables users to create subscriptions that provide email notification when updated reports are published. It also provides administrators with the capability to monitor access and monitor system performance. Details regarding installation, access, and management of Tableau Server are covered in Chapters 11 and 12. Tableau's native automation tools are covered in Chapter 13.

TABLEAU ONLINE

More people want to eliminate the need for procuring, installing, and managing hardware at one or more of their own business locations (on premise). As a result, cloud solutions are becoming more accepted, and Tableau Online is Tableau Software's cloud-based server option. Tableau Online can be turned on quickly and is accessed via password. It offers a hassle-free way to publish workbooks for others to interact with via the web.

TABLEAU PUBLIC

Tableau Public is a free, hosted web service that can be used to publish Tableau Reports on the web. Commonly used content management systems such as WordPress, Tumblr, and Typepad are supported. Tableau's licensed desktop editions can also publish content to Tableau Public. Tableau also offers a free Public desktop edition for creating and publishing reports. With the release of Tableau V9.0, Tableau Software increased the capabilities of Tableau Public. While you can still only publish your work to Tableau Public's server, your storage limits have been increased, and you're given more control over how your work is consumed:

- Storage space on Tableau Public is now 10GB per named user.

- The data row limit is now 10 million records.
- You can now prevent downloading of the source workbook through a settings adjustment in your account.

While Tableau Public isn't a substitute for Tableau Desktop and Tableau Server, this free product gives you the ability to learn Tableau Desktop (through the Tableau Public Desktop) and share your work with people using Tableau Public.

TABLEAU READER

Think of Tableau Reader as you would Adobe Reader. It's a free desktop application that allows you to consume Tableau Desktop reports that have been saved as packaged workbooks. This might be a good way to share analysis with business partners who don't have access to Tableau Desktop.

TABLEAU MOBILE

Tableau Mobile is available for iOS. While the Tableau Mobile app isn't required for accessing workbooks published on Tableau Server, these apps provide an optimized environment for consuming workbooks. And (provided your server administrator had enabled web-editing) you can create new views and new calculations from your mobile device. At the time of this writing, the iOS app is available only on the iPad. This facility will be provided for Android later in 2015.

PROJECT ELASTIC

Project Elastic is a new product under development that will be a separate application for authoring, editing, and consuming data totally within a mobile application. As this book is going to press Tableau has not provided a specific release date. The "whisper" date is some time in 2016.

POWER TOOLS FOR TABLEAU

InterWorks has been a Tableau partner since 2008, helping many businesses, government entities, and educational intuitions deploy Tableau Software by creating non-standard automation, providing performance-tuning solutions, and offering other customizations that are not part of Tableau Software's standard design.

Power Tools for Tableau was developed by InterWorks to address the most common client requests with a standardized set of add-on tools.

Power tools for Tableau consists of the following applications.

WORKBOOK TOOLS

- Workbook Merge
- Style Management
- Data Source Auditing
- Best Practice Analyzer
- Performance Analyzer

WORKBOOK SDK

Workbook SDK is a well-documented set of API tools that enable batch automation and programmatic access to Tableau workbooks.

ENTERPRISE DEPLOYMENT

Enterprise Deployment is a toolset to add, edit, or remove Tableau environments; manage transformation; and provide guided configuration.

REMOTE FOR TABLEAU

There are also mobile apps for iOS and Android that allow server administrators to manage Tableau Server remotely via a tablet computer or cell phone. For more information on these applications, see `http://powertoolsfortableau.com/` or email `powertools@interworks.com`.

APPENDIX B

Supported Data Source Connections

Tableau supports a wide variety of connections for both the Windows and Mac OS X editions. The connections listed in this appendix are updated through Tableau 9.1. Consult Tableau Software's Support website for Drivers and Activation for the latest connections: `www.tableau.com/support/drivers`.

WINDOWS CONNECTIONS

Tableau supports the following Windows connections:

- Amazon Aurora
- Amazon EMR (Elastic Map Reduce drivers for Hive and Impala)
- Amazon Redshift (32- and 64-bit)
- Aster Data nCluster (Teradata)
- Cloudera Hadoop Hive
- Cloudera Hadoop Impala
- DataStax Enterprise
- ExaSol EXASolution
- Firebird
- Google Analytics
- Google BigQuery (32- and 64-bit)
- Google Cloud SQL
- Greenplum (Pivotal)

- Hortonworks Hadoop Hive
- HP Vertica
- IBM BigInsights (32- and 64-bit)
- IBM DB2 (32- and 64-bit)
- IBM Netezza 4.6 or later
- MarkLogic (32- and 64-bit)
- MapR
- Microsoft Access 2007 or later
- Microsoft Azure SQL Data Warehouse (32- and 64-bit)
- Microsoft Azure SQL Database (32- and 64-bit)
- Microsoft Excel 2007 or later
- Microsoft PowerPivot (32- and 64-bit)
- Tableau Add-in for Microsoft PowerPivot (32- and 64-bit)
- Microsoft SQL Server 2005 or later
- Microsoft SQL Server Analysis Services 2000 and later (32- and 64-bit)
- Microsoft SQL Server PDW (32- and 64-bit)
- Microsoft SQL Server PowerPivot for MS Excel 2008, 2010 (32- and 64-bit)
- MySQL
- Odata
- ODBC
- Oracle Database (32- and 64-bit)
- Oracle Essbase (32- and 64-bit)
- ParAccel
- PostgreSQL
- Progress OpenEdge
- Salesforce.com, including Force.com and Database.com
- SAP BW
- SAP HANA
- SAP Sybase ASE 15.5 or later
- SAP Sybase IQ 15 or later

- Spark SQL
- Spark on Azure HDInsight
- Splunk
- Tableau Data Extract
- Teradata
- Teradata OLAP
- Teradata Unity
- Text file—comma-separated value (`.csv`) files
- Vectorwise

MAC OS X CONNECTIONS

Tableau Software supports the following data source connections for OS X:

- Amazon Aurora
- Amazon EMR (Elastic Map Reduce)
- Amazon Redshift
- Cloudera Hive
- Cloudera Impala
- Firebird
- Google Analytics
- Google BigQuery
- Google Cloud SQL
- Hortonworks Hadoop Hive
- HP Vertica
- MapR Hadoop Hive
- Microsoft Excel 2007 or later
- Microsoft SQL Server 2005 or later
- Microsoft Windows Azure Marketplace Data
- MySQL
- OData
- Oracle

- Pivotal Greenplum
- PostgreSQL
- Salesforce.com, including Force.com and Database.com
- Spark SQL
- Tableau Data Extract
- Tableau Server
- Teradata
- Teradata Unity
- Text file—comma-separated value (`.csv`) files
- ODBC 3.0 (and additional data sources compliant with this standard)

APPENDIX C

Keyboard Shortcuts

These keyboard shortcuts were taken from the Tableau Desktop online manual.

TABLE C-1 General Keyboard Shortcuts

	KEYBOARD SHORTCUT	
DESCRIPTION	**WINDOWS**	**MAC OS X**
Select all data	Ctrl+A	Command+A
Use Rectangular Selection tool	A	A
Lock and Unlock Rectangular Selection tool	Shift+A	Shift+A
Smaller cell size	Ctrl+B	Command+B
Bigger cell size	Ctrl+Shift+B	Command+Shift+B
Copy selected data	Ctrl+C	Command+C
Place selected field on Columns shelf	Alt+Shift+C	Option+Shift+C
Connect to data source	Ctrl+D	Command+D
Use Lasso Selection tool	D	D
Lock and unlock Lasso Selection tool	Shift+D	Shift+D
Describe sheet	Ctrl+E	Command+E
Makes the find command in the data pane active	Ctrl+F	Command+F
Place selected field on Filters shelf	Alt+Shift+F	Option+Shift+F
Enter/exit full screen		Control+Command+F
Switch in and out of presentation mode	F7, Ctrl+H	Option+Return
Place selected field on Size	Alt+Shift+I	Option+Shift+I
Flip orientation of column labels at bottom of view	Ctrl+L	
Place selected field on Detail	Alt+Shift+L	Option+Shift+L
New worksheet	Ctrl+M	Command+T
Turn off and on pan and zoom in a map	Ctrl+M+A	

(continues)

(continued)

	KEYBOARD SHORTCUT	
DESCRIPTION	**WINDOWS**	**MAC OS X**
New workbook	Ctrl+N	Command+N
Open file	Ctrl+O	Command+O
Place selected field on Color	Alt+Shift+O	Option+Shift+O
Print	Ctrl+P	Command+P
Place selected field on Pages shelf	Alt+Shift+P	Option+Shift+P
Place selected field on Rows shelf	Alt+Shift+R	Option+Shift+R
Save file	Ctrl+S	Command+S
Use Radial Selection tool	S	S
Lock and unlock Radial Selection tool	Shift+S	Shift+S
Place selected field on Shape	Alt+Shift+S	Option+Shift+S
Place selected field on Text/Label	Alt+Shift+T	Option+Shift+T
Paste clipboard	Ctrl+V	Command+V
Swap rows and columns	Ctrl+W	Control+Command+W
Cut text selection (for example, in captions, titles, formulas, and so on)	Ctrl+X	Command+X
Place selected field on Rows shelf	Alt+Shift+X	Option+Shift+X
Redo	Ctrl+Y	Command+Shift+Z
Place selected field on Columns shelf	Alt+Shift+Y	Option+Shift+Y
Undo	Ctrl+Z	Command+Z
Clear the current worksheet	Alt+Shift+Backspace	Option+Shift+Delete
Make rows narrower	Ctrl+Left Arrow	Control+Command+ Left Arrow
Make rows wider	Ctrl+Right Arrow	Control+Command+Right Arrow
Make columns shorter	Ctrl+Down Arrow	Control+Command+ Down Arrow
Make columns taller	Ctrl+Up Arrow	Control+Command+ Up Arrow
Show Me!	Ctrl+1, Ctrl+Shift+1	Command+1
Add the selected field to the sheet (only works with a single field)	Enter	Return
Opens the Help	F1	Shift+Command+ Question Mark
Deletes the selected sheet (on a dashboard)	Ctrl+F4	
Closes the current workbook	Alt+F4	Command+W

(continues)

(continued)

	KEYBOARD SHORTCUT	
DESCRIPTION	**WINDOWS**	**MAC OS X**
Starts and stops forward playback on the pages shelf	F4	
Starts and stops backward playback on the pages shelf	Shift+F4	
Refreshes the data source	F5	Command+R
Skip forward one page	Ctrl+Period	Command+Period
Skip backward one page	Ctrl+Comma	Command+Comma
Cycle forward through open worksheets	Ctrl+Tab, Ctrl+F6	Shift+Command+Close Brace
Cycle backward through open worksheets	Ctrl+Shift+Tab, Ctrl+Shift+F6	Shift+Command+Open Brace
Run update	F9	Shift+Command+0
Toggles Automatic Updates on and off	F10	Option+Command+0
Reverts workbook to last saved state	F12	Option+Command+E
Clears the selection (Desktop and Reader only)	Esc	Esc

TABLE C-2 Navigation and Selection Shortcuts

	KEYBOARD/MOUSE ACTION	
DESCRIPTION	**WINDOWS**	**MAC**
Selects the mark	Click	Click
Selects a group of marks	Drag	Drag
Adds individual marks to the selection	Ctrl+click	Command+click
Adds a group of marks to the selection	Ctrl+drag	Command+drag
Pans around the view	Shift+drag, click and hold+drag	Shift+drag, click and hold+drag
Zooms into a point in the view (requires zoom mode if not map)	Double-click, Ctrl+Shift+click	Double-click, Shift+Command+click
Zooms out from a point on a map (requires zoom mode if not a map)	Ctrl+Shift+Alt+click	Shift+Option+Command+click
Zooms out	Shift+double-click	Shift+double-click
Zooms into an area in the view (requires zoom mode if not a map)	Ctrl+Shift+drag	Shift+Command+drag
Zooms in and out on a map (Desktop and Reader only)	Ctrl+scroll	Command+scroll
Drags a row and scrolls through a long list simultaneously	Click+drag to bottom of pane+hold	Click+scroll, Command+hold

TABLE C-3 Field Selection Shortcuts

	KEYBOARD/MOUSE ACTION	
DESCRIPTION	**WINDOWS**	**MAC**
Opens the Drop Field menu	Right-click+drag to shelf	Option+drag to shelf
Copies a field in the view to be placed on another shelf or card	Ctrl+drag	Command+drag
Adds a field to the view	Double-click	Double-click

APPENDIX D

Recommended Hardware Configurations

Tableau provides minimum system specifications on its website, and those specifications are presented in the material that follows. Analysts who build reports should have better equipment. More internal memory will have a significant positive effect on speed.

Install 4MB to 8MB of internal memory for the best performance. Tableau's rendering engine will take advantage of modern graphics cards as well. Solid-state disk drives outperform physical hard disks. But don't outfit your report-building analysts with state-of-the-art equipment if the majority of your user base is using four-year-old junk. What performs well on a well-appointed computer may not be as enjoyable an experience on a dated system.

TABLEAU DESKTOP FOR WINDOWS: PROFESSIONAL AND PERSONAL EDITIONS

- Microsoft Windows Vista SP2 or newer (32- and 64-bit)
- Microsoft Server 2008 R2 or newer (32- and 64-bit)
- Intel Pentium 4 and AMD Opteron processor or newer (SSE2 or newer)
- 2GB internal memory
- 750MB minimum free disk space
- Internet Explorer 8 or newer

TABLEAU DESKTOP FOR MAC OS X: PROFESSIONAL AND PERSONAL EDITIONS

- IMac/MacBook computers 2009 or newer
- OX X 10.9 or newer

VIRTUAL ENVIRONMENTS

- Citrix environments, Microsoft Hyper-V, and VMware.
- All of Tableau's products operate in virtualized environments when they are configured with the proper underlying Windows operating system and minimum hardware requirements.

TABLEAU SERVER

The data sources are the same as Tableau Desktop. See Appendix B for a complete list.

SYSTEM REQUIREMENTS

- Microsoft Windows Server 2012 R2, 2012, 2008, 2008R2, 2003 SP2 with platform update, Windows 8.1, 8, or 7 on x86 or x64 chipsets
- 32-bit or 64-bit version of Windows
- Minimum of a Pentium 4 or AMD Opteron processor
- 32-bit color depth recommended
- Internet Protocol version 6 (IPv6), version 4 (IPv4)

WEB BROWSERS

- Android Browser (Android 3.2 or later)
- Apple Safari 3.*x* or later, including Safari on the iPad (iOS 5.1.1 or later)
- Microsoft Internet Explorer 8 or later
- Mozilla Firefox 3.*x* or later (not supported on mobile devices)
- Google Chrome, including on Android devices
- Tableau Mobile iPad (at the Apple App Store) and Android App (at the Google Play Store)

HARDWARE GUIDELINES

The minimum specifications are suggested only for prototyping and testing of Tableau Server. The installer checks for the minimum system requirements and will not proceed on computers with less than these hardware minimums:

64-bit computers

- 4 cores
- 8GB system memory
- 15GB minimum free disk space

32-bit computers

- 2 cores
- 4GB system memory
- 15GB minimum free disk space

The minimum configuration recommended for production usage of Tableau Server is based on these hardware specifications:

Single computer

- 64-bit processor
- 8 physical core, 2.0 GHz or higher CPU
- 32GB system memory
- 50GB minimum free disk space

MULTI-NODE AND ENTERPRISE DEPLOYMENTS

Contact Tableau for sizing and technical guidance or a certified Tableau Gold Partner. Search Tableau's website for scalability white papers and quick-start guides related to hardware configuration.

TABLEAU SERVER USER AUTHENTICATION AND SECURITY

Tableau Server supports Microsoft Active Directory, SAML 2.0 and the built-in Tableau users system, for user authentication and group membership definitions. Tableau Server also provides Kerberos support for Microsoft SQL Server, Microsoft SQL Server Analysis Services (SSAS) and Cloudera Impala.

VIRTUAL ENVIRONMENTS

All of Tableau's products operate in virtualized environments when they are configured with the proper underlying Windows operating system and minimum hardware requirements.

- Citrix environments
- Microsoft Hyper-V
- Parallels
- VMware
- All of Tableau's

INTERNATIONALIZATION

All of Tableau's products (with the exception of the command-line tools tabcmd and tabadmin) are Unicode-enabled and compatible with data stored in any language. The user interface and supporting documentation are in English, French, German, Spanish, Brazilian Portuguese, Japanese, Korean, and Simplified Chinese.

APPENDIX E

Understanding Tableau Functions

If you are accustomed to creating SQL statements in a database, the functions and syntax of Tableau's calculated values should look familiar. If you are a spreadsheet expert, the syntax will be new but should not pose a significant challenge for you to learn.

Tableau's formula-editing window provides help and error-checks the syntax of the formulas you create. Even if you have no experience, with a little practice you'll find that you use some functions frequently. Table E-1 groups functions into thirteen categories.

TABLE E-1 Tableau Function Categories

FUNCTION CATEGORY	CATEGORY CAPABILITIES
Aggregate	Mathematical and statistical summaries of your data.
Date	Calculate and parse date fields.
Google Big Query	Functions that work with Big Query data sources only.
Hadoop Hive	Functions that work with Hadoop Hive data sources only.
Level of Detail (LOD)	Level of detail expressions support aggregation in calculations at dimension levels different from the view level. Computed at the data source unlike table calculations, totals, or reference lines.
Logical	Conditional operations based on your data.
Number	Arithmetic and trigonometric operations.
RawSQL Pass Through	Pass SQL statements directly to the data source and then excute the statement within the data source. These functions do not work with every data source supported by Tableau.

(continues)

(continued)

FUNCTION CATEGORY	CATEGORY CAPABILITIES
String	Functions for manipulating strings.
String Pattern	Specialized string patter functions (REGEXP) that work with a subset of datasources including Text files, Google Big Query, PostgreSQL, Tableau Data Extracts, Microsoft Excel, Saleforce, and Oracle.
Table Calculation	Functions that are executed within Tableau using the structure of the visualization for the calculation.
Type Conversion	Change values from one data type to another.
User	Information on the identify, domain, and membership of the current Tableau user.

ADAPTED FROM THE TABLEAU MANUAL

In the first edition of this book, 111 functions were listed. We now have 167 functions and qualifiers listed. Some of these functions are not available for every data source. Some can be used only with specific tools such as the R Project for Statistical Computing, Hadoop Hive, or Google Big Query. Restrictions on the use of any function in the list are indicated in the function reference.

Tableau Software provides abbreviated help for each function within the formula editing window, the product manual, and the website. This appendix supplements those resources by providing:

- An alphabetically sorted listing of every function by category
- An alphabetically sorted and numbered list with brief function descriptions
- Detailed discussion of each function with syntax examples

ORGANIZATION AND KEY FOR APPENDIX E

Examples are organized alphabetically by function name. Color encoding is used to identify fields, functions, and parameter entries. Care has been taken to match the color hues as they appear in Tableau's formula editor.

Each entry contains one to three examples. There are a few (connection-specific functions) that don't include formula examples. In those cases we refer you to the vendor manual. Examples are listed as basic, intermediate, or advanced. Please note that some function types—RAW SQL for example—are inherently more complicated than basic functions such as those for aggregation or dates. The difficultly is gauged within the function category only.

WHAT IS RAW SQL AND WHY DO YOU NEED IT?

The RAW SQL functions are a special type of function in Tableau known as a pass-through function. These functions allow the user to send statements to the underlying database that are not evaluated by Tableau. This allows the user to call database functions that Tableau is unaware of. Tableau is aware of many built-in database functions and has mapped many of them to functions within Tableau, but depending on your data source, there are probably functions that Tableau doesn't yet support. In addition to built-in database functions, the RAW SQL function allows you to call any function that the underlying data source supports—including user-defined functions. This makes the RAW SQL functions particularly powerful.

The name of each RAW SQL function is based on the type of function being passed to the database (Aggregate or Scalar) and the type of the return value. For example, `RAWSQLAGG_INT()` passes an aggregate function and returns an integer value, and `RAWSQL_REAL()` passes a scalar function and retruns a real value. The valid return types are:

- `BOOL`: A Boolean value.
- `DATE`: A date value. Note that date types in databases usually omit time.
- `DATETIME`: A date-time value. Note that date-time types usually include date and time.
- `INT`: An integer value. Numbers without a decimal component.
- `REAL`: A numeric value. Numbers with a decimal component.
- `STR`: A string value. Text data.

Choosing the incorrect scalar or aggregate function will cause an error within Tableau. Keep in mind that RAWSQL functions will not work for published data sources on Tableau Server.

Another thing to keep in mind when you are working with RAW SQL functions is that your underlying database will not understand the dimension and measure names within Tableau. To pass a dimension or measure into the RAW SQL expression, you must use the substitution syntax provided by Tableau. This syntax is similar to substitution syntax seen in other languages. This is demonstrated in the example that follows:

```
RAWSQL_INT("1000 + %1", [Order ID])
```

In this example, the `RAWSQL_INT` function is being used to pass a simple expression to the database. The `%1` will be replaced by the value of `Order ID` in the expression. Notice that this example uses the scalar function and an integer return type.

Using the RAW SQL functions will let you expand the capabilities of Tableau in many ways. If you can write a function to perform the operation you require at the database level, then you can expose it to Tableau with these functions. Keep in mind that whenever you come across examples of RAW SQL usage that the examples are dependent upon the functions present in the database. For the examples in the remainder of this section, you will use a copy of the Superstore Orders dataset included with Tableau Desktop that has been loaded into a SQL Server 2012 instance. Some of the expressions used to demonstrate the pass-through queries may not work with your data sources.

R INTEGRATION VIA SCRIPT FUNCTIONS

Tableau added four new specialized functions that are for the statistical software tool R. These functions are all table calculations. If you are an R expert learning Tableau, it is important to understand how table calculations work before using them.

OTHER SPECIALIZED FUNCTIONS

New functions have also been added that are for specific data sources such as Hadoop Hive or Google Big Query. In some cases, the functions work with only a subset of data sources, including text files, Microsoft Excel, Tableau Data Extracts, PosgreSQL, Salesforce, and Oracle data sources. Those limitations are indicated in the alphabetical function list example details. After this text is released, Tableau may expand the list of data sources supported by a function. Refer to Tableau's online manual for the latest list of functions and data sources supported by each function.

ALPHABETICAL FUNCTION LIST—SUMMARY

Table E-2 shows every Tableau function available for the typical user. Depending on your data source, additional functions may be available from a particular database. Consult your database manual for additional commands not listed here. The remainder of Appendix E provides detailed explanations of each function. Code examples are provided for each function and are classified as basic, intermediate, and advanced.

TABLE E-2 Alphabetical Function List

#	FUNCTION NAME	TYPE FUNCTION
1	ABS	Number
2	ACOS	Number
3	AND	Logical
4	ASCII	String
5	ASIN	Number
6	ATAN	Number
7	ATAN2	Number
8	ATTR	Aggregate
9	AVG	Aggregate
10	CASE	Logical
11	CEILING	Number
12	CHAR	String
13	CONTAINS	String
14	COS	Number
15	COT	Number
16	COUNT	Aggregate
17	COUNTD	Aggregate
18	DATE	Type Conversion
19	DATEADD	Date
20	DATEDIFF	Date
21	DATENAME	Date
22	DATEPARSE	Date
23	DATEPART	Date
24	DATETIME	Type Conversion
25	DATETRUNC	Date
26	DAY	Date
27	DEGREES	Number
28	DIV	Number
29	DOMAIN	Google BigQuery
30	ELSE	Logical
31	ELSEIF	Logical
32	END	Logical
33	ENDSWITH	String

#	FUNCTION NAME	TYPE FUNCTION
34	EXCLUDE	Logical
35	EXP	Number
36	FIND	String
37	FINDNTH	String
38	FIRST	Table Calculation
39	FIXED	LOD Aggregate
40	FLOAT	Type Conversion
41	FLOOR	Number
42	FULLNAME	User
43	GET_JSON_OBJECT	Hadoop Hive
44	GROUP_CONCAT	Google BigQuery
45	HEXBINX	Number
46	HEXBINY	Number
47	HOST	Google BigQuery
48	IF	Logical
49	IFNULL	Logical
50	IIF	Logical
51	INCLUDE	LOD Aggregate
52	INDEX	Table Calculation
53	INT	Type Conversion
54	ISDATE	Date, logical, String
55	ISFULLNAME	User
56	ISMEMBEROF	User
57	ISNULL	Logical
58	ISUSERNAME	User
59	LAST	Table Calculation
60	LEFT	String
61	LEN	String
62	LN	Number
63	LOG	Number
64	LOG2	Google BigQuery

(continues)

(continued)

#	FUNCTION NAME	TYPE FUNCTION
65	LOOKUP	Table Calculation
66	LOWER	String
67	LTRIM	String
68	LTRIM_THIS	Google BigQuery
69	MAKEDATE	Type Conversion
70	MAKEDATETIME	Type Conversion
71	MAKETIME	Type Conversion
72	MAX	Aggregate, date, number, string
73	MEDIAN	Aggregate
74	MID	String
75	MIN	Aggregate, date, number, string
76	MONTH	Date
77	NOT	Logical
78	NOW	Date
79	OR	Logical
80	PARSE_URL	Hadoop Hive
81	PARSE_URL_JQUERY	Hadoop Hive
82	PERCENTILE	Aggregate
83	PI	Number
84	POWER	Number
85	PREVIOUS_VALUE	Table Calculation
86	RADIANS	Number
87	RANK	Table Calculation
88	RANK_DENSE	Table Calculation
89	RANK_MODIFIED	Table Calculation
90	RANK_PERCENTILE	Table Calculation
91	RANK_UNIQUE	Table Calculation
92	RAWSQL_BOOL	Pass Through
93	RAWSQL_DATE	Pass Through
94	RAWSQL_DATETIME	Pass Through
95	RAWSQL_INT	Pass Through
96	RAWSQL_REAL	Pass Through

#	FUNCTION NAME	TYPE FUNCTION
97	RAWSQL_STR	Pass Through
98	RAWSQLAGG_BOOL	Pass Through
99	RAWSQLAGG_DATE	Pass Through
100	RAWSQLAGG_ DATETIME	Pass Through
101	RAWSQLAGG_INT	Pass Through
102	RAWSQLAGG_REAL	Pass Through
103	RAWSQLAGG_STR	Pass Through
104	REGEXP_EXTRACT	String
105	REGEXP_EXTRACT_NTH	String
106	REGEXP_MATCH	String
107	REGEXP_REPLACE	String
108	REPLACE	String
109	RIGHT	String
110	ROUND	Number
111	RTRIM	String
112	RTRIM_THIS	Google BigQuery
113	RUNNING_AVG	Table Calculation
114	RUNNING_COUNT	Table Calculation
115	RUNNING_MAX	Table Calculation
116	RUNNING_MIN	Table Calculation
117	RUNNING_SUM	Table Calculation
118	SCRIPT_BOOL	Table Calculation
119	SCRIPT_INT	Table Calculation
120	SCRIPT_REAL	Table Calculation
121	SCRIPT_STRING	Table Calculation
122	SIGN	Number
123	SIN	Number
124	SIZE	Table Calculation
125	SPACE	String
126	SPLIT	String
127	SQRT	Number
128	SQUARE	Number
129	STARTSWITH	String

(continues)

(continued)

#	FUNCTION NAME	TYPE FUNCTION
130	STDEV	Aggregate
131	STDEVP	Aggregate
132	STR	Type Conversion
133	SUM	Aggregate
134	TAN	Number
135	THEN	Logical
136	TIMESTAMP_TO_USEC	Google BigQuery
137	TLD	Google BigQuery
138	TODAY	Date
139	TOTAL	Table Calculation
140	TRIM	String
141	UPPER	String
142	USEC_TO_TIMESTAMP	Google BigQuery
143	USERDOMAIN	User
144	USERNAME	User
145	VAR	Aggregate
146	VARP	Aggregate
147	WHEN	Logical

#	FUNCTION NAME	TYPE FUNCTION
148	WINDOW_AVG	Table Calculation
149	WINDOW_COUNT	Table Calculation
150	WINDOW_MAX	Table Calculation
151	WINDOW_MEDIAN	Table Calculation
152	WINDOW_MIN	Table Calculation
153	WINDOW_PERCENTILE	Table Calculation
154	WINDOW_STDEV	Table Calculation
155	WINDOW_STDEVP	Table Calculation
156	WINDOW_SUM	Table Calculation
157	WINDOW_VAR	Table Calculation
158	WINDOW_VARP	Table Calculation
159	XPATH_BOOLEAN	Hadoop Hive
160	XPATH_DOUBLE	Hadoop Hive
161	XPATH_FLOAT	Hadoop Hive
162	XPATH_INT	Hadoop Hive
163	XPATH_LONG	Hadoop Hive
164	XPATH_SHORT	Hadoop Hive
165	XPATH_STRING	Hadoop Hive
166	YEAR	Date
167	ZN	Logical, number

1. ABS

The ABS function returns the absolute value of the given number. The absolute value can also be seen as its distance from zero. This function is useful when you want to find out the difference between two values, regardless of whether that difference is positive or negative.

```
ABS(number)
```

`Number` = **any given number**

Basic

```
ABS([Budget Variance])
```

This function returns the sum of all absolute values for all the rows in the database for `Budget Variance`**.**

Intermediate

```
ABS(SUM([Budget Sales])-SUM([Sales]))/SUM([Budget Sales])
```

This example provides the absolute value of the variance between the sum of `Budget Sales` and the sum of `Sales`, and is expressed in a percentage from `Budget Sales`. The ABS function can be used to highlight exceptions when comparing to variance tolerance levels.

2. ACOS

The ACOS function returns the arc cosine of the given number. This is the inverse of the COS function.

```
ACOS(number)
```

`Number` **= any given number between –1 and 1**

Basic

```
ACOS(0.5)
```

This function returns 1.0471975511966 radians.

Intermediate

```
DEGREES(ACOS(0.5))
```

This function calculates the arccosine of number 0.5 and converts the result to degrees. The return is 60 degrees.

Advanced

This is an application of combined trigonometry functions in Tableau. This syntax calculates the distance between two geographical locations, whereby 3959 is the value of the average radius of the earth.

```
3959 * ACOS(SIN(RADIANS([Lat1])) *  SIN(RADIANS([Lat2]))
+ COS(RADIANS([Lat1])) * COS(RADIANS([Lat2])) *
COS(RADIANS([Long2])– RADIANS([Long1])).
```

Location1: `Lat`, `Long`. **Location2:** `Lat2`, `Long2`

```
3959 * ACOS(SIN(RADIANS(36.105143)) * SIN(RADIANS(36.113231))
+ COS(RADIANS(36.105143)) * COS(RADIANS(36.113231))
* COS(RADIANS(-95.975677) - RADIANS(-97.103813)))
```

The return of this function is 62.98. The distance between the Stillwater and Tulsa InterWorks office locations.

3. AND

The AND qualifier allows multiple expressions to be combined and evaluated within one calculated field. If the two expressions on either side of the AND statement are true, or represent a Boolean value of 1 instead of 0, then the entire statement is considered to be true. If one or both of the expressions on either side of the AND statement are false, or represent a Boolean value of 0 instead of 1, then the entire statement is considered to be false.

Basic

```
IF SUM([Sales]) > 10,000 AND AVG([Discount]) > .1
THEN "Review" ELSE "OK"
END
```

This formula considers both Sales over 10,000 and discounts that exceed 10% returning Review if both conditions are met and OK if both conditions are not met.

Intermediate

```
IF (DATEPART('month', [Order Date]) = 6
AND DATEPART('year',[Order Date]) = 2014)
AND [Profit] < 0 THEN "Review" ELSE "OK"
END
```

This formula evaluates the data source for month and year to return June of 2014, then evaluates profit that is less than 0 and returns Review. If profit is greater than or equal to 0 OK is returned.

4. ASCII

ASCII is a character-encoding scheme that allows English characters, numbers, and symbols to be encoded into a corresponding number in the ASCII character set. The ASCII function returns the ASCII code for the first character in a given string. A Standard ASCII character set comprises 128 characters. These 128 ASCII characters can then be divided further into 4 equal groupings of 32 characters. The ASCII groups contain the following:

- **0–31:** Non-printing characters/control characters
- **32–63:** Numeric values, punctuation characters, and special characters
- **64–95:** Uppercase alphabet characters and special symbols
- **96–127:** Lowercase alphabet characters and special symbols

```
ASCII(String)
```

Return the ASCII code for the first character of string

Basic

```
ASCII([Customer])
```

This function example returns the ASCII character for the first character within the customer name string. The value returned will range from 65 for names beginning with a capital A to 90 for names beginning with a capital Z.

Intermediate

```
IIF (ASCII([Customer Name])<32, 'Non Printable Characters','Printable Characters')
```

This formula example provides a basic data validation mechanism for the ASCII printable characters in the customer name field. If any non printable characters were included at the beginning of a customer name this formula could be used to display those records.

Advanced

```
ASCII(MID([Customer Name],FIND([Customer Name]," ")))
```

This function example uses three other Tableau functions within its logic. The `Customer` field contains the customer name with a space between the Forename and Surname. We can use this formula's logic to determine if any custom names include a character other than a blank space between the first and last names of the customer name.

5. ASIN

The `ASIN` function returns the arcsine, in radians, of the given number. This is the inverse of the `SIN` function.

```
ASIN(number)
```

`Number` = any given number between –1 and 1

Basic

```
ASIN(1)
```

This function returns the arcsine of value 1. The result is 1.5707963267949 radians.

Intermediate

```
DEGREES(ASIN(1))
```

This function calculates the arcsine of 1 and converts the result to degrees. The return is 90 degrees.

6. ATAN

The `ATAN` function returns the arctangent of the given number. This is the inverse of the `TAN` function. The result is given in radians in the range between –π/2 and π/2.

```
ATAN(number)
```

`Number` = any given number, where the number is given in radians

Basic

```
ATAN(1)
```

This function returns 0.785398163397448 radians, which is equal to π/4.

Intermediate

```
DEGREES(ATAN(1))
```

This function calculates the arctangent of number 1 and converts the result to degrees. The return is 45 degrees.

7. ATAN2

The `ATAN2` function returns the arctangent of two given numbers (x and y). The result is in radians in the range between –π and π, excluding –π.

```
ATAN2(y number,x number)
```

The y number = any given number, and the x number = any given number.

Basic

```
ATAN2(1,1)
```

This function calculates the arctangent where y and x values are 1. The return is 0.785398163397448 radians, which is equal to π/4. If both the x and y values are 0, then this will return a NULL value.

Intermediate

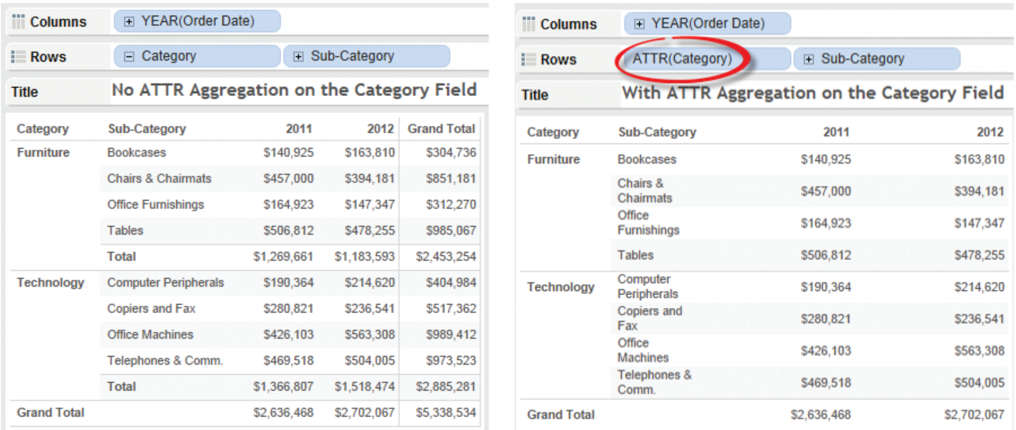
```
DEGREES(ATAN2(-1,-1))
```

This function calculates the arctangent of point –1, –1 and converts the result to degrees. The return is –135 degrees.

8. ATTR

The ATTR function evaluates all of the members contained within the specified field and returns a single value (if all of the values are identical) or the symbol * if more than one value exists in the set. The * symbol is meant to denote a special kind of NULL—one containing many values, instead of the more typical use of NULL, no values.

When the ATTR function is applied to a dimension that is expressed in a hierarchal view of the data, it will treat that field as a label and will cause aggregate values to be calculated based on the remaining dimensions. Figure E-1 shows this result.



Category	Sub-Category	2011	2012	Grand Total
Furniture	Bookcases	$140,925	$163,810	$304,736
	Chairs & Chairmats	$457,000	$394,181	$851,181
	Office Furnishings	$164,923	$147,347	$312,270
	Tables	$506,812	$478,255	$985,067
	Total	$1,269,661	$1,183,593	$2,453,254
Technology	Computer Peripherals	$190,364	$214,620	$404,984
	Copiers and Fax	$280,821	$236,541	$517,362
	Office Machines	$426,103	$563,308	$989,412
	Telephones & Comm.	$469,518	$504,005	$973,523
	Total	$1,366,807	$1,518,474	$2,885,281
Grand Total		$2,636,468	$2,702,067	$5,338,534

Category	Sub-Category	2011	2012
Furniture	Bookcases	$140,925	$163,810
	Chairs & Chairmats	$457,000	$394,181
	Office Furnishings	$164,923	$147,347
	Tables	$506,812	$478,255
Technology	Computer Peripherals	$190,364	$214,620
	Copiers and Fax	$280,821	$236,541
	Office Machines	$426,103	$563,308
	Telephones & Comm.	$469,518	$504,005
Grand Total		$2,636,468	$2,702,067

FIGURE E-1 *ATTR function example*

A good explanation of ATTR can be found in a Tableau Forum entry by Joe Mako[1] in which Joe expresses the logic used by the ATTR function using this formula:

```
IF MIN([field])=(MAX([field]) THEN MIN([field]) ELSE "*" END
```

To restate Joe's logic, if the minimum value and the maximum value of the set of numbers returned from the database are the same, then use the minimum value, and if they are not the same, return the * symbol.

Basic

```
ATTR([field])
```

The result of this formula would be a single member or the * symbol. The * symbol donotes that the result has more than one member.

Basic

```
ATTR([Sub-Category])
```

The ATTR aggregate function returns the value of the expression if it has a single value for all the rows. If the result of the expression is more than one value it will return an asterisk. Null values are ignored. The basic ATTR formula applied to the dimension will return an asterisk for any Sub-Category that has more than one Sub-Category member. This example returns the asterisk for Sub-Category because the Superstore dataset includes more than one subcategory member.

9. AVG

The `AVG` function returns the mean of the expression. It is calculated as the sum of all of the numbers for the expression divided by the count of the number of records in that expression. For example, if you have a set containing 24, 30, 15, 5, 18 the average of those five numbers will be: (24 +30+15+5+16)/5 =15.

Basic

```
AVG([Discount])
```

The result is the mean discount for whatever level of details is exposed in the view.

Intermediate

```
AVG(DATEDIFF('day',[Order Date],[Ship Date]))
```

This will calculate the number of days between the order date and the ship date, and then provide the mean for the level of detail displayed in the view.

Advanced

```
AVG(IF(DATEDIFF('day',[Order Date],[Ship Date])<=[Time to Ship Goal])
THEN 1 ELSE 0 END)
```

This calculation compares the number of days it takes to ship an item to a parameter `(`*`variable`*`) [Time to Ship Goal]`**. If the time is less than or equal to the parameter, a 1 is returned. If it's greater than the parameter, a 0 is returned. Then, the mean of those values is calculated the proportion of 1s that shipped within the time specified by the selected parameter value. This is a useful trick: The mean of a set of 0s and 1s is the proportion of 1s so multiply by 100 if you prefer percentage.**

10. CASE

The `CASE` function is provided with an expression/data field, which can be defined as the CASE statement source field. The data values located within this source field are compared against a sequence of values specified in the `WHEN` clauses. If any of the values within the expression match a `WHEN` specification, the corresponding THEN value is returned. If no match is found, then the default return expression, specified in the ELSE clause, is used. If there is no ELSE clause, **NULL** is returned when no match is found.. The CASE function can be duplicated using IF or IIF functions. Typically CASE is easier to use and more concise.

```
CASE expression WHEN value1 THEN return1 WHEN value2 THEN return2 ......
ELSE default return
END
```

Basic

```
CASE [Month]
WHEN 1 THEN "January"
WHEN 2 THEN "February"
WHEN 3 THEN "March"
WHEN 4 THEN "April"
ELSE "Not required"
END
```

This example will look at the month field—containing integers—and return the corresponding month strings defined.

Intermediate

```
CASE (LEFT([Customer Name],1))
WHEN 'A' THEN 'Customer name starts with A'
WHEN 'B' THEN 'Customer name starts with B'
WHEN 'C' THEN 'Customer name starts with C'
ELSE 'Customer names not starting with A, B or C'
END
```

The intermediate formula evaluates the first character in the Customer Name field then groups customer names by the first character of the name.

11. CEILING

The CEILING function rounds numbers up to the next highest integer. This function will be applied according to the level of aggregation specified in the equation, meaning SUM must be present to produce a rounded summation.

Basic

```
CEILING(41.09)
```

This formula will return the value of 42, which is the next highest integer.

Intermediate

```
CEILING(SUM([Sales]))
```

This equation calculates the sum of the referenced sales values and then rounds the result up to the next highest integer.

12. CHAR

`CHAR` is a function that changes an ASCII code into its relevant String character. `ASCII` and `CHAR` functions perform the reverse of each other and both are fundamentally linked.

```
CHAR(Number)
```

Basic

```
CHAR(65)
```

This function example returns the corresponding String value for the integer value 65. 65 in the ASCII character table is A.

Intermediate

```
CHAR(IIF (ASCII([Customer])> 96
and ASCII([Customer])<= 122,
ASCII([Customer])-32,
ASCII([Customer])))
```

This function example provides an effective way of ensuring the `CHAR` output is `CASED` correctly. If it is a lowercase letter (96 to 122), make it uppercase by subtracting 32.

Advanced

```
CHAR(ASCII([Customer]))+ "." +
CHAR(ASCII(LTRIM(MID([Customer],FIND([Customer]," ")))))+ "."
```

The function example shown displays the customer initials that have been obtained from the `Customer` field, separated by a full stop. Example output would be `A.D.` for `Customer = Aaron Day`. The additional String functions are detailed within this section of the book.

13. CONTAINS

The `CONTAINS` function gives the user the ability to search for any sequence of characters (`SUBSTRING`) that may be present within a searchable string. The `CONTAINS` function returns a Boolean value of True or False.

```
CONTAINS(String, Substring)
```

Basic

```
CONTAINS([City],"New")
```

This function returns a True value for all the Customer Cities held within the data that has `New` within its name.

14. COS

The `COS` function returns the cosine of a given number specified in radians.

```
COS(number)
```

`Number` **= any given number, where the number is in radians**

Basic

```
COS(PI()/8)
```

This function calculates the cosine of π/8 radians. This function returns 0.923879532511287.

Intermediate

```
COS(RADIANS(60))
```

In this function, the number shown is specified in degrees. First the 60 degrees are converted to radians before the cosine is calculated. The result is 0.5.

Advanced

This is an application of combined trigonometry functions in Tableau. This syntax calculates the distance between two geographical locations, whereby 3959 is the value of the average radius of the earth.

```
3959 * ACOS(SIN(RADIANS([Lat1])) *  SIN(RADIANS([Lat2]))
+ COS(RADIANS([Lat1])) * COS(RADIANS([Lat2])) *
COS(RADIANS([Long2])– RADIANS([Long1])).
```

Location1: `Lat`, `Long`. **Location2:** `Lat2`, `Long2`

```
3959 * ACOS(SIN(RADIANS(36.105143)) * SIN(RADIANS(36.113231))
+ COS(RADIANS(36.105143)) * COS(RADIANS(36.113231))
* COS(RADIANS(-95.975677) - RADIANS(-97.103813)))
```

The return of this function is 62.98. The distance between the Stillwater and Tulsa InterWorks office locations.

15. COT

Returns the cotangent of a given number specified in radians. The number is expressed in radians.

`COT(number)`

`Number` = **any given number, where the number is given in radians**

Basic

`COT(PI( )/4)`

This function calculates the cotangent of π/4 radians. This function returns 1.

Intermediate

`COT(RADIANS(45))`

In this function, the number is specified in degrees, in this case 45. First the degrees are converted to radians before the cotangent is calculated. The result is also 1.

16. COUNT

This function returns the count of the items in a group. NULL values are not counted.

Basic

`COUNT([Ship Date])`

The formula returns the count of records that have a ship date value. Records missing a ship date entry will be NULL and the record will not be counted.

Intermediate

`COUNT(IIF([Discount]=0,1,NULL))`

This example returns the number of records that have zero discount.

Advanced Example

`COUNT(IIF([Discount]=0,1,NULL))/COUNT([Number of Records])`

This formula will look at the discount field; if the value is 0, it will not be counted. For rows that return a non-null amount, the resulting count is then divided by the total number of records to derive the percentage of items that have a discount.

17. COUNTD

Count distinct returns the number of distinct items in a group. NULL values are not counted. Each unique value is counted only once.

Basic

```
COUNTD([Customer Name])
```

This formula returns the count of unique customer names. Any record where `Customer Name` is NULL will not be counted. It will count each unique name only once regardless of the number of records including the name.

Intermediate

```
COUNTD([City]+[State])
```

This formula combines the city and state fields to create a new field in order to count the number of unique city-state combinations. Note that the + sign concatenates the city and state fields together.

Advanced

```
COUNTD(IF([Country]=[Country Parameter])
THEN [Customer Name] ELSE NULL END)
```

The formula counts the unique customer instance for a selected country. The parameter (variable) permits the user to select the country to count customer names. Any other country is viewed as `NULL` so it will not be counted.

18. DATE

The `DATE` function converts a given input into a date. This is similar to the `DATETIME` function, but doesn't include time. This is especially useful when you have string dates in your data source or are building your own dates using other data sources.

Basic

```
DATE("March 15, 2013")
```

This example converts the string to a date value of March 15, 2013.

Intermediate

```
DATE([DateString])
```

This formula returns a date if `[DateString]` is a valid date type; otherwise it will return NULL.

Advanced

```
DATE(STR([Year]) + '/'+ STR([Month]) + '/' + STR([Day]))
```

This formula returns a date constructed from various components in the data source. This is especially useful when the data source has date elements in it, but no true date dimension. This allows you to create that date dimension so you can use Tableau's auto-generated date hierarchy in views.

19. DATEADD

The `DATEADD` function adds a specified time period to a given date. This function is useful when you want to calculate new dates based on another date in your dataset, to create reference lines in time series analysis, or to create dimensions to use for filtering.

```
DATEADD(date_part, increment, date, start_of_week)
```

The `date_part` specifies the type of time period that is being added. It is always specified in single quotes and lowercase (for example, `'day'`). Increment specifies the exact amount of time to add. Table A-3 below displays the `date_part` values that can be used with date functions.

TABLE E-3 Valid Date Function Date Parts

DATE_PART	VALUES
'year'	Four-digit year
'quarter'	1–4
'month'	1–12 or "January," "February," and so on
'dayofyear'	Day of the year: Jan 1 is 1, Feb 1 is 32, and so on
'day'	1–31
'weekday'	1–7 or "Sunday," "Monday," and so on
'week'	1–52
'hour"	0–23
'minute'	0–59
'second'	0–60

SOURCE: TABLEAU DESKTOP MANUAL

The date variable in the formula is the base date used for the addition. This value can be a constant value (#2015-06-23#), field, parameter, or another function that returns a date.

Basic

```
DATEADD('day',3,[Order Date])
```

This formula returns a date three days after the given Order `Date`. In this case, if Order `Date` is equal to December 9, 2015, the function returns December 12, 2015.

Intermediate

```
DATEADD('day', -30, TODAY())
```

This formula returns a date 30 days before today's date. If today is March 18, 2013, then this function returns February 16, 2013. This can be very useful in filtering for specific periods (such as 30 rolling days) or highlighting a period on a timeline where data may be uncertain.

Advanced

```
DATEADD('month', -12, WINDOW_MAX(MAX([Date])))
```

The advanced example formula returns a date that is 12 months before the last date in the defined window (see the `WINDOWMAX` calculation for more details). It's common in many businesses for data to be days, weeks, or even months old. This method gives you a date 12 months from the end of the dataset, no matter how big (or small) the gap is between the end of the dataset and the current date.

20. DATEDIFF

The `DATEDIFF` function calculates the time between two given dates. This is useful for creating additional metrics or dimensions for your analysis. It returns an integer value of `date2-date1` expressed in units of `date_part`.

```
DATEDIFF(date_part, date1, date2, start_of_week)
```

`DatePart` specifies the type of time period that is being returned. It is always specified in single quotes and lowercase (for example: `'day'`). See the `DATEADD` entry for valid datepart values. `Date1` and `Date2` are the actual dates used for subtraction. The values can be constants, fields, parameters, other functions that return dates, or combinations of any of these.

Basic

```
DATEDIFF('day', #June 3, 2012#, #June 5, 2012#)
```

This formula uses the `DATADIFF` function to calculate the difference in days between the two date literal values—which in this example results in the answer of 2 days. Keep in mind that you can use date fields in your dataset to specify flexible date values as well.

Intermediate

```
DATEDIFF('day',[Ship Date], TODAY())
```

This formula returns the number of days between the two dates. This is a common setup for aging reports. The example derives the current age of the invoice in days. If today is November 2, 2015 and you have an invoice with an `[Ship Date]` of October 30, 2015, the function returns 3 (days). This can be very useful, especially when combined with Bins.

Advanced

```
CASE [Parameter].[Date Unit]
WHEN 'Day' THEN DATEDIFF('day',[OrderDate],[ShipDate])
WHEN 'Week' THEN DATEDIFF('week',[OrderDate],[ShipDate])
END
```

This formula returns an integer, either the number of days or weeks between the two dates. In this example, you're using a parameter, `[Date Unit]`. This parameter is a string type that allows the user to input the time period they want returned in the result (days or weeks). This allows the user to determine the best way to express the answer.

21. DATENAME

The `DATENAME` function returns part of date as text. This function is useful for creating custom labels that go beyond what Tableau formatting can provide.

```
DATENAME(DatePart, Date)
```

The `DatePart` operator defines exactly how you want the date to be expressed, such as week, month, or year. See `DATEADD` for a list of `date_part` values. `Date` is the actual date you want to extract from.

Basic

```
DATENAME('month', #2012-06-03#)
```

This formula returns `June` from the literal date of June 3, 2012.

Intermediate

```
DATENAME('month',[StartDate]) + ' to ' + DATENAME('month',[EndDate])
```

This formula returns a string describing the start and end dates for each row in the data source. If `[StartDate]` is equal to January 1, 2013 and `[EndDate]` is equal to March 1, 2013 then the function will return January to March.

Advanced

```
DATENAME('month', TOTAL(MIN([Date]))) + ' ' +
DATENAME('year', TOTAL(MIN([Date]))) + ' to ' +
DATENAME('month', TOTAL(MAX([Date]))) + ' ' +
DATENAME('year', TOTAL(MAX([Date])))
```

This formula returns a string describing the first and last date that exists in the view. The key to this is using the `TOTAL` function with `MIN` and `MAX` (see those function entries for more details). This is useful when you'd like to have a header, annotation, or other label that describes the period of analysis. If `TOTAL(MIN([Date]))` is equal to January 1, 2013 and `TOTAL(MAX([Date]))` is equal to March 1, 2013, then the function will return January 2013 to March 2013.

22. DATEPARSE

The `DATEPARSE` function converts a string field into a date/datetime field. This is especially useful when you have string dates in your data source or are building your own dates using other data sources. Converting string dates to a true date/datetime field allows you to use Tableau's autogenerated date hierarchy in views. The `DATEPARSE` function is available for non-legacy Microsoft Excel and text file connections, MySQL, Oracle, PostgreSQL, and Tableau data extracts data sources. Your computer system will control locale-specific formats.

```
DATEPARSE(format, [String])
```

- `Format` provides a map of how the string field in the data source is laid out. For example, if your data looks like `"20150807"` then set up the function with `"yyyyMMdd"` in the format section. Double quotes (`""`) should wrap the combination of symbols in your format. Note that capital `M` is used to designate "months" and lowercase `m` designates "minutes." The symbols to be used for the format are defined by International Components for Unicode (ICU) formatting language. For a complete list of the syntax for these date symbols refer to the ICU website at: `http://userguide.icu-project.org/formatparse/datetime`.
- `String` references the existing string field in the data source you are converting to a date/datetime field.

Basic

```
DATEPARSE("MMyyyy", "082015")
```

This example converts the string to a date value of August 1, 2015. The date format displayed can be controlled by selecting Default Properties ➢ Date Format for the newly created field.

Intermediate

```
DATEPARSE("MMddyyyy hh:mma", "08072015 10:07AM")
```

This formula returns a date value of August 7, 2015 10:07 AM. The date format displayed can be controlled by selecting Default Properties ➢ Date Format for the newly created field.

Advanced

```
DATEPARSE("MMddyy", RIGHT("Customer 1 – 080715", 6))
```

This formula identifies which part of the string file to convert to a date field. The result is a new date field August 7, 2015.

23. DATEPART

The `DATEPART` function returns part of a date as an integer. This can be useful for certain calculations when you need to parse portions of a date.

```
DATEPART(DatePart, date)
```

`DatePart` defines the portion of the date you require, such as week, month, or year. `Date` is the actual date you want to extract from the original date.

Basic

```
DATEPART('month', #June 3, 2012#)
```

This formula returns 6, which represents the sixth month, June.

Intermediate

```
DATEPART('dayofyear',[Date])
```

This example returns a numeric representation of the day of the year (in other words, how many days into the year it is). For example, if the `[Date]` is June 3, 2012, it will return 155.

Advanced

```
IF DATEPART('hour',[Datetime]) < 12 THEN 'Morning'
ELSEIF DATEPART('hour',[Datetime]) < 16 THEN 'Afternoon'
ELSEIF DATEPART('hour',[Datetime]) < 21 THEN 'Evening'
ELSE 'Night'
END
```

This formula is using the hour to categorize the time into different grouping. In this case, if `[Datetime]` is between midnight and noon, it will categorize it as `Morning`. If `[Datetime]` is between noon and 4 p.m., it will say it's the `Afternoon`. If `[Datetime]` is between 4 p.m. and 9 p.m. it will be the `Evening`, while anything else will be `Night`. This function can be useful for categorizing time periods such as shifts in manufacturing, or day parts in advertising. Note that `DATEPART` returns a 24-hour time period (for example, 1 p.m. will return 13).

24. DATETIME

The `DATETIME` function converts a given input into a date and time. This is similar to the `DATE` function but includes the time. This is especially useful when you have string dates in your data source, separate fields for date and time, or are building dates from other sources.

Basic

```
DATE("March 15, 2013 5:30 PM")
```

This example converts the string to a date time value of March 15, 2013 at 5:30 PM.

Intermediate

```
DATETIME(STR([Date]) + ' ' + [Time]),
```

This example returns a date time. The `[Date]` field is simply a date with no time. The formula first converts `[Date]` to a string and then adds the `[Time]` component from another dimension. This is accomplished through concatenation of the two separate fields. Once concatenated, the result is converted into date time.

Advanced

```
DATE(STR([Year]) + '/' + STR([Month]) + '/' + STR([Day]) + ' ' +
STR([Time]))
```

This example returns a `Date` time constructed with various components from the data source. This is especially useful when the data source has date elements in it but no true date time dimension. This formula converts three different fields into a single date dimension to take advantage of Tableau's built-in date hierarchy.

25. DATETRUNC

This function returns a date—truncated to the nearest specified date part. Think of this as an aggregating method for converting time into the desired level of detail while maintaining the date format. The date returned will always be the first day in the time period.

```
DATETRUNC(DatePart, date)
```

The `DatePart` defines the date aggregation displayed (week, month, or year, and so on). `Date` is the actual date used to extract the desired `DatePart`.

Basic

```
DATETRUNC('Month', #March 14, 2013#)
```

This formula returns the date March 1, 2013 presuming that you define the date granularity in the view as month/day/year. Because you selected `Month` as your datepart, it returns the first of the month.

Intermediate

```
DATETRUNC('week', MIN([Date]))
```

This formula returns a date that is the start of the week of `MIN ([Date])`. In this case, the date represents the start of the first date for any mark in the view. This function could be useful when looking at dimensions for which the start date is important, such as customer sales or internal projects.

Advanced

```
CASE [Parameter].[Date Unit]
WHEN 'Day' THEN DATETRUNC('day', [Date])
WHEN 'Week' THEN DATETRUNC('week', [Date])
WHEN 'Month' THEN DATETRUNC('month', [Date])
END
```

This formula returns a flexible expression of the date through the use of a parameter. The parameter, `[Date Unit]`, is a string type that allows the user to input the time period he or she wants to aggregate the result. This is ideal when the same report needs to be viewed at different levels of detail—for instance, a trend report that users would like to see the monthly trend—but also drill down into other specific time periods, such as week or day.

26. DAY

This function returns an integer representing the day of the month for the given date. This is a shortened form of `DATEPART ('day', [Date])`.

```
DAY(Date)
```

Date is the date you want to extract from.

Basic

```
DAY(#March 14, 2013#)
```

This formula returns 14

Intermediate

```
DAY(DATEADD('day',[Date], 5 ))
```

This formula returns the day [Date] + five days. If [Date] is March 14, 2013 then this function will return 19.

Advanced

```
CASE [Parameter].[Date Unit]
WHEN 'Day' THEN DAY([Date])
WHEN 'Month' THEN MONTH([Date])
WHEN 'Year' THEN YEAR([Date])
END
```

This example returns an integer. The parameter for the [Date Unit] controls the specific date level that will be returned in the answer set. If the user selects Day, the function will return the day of [Date].

27. DEGREES

The DEGREES function converts a given number in radians to degrees.

```
DEGREES(number)
```

Number = any given number

Basic

```
DEGREES(PI()*2)
```

This function converts the radian value of 2π. The result is 360 degrees. The conversion from radians to degrees can also be calculated with the PI () function: (PI ()*2) * (180/PI ()), which gives the same return.

28. DIV

The DIV function accepts a numerator and denominator as input and then outputs the integer portion of the result. Any remainder left over from the division is not included in this output.

Basic

```
DIV(16,5)
```

This function will result in an integer output of 3. Dividing 16 by 5 will not result in an integer value, but if the remainder is removed from the result, you are left with an integer value of 3.

Advanced

```
DIV(INT(SUM([Sales])), COUNTD([Customer ID]))
```

This formula converts the sum of sales to an integer value and divides that number by the distinct count of customer IDs within the dataset, resulting in an approximate integer value of dollar sales per customer. The conversion of sales to an integer value is necessary because the `DIV` function accepts only integers as inputs.

29. DOMAIN

Returns the domain of a given URL. The URL must include protocol for this function to work properly. This function only works with Google Big Query data sources.

```
DOMAIN([URL])
```

Basic

```
DOMAIN('http://www.twitter.com/DGM885')
```

The formula returns the domain for a given URL. In the example, a literal string is used to define the URL. A field containing the URL will work as well. The answer from the basic example is `twitter.com`.

30. ELSE

The `ELSE` qualifier can be used in conjunction with an `IF/THEN` statement to provide a default value or expression in the case that a specified `IF` statement evaluates to false.

Basic

```
IF [Sales] >= [Quota] THEN "Goal Met"
ELSE "Needs Improvement"
END
```

This `IF/THEN` statement compares a given sales value to a given quota value. If sales are greater than or equal to the quota, then the goal has been met. If sales are not greater than or equal to the quota, the `ELSE` statement flags the corresponding sales values as needing improvement.

Intermediate

```
IF [Order Priority] = "Critical" or [Order Priority] = "High"
THEN [Shipping Cost] * 2
ELSE [Shipping Cost] * 1.5
END
```

This formula assigns a new value for shipping cost depending on the importance of the order priority. Orders with critical or high priority have their shipping cost multiplied by 2, whereas orders with any other level of priority default to a value of 1.5 times the shipping cost. In this fashion, `ELSE` statements can be used with mathematical operations.

31. ELSEIF

The `ELSEIF` qualifier makes it possible to use more than one `IF` statement inside one calculated field. Multiple `IF` statements cannot be used inside a single calculated field, but any number of `ELSEIF` statements may be stacked underneath an `IF` statement to provide alternative mathematical or logical comparisons. Given a situation where the expression in the first `IF` statement is not true, each consecutive `ELSEIF` statement will be evaluated until one does evaluate to true or until an `ELSE` statement is encountered. Also see the `CASE` function.

Basic

```
IF [Profit Ratio] >= .25 then "Excellent"
ELSEIF [Profit Ratio] >= .1 then "Decent"
ELSEIF [Profit Ratio] >= 0 then "Needs Improvement"
ELSE "Urgent"
END
```

This equation uses a combination of `IF`, `ELSEIF`, and `ELSE` statements to categorize a broad range of profit ratios into four easily managed groups. First, the algorithm checks to see if values have a profit ratio that is greater than or equal to 25 percent. If that condition is satisfied, then the value is considered to be excellent. If the profit ratio value being examined does not meet the first criterion, then the algorithm jumps down to the first `ELSEIF` statement. This logic is repeated for each value passed into the equation, and any profit ratios below zero are flagged as urgent by default.

32. END

The `END` statement is not a function that stands alone, but rather an essential piece of any `IF/THEN` evaluation or `CASE` statement. It behaves much like a period does in a regular sentence, signaling that a phrase or expression is complete. For the `END` statement, these phrases and expressions are logical and mathematical comparisons.

Basic

```
IF YEAR([Order Date]) = 2013 THEN [Sales] END
```

This equation checks if the year of an order date is 2013. If the condition is true, then the sales value is recorded. The `END` statement indicates that the comparison is complete. Without `END`, the calculated field would generate an error.

33. ENDSWITH

The `ENDSWITH` function does the same task as the `STARTSWITH` function; however, its focus is on the end of a string.

```
ENDSWITH(String, Substring)
```

BASIC

```
ENDSWITH([City],"Orleans")
```

This function returns a true value only if the `END` of the search field matches the substring. The `STARTSWITH` and `ENDSWITH` functions ignore any leading or trailing spaces that may be present in the string field.

34. EXCLUDE

One of the three key words used in Level of Detail calculations. The `EXCLUDE` keyword will ignore the dimensions listed from the dimensionality of the worksheet. This keyword is most useful when trying to visualize a measure at a coarser level of detail than the worksheet. You can set the level of detail to exclude one more more dimensions.

```
{ EXCLUDE [Dimension 1], [Dimension 2],... : AGG([Measure]))}
```

`AGG` represents any aggregation.

BASIC

```
{ EXCLUDE [State]: SUM([Sales])}
```

This calculation will give the total sales at any level of detail, not including `State`. If `State` is present in the view, it will be ignored.

35. EXP

The `EXP` function is the inverse of the `LN` function. `EXP` returns "e" raised to the power of the given number, where "e" has the value 2.71828182845905. In Tableau, the return is accurate to 14 decimal places.

```
EXP(number)
```

`Number` = **any given number**

Basic

```
EXP(2)
```

The result of this function is 7.38905609893065, where *e* is multiplied by itself. Tableau will first calculate the function before the return is truncated to 14 decimal places.

36. FIND

The `FIND` function returns the index position of a `substring` contained within a selected `string`, or 0 if the `substring` isn't found. If the optional argument `start` is added, the function ignores any instances of substring that appear before the index position `start`. The first character in the string is position 1.

```
FIND(String,Substring,[start])
```

Basic

```
FIND(([City],"City")
```

The FIND function locates the starting position of defined string within a specified field or literal. In the preceeding example the City field will be searched for the string "City" and the starting positions for each city that contains the string "City" will be located and the starting postion of the string computed. The Superstore data set includes many city names that include the string "City" including New York City (position 10), Oklahama City (position 10), Johnson City (position 10, Garden City (postion 8), Texas City (postion 7) and so on.

Intermediate

```
FIND(([Customer] ,"'",4)
```

This function will search for any customers that have an apostrophy contained within their name starting at the 4th position or after. Any apostrophies that appear before position 4 will not be returned in the result. For example, a customer name Ti's Company would not be returned in the result. A customer named Billy's Company would return the result 6.

Advanced

```
IIF(CONTAINS([Customer],"'"),
(RIGHT([Customer],LEN([Customer]),FIND([Customer],"'",4)+2)),NULL)
```

The function example shows how combined string functions can be useful within Tableau. The function first searches for any field with an apostrophe. If this returns a true value, the combination of `RIGHT`, `LEFT`, and `FIND` functions finds the apostrophe, determines its position, moves the indexing position back two places, and extracts all string characters from this new index position. In a nutshell, you are looking to extract surnames that have an apostrophe within them. If the `CONTAINS` function returns a `False` value then the field is given a `NULL` value.

37. FINDNTH

The `FINDNTH` function searches a given string and counts the number of times a specified character or set of characters occurs within the string. One of the inputs to this function is the "occurrence," otherwise known as the "nth" time the specified character or characters present themselves in the string. The `FINDNTH` function outputs the index value of the "nth" occurrence within the string.

Basic

```
FINDNTH("Jon Doe", "o", 2)
```

The output of the preceding function will be 6. The `FINDNTH` function searches the given "Jon Doe" string for the second occurrence of the letter "o," which has an index value of 6 in the string. In Tableau, index values start at 1 at the beginning of a string and each individual character has its own index value.

Intermediate

```
FINDNTH("Is it hot or is it cold?", "it", 2)
```

The output of this function will be 17. The index of the first character of the "nth" occurrence is the value that will be output by this function if searching for a combination of characters instead of a single character. That is why the result in this example is 17 and not 18, which is the index where the substring "it" is complete.

38. FIRST()

The `FIRST()` table calculation function returns the number of rows back to the first row of the view/partition. This function does not require any arguments.

Basic

```
FIRST()
```

This function returns the amount of rows back to the first row.

Intermediate

```
WINDOW_AVG(SUM([Sales]),FIRST(),LAST())
```

This function returns the average of the sum of sales from the first row to the last row of the window (or frame). If the values are incorrect or inconsistent, ensure that you are properly using the Compute using option of the table calculation.

Advanced

```
IF FIRST()=0 THEN WINDOW_AVG(SUM([Sales]),0,IIF(FIRST()=0,LAST(),0)) END
```

This function assumes there are a large number of marks in the view or that the user is dealing with a big dataset. Through the use of if/then logic, table scans, which cause noticeable performance degradation, can be bypassed. In the end, the calculation takes the average of the sum of sales for the specified window (or frame).

39. FIXED

`FIXED` is one of the three key words used in Level of Detail calculations. The `FIXED` keyword will fix a calculation at a particular level of detail, regardless of the level of disaggregation in the worksheet the calculation is used in. You can set the level of dimensionality in the calculation to use zero, one, or multiple dimensions.

```
{ FIXED [Dimension 1], [Dimension 2],... : AGG([Measure]))}
```

`AGG` represents any aggregation.

Basic

```
{ FIXED : MIN([Order Date])}
```

This returns the date of the first transaction.

Intermediate

```
{ FIXED [Customer Name] : MIN([Order Date])}
```

This will give the date of the first order placed for each customer.

40. FLOAT

The `FLOAT` function returns a floating point number, or in other words, a decimal number.

Basic

```
FLOAT(5)
```

This example returns a floating decimal number of 5.000.

Intermediate

```
INT([Teachers]) + FLOAT([Students])
```

This formula returns a decimal number that represents the total number of `Teachers` and `Students`. Despite one of the dimensions being an integer, Tableau will return the type with the most precision.

Advanced

```
FLOAT(MID(2,[DollarString]))
```

In this formula, the field `[DollarString]` contains a string that represents a dollar amount, $ 5.00. The `MID` function is used to dispose of the dollar sign and get only the number; then the `FLOAT` function is used to convert the string to a floating decimal point number that will be conducive for further calculations.

41. FLOOR

The `FLOOR` function returns the nearest integer less than or equal to the specified number. This function is useful when you want to round a decimal down to the nearest integer.

Basic

```
FLOOR(123.55)
```

This formula returns the integer 123.

Advanced

```
FLOOR(FLOAT(REGEXP_EXTRACT('abc 123.55','(\d+\.\d+)')))
```

In this example, you want to evaluate the string `abc 123.55` and extract only the numeric portion of this literal rounded down to the nearest integer. Using the `REGEXP_EXTRACT` function, you extract the numeric portion of the string and cast the REGEXP expression result using the `FLOAT` function. Lastly, you use the `FLOOR` function to round down to the nearest integer.

42. FULLNAME()

The following assumptions are used for the examples:

User 1

- Full name: Malcolm Reynolds
- Active Directory name: DOMAIN\m.reynolds

User 2

- Full name: River Tam
- Active Directory name: DOMAIN\r.tam

User 3

- Full name: Jayne Cobb
- Active Directory name: DOMAIN\j.cobb

`FULLNAME()` returns the full name of the user logged on to Tableau Server. For example, if Malcolm is the user currently logged into Tableau Server, `FULLNAME()` will return Malcolm Reynolds. In design mode, the author has the ability to impersonate any registered user on the server. Expression = a valid discrete argument.

Basic

```
FULLNAME()='River Tam'
```

This will return a Boolean result, which depends on whether or not River Tam is the user logged onto Tableau Server. If the FULLNAME () of the user is River Tam the formula will return true. If the name does not match it will return false.

Intermediate

```
FULLNAME()=[Sales Person]
```

Returns a Boolean result (true/false) if the logged-in user matches the `[Sales Person]` dimension on a given row of data. This can be used to provide a global filter to enforce row-level security.

Advanced

```
CASE FULLNAME()
WHEN [Sales Person] Then 'True'
WHEN [Junior Manager] Then 'True'
WHEN [Snr Manager] Then 'True'
ELSE 'False'
END
```

This will compare `FULLNAME()` to several fields in the database returning `True` whenever the logged on user matches a name in either the `Sales Person`, `Junior Manager`, or `Snr Manager` field. This can be applied to a global filter to enforce row level security.

43. GET_JASON_OBJECT

This is a Google Big Query–specific function that returns a JSON object within the JSON string based on the JSON path. Refer to the Apache Language Manual UDF for details at `https://cwiki.apache.org/confluence/display/Hive/LanguageManual+UDF`.

```
GET_JSON_OBJECT(JSON string, JSON path)
```

44. GROUP_CONCAT

This is a Google Big Query-specific function that concatenates values from each record into a single comma-delimited string. This function acts like a `SUM()` function for strings.

```
GROUP_CONCAT(expression)
```

Basic

```
GROUP_CONCAT(Region)="Central,East,West"
```

45. HEXBINX

The `HEXBINX` function accepts two variables that will be treated as a horizontal axis and vertical axis, or as Cartesian coordinates. The values within the designated axis variables are then placed in hexagonal bins and can be plotted. The output of the `HEXBINX` function will become the new horizontal axis.

Basic

```
HEXBINX([Lon],[Lat])
```

Converts a set of longitude and latitude coordinates into hexagonal bins. Note that these are not the autogenerated longitude and latitude values that Tableau often creates by default. The autogenerated coordinates cannot be used in the `HEXBIN` functions, so in this example latitude and longitude are generated outside of Tableau.

Intermediate

```
HEXBINX([Lon]*[Scalar Value],[Lat]*[Scalar Value])/[Scalar Value]
```

This formula uses a parameter to allow the user to define the sizes of the hexagonal bins. As the scalar value parameter increases, the number of hexagonal bins displayed in the view will increase as well.

46. HEXBINY

The `HEXBINY` function accepts two variables that will be treated as a horizontal axis and vertical axis, or as Cartesian coordinates. The values within the designated axis variables are then placed in hexagonal bins and can be plotted. The output of `HEXBINY` function will become the new vertical axis.

Basic

```
HEXBINY([Lon],[Lat])
```

Converts a set of longitude and latitude coordinates into hexagonal bins. Note that these are not the autogenerated longitude and latitude values that Tableau often creates by default. The autogenerated coordinates cannot be used in the `HEXBIN` functions, so in this example latitude and longitude are generated outside of Tableau.

Intermediate

```
HEXBINY([Lon]*[Scalar Value],[Lat]*[Scalar Value])/[Scalar Value]
```

This formula uses a parameter to allow the user to define the sizes of the hexagonal bins. As the scalar value parameter increases, the number of hexagonal bins displayed in the view will increase as well.

47. HOST

This is a Google Big Query–specific function that returns the host name as a string for a given URL.

```
HOST(string_URL)
```

Basic

```
HOST('http://www.google.com:80:/index.html')='www.google.com:80'
```

This formula returns the `www.google.com:80` host for the URL string provided within the formula.

48. IF

The `IF` statement is is a logical function that allows you to test IF, THEN, ELSE conditions and return a result that meets the specified conditions.

```
IF test THEN value END / IF test THEN value ELSE else END
```

Basic

```
IF [Order Quantity] > 10 THEN "Bulk Buy" ELSE "Non Bulk" END
```

This example provides simple segmenting of the order quantities greater than ten units to be named Bulk Buys, while all lesser order quantities are classified as Non Bulk. Adding more `ELSEIF` clauses provides a means for adding more logical criteria.

```
IF test1 THEN value1 ELSEIF test2 THEN value2 ELSE else END
```

Intermediate

```
IF [Ship Mode] = "Regular Air"
THEN "Customs Required"
ELSEIF [Ship Mode] = "Express Air"
THEN "Express Customs"
ELSE "No Customs" END
```

This formula evaluates the `[Ship Mode]` field. If it contains `"Regular Air"` then the answer returned will be `"Customs Required"`. If it contains `"Express Air"`, the answer returned will be “`Express Customs"`. Any other value in the field will return `"No Customs"`.

49. IFNULL

The `IFNULL` statement is a simple reference function against a field. It contains two expressions. The first expression is a testing expression, and the second is the override expression. If the first expression is NULL, then it returns the override expression as the result. If the first expression is not NULL then it retains that value.

```
IFNULL(expresson1,expression2)
```

This formula will evaluate `expression1`. If the `expression1` is NULL, then `expression2` is returned. If `expression1` is not NULL, then it returns the `expression1` value.

Basic

```
IFNULL([Customer],"Unidentified")
```

If the customer field *does* have a `NULL` value then it will return the string `Unidentified`; otherwise, the value of `[Customer]` will be returned.

50. IIF

The `IIF` function uses similar logic to the `IF` statement; however, its arguments and return values are not as flexible. The `IIF` statement contains a `TEST` argument, followed by a `THEN` statement, and an `ELSE` statement. The test argument is first calculated; if the result is `TRUE`, then it returns the `THEN` statement as an answer. If the result of the argument is `FALSE`, then it returns the ELSE statement as an answer. An additional UNKNOWN value can be added to the end of an IIF statement should the TEST argument not return either a TRUE or FALSE value.

```
IIF(test,then,else)
or
IIF(test, then, else,[unknown])
```

Basic

```
IIF(1<2,"True","False")
```

The expression will return True.

Intermediate

```
IIF([Time to Ship]>12,"Within SLA", "Outside SLA" )
```

If the time to ship is greater than 12, then the function returns Within SLA. Otherwise the function returns Outside SLA.

Advanced

```
IIF([Order Date]< Today()-14 and [Ship Mode] = "N"
,"High Priority",IIF([Order Date]<Today()-4 and
 [Ship Mode] = "N","Medium Priority","Low Priority"))
```

This more complex example checks first for order dates that are 14 days before today's date and with a ship mode of N, returning High Priority in this case. More recent orders that are within four days of today's date are classified as Medium priority if ship mode is N. Otherwise, they are classified as low priority orders. Nested IIF statements can be used to embed intricate logic in comparisons.

51. INCLUDE

One of the three key words used in Level of Detail calculations. The INCLUDE keyword will add the dimensions listed from the dimensionality of the calculation to calculate a result at a more granular level of detail than present in the view.

```
{INCLUDE [Dimension 1], [Dimension 2],... : AGG([Measure]))}
```

Basic

```
AVG({ INCLUDE [Sub-Category]: AVG([Sales]))})
```

This calculation will determine the average sales at the level of detail of Sub-Category, then average those values to give the average of averages. This is analogous to creating a table with average sales values and totals computed using average.

52. INDEX()

The `INDEX` function returns the row number of the current row within the window (pane) or partition. This function does not require any arguments.

Basic

```
INDEX()
```

This table calculation function is useful for creating ranking lists or as a row number function.

Intermediate

```
Index() <= 5
```

This table calculation allows a user to filter for the TOP X fields within the view/partition. For example, if you need TOP 5 products within a category, this function will provide such a filter. Ensure that you are properly using the "Compute using" option of the table calculation.

Advanced

```
IF INDEX()=1 THEN WINDOW_AVG(SUM([Sales]),0,LAST(),0)) END
```

This function assumes there is a large number of marks on the view and/or that the source dataset is large. The if/then logic bypasses table scans which can cause noticeable performance degradation. The calculation takes the average of the sum of sales for the specified window (or pane). `LAST()` returns the number of rows from the current row to the last row in the view/partition.

53. INT

The function `INT` converts a value to an integer. If the value is a floating point number, it will round down to the nearest integer (this can be used as a `FLOOR` function).

Basic

```
INT(3.7)
```

This formula returns the integer 3.

Intermediate

```
INT([Date])
```

This formula returns the integer representation of a date. This can be useful in certain calculations, or to create a `BIN` on a date. (Tableau will only allow you to create bins on measures.)

Advanced

```
INT(MID(4,[QtyString]))
```

In this example, `[QtyString]` is a string that contains a quantity that you want to use in additional calculations, QTY 5. The `MID` function is used to dispose of the text `(QTY)` and return the number only. The `INT` function then converts the string into an integer.

54. ISDATE

The function `ISDATE` checks to see if a text is a valid date. The resulting output is a Boolean value. If the date string is valid, it will return `TRUE`; otherwise, an invalid date string returns a `FALSE` value.

```
ISDATE(Text)
```

`Text` is the value you want to test.

Basic

```
ISDATE("This is not a date")
```

This formula returns `FALSE`.

Intermediate

```
ISDATE("01 January 2013")=TRUE
ISDATE("1st January 2012")=FALSE
ISDATE("1/9/2012")=TRUE
```

This formula returns `TRUE` values for strings that are appropriately formatted as dates. The `FALSE` occurs because `st` is not being recognized as a valid date format.

Advanced

```
ISDATE(STR([Year]) + '/' + STR([Month]) + '/' + STR([Day]))
```

This formula returns `TRUE` if the date constructed is valid.

55. ISFULLNAME()

The following assumptions are used for the examples:

User 1

- Full name: Malcolm Reynolds
- Active Directory name: DOMAIN\m.reynolds

User 2

- Full name: River Tam
- Active Directory name: DOMAIN\r.tam

User 3

- Full name: Jayne Cobb
- Active Directory name: DOMAIN\j.cobb

`ISFULLNAME()` returns a Boolean (true/false) value when the string or dimension specified in the brackets matches the user's full name for the user logged on to Tableau Server. In design mode, the author has the ability to impersonate any registered user on the server.

Basic

```
ISFULLNAME('River Tam')
```

Returns a Boolean depending on whether or not River is the user logged onto Tableau Server. Unlike using `FULLNAME()` within an `IF` or `CASE` statement, using `ISFULLNAME()` requires that you input the string value manually. You cannot reference another dimension.

Intermediate

```
IF ISFULLNAME('Malcolm Reynolds') THEN 'Management'
ELSEIF ISFULLNAME('River Tam') THEN 'Sales'
ELSEIF ISFULLNAME('Jayne Cobb') THEN 'Public Relations'
ELSE 'Unknown'
END
```

The `ISFULLNAME()` function is used to put the users into logical groups. It could be used to customize dynamic titles or color schemes.

56. ISMEMBEROF()

This user function returns a Boolean (true/false) based on the logged-in user's group membership defined on Tableau Server.

Basic

```
ISMEMBEROF('Sales')
```

If the user is a member of the group (local or Active Directory) Sales then the function returns a true value; otherwise it returns false. This could be used to manage row level security.

Intermediate

```
IF ISMEMBEROF('Management') THEN 'Access Permitted'
ELSEIF ISFULLNAME('Sales') THEN 'Access Permitted'
ELSE 'Access Denied'
END
```

This example is being used to drive permissions for a sales dashboard while providing row-level access only to sales team members and senior management. Other staff could access the dashboard but would return no data.

57. ISNULL

The ISNULL statement is a simple Boolean function. It returns `TRUE` if expression is `NULL`; returns `FALSE` if it is not `NULL`.

```
ISNULL(expression)
```

Basic

```
ISNULL([Customer])
```

The `ISNULL` example will return a FALSE value if a customer name exists or a TRUE value if the Customer field is `NULL`.

58. ISUSERNAME()

The following assumptions are used for the examples:

User 1

- Full name: Malcolm Reynolds
- Active Directory name: DOMAIN\m.reynolds

User 2

- Full name: River Tam
- Active Directory name: DOMAIN\r.tam

User 3

- Full name: Jayne Cobb
- Active Directory name: DOMAIN\j.cobb

Returns `TRUE` if the current user's username matches the specified username or `FALSE` if it does not match.

Basic

```
ISUSERNAME('j.cobb')
```

Returns a Boolean `TRUE` if Jayne is logged on the server and `FALSE` if not the current user.

59. LAST()

This table calculation function does not require any arguments.

Basic

```
LAST()
```

This function returns the amount of rows from the current row to the last row in the view/partition.

Intermediate

```
WINDOW_COUNT(SUM([Sales]),FIRST(),LAST())
```

This function returns the count of the sum of sales from the first row to the last row of the window (or frame). Note that if the values are incorrect or inconsistent, ensure that you are properly using the "Compute using" option of the table calculation.

Advanced

```
IF INDEX()=1 THEN WINDOW_AVG(SUM([Sales]),0,LAST(),0)) END
```

This function assumes a large amount of marks on the view and/or that the user is dealing with big data. Through if/then logic, you can bypass the table scans, which cause noticeable performance degradation. In the end, the calculation takes the count of the sum of sales for the specified window (or frame). Bonus: Try making the window frames dynamic with a parameter.

60. LEFT

`LEFT` is a String function that returns the left-most characters from its designated string. This function can be used to create new dimensions directly or combined to create advanced calculated fields.

```
LEFT(String,Number)
```

Basic

```
LEFT([Customer Zip Code],3)
```

> **The function example is a simple way of identifying the sectional center facility used within the current U.S. zip code system. `LEFT` can be used in more advanced queries when only the beginning sections of a string are required to be separated or queried.**

61. LEN

Returns the length of a string as an integer. Note that `LEN` counts spaces between string characters to contribute to the `LEN` total value.

```
LEN(String)
```

Basic

```
LEN("Bob Hope")
```

> **The value returned for the preceding calculation is 8, 3 for Bob, 1 for the space, and 4 for Hope.**

Advanced

The Advanced `FIND` example that was provided previously uses the `LEN` statement to help complete the calculation.

```
RIGHT([Customer],LEN([Customer])-FIND([Customer],"'",4)+2)
```

> **Calculate the length of the field `[Customer]`, subtract the index value for the location of any apostrophes within `[Customer]`, and take the `right`-most number of characters. The number of characters is the result of the simple subtraction that is taking place.**

62. LN

The `LN` function returns the natural logarithm of a number. This is the logarithm to the base *e*, where *e,* has the value 2.71828182845905. In Tableau, the return is accurate to 14 decimal places. The natural logarithm of the expression is the power to which *e* would have to be raised to equal the expression.

```
LN(number)
```

> **`Number` = any given number greater than zero. If the number is less than or equal to zero, the `LN` function returns a NULL.**

Basic

```
LN(7.38905609893065)
```

> **The result of this function is 2. In this case, *e* would need to be raised to the power of 2 to equal the expression.**

63. LOG

The `LOG` function returns the logarithm of a number for the given base. The logarithm of the expression is the power to which the base would have to be raised to equal the expression. If the base value is omitted, then base 10 is used.

```
LOG(number, [base])
```

`Number` = any given number greater than zero. If the number is less than or equal to zero, the `LOG` function returns a NULL, and `[base]` = any given number (not required).

Basic

```
LOG(1000)
```

The return of this function is 3, as base 10 needs to be raised to the power of 3 to return the expression. In other words, the calculation 10*10*10 would return 1000.

```
LOG(8,2)
```

This function also returns 3. In this case, base 2 is used, and 2 should to be raised to the power of 3 to return the expression. In other words, the calculation 2*2*2 would return 8.

64. LOG2

A Google Big Query–specific function that returns the logarithm base 2 of a number.

```
LOG2(number)
```

Basic

```
LOG2(16)
```

`Number` = any given number greater than zero. The answer 4.00 is returned by the basic example.

65. LOOKUP

A table calculation function that returns the value of the expression in the target row, specified as the relative offset from the current row. Use `FIRST() + n` and `Last() -n` as part of the offset definition for the target relative to the first or last rows in the partition. If offset is omitted, the row to compare to can be set on the field menu. This function returns `NULL` if the target row cannot be determined.

`LOOKUP(expression,[offset])`

> `Expression` = **Any valid aggregate calculation (for example,** `SUM([Sales]))` `[offset]` = **target row from first/last).**

Basic

`LOOKUP(SUM([Sales]),2)`

> **This example returns** `SUM([Sales])` **in each row for the sales from the forward looking or future sales. Put simply, it grabs the value of sales from two rows down.**

Intermediate

`LOOKUP(SUM([Sales]), FIRST()+1)`

> **This function returns** `SUM([Sales])` **in the second row of the view/partition. Notice the other table calculations can be used as a helper in this function.**

Advanced

`LOOKUP( MIN([Region]),0)`

> **This formula returns the minimum value within each region contained in the data.**

66. LOWER

This function allows the user to lowercase all characters within a string. The `LOWER` function will only change the uppercased characters that exist in a string and thus ignore all lowercase characters that already exist.

`LOWER(String)`

Basic

`LOWER("BatMan")`

> **The function example will output the following all-lowercase string:** `batman`**.**

67. LTRIM

The `LTRIM` function removes leading spaces that may be present within the data. This function can be used as a data cleansing function so that the data is consistent and set correctly.

`LTRIM (string)`

Basic

```
LTRIM("    Bob Hope" )
```

The output for this would simply be `"Bob Hope"`. You need to remove leading spaces as these can cause a number of issues if you try to apply any additional functions to the data. An example of this would be:

```
LEFT("    Bob Hope", 4) which result is "    ".
```

68. LTRIM_THIS

A Google Big Query–specific function that removes all of the leftmost characters from the first string that match the second string. It is case sensitive.

```
LTRIM_THIS(string1, string2)
```

Basic

```
LTRIM_THIS('Remove Me',' Me')
```

The function example will = `Remove`. The matching stings in the first and second strings are the space and the word `Me`.

69. MAKEDATE

The `MAKEDATE` function provides a simple way to create a date given three basic date parts: a year, a month number, and a day number. This is a useful function for avoiding complex variable type conversions if attempting to construct a date variable out of separate date parts.

```
MAKEDATE(year, month, day)
```

Basic

```
MAKEDATE(2015, 6, 18)
```

The function reads in the three different date parts of year, month, and day and returns a Tableau date.

70. MAKEDATETIME

Returns a datetime that combines a date and a time. The date can be a date, datetime, or a string type. The time must be a datetime. This function is available only for MySQL connections.

Basic

```
MAKEDATETIME([Order Date], #02:32:59#)
```

This function applies a time stamp of "2:32:59 AM" to every date value within the order date variable, thus modifying each entry to the datetime format.

71. MAKETIME

The `MAKETIME` function allows users to define their own time values using a calculated field. There are three inputs required: hours, minutes, and seconds. The output is formatted as hh:mm:ss.

Basic

```
MAKETIME(15,37,59)
```

Result is a time value that appears as 3:37:59 PM.

72. MAX

The `MAX` function is normally reserved for numbers, but this function can also be used on strings and dates. When `MAX` is applied to strings, the `MAX` value returns the string that is highest within the data's sort sequence for that particular string.

```
MAX(a,b)
```

Basic

```
MAX("Maureen","William")
```

It considers the first name in alphabetical order to be the lowest and the last name to be the highest, and thus `"William"` is returned. If any of the strings used in the comparison logic have a value of `NULL`, then `NULL` will be the resulting output.

Intermediate

```
MAX([Sales])
```

This example returns the maximum value across all the rows in the database.

```
MAX([Sales],[Profit])
```

This example returns the maximum value between sales and profit for every row in the database.

The MAX function can also be used as a String function or Date function, whereby expression1 and expression2 are string or date data types respectively.

Advanced

```
MAX(ABS([Sales]-[Sales est]))
```

This formula returns the difference between the estimated sales and actual sales. In this example, the largest differences (positive or negative) would be derived and displayed for any dimension expressed in the view.

73. MEDIAN

This function returns the median of a single expression. MEDIAN can be used with numeric fields only. NULL values are ignored. For Tableau Desktop workbooks created by versions before 8.2, if the data source is Excel, Access, or a text file , this function will not be available unless the data source is extracted.

Basic

```
MEDIAN([Discount])
```

This formula will return the MEDIAN value of Discount for each level of detail in view.

Intermediate

```
MEDIAN(DATEDIFF('day',[Order Date],[Ship Date])
```

This formula returns the MEDIAN difference between the order date and ship date, and expresses the result in days for each partition in the view.

74. MID

The MID function returns a partial string as its output. The MID function allows extraction of specific segments from within a string. This function requires an index position, from which the extraction begins. The function then extracts all parts of the string from the index position onwards, or an optional argument can be used to only extract a certain number of characters from the start index position.

```
MID(string,start,[Length])
```

Basic

```
MID("Michael Gilpin",9)
```

The function example output will be Gilpin. The `MID` function only begins extracting the string at index point 9. Michael contains 7 characters, and the space is also classed as a character: hence the extraction begins at the G and returns the remaining data from within that string.

Intermediate

```
MID("Michael Gilpin",9,4)
```

The function example is similar to the first example, but an additional argument has been applied. This adds another option to the function. This example begins extracting the string at index position 9, the letter G; the length function that has now been added then limits the extract to four characters starting at the index position. The result of this new calculation is Gilp.

75. MIN

The `MIN` function is similar to the `MAX` function, whereby this function returns the minimum value when applied to a single field in an aggregate calculation. The `MIN` function can also be applied to return the minimum of two arguments. These arguments must be of the same type. When used with two arguments, the function returns NULL if either argument is NULL.

```
MIN(expression1, expression2)
```

`Expression1` = A valid number or aggregate calculation. `Expression2` = A valid number or aggregate calculation (not required).

Basic

```
MIN([Sales])
```

This function returns the minimum value across all rows in the database.

```
MIN([Sales],[Profit])
```

This function returns the minimum value between Sales and Profit for every row in the database. Note that this function can be used as a String or Date type function, where `expression1` and `expression2` are string or date data types.

Intermediate

```
MIN([Shipping Cost],[Maximum Shipping Cost])
```

Another use case for `MIN` allows you to pass in two fields and return the minimum value between the two. In this example, a parameter (variable) is used to permit the user to select the maximum allowable value for comparison.

Advanced

```
DATEDIFF('day',MIN([Order Date]),MAX([Ship Date]))
```

This formula derives the difference between the minimum order date and the maximum ship date. This could be used in conjunction with an order identification dimension to calculate how long it takes for the entire order to be filled if there were multiple shipments.

76. MONTH

This function returns an integer representing the month of any given date. This is a shortened form of `DATEPART` ('month', date).

```
MONTH(Date)
```

The `Date` is the date the function will use to extract the month.

Basic

```
MONTH(#March 14, 2013#)
```

This formula returns 3.

Intermediate

```
MONTH(DATEADD('day',[Date],5))
```

This formula returns the month that `[Date]` + 5 days lands on. If the `[Date]` is March 30, 2013 the function will return 4 (since `[Date]` + 5 days is April 4, 2012).

Advanced

```
CASE [Parameter].[Date Unit]
WHEN 'Day' THEN DAY([Date])
WHEN 'Month' THEN MONTH([Date])
WHEN 'Year' THEN Year([Date])
END
```

This formula returns an integer. In this example, a parameter for the `[Date Unit]` is used to control the level of detail of the result: day, month, or year.

77. NOT

The `NOT` statement, when placed before a mathematical or logical expression, negates the evaluation of the given expression. This function is useful when a desired outcome is most easily achieved by specifying values that should not be included in the resulting output.

Basic

```
IF NOT QUARTER([Order Date]) = 1 THEN [Profit] END
```

This formula singles out profits whose orders occurred in the first quarter of the year. The NOT statement that appears early on in the equation indicates that these first quarter profits will be excluded from the result.

Advanced

```
IF NOT ([Segment] = "Home Office"
AND ([Order Priority] = "Low" OR [Order Priority] = "Medium"))
THEN [Shipping cost] END
```

This equation looks for values associated with the Home Office customer segment that had an order priority that was specified as either low or medium. Without the NOT statement, this algorithm would return any data points that satisfy both of those conditions. However, because the NOT statement is included the algorithm, it modifies the AND statement to a NAND operator and thus excludes any data points that satisfy the conditions on either side of the AND statement.

78. NOW

The NOW function returns the current date and time.

Basic

```
NOW( )
```

Presuming today's date is March 12, 2013 at 3:04 PM, the preceding formula will return the date March 12, 2013 03:04:00 PM.

Intermediate

```
DATEADD('hour', -5, NOW())
```

This formula returns a date time that is 5 hours before the current date time. If today is March 18, 2013 at 3PM. then this function will return March 18, 2013 at 10 a.m. This can be very useful in filtering for specific time periods.

79. OR

The OR statement, when placed between two mathematical or logical expressions, evaluates the combined output of both expressions to a single Boolean output. If either of the expressions is TRUE, the final OR output is TRUE. If both expressions are FALSE, the OR output is FALSE. This function is useful when multiple conditions are eligible to trigger a desired calculation or outcome.

Basic

```
IF [Ship Mode] = "First Class" OR [Market] = "EMEA" THEN [Profit] END
```

This forumula extracts all profit values for orders that were shipped via first class service OR were ordered within the Europe/Middle East/Africa market. This would be more difficult to achieve using filters, but as shown here it is easily done using the `OR` statement.

Intermediate

```
IF ([Hours Worked]/[Hours Scheduled] < 0.8)
OR ([Hours Worked]/[Hours Scheduled] > 1.2)
THEN "Needs Attention" ELSE "Reasonable" END
```

The equation considers a ratio of hours worked to hours scheduled. This example assumes that a 20 percent deviation from the scheduled working hours falls within the reasonable range. However, any deviation of a greater magnitude will be classified as needing attention. The `OR` statement allows either of the two independent mathematical expressions to flag the `Needs Attention` classification.

80. PARSE_URL

A Hadoop Hive–specific function that returns a component of the given URL string where the component is defined by `url_part`. Valid `url_part` values include: `HOST`, `PATH`, `QUERY`, `REF`, `PROTOCOL`, `AUTHORITY`, `FILE`, and `USERINFO`.

```
PARSE_URL(string,url_part)
```

Basic

```
PARSE_URL('http://www.tableau.com','HOST')
```

The formula returns the URL for Tableau Software, `www.tableau.com`.

81. PARSE_URL_QUERY

A Hadoop Hive–specific function that returns the value of the specified query parameter in the given URL string. The query parameter is defined by the key.

```
PARSE_URL_QUERY(string,url_part)
```

Basic

```
PARSE_URL_QUERY('http://www.tableau.com?page=1&cat=4','page')
```

The formula returns value of the query parameter for the page. In this example, the answer = “1”.

82. PERCENTILE

The PERCENTILE function requires an input variable and a user-defined value between 0 and 1, which represents the decimal form of a desired percentage. The function applies the specified percentage to the numerical range that exists within the input variable and outputs the corresponding result. A 0 percent specification is equivalent to the minimum value, a 100 percent specification is equivalent to the maximum value, and a 0.5 percent specification is equivalent to the median value.

Basic

```
PERCENTILE([Sales],0.5)
```

This function will output the median value within the sales variable. The median is the exact midpoint of any given dataset, just as 0.5 is the midpoint between 0 and 1. The value of 0.5 represents the 50 percent mark between the minimum and maximum value in the sales variable, and so the output of this PERCENT example will be the middle, or median, of the sales data.

83. PI

The function PI returns the mathematical constant pi, also expressed with the symbol π. The value is approximately equal to 3.14159265358979. In Tableau, the return is accurate to 14 decimal places.

```
PI()
```

The PI function returns the constant value 3.14159265358979. There is no expression for this function—simply add the open and close parentheses ().

Basic

```
2*PI()*5
```

This function returns 31.4159265358979. This is the formula where PI is used to calculate the circumference of a circle with the formula 2π*radius, where in this example 5 is the value for the radius of the circle.

84. POWER

The POWER function raises the number to the specified power.

```
POWER(number,power)
```

Number **= any given number.** Power **= any given number.**

Basic

```
POWER(4,3)
```

This function raises `4` to the power of `3`. The result is 64. The ^ symbol can also be used in the calculation instead. The calculation 4^3 therefore returns the same result as `POWER (4,3)`.

Intermediate

```
[Profit]*POWER(1+0.12,6)
```

The `POWER` function can be used to apply exponential growth factors over time. This function reflects a growth factor of 12 percent (for which the number expression is 1.12) over 6 periods.

85. PREVIOUS_VALUE

Returns the value of calculation in the previous reow. Returns the given expression if the current row is the first row in the partition.

```
PREVIOUS_VALUE(expression)
```

`Expression` = the previous row or the value of the given expression if the row is the first one in the partition..

Basic

```
SUM([Sales])+ PREVIOUS_VALUE(1)
```

This formula returns the running sum of `Sum([Sales])`.

86. RADIANS

The `RADIANS` function converts the given number from degrees to radians.

```
RADIANS(number)
```

`Number` = any given number

Basic

```
RADIANS(360)
```

This function converts the value of 360 degrees to the radian value. The result is 6.28318530717959, which is equal to 2π. In Tableau, the return is truncated to 14 decimal places. The conversion from degrees to radians can also be calculated with the `PI()` function: `360 * (PI()/180)`, which yields the same return.

87. RANK

The `RANK` table calculation function returns an ordered list of values currently in the window (pane) or partition. This version of the function uses a standard competition ("1,2,2,4") rank and descending order by default. Items that equally compare are given the same rank number, and subsequent numbers are skipped.

```
RANK(expression, ['asc' | 'desc'])
```

`Expression` = **Any valid aggregate calculation (for example,** `SUM ([Sales]));`
`['asc'|'desc']` = **specified rank order [not required]).**

Basic

```
RANK(SUM([Sales]))
```

This returns the numbered rank according to `the total sales within the presented level of detail in the view`**. The default Rank shows in descending order. Null values are ignored in ranking functions.**

88. RANK_DENSE

The `RANK_DENSE` table calculation function returns an ordered list of values currently in the window (pane) or partition. This version of the function uses a dense ("1,2,2,3") rank and descending order by default. Items that equally compare are given the same rank number and subsequent items are assigned the following rank number.

```
RANK_DENSE(expression, ['asc' | 'desc'])
```

`Expression` = **Any valid aggregate calculation (for example,** `SUM ([Sales]));`
`['asc'|'desc']` = **specified rank order [not required]).**

Basic

```
RANK_DENSE (SUM([Sales]),'asc'))
```

This returns the numbered rank according to the sum of sales sorted ascending from lowest to highest sales. The default Rank shows in descending order. \ Null values are ignored in ranking functions.

89. RANK_MODIFIED

The `RANK_MODIFIED` table calculation function returns an ordered list of values currently in the window (pane) or partition. This version of the function uses a modified competition ("1,3,3,4") rank and descending order by default. A rank value is skipped, creating a gap before equally comparable items.

```
RANK_MODIFIED(expression, ['asc' | 'desc'])
```

`Expression` = **Any valid aggregate calculation (for example,** `SUM ([Sales]));`
`['asc'|'desc']` = **specified rank order (not required).**

Basic

```
RANK_MODIFIED(SUM([Sales]))
```

This returns the numbered rank according to `the sum of sales`**. The default Rank shows in descending order. Null values are ignored in ranking functions. Identical values are assigned an identical rank.**

90. RANK_PERCENTILE

The `RANK_PERCENTILE` table calculation function returns an ordered list of percentages currently in the window (pane) or partition. This version of the function returns a percentile ("25,75,75,100") rank and ascending order by default. Items are assigned percentages according to their position in the frequency distribution. Note that using the percent number format provides accurate results.

```
RANK_PERCENTILE(expression, ['asc' | 'desc'])
```

`Expression` = **Any valid aggregate calculation (for example,** `SUM ([Sales]));`
`['asc'|'desc']` = **specified rank order [not required]).**

Basic

```
RANK_PERCENTILE(SUM([Sales]))
```

This returns the percentile rank according to `the sum of sales`**. The default Rank shows in ascending order. Null values are ignored in ranking functions.**

91. RANK_UNIQUE

The `RANK_UNIQUE` table calculation function returns an ordered list of values currently in the window (pane) or partition. This version of the function uses an ordinal ("1,2,3,4") rank and descending order by default. Items that equally compare receive separate numbers assigned by the sort order, typically alphabetical.

```
RANK_UNIQUE(expression, ['asc' | 'desc'])
```

`Expression` = **Any valid aggregate calculation (for example,** `SUM ([Sales]));`
`['asc'|'desc']` = **specified rank order [not required]).**

Basic

```
RANK_UNIQUE(SUM([Sales]))
```

This returns the numbered rank according to the sum of sales. The default Rank shows in descending order. Null values are ignored in ranking functions.

92. RAWSQL_BOOL()

The `RAWSQL_BOOL()` function is a pass-through function that allows the user to send an arbitrary expression to the underlying data source. The expression must return a scalar value of a type that Tableau can convert into a Boolean. This expression will not be checked in any way by Tableau and may produce an error at the data source level. The user must respect the syntax conventions of the data source when constructing the expression. The following is the generalized syntax for the function:

```
RAWSQL_BOOL("expr",[arg1],...[argN])
```

The `expr` in quotes is the expression to be passed through to the data source. `N` number of arguments can be specified in a comma-separated list as shown. These arguments are referenced in the expression with a %1, %2, and % `N` syntax.

Basic

```
RAWSQL_BOOL("%1=%2",[Order Date],[Ship Date])
```

This example checks whether or not the `[Order Date]` and `[Ship Date]` fields are equivalent. The function will return true if they are equivalent and false if they are not.

Intermediate

```
RAWSQL_BOOL("%1='Oklahoma' AND %2 > 100.00",[State],[Sales])
```

This formula checks whether or not the `[State]` field is equal to Oklahoma and if the `[Sales]` field is greater than 100.00. If both of these are true, the function will return true. It will return false if either is false.

Advanced

```
RAWSQL_BOOL("PATINDEX('%Henry%',%1)>0 AND %2>100.00",[Customer
Name],[Sales])
```

This formula performs the SQL Server `PATINDEX()` function on the `[Customer Name]` field to look for the presence of the substring Henry. If that string is contained in the `[Customer Name]` field and the `[Sales]` field is greater than 100.00 the function returns true. If either condition is false the function returns false.

93. RAWSQL_DATE()

The `RAWSQL_DATE()` function is a pass-through function that allows the user to send an arbitrary expression to the underlying data source. The expression must return a scalar value of a type that Tableau can convert into a date. Tableau will ignore any time component if a date-time is returned. This expression will not be checked in any way by Tableau and may produce an error at the data source level. The user must respect the syntax conventions of the data source when constructing the expression. The generalized syntax for the function is as follows:

```
RAWSQL_DATE("expr",[arg1], ...[argN])
```

The `expr` in quotes is the expression to be passed through to the data source. `N` number of arguments can be specified in a comma-separated list as shown. These arguments are referenced in the expression with a %1, %2, and %`N` syntax.

Basic

```
RAWSQL_DATE("%1 + 10", [Order Date])
```

This example adds 10 days to the `[Order Date]` value.

Intermediate

```
RAWSQL_DATE("COALESCE(%2, %1)", [Order Date], [Ship Date])
```

This example uses the SQL Server `COALESCE()` function to choose the first value from `[Order Date]` and `[Ship Date]` that is not NULL.

Advanced

```
RAWSQL_DATE("CASE WHEN %1 = 'Critical' THEN %2+2
WHEN %1 = 'High' THEN %2+3
WHEN %1 = 'Medium' THEN %2+4
ELSE %2+10 END", [Order Priority], [Order Date])
```

This example uses a SQL Server `CASE` statement to add a different number of days to the `[Order Date]` field depending on which `[Order Priority]` value is found.

94. RAWSQL_DATETIME()

The `RAWSQL_DATETIME()` function is a pass-through function that allows the user to send an arbitrary expression to the underlying data source. The expression must return a scalar value of a type that Tableau can convert into a date time. This expression will not be checked in any way by Tableau and may produce an error at the data source level. The user must respect the syntax conventions of the data source when constructing the expression. The generalized syntax for the function is as follows:

```
RAWSQL_DATETIME("expr", [arg1], ...[argN])
```

The `expr` in quotes is the expression to be passed through to the data source. `N` number of arguments can be specified in a comma-separated list as shown. These arguments are referenced in the expression with a %1, %2, and % `N` syntax.

Basic

```
RAWSQL_DATETIME("%1 + '06:30:00'", [Order Date])
```

In this example, you're adding the time literal `'06:30:00'` (6 hours, 30 minutes, 0 seconds) to the `[Order Date]` value.

Intermediate

```
RAWSQL_DATETIME("DATETIMEFROMPARTS(2013,2,24,9,40,35,0)")
```

This example uses the SQL Server function `DATETIMEFROMPARTS()` to build a date-time value from time parts. The resulting date-time literal in SQL Server would be expressed as `2013-02-24 09:40:35:000`.

Advanced

```
RAWSQL_DATETIME("CASE WHEN %2 = 'East'THEN %1 + '01:00:00'
WHEN %2 = 'West' THEN %1—'02:00:00'
ELSE %1 END", [Order Date], [Region])
```

This example uses the SQL Server `CASE` statement to add or subtract a time literal depending on the value in the `[Region]` field. When the `[Region]` is `East`, you will add 1 hour and when the `[Region]` is `West`, you will subtract 2 hours. If the `[Region]` is any other value it will simply return `[Order Date]` without any change.

95. RAWSQL_INT()

The `RAWSQL_INT()` function is a pass-through function that allows the user to send an arbitrary expression to the underlying data source. The expression must return a scalar value of a type that Tableau can convert into an integer. This expression will not be checked in any way by Tableau and may produce an error at the data-source level. The user must respect the syntax conventions of the data source when constructing the expression. The generalized syntax for the function is as follows:

```
RAWSQL_INT("expr",[arg1],...[argN])
```

The `expr` in quotes is the expression to be passed through to the data source. `N` number of arguments can be specified in a comma-separated list as shown. These arguments are referenced in the expression with a %1, %2, and % `N` syntax.

Basic

```
RAWSQL_INT("1+2")
```

This formula returns the addition of the integers 1 and 2. If the result is not 3, something has gone seriously awry.

Intermediate

```
RAWSQL_INT("CEILING(%1)",[Unit Price])
```

This example uses the SQL Server function `CEILING()`, which takes in numeric values and returns the smallest integer that is larger than the argument. You're passing `[Unit Price]` into the `CEILING()` function.

Advanced

```
RAWSQL_INT("DATEDIFF(day,COALESCE(%2,%1),GETDATE())",[Order Date],[Ship
Date])
```

In this example, you're using the SQL Server functions `DATEDIFF()`, `COALESCE()`, and `GETDATE()` to return the difference in days between the current SQL Server date time and either the `[Ship Date]` or the `[Order Date]`, depending on whether either value is NULL.

96. RAWSQL_REAL()

The `RAWSQL_REAL()` function is a pass-through function that allows the user to send an arbitrary expression to the underlying data source. The expression must return a scalar value of a type that Tableau can convert into a number. This expression will not be checked in any way by Tableau and may produce an error at the data-source level. The user must respect the syntax conventions of the data source when constructing the expression. The generalized syntax for the function is as follows:

```
RAWSQL_REAL("expr",[arg1],...[argN])
```

The `expr` in quotes is the expression to be passed through to the data source. `N` number of arguments can be specified in a comma-separated list as shown. These arguments are referenced in the expression with a %1, %2, and % `N` syntax.

Basic

```
RAWSQL_REAL("5.39 + 3.56")
```

This example adds the two numeric values 5.39 and 3.56 together. The result is 8.95.

Intermediate

```
RAWSQL_REAL("RAND()")
```

This example uses the SQL Server function `RAND()` to generate a pseudo random number. The generated value is a numeric type between 0 and 1.

Advanced

```
RAWSQL_REAL("ROUND(CASE WHEN %1 = 'East' THEN %2 * 1.15
WHEN %1 = 'West' THEN %2 * 0.85
ELSE %2 END, 2)",[Region],[Sales])
```

This formula uses a SQL Server `CASE` statement to selectively multiply the `[Sales]` value by either 1.15 or 0.85, depending on what the `[Region]` value is. You then use the `ROUND()` function to round the result to 2 decimal places.

97. RAWSQL_STR()

The `RAWSQL_STR()` function is a pass-through function that allows the user to send an arbitrary expression to the underlying data source. The expression must return a scalar value of a type that Tableau can convert into a string. This expression will not be checked in any way by Tableau and may produce an error at the data-source level. The user must respect the syntax conventions of the data source when constructing the expression. The generalized syntax for the function is as follows:

```
RAWSQL_STR("expr",[arg1],...[argN])
```

The `expr` in quotes is the expression to be passed through to the data source. `N` number of arguments can be specified in a comma-separated list as shown. These arguments are referenced in the expression with a %1, %2, and %`N` syntax.

Basic

```
RAWSQL_STR("'Trivial Case'")
```

This basic example defines the string literal `Trivial Case` with the `RAWSQL_STR` function. This is truly the trivial case.

Intermediate

```
RAWSQL_STR("%1 + '-' + CONVERT(varchar, %2)",[State],[Zip Code])
```

This example concatenates the value in the `[State]` field with the string literal `'-'` and the string representation of the value in the `[Zip Code]` field. The `CONVERT(varchar, %2)` is necessary in SQL Server to avoid a type error.

Advanced

```
RAWSQL_STR("STUFF(%1,CHARINDEX(' ', %1), 0,' ''' + %2 + ''' ')",
[Customer Name],[State])
```

This example uses the SQL Server `CHARINDEX()` function to locate the first instance of a space in the value of the `[Customer Name]` field and then uses the SQL Server `STUFF()` function to insert the value in the `[State]` field there. Observant readers might notice that the case where the blank space isn't found in the `[Customer Name]` hasn't been handled. If that happens, the `[State]` will just get tacked on to the front of the `[Customer Name]`.

98. RAWSQLAGG_BOOL()

The `RAWSQLAGG_BOOL()` function is a pass-through function that provides a means to send an arbitrary expression to the underlying data source. The expression must return an aggregate value that Tableau can convert into a Boolean. This expression will not be checked in any way by Tableau and may produce an error at the data source level. The user must respect the syntax conventions of the data source when constructing the expression. The generalized syntax for the function is as follows:

```
RAWSQLAGG_BOOL("agg_expr", [arg1], ...[argN])
```

The `agg_expr` in quotes is the expression to be passed through to the data source. `N` number of arguments can be specified in a comma-separated list as shown. These arguments are referenced in the expression with a %1, %2, and %`N` syntax.

Basic

```
RAWSQLAGG_BOOL("SUM(%1) = SUM(%2)", [Sales],[Profit])
```

This example compares the sum of the `[Sales]` values with the sum of the `[Profit]` values and returns true if they are equal. It returns false if they are not.

Intermediate

```
RAWSQLAGG_BOOL("SUM(CASE WHEN %1='Oklahoma' THEN %2 ELSE 0 END)
> 100.00", [State], [Sales])
```

This formula uses the sum of the `[Sales]` values when `[Region] = 'Oklahoma'` and then checks whether or not it is greater than 100.00. The calculation returns true if the sum is greater than 100.00 and false if it is not.

Advanced

```
RAWSQLAGG_BOOL("SUM(CASE WHEN PATINDEX('%Henry%', %1) > 0 THEN %2 ELSE 0
END)
> 100.00",[Customer Name],[Sales])
```

This example performs the SQL Server `PATINDEX()` function on the `[Customer Name]` field to look for the presence of the substring `Henry`. It sums the `[Sales]` values where that string is contained in the `[Customer Name]` field. If that sum is greater than `100.00`, the function returns true. Otherwise it returns false.

99. RAWSQLAGG_DATE()

The `RAWSQLAGG_DATE()` function is a pass-through function that allows the user to send an arbitrary expression to the underlying data source. The expression must return an aggregate value that Tableau can convert into a date. This expression will not be checked in any way by Tableau and may produce an error at the data-source level. The user must respect the syntax conventions of the data source when constructing the expression. The generalized syntax for the function follows:

```
RAWSQLAGG_DATE("agg_expr",[arg1],...[argN])
```

The `agg_expr` in quotes is the expression to be passed through to the data source. `N` number of arguments can be specified in a comma-separated list as shown. These arguments are referenced in the expression with a %1, %2, %`N` syntax.

Basic

```
RAWSQLAGG_DATE("MIN(%1)",[Order Date])
```

This example returns the minimum value of the `[Order Date]` field.

Intermediate

```
RAWSQLAGG_DATE("MAX(COALESCE(%2, %1))",[Order Date],[Ship Date])
```

This formula returns the maximum value of the SQL Server function `COALESCE()` on `[Order Date]` and `[Ship Date]`. The `COALESCE()` function returns the first non-NULL field from its argument list.

Advanced

```
RAWSQLAGG_DATE("MAX(CASE WHEN %1 = 'Critical' THEN COALESCE(%3, %2)
END)",
[Order Priority],[Order Date],[Ship Date])
```

This formula returns the maximum value of the SQL Server function `COALESCE()` on `[Order Date]` and `[Ship Date]` when `[Order Priority]` is `Critical`. The `COALESCE()` function returns the first non-NULL field from its argument list.

100. RAWSQLAGG_DATETIME()

The `RAWSQLAGG_DATETIME()` function is a pass-through function that allows the user to send an arbitrary expression to the underlying data source. The expression must return an aggregate value that Tableau can convert into a date time. This expression will not be checked in any way by Tableau and may produce an error at the data source level. The user must respect the syntax conventions of the data source when constructing the expression. The generalized syntax for the function is as follows:

```
RAWSQLAGG_DATETIME("agg_expr",[arg1],...[argN])
```

The `agg_expr` in quotes is the expression to be passed through to the data source. `N` number of arguments can be specified in a comma-separated list as shown. These arguments are referenced in the expression with a %1, %2, and % `N` syntax.

Basic

```
RAWSQLAGG_DATETIME("MAX(%1)",[Ship Date])
```

This formula returns the maximum value, or most recent date within the `[Ship Date]` field.

Intermediate

```
RAWSQLAGG_DATETIME("MAX(%2–%1)",[Order Date],[Ship Date])
```

This formula returns the maximum of the date-time difference between `[Ship Date]` and `[Order Date]`. In SQL Server, this difference will be returned as a date time that is offset from 1900–01–01.

Advanced

```
RAWSQLAGG_DATETIME("MAX(CASE WHEN %2 = 'East' THEN %1 + '01:00:00'
WHEN %2 = 'West' THEN %1–'02:00:00'
ELSE %1 END)",[Order Date],[Region])
```

This example uses the SQL Server `CASE` statement to add or subtract a time literal depending on the value in the `[Region]` field. When the `[Region]` is `East`, one hour is added, or when the `[Region]` is `West` two hours are subtracted. If the `[Region]` is any other value we simply return `[Order Date]` without any change. The maximum value from the `CASE` statement across all records is returned. This all works perfectly if you process orders in the central time zone.

101. RAWSQLAGG_INT()

The `RAWSQLAGG_INT()` function is a pass-through function that allows the user to send an arbitrary expression to the underlying data source. The expression must return an aggregate value that Tableau can convert into an integer. This expression will not be checked in any way by Tableau and may produce an error at the data source level. The user must respect the syntax conventions of the data source when constructing the expression. The generalized syntax for the function is as follows:

```
RAWSQLAGG_INT("agg_expr",[arg1],...[argN])
```

The `agg_expr` in quotes is the expression to be passed through to the data source. N number of arguments can be specified in a comma-separated list as shown. These arguments are referenced in the expression with a %1, %2, %N syntax.

Basic

```
RAWSQLAGG_INT("FLOOR(SUM(%1))",[Sales])
```

This example uses the SQL Server function `FLOOR()` to return the greatest integer value less than the sum of the `[Sales]` field.

Intermediate

```
RAWSQLAGG_INT("CEILING(STDEV(%1))",[Unit Price])
```

This example uses the SQL Server function `CEILING()` to return the smallest integer value greater than the standard deviation of the `[Unit Price]` field.

Advanced

```
RAWSQLAGG_INT("AVG(DATEDIFF(day, COALESCE(%2, %1), GETDATE()))",
[Order Date],[Ship Date])
```

This example returns the average difference in whole days between the current date time on the SQL Server and the first non-NULL value in `[Ship Date]` and `[Order Date]` returned by the `COALESCE()` function.

102. RAWSQLAGG_REAL()

The `RAWSQLAGG_REAL()` function is a pass-through function that allows the user to send an arbitrary expression to the underlying data source. The expression must return an aggregate value that Tableau can convert into a number. This expression will not be checked in any way by Tableau and may produce an error at the data source level. The user must respect the syntax conventions of the data source when constructing the expression. The generalized syntax for the function is as follows:

```
RAWSQLAGG_REAL("agg_expr",[arg1],...[argN])
```

The `agg_expr` in quotes is the expression to be passed through to the data source. N number of arguments can be specified in a comma-separated list as shown. These arguments are referenced in the expression with a %1, %2, and % `N` syntax.

Basic

```
RAWSQLAGG_REAL("SUM(%1)",[Profit])
```

This example returns the sum of the `[Profit]` values with the decimal component intact.

Intermediate

```
RAWSQLAGG_REAL("VAR(%1-%2)",[Product Base Margin],[Discount])
```

This example uses the SQL Server function `VAR()` to return the statistics variance of the `[Product Base Margin]` value minus the `[Discount]` value.

Advanced

```
RAWSQLAGG_REAL("ROUND(SUM(CASE WHEN %3='East' THEN %1*%2*0.85
WHEN %3='West' THEN %1*%2*1.15
ELSE %1*%2 END),4)",[Unit Price],[Order Quantity],[Region])
```

In the advanced example, the sum of a SQL Server `CASE` statement selectively multiplies the `[Unit Price]` value and the `[Order Quantity]` value by either 1.15 or 0.85 depending on what the `[Region]` value comes from. The formula then uses the `ROUND()` function to round the result to four decimal places.

103. RAWSQLAGG_STR()

The `RAWSQLAGG_STR()` function is a pass-through function that allows the user to send an arbitrary expression to the underlying data source. The expression must return an aggregate value that Tableau can convert into a string. This expression will not be checked in any way by Tableau and may produce an error at the data-source level. The user must respect the syntax conventions of the data source when constructing the expression. The generalized syntax for the function is as follows:

```
RAWSQLAGG_STR("agg_expr",[arg1],...[argN])
```

The `agg_expr` in quotes is the expression to be passed through to the data source. `N` number of arguments can be specified in a comma-separated list as shown. These arguments are referenced in the expression with a %1, %2, and % `N` syntax.

Basic

```
RAWSQLAGG_STR("MIN(%1)",[Order Date])
```

This example returns the minimum value of the `[Order Date]` field. It's worth noting that it returns it as a string value. The expression that the `RAWSQL` function is calling can return a different value than the function returns as long as it is convertible into the outer function return type.

Intermediate

```
RAWSQLAGG_STR("MAX(LEFT(%1, 3))",[City])
```

In this example, the function uses the SQL Server function `LEFT()` to get the left-most three characters from the `[City]` field and then returns the maximum value of that string.

Advanced

```
RAWSQLAGG_STR("CASE WHEN (SUM(%1)/SUM(%2)) > 0
THEN 'Compliant' ELSE 'Noncompliant' END",[Profit],[Sales])
```

This example calculates a profit ratio by dividing the sum of `[Profit]` by the sum of `[Sales]` and then returns the string `Compliant` if this value is greater than 0. It returns `Noncompliant` if the calculated value is less than 0.

104. REGEXP_EXTRACT

`REGEXP_EXTRACT` is a function that only works with text files, Hadoop Hive, Google BigQuery, PostgreSQL, Tableau Data Extracts, Microsoft Excel, Salesforce, and Oracle datas sources.

The function returns a string based on the given pattern. If the pattern returns more than one result, the function fails. For Tableau Data Extracts, the pattern must be a constant.

For information on regular expression syntax, see your data source's documentation. For Tableau extract files, syntax should conform to the International Components for Unicode (ICU) standard, an open source project of mature C/C++ and Java libraries for Unicode support, software internationalization, and software globalization. Search for the Regular Experessions page in the online ICU User Guide.

`REGEXP_EXTRACT(string,pattern)`

Basic

`REGEXP_EXTRACT('abc 123','[a-z]+\s+(\d+)')`

This example returns the portion of the literal string that matches the pattern definition `'123'`.

Intermediate

`REGEXP_EXTRACT('ABC20DEF','20(.*)')`

The intermediate example will return `'DEF'`.

105. REGEXP_EXTRACT_NTH

`REGEXP_EXTRACT_NTH` is a function that works only with text files, Hadoop Hive, Google BigQuery, PostgreSQL, Tableau Data Extracts, Microsoft Excel, Salesforce and Oracle datas sources.

The function returns the portion of the string that matches the regular expression pattern. The substring is matched to the nth capturing group, where n is the given index. If the index is 0, the entire string is returned. For Tableau Data Extracts, the pattern must be a constant.

For information on regular expression syntax, see your data source's documentation. For Tableau extract files, syntax should conform to the International Components for Unicode (ICU) standard, an open source project of mature C/C++ and Java libraries for Unicode support, software internationalization, and software globalization. Search for the Regular Experessions page in the online ICU User Guide.

`REGEXP_EXTRACT_NTH(string,pattern,index)`

Basic

`REGEXP_EXTRACT_NTH('abc 123','[a-z]+\s+(\d+)',1)`

This example returns the portion of the literal string that matches the pattern definition and index `'123'`.

106. REGEXP_MATCH

REGEXP_MATCH is a function that works only with text files, Hadoop Hive, Google BigQuery, PostgreSQL, Tableau Data Extracts, Microsoft Excel, Salesforce, and Oracle datas sources.

The function returns a Boolean value (true/false) if a substring of the specified string matches the regular expression pattern. For Tableau Data Extracts, the pattern must be a constant.

For information on regular expression syntax, see your data source's documentation. For Tableau extract files, syntax should conform to the International Components for Unicode (ICU) standard, an open source project of mature C/C++ and Java libraries for Unicode support, software internationalization, and software globalization. Search for the Regular Experessions page in the online ICU User Guide.

```
REGEXP_MATCH(string,pattern)
```

Basic

```
REGEXP_MATCH('ABC20DEF','[0-9]')
REGEXP_MATCH('ABCDEF','[0-9]')
```

This example compares the string with the pattern definition and if there is a match returns true. The first example returns TRUE. The second returns FALSE.

107. REGEXP_REPLACE

REGEXP_REPLACE is a function that works only with text files, Hadoop Hive, Google BigQuery, PostgreSQL, Tableau Data Extracts, Microsoft Excel, Salesforce, and Oracle datas sources.

The function replaces the characters within a string by using a matching pattern. For Tableau Data Extracts, the pattern must be a constant.

For information on regular expression syntax, see your data source's documentation. For Tableau extract files, syntax should conform to the International Components for Unicode (ICU) standard, an open source project of mature C/C++ and Java libraries for Unicode support, software internationalization, and software globalization. Search for the Regular Experessions page in the online ICU User Guide.

```
REGEXP_REPLACE(string,pattern,replacement)
```

Basic

```
REGEXP_REPLACE('ABC20DEF','[0-9]', '*')
```

This example scans for the string pattern specified in the formula and replaces the characters with the replacerment string. The basic formula example returns `ABC**DEF`.

Intermediate

```
REGEXP_REPLACE('abc 123','\s','-')
```

This example scans for the string pattern specified in the formula and replaces the characters with the replacerment string. This intermediate example returns `'abc-123'`.

108. REPLACE

The `REPLACE` function is an advanced function that allows specified data replacement within a string field. This does *not* change the data at the source level by using this function, but instead merely creates a new field that includes the replacement strings. The function searches a string field to find the stated substring. Once the substring is found, the replacement string replaces the substring data.

```
REPLACE(String,Substring,Replacement)
```

Basic

```
REPLACE("[Order Priority]","Not Specified","High")
```

The function example searches the `Order Priority` field to find any orders that have not got a specified priority within the database. The `REPLACE` function then replaces the `Not Specified` orders to a new status of `High`.

Intermediate

```
IIF([Order Date] < dateadd('month',-2,today()) ,
REPLACE([Order Priority],"Not Specified","High"),[Order Priority])
```

The function example is being used to determine the validity of each Order Priority assigned to every Order id. The calculation only investigates orders that are older than 2 months. The `REPLACE` stage then assigns a new `Order Priority` to all orders older than 2 months and that have a `Not Specified` priority. The function replaces the `Not Specified` with High. The `REPLACE` function is highly dependent on the source data. The `Replace` function is not available against all data sources. Using a data extract does allow a more robust solution when planning on using the `REPLACE` function.

109. RIGHT

`RIGHT` is a String function that returns the rightmost characters from its designated string. This function can be used to create new dimensions directly or combined to create advanced calculated fields. This has the same principles as the `LEFT` function.

```
RIGHT(String,Number)
```

Basic

```
RIGHT([Customer Zip Code],2)
```

The function example is an additional way of using U.S. zip Codes. The `RIGHT` function helps identify the area of the city or metropolitan area that is coded by the last two digits.

110. ROUND

The `ROUND` function rounds numbers to the number of digits as specified with the decimals argument within the function. The `decimals` argument specifies how many decimal points of precision to include in the final result, although it is not required. If the decimals variable is not included, then the number is rounded to the nearest integer. Tableau uses the following rounding rules:

- If the value of the number to the right of the rounding digit is less than five, the rounding digit is left unchanged.
- If the value of the number to the right of the rounding digit is five or higher, the rounding digit is raised by one.

```
ROUND(number, [decimals])
```

`Number` **= Any given number.** `Decimals` **= Any given number (not required)**

Basic

```
ROUND([Sales])
```

This function returns all sales values in the database to the nearest integer. If this function is used to round the sales for one year, then it will first round the results for every row in the database before the aggregation is applied.

Intermediate

```
ROUND(SUM([Profit])/SUM([Order Quantity]),2)
```

This function will first calculate the sum of profit divided by sum of order quantity and then round the result to two digits. If the sum was not used, and non-aggregate arguments were used instead, then the function would round all underlying rows in the database first before any result is aggregated, as defined by the dimension.

111. RTRIM

The `RTRIM` function removes trailing spaces that may be present within the data. This function, like the `LTRIM` function, can be used as a data cleansing function so that the data is consistent and set correctly.

```
RTRIM(String)
```

Basic

```
RTRIM("Ruby Young  ")
```

The output for this would simply be Ruby Young. Just like `LTRIM`, trailing spaces may cause an issue if using additional functions and trying to blend data from different sources.

112. RTRIM_THIS

This is a Google Big Query–specific function that removes all of the rightmost characters from the first string that match the second string. It is case sensitive.

```
RTRIM_THIS(string1, string2)
```

Basic

```
RTRIM_THIS('Remove me', ' me'
```

The output for this would simply be `Remove`. The function will remove the space in front of `me` and the string `me`.

113. RUNNING_AVG

This is a table calculation function that returns the running average of the provided expression from the first to the current row of the view/partition.

```
RUNNING_AVG(expression,[start],[end])
```

`Expression` **= Any valid aggregate calculation (for example:** `SUM([Sales])`**)**

Basic

```
RUNNING_AVG(SUM([Sales]))
```

This table calculation function returns the running average of the sum of sales within the window (or frame). Note: When the `[start]` and `[end]` arguments are omitted, the entire frame (window/pane) is used. If the values are incorrect or inconsistent, ensure that you are properly using the "Compute using" option of the table calculation.

Intermediate

```
RUNNING_AVG(SUM([Sales]), FIRST(),LAST())
```

This formula returns the running average of the sum of sales from the first row to the last row of the window (or pane). If the values are incorrect or inconsistent, ensure that you are properly using the "Compute using" option of the table calculation.

Advanced

```
IF INDEX()=1 THEN RUNNING_AVG( SUM([Sales]) ) ELSE NULL END
```

This function assumes there are a large number of marks on the view and/or that the user is dealing with big data. Through if/then logic, you can bypass the table scans, which cause noticeable performance degradation. In the end, the calculation takes the running average of the sum of sales for the specified window (or frame) and does not repeat the values.

114. RUNNING_COUNT

This table calculation function returns the running count of the provided expression from the first to the current row of the view/partition.

```
RUNNING_COUNT(expression,[start],[end])
```

`Expression` = **Any valid aggregate calculation (e.g.: `SUM([Sales])`)**

Basic

```
RUNNING_COUNT(SUM([Sales]))
```

This formula returns the running count of the sum of sales within the window (or frame). Note that when the `[start]` and `[end]` arguments are omitted, the entire frame is used. If the values are incorrect or inconsistent, ensure that you are properly using the "Compute using" option of the table calculation.

Intermediate

```
RUNNING_COUNT(SUM([Sales]), FIRST(),LAST())
```

This formula returns the running count of the sum of sales from the first row to the last row of the window (or frame). If the values are incorrect or inconsistent, ensure that you are properly using the "Compute using" option of the table calculation.

Advanced

```
IF INDEX()=1 THEN RUNNING_COUNT(SUM([Sales]),0, IIF(INDEX()=1,LAST(),0))
END
```

This formula assumes there are a large number of marks on the view or that the user is dealing with a large dataset. Using if/then logic, the formula can bypass the table scans, which can cause noticeable performance degradation. In the end, the calculation returns the running count of the sum of sales for the specified window (or frame). Bonus: Try making the window frames dynamic with a parameter.

115. RUNNING_MAX

This table calculation function returns the running maximum of the provided expression from the first to the current row of the view (partition).

```
RUNNING_MAX(expression,[start],[end])
```

`Expression` = **Any valid aggregate calculation (for example `SUM ([Sales])`)**

Basic

```
RUNNING_MAX( SUM([Sales]) )
```

This formula returns the running maximum of the sum of sales within the window (or pane). When the [start] and [end] arguments are omitted, the entire frame is used. If the values are incorrect or inconsistent, ensure that you are properly using the "Compute using" option of the table calculation.

Intermediate

```
RUNNING_MAX( SUM([Sales]) ), FIRST(),LAST())
```

This formula returns a running maximum of the sum of sales from the first row to the last row of the window (or frame). If the values are incorrect or inconsistent, ensure that you are properly using the "Compute using" option of the table calculation.

Advanced

```
IF INDEX()=1 THEN RUNNING_MAX( SUM([Sales]) ) ELSE NULL END
```

This formula assumes there are a large number of marks on the view or that the user is dealing with a large dataset. Through if/then logic, you can bypass the table scans, which cause noticeable performance degradation. The calculation returns the running maximum of the sum of sales for the specified window (or pane) and does not repeat the values.

116. RUNNING_MIN

This table calculation function returns the running minimum of the provided expression from the first to the current row of the view or partition.

`RUNNING_MIN(expression,[start],[end])`

`Expression` = **Any valid aggregate calculation (e.g.: `SUM ([Sales])`)**

Basic

`RUNNING_MIN(SUM([Sales]))`

This formula returns the running minimum of the sum of sales within the window (or frame). When the `[start]` and `[end]` arguments are omitted, the entire frame is used. If the values are incorrect or inconsistent, ensure that you are properly using the "Compute using" option of the table calculation.

Intermediate

`RUNNING_MIN(SUM([Sales]) ), FIRST(),LAST())`

This formula returns the running minimum of the sum of sales from the first row to the last row of the window (or frame). If the values are incorrect or inconsistent, ensure that you are properly using the "Compute using" option of the table calculation.

Advanced

`IF INDEX()=1 THEN RUNNING_MIN (SUM([Sales]))ELSE NULL END`

This formula assumes there are a large number of marks on the view or that the user is dealing with a large dataset. The use of if/then logic bypasses table scans, which cause noticeable performance degradation. In the end, the calculation returns the running minimum of the sum of sales for the specified window (or frame) and does not repeat the values.

117. RUNNING_SUM

This table calculation returns the running sum of the provided expression from the first to the current row of the view/partition.

`RUNNING_SUM(expression,[start],[end])`

`Expression` = **Any valid aggregate calculation (for example, `SUM ([Sales])`)**

Basic

```
RUNNING_SUM(SUM([Sales]))
```

This formula returns the running sum of the sum of sales within the window (or frame). Note that when the `[start]` and `[end]` arguments are omitted, the entire pane is used. If the values are incorrect or inconsistent, ensure that you are properly using the "Compute using" option of the table calculation.

Intermediate

```
RUNNING_SUM( SUM([Sales]) ), FIRST(),LAST())
```

This formula returns the running sum of the sum of sales from the first row to the last row of the window (or pane). Note that if the values are incorrect or inconsistent, ensure that you are properly using the "Compute using" option of the table calculation.

Advanced

```
IF INDEX()=1 THEN RUNNING_SUM ( SUM([Sales]) ) ELSE NULL END
```

This formula assumes there are a large number of marks on the view or that the user is dealing with a large dataset. The use of if/then logic bypasses the table scans, which can cause noticeable performance degradation. In the end, the calculation returns the running sum of the sum of sales for the specified window (or frame) and does not repeat the values.

Functions 118, 119, 120, and 121 are the core of R integration in Tableau. Each invokes the R console and can send data from Tableau to R for manipulations and calculations. These functions are all *table calculations*, so it is important to understand how table calculations work before using them.

Each function is named after the type of data Tableau expects to receive from the R console. For example, the `SCRIPT_REAL` function expects that R will return a real number. If it does not, the function will fail and Tableau will show an error. Thus it is important to think about the type of data you want R to return.

Before getting into the syntax, it is critical to know how Tableau passes data to and receives data from R. Because the functions are table calculations, Tableau sends vectors of data along the specified level of partition in the table calculation. As with other table calculations, the dimensions in the view determine the level of aggregation of each row, *unless* the Aggregate Measures option is turned off.

Once Tableau sends the data to R and the R script has been run, Tableau receives data back in vectors of the same length that it originally sent. This means that if the function sent ten rows of data to R, and the R script simply returns the number 2, Tableau will receive the number 2 back ten times.

The script functions have two parts: the R script, and arguments specifying the data that Tableau is to send to R. Consider this example:

```
SCRIPT_REAL(".arg1 + .arg2",SUM([Sales]),SUM([PROFIT]))
```

In the R code, the value ".arg#" is used to represent corresponding data values in Tableau that are being passed to the R console. In this example, .arg1 represents SUM([Sales] in the R code, while .arg2 represents SUM([PROFIT]). For a data value to be passed to R through this function, *it must be aggregated even if the level of detail at which you are passing data to R is not*. The .arg# values can also represent parameters thereby creating dynamic R scripts.

118. SCRIPT_BOOL

```
SCRIPT_BOOL ("insert R code here", .arg1, .arg2, ... .argN)
```

119. SCRIPT_INT

```
SCRIPT_INT("insert R code here", .arg1, .arg2, ... .argN)
```

120. SCRIPT_REAL

```
SCRIPT_REAL ("insert R code here", .arg1, .arg2, ... .argN)
```

121. SCRIPT_STRING

```
SCRIPT_STRING ("insert R code here", .arg1, .arg2, ... .argN)
```

Advanced

```
SCRIPT_REAL("df <- data.frame(tire_size = .arg1, mpg = .arg2);
fit <- lm(mpg ~ tire_size, data = df);
scores <- predict(fit, df);
scores", SUM ([Tire Size]), SUM ([MPG])
)
```

This advanced function uses the fields Tire Size and MPG, then computes a simple linear regression model with Tire Size as the independent variable and MPG as the dependent variable. This would allow a user to plot a line of best fit. Note that using data frames and renaming your data can help make it easier to use R integration. This way, as with the preceding example, you do not have to continually use .arg1, and can instead just use tire_size.

122. SIGN

The SIGN function is used to highlight whether the value of the result is positive, negative, or equal to zero. The returned values are -1 if the number is negative, 0 if the number is zero, or 1 if the number is positive.

```
SIGN(number)
```

Number = Any given number

Basic

```
SIGN(-21)
```

The result of this function is `-1` because the expression `-21` is a negative value.

Intermediate

```
IF SIGN(SUM([Profit]))=1
THEN "Profit"
ELSEIF SIGN(SUM([Profit]))=-1
THEN "Loss"
ELSE "Break-Even"
END
```

The `SIGN` function is helpful in combination with a logical function. In this example, the logical `IF` function is used, which will return a string to indicate the level of profitability.

123. SIN

The `SIN` function returns the sine of a given number specified in radians.

```
SIN(number)
```

`Number` = Any given number, where the number is in radians

Basic

```
SIN(PI()/4)
```

This function calculates the sine of π/4 radians. This function returns 0.707106781186547 radians.

Intermediate

```
SIN(RADIANS(90))
```

In this function, the number is known in degrees—in this case `90`. First the degrees are converted to radians before the sine is calculated. The result is 1.

124. SIZE()

This table calculation function returns the total number of rows in the view/ partition.

```
SIZE()
```

This function does not require any arguments.

Basic

```
SIZE()
```

The function returns the number of rows in the view/partition.

Intermediate

```
WINDOW_SUM(SUM([Sales]))/SIZE()
```

This formula first takes the sum of sales for the view or partition and then divides that result by the total number of rows within the partition.

Advanced

```
IF INDEX()=1 THEN WINDOW_SUM(SUM([Sales]))/SIZE() ELSE NULL END
```

This formula assumes there are a large amount of marks on the view or that the user is dealing with a large dataset. Through the use of if/then logic, table scans can be bypassed that can cause noticeable performance degradation. In the end, the calculation takes the sum of sales for the view/partition and then divides by the total rows in the partition. Bonus: Try making the window frames dynamic with a parameter.

125. SPACE

The `SPACE` function is a simple function allowing the user to create a string of spaces that can then be used within other calculations.

```
SPACE(number)
```

Basic

```
SPACE(4)
```

The function example simply creates a string of four blank spaces. This string of spaces can now been used in other calculated fields.

Intermediate

```
[Customer]+SPACE(2)+[City]+SPACE(2)+[Zip Code]
```

The function example output would look like: Andrew Roberts Fresno 93727. Without the `SPACE` values, the output would look like this: Andrew RobertsFresno93727.

126. SPLIT

The SPLIT function cuts a string into multiple sections based on a user-specified delimiter or a character used to indicate the point of separation. A token number, which is assigned positive values if counting from the beginning of the string or negative values if counting backwards from the end of the string, must also be provided to indicate which piece of the string to include as the final result.

```
SPLIT(string, delimiter, token number)
```

Basic

```
SPLIT("Hi, how are you?", " ", -2)
```

This function results in the following output: are. It searches through the string and identifies each set of characters existing between the specified delimiter as a substring with an associated index number. In this case, the delimiter is a single space, which is included as the second argument in the function. The index specified by the user is -2, or the second set of characters encountered starting from the end of the string. Thus the output is the second set of characters counting backwards from the end of the given string.

Intermediate

```
INT(TRIM(SPLIT([Product ID], "-", -1)))
```

This function extracts a portion of a product ID, which was originally stored as a string, removes leading and trailing spaces that may be present following the split function, and finally converts the resulting characters from a set of strings to an integer. In this case, the delimiter is a hyphen and the desired portion of the original product ID string is the first set of characters counting backwards from the end of the string.

127. SQRT

The SQRT function is the inverse of the SQUARE function. It returns the square root of a number. It gives the same return when using the POWER function when raising the number to the power of 0.5.

```
SQRT(number)
```

Number = **Any given number greater than zero. If the number is less than or equal to zero, the SQRT function returns a NULL.**

Basic

```
SQRT(49)
```

The result of this function is 7. The result can be achieved with the `POWER` function. In that case, the function is `POWER` (49,0.5), where `49` is raised to the power of 0.5. The ^ symbol can also be used in the calculation instead. The calculation 49^0.5 also returns the same result in this case.

128. SQUARE

The `SQUARE` function returns the square of the number. In other words, it multiplies the expression by itself. It gives the same return when using `POWER` function when raising the number to the power of two.

```
SQUARE(number)
```

`Number` = **Any given number**

Basic

```
SQUARE(7)
```

This function multiplies 7 by itself, so 7*7 = 49. Similar to the `POWER` function, it raises `7` to the power of 2. The ^ symbol can also be used in the calculation instead. The calculation 7^2 returns the same result in this case.

129. STARTSWITH

The `STARTSWITH` function is similar in its approach to the `CONTAINS` function, but it has limits on the way it searches the string. Whereas the `CONTAINS` function searches the full length of the string for the specified substring, the `STARTSWITH` function only searches the very beginning of the string.

```
STARTSWITH(String, Substring)
```

BASIC

```
STARTSWITH([City],"New")
```

The function example searches the first three characters in the `CITY` string and will only return a `True` value if these characters match the substring and are in order. Examples would be New York, New Orleans, New Jersey.

130. STDEV

This function returns an estimate of the population standard deviation based on a sample of data from the populaton. It uses `N-1` in the denominator to adjust for bias retlated to small sample size.

Basic

```
STDEV([Sales])
```

This formula will return the standard deviation of sales. One possible use would be to use the result for reference lines on a time series chart.

Intermediate

```
AVG([Sales])+STDEV([Sales])
```

This example shows how you can combine the standard deviation calculation with the average calculation to set limits based on standard deviations.

Advanced

```
AVG([Sales])+(([Number of deviations])*STDEV([Sales]))
```

This formula uses a parameter to change the number of standard deviations from the mean value you wish to calculate. The number of deviations could be limited to the values (1, 2, and 3). 95 percent confidence limits would use + and – 1.96 standard deviations.

131. STDEVP

This function returns the statistical standard deviation of the expression without adjusting for small sample bias. Use `STDEVP` if the expression includes the entire population, even if there are a small number of values.

Basic

```
STDEVP([Sales])
```

This formula will return the standard deviation of Sales.

Intermediate

```
AVG([Sales]) + STDEVP([Sales])
```

This example shows how you can combine the standard deviation function with the average function to set limits based on standard deviations..

Advanced

```
AVG([Sales])+ (([Number of deviations])*STDEV([Sales]))
```

This formula uses a parameter to change the number of standard deviations from the mean value you wish to calculate. The number of deviations could be limited to the values (1, 2, and 3). 95 percent confidence limits would use + and – 1.96 standard deviations.

132. STR

Returns a string for a given expression.

Basic

```
STR(5.0)
```

This formula changes a number into a string value of `5.0`.

Intermediate

```
"Total Products = " + STR([Qty])
```

This formula returns a string. In this example, a custom label is being created using a number field `[Qty]`. If `[Qty]` has a value of 10, then this function returns `Total Products = 10`.

Advanced

```
STR([StartDate]) + ' to ' + STR([EndDate])
```

This formula returns a string describing the period (start to end). This is useful when you'd like to have an annotation, ToolTip, or other label that describes the period of analysis in a custom formatted way. If `[StartDate]`) is equal to January 1, 2013 and `[EndDate]` is equal to March 1, 2013 then the function will return January 2013 to March 2013.

133. SUM

This `SUM` function returns the sum of all the values in the expression. `SUM` can be used with numeric fields only. NULL values are ignored.

Basic

```
SUM([Sales])
```

This formula returns the total sales value for whatever dimension is being displayed in the view.

Intermediate

```
SUM([Sales])*[Commission Rate]
```

This example shows how to use the `SUM` of sales and multiply it by a parameter (variable) value to derive a new amount that can be dynamically changed by the parameter control. This example gives us a total amount of commissions for the level of detail displayed in the view.

Advanced

```
SUM([Sales]) / COUNTD([Customer ID])
```

This formula derives the average sales per unique customer.

134. TAN

The `TAN` function returns the tangent of a given number specified in radians.

```
TAN(number)
```

`Number` = Any given number, where the number is in radians

Basic

```
TAN(PI()/4)
```

This function calculates the tangent of π/4 radians. This function returns 1 radian.

Intermediate

```
TAN(RADIANS(45))
```

In this function, the number is known in degrees—in this case 45. First, the degrees are converted to radians before the sine is calculated. The result is also 1.

135. THEN

The `THEN` qualifier is used in logical expressions to transition from evaluating whether or not an expression is true to triggering a resulting action. It is an essential element of `IF/ELSEIF/ELSE/THEN` statements and `CASE` statements.

Basic

```
IF [Profit] >= 0 THEN "Profitable" ELSE "Unprofitable" END
```

This equation will assign the tag `"Profitable"` to any values where profits are greather than or equal to zero, and the tag `"Unprofitable"` to any values to any sub-zero profits. It is the `THEN` statement that initiates the action following the evaluation of the mathematical expression. This `IF` statement cannot function without an associated `THEN` statement.

136. TIMESTAMP_TO_USEC

A Google Big Query–specific function that converts a timestamp data type to a Unix timestamp in microseconds. Often when working with dates you will need to convert to a datetime first.

```
TIMESTAMP_TO_USEC(expression)
```

Basic

```
TIMESTAMP_TO_USEC(#2012-10-01 01:02:03#)
```

The example converts the literal datetime to a Unix timestamp in microseconds. The answer is 1349053323000000.

137. TLD

A Google Big Query–specific function that returns a top-level domain of a URL string. The URL must include protocol to work.

```
TLD(string_url)
```

Basic

```
TLD("https://www.twitter.com/DGM885")
```

The example returns the top-level domain `.com`.

138. TODAY

The `TODAY` function returns the current date. This is similar to `NOW()` but does not include the time component.

Basic

```
TODAY()
```

This function works as follows: If today's date is March 12, 2013 at 3:04 p.m. the function will return the date March 12, 2013.

Intermediate

```
DATEADD('day', -30, TODAY())
```

This formula returns a date 30 days before today's date. If today is March 18, 2013 then this function will return February 16, 2013. This can be very useful in filtering for specific periods (like 30 rolling days) or highlighting a period on a timeline where data may be uncertain.

139. TOTAL

`TOTAL(Expression)`

`Expression` = Any valid aggregate calculation (for example, `SUM ([Sales])`)

Basic

`TOTAL(SUM([Sales]))`

This formula returns the total across all rows in the database for the window (or pane). It is the same concept as the `SUM(Expression)` function but applies it selectively within the window (or pane).

Intermediate

`SUM([Sales])/TOTAL(SUM([Sales]))`

This formula returns the sum of sales and divides it by the total across all rows in the database to get a percent of total. If the values are incorrect or inconsistent, ensure that you are properly using the "Compute using" option of the table calculation.

Advanced

```
WINDOW_MAX(SUM([Sales])/TOTAL(SUM([Sales])))=(SUM([Sales])/TOTAL(SUM([Sa
les])))
```

This formula highlights the *maximum* percent of total value within partition when it equals the percent of total portion of the partition. If the values are incorrect or inconsistent, ensure that you are properly using the "Compute using" option of the table calculation.

140. TRIM

The `TRIM` function encompasses the logic of both `LTRIM` and `RTRIM` into one function.

`TRIM(String)`

Basic

`TRIM("  Gemma Palmer     ")`

The function example output for this would simply be Gemma Palmer. Best practice would be to use the `TRIM` function if you have any concerns about leading or trailing spaces in imported data.

141. UPPER

This function allows the user to change all characters to uppercase within a string. The `UPPER` function will only change the lowercased characters that exist in a string and thus ignore all uppercase characters that already exist.

```
UPPER(String)
```

Basic

```
UPPER("BatMan")
```

The function example retursn all uppercase letters in the string: `BATMAN`.

142. USEC_TO_TIMESTAMP

A Google Big Query–specific function that converts a Unix timestamp in microseconds to a `TIMESTAMP` data type.

```
USEC_TO_TIMESTAMP(expression)
```

Basic

```
TIMESTAMP_TO_USEC(1349053323000000)
```

The example converts the Unix timestamp in microseconds to a `DATETIME` literal. The answer #2012-10-01 01:02:03#.

143. USERDOMAIN

This function returns the domain of the person currently logged into Tableau Server. If the user is not logged on to Server, the function returns the Windows domain. This function can be used in conjunction with other user functions when you desire to create security based on username and domain.

Refer to the assumptions in the USERNAME() section that follows for the user and domain data.

Basic

```
USERDOMAIN()
```

If a company had two subsidiaries, Retail and Wholesale, which had separate domains (`RETAIL.local` and `WSALE.local`) then it would return the domain name for the logged-on user.

Intermediate

```
CASE USERDOMAIN()
WHEN 'RETAIL' THEN 'Access Granted'
WHEN 'WSALE  THEN 'Access Denied'
END
```

This formula would return either `Access Granted` or `Access Denied` at row level and could be used to drive row-level security across two separate domains.

Advanced

```
IF USERDOMAIN() = 'WSALE' THEN
IF ISMEMBEROF('Report Viewer')Then
'Access Granted'
ELSE
'Access Denied'
END
ELSEIF USERDOMAIN() = 'RETAIL' THEN
IF ISMEMBEROF('Management') THEN
'Access Granted'
ELSE
IF FULLNAME() = [Sales Person] THEN
'Access Granted'
ELSE
'Access Denied'
END
END
ELSE
'Access Denied'
END
```

This formula returns either an `Access Denied` or an `Access Granted` at row level, which could be used in a filter to apply row-level security. The statement is comparing domains, groups, and users before assigning permissions to access each row of data.

144. USERNAME()

The following assumptions used for the examples:

User 1

- Full name: Malcolm Reynolds
- Active Directory name: DOMAIN\m.reynolds

User 2

- Full name: River Tam
- Active Directory name: DOMAIN\r.tam

User 3

- Full name: Jayne Cobb
- Active Directory name: DOMAIN\j.cobb

`USERNAME()` returns the username of the user logged onto the server. If the user Malcolm was logged onto the server, then `USERNAME()` would return `m.reynolds`.

`Expression` = Any valid discrete argument

Basic

```
USERNAME()='m.reynolds'
```

This example returns a Boolean (true/false) value of true if Malcolm Reynolds is logged-on the system, and false if he is not.

Intermediate

```
USERNAME()=[MANAGER]
```

A row-level security argument that compares the `USERNAME()` function result with the `[MANAGER]` field in the dataset. This would be useful for data in which users are only permitted to view their own data.

Advanced

```
IF ISMEMBEROF('Management')then 'Access Permitted'
ELSEIF USERNAME()=[Manager]then 'Access Permitted'
ELSE 'Access Denied' END
```

This formula returns either `Access Permitted` or `Access Denied` at row level. When applied as a filter to show only `Access Permitted` rows, if the user is a member of the group management, then they can see all rows of data unless the user can only see lines of data, which are tagged with their username as the manager.

145. VAR

This function returns the statistical variance of the expression without adjusting for small sample bias. Use STDEV if the expression includes the entire population, even if there are a small number of values.

Basic

```
VAR(expression)
```

This function returns statistical variance of the expression—a measure related to the spread of values.

146. VARP

This aggregate function returns the statistical variance of the values in the given expression based on a biased sample of the population. Variance is a measure of dispersion and is calculated using the average of the squared deviations from the mean. Thinking about statistical variance, this function seems like a weigh-station on the journey to arriving at standard deviation—a more commonly used dispersion measure—that is the square root of variance. In normally distributed sets of data, standard deviation implies specific value ranges that are useful for plotting control charts. Variance by itself seems to have less practical use cases. If you have one, please share it.

```
VAR(expression)
```

This function returns the statistical variance of the expression based on the entire population.

147. WHEN

The `WHEN` qualifier is used in conjunction with a `CASE` statement and identifies specific scenarios, also known as cases, that the `CASE` structure will encounter and interact with. The `WHEN` statement points to each individual scenario and specifies the appropriate action to take for each potential case.

Basic

```
CASE MONTH([Order Date])
WHEN 1 THEN "Manager A"
WHEN 2 THEN "Manager B"
WHEN 3 THEN "Manager C"
WHEN 4 THEN "Manager A"
WHEN 5 THEN "Manager B"
WHEN 6 THEN "Manager C"
WHEN 7 THEN "Manager A"
WHEN 8 THEN "Manager B"
WHEN 9 THEN "Manager C"
WHEN 10 THEN "Manager A"
WHEN 11 THEN "Manager B"
WHEN 12 THEN "Manager C"
END
```

The preceding `CASE` statement assigns each month value to one of three different managers: A, B, or C. In this example, we assume that there exists a situation where three different managers are responsible for some performance metric depending on the month of the year. In this instance, the `CASE` statement allows for assigning specific months to specific managers.

Intermediate

```
CASE [Performance Metric]
WHEN "Sum of Sales" THEN SUM([Sales])
WHEN "Sum of Profit" THEN SUM([Profit])
WHEN "Quantity Sold" THEN SUM([Quantity])
WHEN "Average Shipping Cost" THEN AVG([Shipping Cost])
WHEN "Average Discount" THEN AVG([Discount])
END
```

This `CASE` statement interacts with a parameter instead of a field. Depending on the selection made by the user, the `CASE` statement identifies which performance metric will be passed as the output of this function. If used in a view with a chart where the Columns shelf holds a date and the Rows shelf holds the calculated field containing this `CASE` statement, changing the parameter selection within the view will result in a chart showing the specified performance metric over time.

148. WINDOW_AVG

This function returns the average for a given expression over a window (or pane) specified. Note that performance is affected with an increase in marks; if the dataset is large, the advanced method will offer better performance and scalability.

```
WINDOW_AVG(expression,[start],[end])
```

`Expression` = Any valid aggregate calculation. For example: `SUM ([Sales]))`. `[start]` = Start of window (not required), `[end]` = End of window (not required).

Basic

```
WINDOW_AVG(SUM([Sales]))
```

This formula returns the average of the sum of sales within the window (or pane). Note that when the `[start]` and `[end]` arguments are omitted, the entire frame is used. If the values are incorrect or inconsistent, ensure that you are properly using the "Compute using" option of the table calculation.

Intermediate

```
WINDOW_AVG(SUM([Sales]),FIRST,()LAST())
```

This formula returns the average of the sum of sales from the first row to the last row of the window (or pane). This is the default used when `[start]` and `[end]` are not specified. Note that if the values are incorrect or inconsistent, ensure that you are properly using the "Compute using" option of the table calculation.

Advanced

```
IF INDEX()=1 THEN WINDOW_AVG(SUM([Sales]),0,IIF(INDEX()=1,LAST(),0))END
```

This formula assumes a large number of marks are on the view or that the user is dealing with a big dataset. Through the use of if/then logic, table scans can be bypassed, which can cause noticeable performance degradation. In the end, the calculation takes the average of the sum of sales for the specified window (or pane).

149. WINDOW_COUNT

This table calculation function returns the count for a given expression with a window (or pane) the user specifies. Note that performance is affected with an increase in marks; if the dataset is large, the advanced method will offer better performance and scalability.

```
WINDOW_COUNT(expression,[start],[end])
```

`Expression` **= Any valid aggregate calculation (for example,** `SUM ([Sales]))`**,** `[start]` **= start of window [not required];** `[end]` **= end of window [not required]).**

Basic

```
WINDOW_COUNT(SUM([Sales]))
```

This formula returns the count of the sum of sales within the window (or pane). Note that when the `[start]` and `[end]` arguments are omitted, the entire frame is used. If the values are incorrect or inconsistent, ensure that you are properly using the "Compute using" option of the table calculation.

Intermediate

```
WINDOW_COUNT(SUM([Sales]),FIRST(),LAST())
```

This formula returns the count of the sum of sales from the first row to the last row of the window (or pane). Note that if the values are incorrect or inconsistent, ensure that you are properly using the "Compute using" option of the table calculation.

Advanced

```
IF INDEX()=1 THEN WINDOW_COUNT(SUM([Sales]),0, IIF(INDEX()=1,LAST(),0)) END
```

This formula assumes a large number of marks are on the view or that the user is dealing with a big dataset. Through the use of if/then logic, table scans can be bypassed that can cause noticeable performance degradation. In the end, the calculation takes the count of the sum of sales for the specified window (or frame). Bonus: Try making the window frames dynamic with a parameter.

150. WINDOW_MAX

This table calculation function returns the maximum value for a given expression within the window (or pane) specified. Note: Performance is affected with an increase in marks; if the dataset is large, the advanced method will offer better performance and scalability.

```
WINDOW_MAX(expression,[start],[end])
```

`Expression` **= Any valid aggregate calculation (for example:** `SUM ([Sales]))`, `[start]` **= start of window not required];** `[end]` **= end of window [not required]).**

Basic

```
WINDOW_MAX(SUM([Sales]))
```

This formula returns the maximum of the sum of sales within the window (or pane). Note that when the `[start]` **and** `[end]` **arguments are omitted, the entire frame is used. If the values are incorrect or inconsistent, ensure that you are properly using the "Compute using" option of the table calculation.**

Intermediate

```
WINDOW_MAX(SUM([Sales]),FIRST(),LAST())
```

This formula returns the maximum of the sum of sales from the first row to the last row of the window (or pane). `First()` **and** `Last()` **are the defaults that makes this is the same as the Basic example. If the values are incorrect or inconsistent, ensure that you are properly using the "Compute using" option of the table calculation.**

Advanced

```
IF MAX([Ship Date]) = WINDOW_MAX( MAX([Ship Date]))
THEN SUM([Sales]) ELSE NULL END
```

This function assumes a large number of marks are in the view or that the user is dealing with a big dataset. Through the use of if/then logic, table scans can be bypassed that can cause noticeable performance degradation. In the end, the calculation takes the maximum of the sum of sales for the specified window (or frame) and does not repeat the values.

151. WINDOW_MEDIAN

This function returns the median for a given expression within a window (or pane) specified by the user. Performance is affected with an increase in marks; if the dataset is large, the advanced method will offer better performance and scalability.

```
WINDOW_MEDIAN(expression,[start],[end])
```

`Expression` = Any valid aggregate calculation (for example: `SUM ([Sales]))`, `[start]` = start of window [not required]; `[end]` = end of window [not required]).

Basic

```
WINDOW_MEDIAN(SUM([Sales]))
```

This formula returns the median of the sum of sales within the window (or pane). Note that when the `[start]` and `[end]` arguments are omitted, the entire frame is used. If the values are incorrect or inconsistent, ensure that you are properly using the "Compute using" option of the table calculation.

Intermediate

```
WINDOW_MEDIAN(SUM([Sales]),FIRST(), LAST())
```

This formula returns the median of the sum of sales from the first row to the last row of the window (or pane). If the values are incorrect or inconsistent, ensure that you are properly using the "Compute using" option of the table calculation.

Advanced

```
IF INDEX()=1 THEN WINDOW_MEDIAN(SUM([Sales]),0, IIF(INDEX()=1,LAST(),0)) END
```

This formula assumes there are a large number of marks on the view or that the user is dealing with a big dataset. Through the use of if/then logic, table scans can be bypassed that can cause noticeable performance degradation. In the end, the calculation takes the median of the sum of sales for the specified window (or frame). Bonus: Try making the window frames dynamic with a parameter.

152. WINDOW_MIN

This table calculation function returns the minimum value for a given expression within a window (or pane) the user specifies. Performance is affected with an increase in marks; if the dataset is large, the advanced method will offer better performance and scalability.

```
WINDOW_MIN(expression,[start],[end])
```

`Expression` = **Any valid aggregate calculation (for example,** `SUM ([Sales]))`, `[start]` **= start of window [not required];** `[end]` **= end of window [not required]).**

Basic

```
WINDOW_MIN(SUM([Sales]))
```

This formula returns the minimum of the sum of sales within the window (or pane). Note that when the `[start]` **and** `[end]` **arguments are omitted, the entire frame is used. If the values are incorrect or inconsistent, ensure that you are properly using the "Compute using" option of the table calculation.**

Intermediate

```
WINDOW_MIN(SUM([Sales]), FIRST(),LAST())
```

This formula returns the minimum of the sum of sales from the first row to the last row of the window (or pane). If the values are incorrect or inconsistent, ensure that you are properly using the "Compute using" option of the table calculation.

Advanced

```
IF INDEX()=1 THEN WINDOW_MIN( SUM([Sales])) ELSE NULL END
```

This formula assumes there are a large number of marks in the view or that the user is dealing with a big dataset. Through the use of if/then logic, table scans can be bypassed, which reduces potentially noticeable performance degradation. In the end, the calculation takes the minimum of the sum of sales for the specified window (or frame) and does not repeat the values.

153. WINDOW_PERCENTILE

This table calculation function returns values corresponding to the specified percentile within the window. The window is defined by means of offsets from the current row. Use `FIRST()+n` and `LAST()-n` for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

```
WINDOW_PERCENTILE(expression,number,[start],[end])
```

`Expression` = **Any valid aggregate calculation (for example,** `SUM([Sales]))`**, number = specified percentile,** `[start]` **= start of window [not required],** `[end]` **= end of window [not required]).**

Basic

```
WINDOW_PERCENTILE(SUM([Profit]),0.95,-2,0))
```

This formula returns the 95th percentile for the sum of profit from the two previous rows to the current row. The offsets are not required if you want to return the entire partition.

154. WINDOW_STDEV

This table calculation function will return the unbiased estimate of the population standard deviation based on a random sample of data in the expression. Performance is affected with an increase in marks; if the dataset is large, the advanced method will have better performance and scalability.

```
WINDOW_STDEV(expression,[start],[end])
```

`Expression` **= Any valid aggregate calculation (for example,** `SUM ([Sales]))`**, [start] = start of window [not required];** `[end]` **= end of window [not required]).**

Basic

```
WINDOW_STDEV(SUM([Sales]))
```

This formula returns the sample standard deviation of the sum of sales within the window (or pane). Note that when the `[start]` **and** `[end]` **arguments are omitted, the entire frame is used. If the values are incorrect or inconsistent, ensure that you are properly using the "Compute using" option of the table calculation.**

Intermediate

```
WINDOW_STDEV(SUM([Sales]),FIRST(),LAST())
```

This formula returns the sample standard deviation of the sum of sales from the first row to the last row of the window (or pane). If the values are incorrect or inconsistent, ensure that you are properly using the "Compute using" option of the table calculation.

Advanced

```
IF INDEX()=1 THEN WINDOW_ STDEV(SUM([Sales]),0, IIF(INDEX()=1,LAST(),0)) END
```

This formula assumes that a large number of marks are in the view or that the user is dealing with a big dataset. Through the use of if/then logic, table scans, which cause noticeable performance degradation, can be bypassed. In the end, the calculation takes the sample standard deviation of the sum of sales for the specified window (or frame). Bonus: Try making the window frames dynamic with a parameter.

155. WINDOW_STDEVP

This function will return the standard deviation of a given expression over a window (or pane) for which the user specifies. Performance is affected with an increase in marks; if the dataset is large, the advanced method will offer better performance and scalability.

```
WINDOW_STDEVP(expression,[start],[end])
```

`Expression` = **Any valid aggregate calculation (for example,** `SUM ([Sales]))`, `[start]` **= start of window [not required];** `[end]` **= end of window [not required]).**

Basic

```
WINDOW_STDEVP(SUM([Sales]))
```

This formula returns the biased standard deviation of the sum of sales within the window (or pane). Note that when the `[start]` **and** `[end]` **arguments are omitted, the entire frame is used. If the values are incorrect or inconsistent, ensure that you are properly using the "Compute using" option of the table calculation.**

Intermediate

```
WINDOW_STDEVP(SUM([Sales]), FIRST(),LAST())
```

This formula returns the biased standard deviation of the sum of sales from the first row to the last row of the window (or pane). If the values are incorrect or inconsistent, ensure that you are properly using the "Compute using" option of the table calculation.

156. WINDOW_SUM

This table calculation function will return the sum for a given expression over a window (or pane) for which the user specifies. Note that performance is affected with an increase in marks; if the dataset is large, the advanced method will offer better performance and scalability.

```
WINDOW_SUM(expression,[start],[end])
```

`Expression` = **Any valid aggregate calculation (for example,** `SUM ([Sales]))`, `[start]` **= start of window [not required];** `[end]` **= end of window [not required]).**

Basic

```
WINDOW_SUM(SUM([Sales]))
```

This formula returns the sum of sales within the window (or pane). Note that when the `[start]` **and** `[end]` **arguments are omitted, the entire frame is used. If the values are incorrect or inconsistent, ensure that you are properly using the "Compute using" option of the table calculation.**

Intermediate

```
WINDOW_SUM(SUM([Sales]) ),FIRST(),LAST)()
```

This formula returns the sum of sales from the first row to the last row of the window (or pane). If the values are incorrect or inconsistent, ensure that you are properly using the "Compute using" option of the table calculation.

Advanced

```
IF INDEX()=1 THEN WINDOW_SUM( SUM([Sales]) ) ELSE NULL END
```

This formula assumes there are a large number of marks in the view or that the user is dealing with a big dataset. Through the use of if/then logic, table scans, which cause noticeable performance degradation, can be bypassed. In the end, the calculation takes the sum of sales for the specified window (or frame) and does not repeat the values.

Advanced

```
IF INDEX()=1 THEN WINDOW_VAR(SUM([Sales]),0, IIF(INDEX()=1,LAST(),0)) END
```

This function assumes there are a large number of marks in the view or that the user is dealing with a big dataset. Through the use of if/then logic, table scans, which cause noticeable performance degradation, can be bypassed. In the end, the calculation takes the sample variance of the sum of sales for the specified window (or frame). Bonus: Try making the window frames dynamic with a parameter.

157. WINDOW_VAR

This table calculation function will return the unbiased estimate of the population variance of a given expression over a window (or pane) for which the user specifies. Performance is affected with an increase in marks; if the dataset is large, the advanced method will offer better performance and scalability.

```
WINDOW_VAR(expression,[start],[end])
```

`Expression` **= Any valid aggregate calculation (for example,** `SUM ([Sales]))`**,** `[start]` **= start of window [not required];** `[end]` **= end of window [not required]).**

Basic

```
WINDOW_VAR(SUM([Sales]))
```

This formula returns the sample variance for the sum of sales within the window (or frame). Note that when the `[start]` **and** `[end]` **arguments are omitted, the entire frame is used. If the values are incorrect or inconsistent, ensure that you are properly using the "Compute using" option of the table calculation.**

Intermediate

```
WINDOW_VAR(SUM([Sales]), FIRST(),LAST())
```

This formula returns the sample variance of the sum of sales from the first row to the last row of the window (or frame). If the values are incorrect or inconsistent, ensure that you are properly using the "Compute using" option of the table calculation.

158. WINDOW_VARP

This table calculation function returns the variance of a given population in the expression. Performance is affected with an increase in marks; if the dataset is large, using the advanced method will have better performance and scalability.

```
WINDOW_VARP(expression,[start],[end])
```

Expression = any valid aggregate calculation (for example, `SUM ([Sales]))`, `[start]` = start of window [not required], `[end]` = end of window [not required]).

Basic

```
WINDOW_VARP(SUM([Sales]))
```

This formula returns the variance for the sum of sales within the window (or frame). Note that when the `[start]` and `[end]` arguments are omitted, the entire frame is used. If the values are incorrect or inconsistent, ensure that you are properly using the "Compute using" option of the table calculation.

Intermediate

```
WINDOW_VARP(SUM([Sales]),FIRST(),LAST())
```

This formula returns the variance of the sum of sales from the first row to the last row of the window (or pane). If the values are incorrect or inconsistent, ensure that you are properly using the "Compute using" option of the table calculation.

Advanced

```
IF INDEX()=1 THEN WINDOW_VARP(SUM([Sales]),0,
IIF(INDEX()=1,LAST(),0)) END
```

This formula assumes there are a large number of marks in the view or that the user is dealing with a big dataset. Through the use of if/then logic, table scans, which cause noticeable performance degradation, can be bypassed. In the end, the calculation takes the biased variance of the sum of sales for the specified window (or frame). Bonus: Try making the window frames dynamic with a parameter.

Functions 159 through 165 are for the core of Tableau's Hadoop Hive functions. See your Hadoop vendor's user manual and Tableau's website for code samples.

159. XPATH_BOOLEAN

```
XPATH_BOOLEAN (XML string, XPATH expression string)
```

160. XPATH_DOUBLE

```
XPATH_DOUBLE (XML string, XPATH expression string)
```

161. XPATH_FLOAT

```
XPATH_FLOAT (XML string, XPATH expression string)
```

162. XPATH_INT

```
XPATH_INT (XML string, XPATH expression string)
```

163. XPATH_LONG

```
XPATH_LONG (XML string, XPATH expression string)
```

164. XPATH_SHORT

```
XPATH_SHORT (XML string, XPATH expression string)
```

165. XPATH_STRING

```
XPATH_STRING (XML string, XPATH expression string)
```

166. YEAR

This date function returns an integer representing the year of any given date. This is a shortened form of `DATEPART` ('year', [Date]).

```
YEAR(Date)
```

`Date` is the time period from which the year is extracted.

Basic

```
YEAR(#March 14, 2013#)
```

The preceding formula returns 2013.

Intermediate

```
YEAR(DATEADD('day', [Date], 5 ))
```

This formula returns the day that [Date] + 5 days lands on. If [Date] is March 14, 2013then this function will return 2013.

Advanced

```
CASE [Parameter].[Date Unit]
WHEN 'Day' THEN DAY([Date])
WHEN 'Month' THEN MONTH([Date])
WHEN 'Year' THEN Year([Date])
END
```

This formula returns an integer. A parameter, [Date Unit], is also used to allow the user to control the date type that will be used to express the answer. If the user selects Year, the function will return the year of [Date].

167. ZN

`ZN(expression)`

Expression = Any given number. The ZN (zero NULL) function is used to return zero values where NULL values exist in the view. If the value is not NULL then the function returns the expression; otherwise it returns zero. Use ZN to avert NULL results.

Basic

`ZN([Profit])`

This function will look at all values for Profit that exist in the view and return Profit value if is not NULL; otherwise it will return zero.

Intermediate

`ZN(SUM([Profit]))-LOOKUP(ZN(SUM([Profit])),-1)`

This function returns the difference in profit from the current row to the value in the relative offset of this row; in this case, the target row is the previous row. The ZN function is applied twice. In the first calculation, the ZN function is applied to return zero if the value is NULL. Then ZN is included in the LOOKUP expression to avoid returning NULL values there as well.

NOTES

1. Joe Mako, Tableau User Forum, "4. Re Attribute?" last modified November 16, 2011, accessed July 20, 2013, http://community.tableausoftware.com/thread/114562?start=0&tstart=0.

APPENDIX F

Companion Website

The book's companion website includes resources to further advance your Tableau knowledge. Point your browser to `www.TableauYourData.com` and take advantage of free resources including:

- Sample workbooks used for the figures and examples in the text
- Video training files
- Feature updates for new Tableau releases
- Useful web resources
- Recommended reading

SAMPLE WORKBOOKS

Organized by chapter and annotated with additional instructions, the sample workbook files complement the book by allowing you to follow along with working copies of the material presented in the text.

VIDEO TRAINING FILES

View live demonstrations of techniques included in the book. The videos will be limited to five-minute durations and focus on one or two topics.

FEATURE UPDATES ON NEW TABLEAU RELEASES

Tableau maintains an aggressive upgrade schedule, typically releasing one major update every year with several significant maintenance releases. Read about the new features and how to take advantage of them to enhance your existing dashboards or how to leverage new chart types and data source connections.

USEFUL WEB RESOURCES

The InterWorks team monitors the best websites, blogs, and social media streams on the web related to data visualization and infographics. Find people who love to share their knowledge with the Tableau community.

RECOMMENDED READING

Books about data visualization, infographics, database design, and new open source tools that will add to your Tableau toolset can be found on the Recommended Reading page. Links to blog posts form the InterWorks team and other experienced Tableau practitioners can also be found here.

In addition, updates regarding Tableau conferences, roadshow events, speaking appearances, and other live events in the Tableau ecosystem can also be found under Recommended Reading. Get active in the community by sharing your knowledge at your local user group.

GLOSSARY

The glossary presented here is an eclectic mix of Tableau terminology, industry jargon, author jargon, and noteworthy terms, items, and individuals not otherwise covered in the text. Some of these terms have more generic industry definitions. They are presented here in the context of Tableau usage.

1000-GIG LAPTOP A mythical vision of the future where a $1,000 laptop can contain 1,000 gigabytes of random access memory that is fully addressable by a 64-bit edition of Tableau Desktop. While this is theoretically possible, it will require significant technology advances in hardware and software to achieve.

32-BIT ARCHITECTURE A computing architecture that is capable of addressing up to 4 gigabytes of physical memory without utilizing physical hard disk space.

64-BIT ARCHITECTURE A computer architecture that is capable of addressing over 16 exabytes of physical memory without utilizing physical hard disk space. While theoretically possible in a desktop computer, hardware limitations confine this kind of processing to server environments today.

ACTION Tableau feature that facilitates filtering, highlighting, or invoking a URL call by selecting information contained in a Tableau dashboard or worksheet so that a dashboard or worksheet view is altered by selecting data element(s) from within a pane.

AGGREGATE FUNCTION A function groups together values in a specific way in the resulting view, but these values are presented in the data source as multiple rows of data. Tableau aggregate functions include `AVG` (average), `COUNT`, `COUNTD`, `MAX`, `MEDIAN`, `MIN`, `STDDEV`, `STDEVP`, `SUM`, `VAR`, and `VARP`. See the function reference for specific definitions and examples.

AGGREGATION The level of detail expressed in a view. Highly aggregated data present less detail. Highly disaggregated data presents more granular (atomic) views of the data.

ANIMATED VISUALIZATION Visualization contained in a dashboard or a workbook that utilizes the Tableau Pages shelf to animate the view by automatically incrementing a filter.

ARTISTIC-BENT A person with artistic-bent values the artistic qualities or aesthetic contained within the presentation of data in dashboards and visualizations.

AXIS LABEL An editable part of the axis heading that provides descriptive information regarding the nature of the data being presented by the mark(s) within the view.

BEHFAR JAHANSHAHI Behfar is the CEO and founder of InterWorks, Inc. Born in Edmond, Oklahoma. He founded InterWorks while attending Oklahoma State University. (His last name is pronounced jah-han-sha-he.)

BETA TESTER Someone who participates in evaluating beta releases of Tableau software and provides valuable, documented feedback on features, functions, and quality issues.

BIG DATA A technology industry buzzword that refers to any particularly large data source you would like to analyze.

BLUE ICON Refers to the color of the icons used to present discrete entities.

BLUE PILL Refers to the color used to represent discrete dimensions, measures, and parameters. Discrete entities result in panes (windows) as more granular views of the data are presented within the visualization.

BOOLEAN Refers to Boolean logic—a part of algebra used to derive true or false statements by using logical operators such as `AND`, `BUT`, `OR`, and `NOT`. Formulas that result in true/false or yes/no answers are sometimes referred to as Boolean results.

BOX PLOT A type of visualization typically used to present ranges of disaggregated data for discrete dimensions within Tableau. While this type of chart isn't part of Tableau's Show Me facility, these charts can be created by disaggregating the data plot and adding a reference distribution and reference lines to the view—creating a box-like appearance around the median value, while also highlighting the minimum and maximum values for each cell in view.

BUBBLE CHART A way to present one-to-many comparisons in Tableau that is generally not favored by data visualization experts because other forms of presenting one-to-many comparisons are more precise.

BULLET GRAPH A chart type invented by Stephen Few that uses the combination of a bar chart, a reference line, and a reference distribution to present actual and comparative data in a space-efficient way. Tableau's Show Me button supports this chart type although one should verify that the bar reflects the actual value and the reference line and reference distribution should present comparative budget, historical, or target data.

BUZZED A new Tableau user who is very excited about Tableau's capabilities.

CALCULATED VALUE New measure or dimension created by defining a formula within Tableau's formula editing menu.

CELL The lowest-level of granularity presented within a view.

CENSUS DATA Information that comes from the United States Census Bureau that is also available in Tableau maps expressed as polygon shapes for state, county, ZIP code, or census block group.

CHOROPLETH MAP (FILLED MAP) Map style in which the shape of a map element (country, state, and so on) is used to depict a value range within the map through the use of color. This chart type is supported by the Show Me feature where it is referred to as a filled map.

CLOUD SERVICES Computing services provided to end users via the Internet because they can be economically delivered with adequate performance and security. Tableau Public is an example of a free cloud computing service. Tableau Public Premium is a fee-based cloud computing service.

COLUMN SHELF The place to position dimension or measure pills so that the data that they represent is expressed horizontally across the view.

COLUMN TOTAL The menu option for expressing the total of a column contained within a Tableau view.

COLUMNAR-ANALYTIC DATABASE A database that is designed to efficiently query and present data from very large datasets very quickly. Tableau has connectors to most of the popular column stores commercially available.

CONTINUOUS DATA TYPE Continuous data types (denoted by green pills in Tableau) present data in continuous, unbroken panes. As more granular levels of the data are exposed in a view, they remain unbroken, whereas discrete data types display the data with breaks between different aggregations of the data. For example, dates presented as discrete data types appear as broken lines within data windows (year, quarter, month, week, and so on). See *Discrete Data Type*.

CORE LICENSE One of Tableau Server's licensing models in which the customer licenses Tableau Server in a way that provides for unlimited users but a limited hardware The license amount is determined by the number of server core memory microprocessors.

COUNT Database function that is used to count every row from the data source presented in the view. It is the equivalent of assigning the value of one to each row in the data source and then summing those values.

COUNT DISTINCT A database function used to count each distinct instance of a value within a set so that repeated entries of the same value are counted only once.

CROSSTAB A visualization style that presents text in grid form, similar to the way numbers are presented in spreadsheets. Crosstabs provide an effective means for looking up specific values.

CUSTOM GEOCODING Tableau provides geographic coordinates of standard geographic entities (state, county, postal code, and so on). If you need to present a specific address on a map, you must supply custom geocoding coordinates for the address. Custom geocoding refers to the process of obtaining customized geographic coordinates.

DASHBOARD Dashboards are an assembly of workbooks within a special view within Tableau. Unlike a workbook, in which the data shelf presents data source fields (dimensions and measures) as well as parameter, sets, and groups, the dashboard design page provides a workspace for placing individual workbooks within view and enables the resulting panes of information to become interactive.

DATA ANALYST A data analyst is a person responsible for gathering data and turning that data into information that other people can use. Frequently, analysts are required to develop insight and provide analysis of the data so that other staff can act on the information provided.

DATA ARCHITECT A data architect is a technical professional with deep understanding of database schema, extract transform and load logic, and other technical aspects of database design.

DATA BLENDING The act of combining data from dissimilar data sources in Tableau through the use of a common dimension. Data blending joins data from a primary and secondary data source via a left outer-join-like association of the members of one or more fields.

DATA CUBE A database construct in which data is pre-aggregated in order to achieve improved performance through pre-calculation of the answer set. As a result, data cubes reduce the amount and granularity of data available for the user to query.

DATA EXTRACT Extracting data in Tableau refers to the act of pulling some or all of the data from a data source into Tableau's proprietary data engine. Extract files provide a compressed (and many times better performing) alternative versus a direct connection to a data source.

DATA QUALITY Refers to the accuracy and completeness of the data source being analyzed.

DATA SERVICE Data services offer data for a fee in industry-standard data formats. Some data services offer public data without charge. For-fee data services typically provide more accurate and complete data sets in more readily consumable forms than free services.

DATA SHELF The Data shelf is the left side of Tableau's worksheet window that displays the data connections available for the analyst to use to build views. You can connect to as many data sources as you like with Tableau.

DATA VISUALIZATION Data visualizations in Tableau refer to the worksheet views of the combination of measures and dimensions added to the column and row shelves and the marks card. Combinations of worksheets displayed in dashboard panes are also referred to collectively as data visualizations.

DATE PART Date part is used to express discrete data types when defining custom date types in Tableau.

DATE VALUE Date value is used to express continuous data types when defining custom date types in Tableau.

DIMENSION Dimensions refer to data types that are typically text, a date, or a key-value number in a data source. Number ranges in histograms are also dimensions.

DIRECT CONNECTION Direct connections in Tableau refer to working directly with a data source as opposed to using Tableau's data extract engine to store the data.

DISAGGREGATION Exposing progressively more detail contained within the database.

DISCRETE DATA TYPE Discrete data types (denoted by blue pills in Tableau) present data in broken, windowed panes as more granular levels of the data are exposed in a view. For example, dates presented as discrete data types appear as broken lines within data windows (panes) whereas time is expressed in more granular ways (year, quarter, month, week, and so on). See *Continuous Data Type*.

DUPLICATE FIELD A duplicate field in Tableau refers to the act of duplicating a dimension or measure from a data source within Tableau by right-clicking the primary field and selecting the Duplicate menu option.

EDWARD TUFTE Edward Tufte is the respected author of books on data visualization.

EXTRACT, TRANSFORM, LOAD (ETL) An ETL process is the act of cleaning suspect source data through computer logic and human intervention.

FACT In a database, a fact refers to a numeric measure to which mathematics may be applied to derive additional insight.

FILMSTRIP VIEW A way of viewing worksheet tabs in Tableau that expresses the information in a small graphic as opposed to text. The filmstrip view can be invoked by selecting the lower-right side of the worksheet view (the up and down arrows) and clicking to expose the filmstrip. Alternatively, the upper-right section of the worksheet that contains the four gray boxes can be selected to view a slide deck–style presentation of the filmstrip. Right-clicking either filmstrip view causes all of the views to be refreshed. This is particularly helpful if you are using Tableau in a presentation because each worksheet will load instantaneously. If you don't refresh all views via the filmstrip, Tableau requires time to render each view individually.

FILTER ACTION An action control that invokes a filter similar to the way a quick filter might, but through the use of the data visualization itself. Filter actions restrict the data presented in other panes within a dashboard based on selection(s) made in an individual pane.

FIXED AXIS A fixed axis is an axis that has been specifically restricted to present a predetermined range of values. It is generally not a good idea to fix the axis range within a visualization connected to an active data source, as the range of values in the data may exceed the range of values in the fixed axis.

FORECAST The act of presenting future estimated values based on historical values. In Tableau, this can be done by right-clicking within the view and selecting the Forecast option. Tableau provides a Best Fit forecast that the user can modify by selecting from a menu of available forecast options including Automatic, Automatic With Seasonality, Trend And Season, Trend Only, Season Only, or No Trend Or Season.

GEOCODING The act of obtaining longitude and latitude for geographic entities being presented on a map. Tableau automatically geocodes standard geographic entities such as country, state, county, and postal code.

GRANULARITY The level of detail provided within a dataset or displayed within a visualization. For example, city is a more granular presentation of geographic data than state.

GREEN ICON Represent dimensions or measures that are continuous.

GREEN PILL Refers to a dimension or measure that is continuous.

GROUP An ad hoc entity in Tableau that is represented by the paper clip icon. Groups refer to combinations or sets of values. Grouping dimension set members is an effective way to deal with many small outlier members of a set.

HADOOP A popular, open source, no-SQL database used to collect and store very large and dynamic datasets without having to predefine a data schema.

HEAT MAP A data visualization in which up to two measures can be displayed by using size and color to express the values. Heat maps are a good way to express many comparisons of a large number of set members to quickly identify outliers.

HIERARCHY Hierarchies provide the ability to express different values within related dimensions in Tableau by expanding and contracting the hierarchy displayed in a view. Tableau provides automatically generated data hierarchies for year, quarter, month, and so on. You can manually combine fields to create custom combinations of dimensions in views.

HIGH AVAILABILITY A Tableau Server configuration that reduces the possibility of downtime through the use of redundant physical hardware and the elimination of single points of failure.

HIGHLIGHT ACTION An action type that provides highlighting of specific selections based on color or shape. Highlight actions can be invoked through color or shape legends or by manual definition through the action menu.

HIGHLIGHT TABLE A crosstab view that uses coloration of the cells to highlight value differences. In Tableau, highlight tables require the selection of one measure only.

HISTOGRAM A visualization type that uses a bar graph to display ranges of values by counting the number of times a particular range of values occurs in the source data.

JAVASCRIPT An interpreted high-level language that is typically used along with HTML and CSS. It is an essential technology enabling the Internet. Tableau has a JavaScript API that can be used to provide interactivity between a Tableau view embedded inside a web page and the rest of the web page. See Chapter 13 for details.

JEDI A Tableau user that is particularly skilled and knowledgeable.

JOIN A database term used to describe the linking of tables through a common key record. Tableau supports inner, left, and right join types through point-and-click selection. Union join types can be achieved by editing the connection script generated by Tableau within Tableau's connection dialog window.

JSON A language-independent data format that was originally derived from JavaScript. There is a Hadoop Hive–specific function called `GET_JSON_OBJECT`, which returns the JSON object within the JSON string based on the JSON path. See Appendix E for details.

KERBEROS A three-way network authentication protocol that uses a trusted third-party service called a Key Distribution Center (KDC) to verify the identity of a computer requesting access to Tableau Server. Developed by the Massachusetts Institute of Technology. Microsoft Windows 2000 and later uses Kerberos as its default authentication method. Tableau Server supports Kerberos-based single sign-on (SSO).

LEVEL OF DETAIL Refers to the level of granularity expressed within the view. More granular details of a dataset are exposed as more dimensions are placed within the view or within the Marks card.

MAINTENANCE RELEASE Minor releases provided to Tableau customer that fix bugs or provide software upgrades that do not constitute major releases.

MARKS CARD Contains the marks buttons for mark type, color, size, text labels, detail (level of detail), and tooltips.

MEASURE Refers to the numbers contained in the data source that you may want to apply math toward, geographic coordinates, and the record count calculation supplied automatically by Tableau. Measures can also be created through calculations you define.

NAMED-USER LICENSE A license type in which the license is determined by the specific user name.

ODBC CONNECTION A generic windows connection to an otherwise unsupported database type.

OLAP CUBE See *Data Cube*.

ONE-TO-MANY COMPARISON A one-to-many comparison refers to any of the chart types used for comparing the different value ranges within the view. Bar charts, bullet graphs, heat maps, highlight tables, histograms, maps, and bubble graphs are typically used for comparing values.

PAGES SHELF The Pages shelf in Tableau is a filter type that can be used to animate views. Animated views are available only in Tableau Desktop or Tableau Reader and are not enabled within Tableau Server or Tableau Public.

PANE Panes denote the breaks between discrete entities expressed in the visualization. They are expressed by light gray border lines.

PARAMETER Parameters are formula variables that appear as quick-filter-like controls on the desktop. Parameters enable self-service business information by allowing information consumers to change the values or dimensions expressed in views.

PERFORMANCE TUNING Refers to the act of improving the load and rendering performance of a visualization or dashboard. Tableau provides performance tuning tools for the desktop (via the Help menu option and the Start Performance Recording selection). The performance recorder analyzes Tableau's log files and creates a Tableau dashboard reflecting the relative speed that queries are executed and rendering is achieved.

PERSISTENT QUERY CACHE Provides a faster response time in Tableau Server by enabling Tableau Data Server to save data in temporary tables.

PIE CHART A popular way to visualize one-to-many comparisons that are generally not favored by the Vizerati.

PILL A pill refers to the visual entity used to place a dimension field or measure field on the Row shelf, Column shelf, or any other shelf within the Tableau worksheet.

POLYGON Polygons are a shape typically used for expressing geographic members on a map. For example, Tableau supplies polygon shapes for country, state, and so on.

PORT NUMBER A communication endpoint that is a logical construct identifying a service. Specific services are identified to the computer by assigning port numbers. See the Tableau Server Administrator Guide for more details.

POWER HELLO The power hello is what happens when you meet Christian Chabot and Kelly Wright of Tableau Software in person. Pure energy and enthusiasm.

POWER TOOLS FOR TABLEAU A suite of software tools that add functionality to Tableau Desktop and Tableau Server. For details, see `http://powertoolsfortableau.com`.

PYTHON A popular, open source, and general-purpose programming language that is frequently cited as a good language for beginning programmers because of its easy-to-understand syntax.

QUICK FILTER Filter elements that are exposed on the desktop within a worksheet or a dashboard.

REFERENCE LINE Reference lines are invoked from the axis and are used to express a statistic or value within a cell, pane, or worksheet.

REMOTE Refers to mobile applications developed by Tableau Partner InterWorks, Inc. The apps provide Tableau Server access remotely from an iOS or Android device.

REPLACE DRAG Refers to the ability to replace a pill by dragging and dropping another pill on top of a pill.

REST API Refers to Representational State Transfer (REST) that is the software architectural style of the Internet. Tableau Server includes a REST API that enables you to manage Tableau Server resources programmatically using HTTP. See Chapter 13 for details.

RIGHT-CLICK-DRAG Right-click-drag refers to the ability to express more granular controls when placing a field into the worksheet.

ROW SHELF The Row shelf is used to express the value row-wise in the visualization.

ROW TOTAL The row total is used to express the grand total of the row expressed in the visualization.

SAML Security Assertion Markup Language is an XML standard providing secure web domains to exchange user authentication and authorization data. See Chapter 11 for details.

SCATTER PLOT A visualization used to compare two measures. More aspects of the dataset can be expressed through the use of shape, color, and size within the scatter plot. Reference lines can be added to express correlation. Scatter plots offer a good way to do ad hoc analysis.

SCHEMA Refers to the architectural design of a database.

SECURE SOCKET LAYERS (SSL) A standard security protocol enabling an encrypted link between a web server and clients. Tableau Server uses SSL for user authentication only. See Chapter 11 for details.

SET In Tableau, a set can be used to express specific combinations of dimensions and facts within the data source. Sets can be static (based on manually selected marks) or dynamically defined based on value ranges.

SHOW ME BUTTON Tableau's Show Me button allows novice users to create data visualizations based on the combination of measures and dimensions selected by the designer without having to understand what shelves to place the pills upon. This facility makes it easy for novices to build data visualizations.

SPARKLINE Tiny information graphics first conceived of by Edward Tufte in his book *Beautiful Evidence*.

STAR SCHEMA Refers to a traditional data warehouse design format in which a fact table (consisting of numbers and key records) is surrounded by joined dimension tables (consisting of key records and text).

STEPHEN FEW Seminal thought leader in the data visualization space, Mr. Few is the author of three influential books on data visualization in the computer age.

SYNTAX Syntax refers to the way formulas are expressed, including the symbols needed for the computer to correctly interpret the instruction.

TABLE CALCULATION Table calculations are a type of calculated value within Tableau that uses the structure of the view itself to derive the solution set.

TABLE CALCULATION FUNCTION Table calculation functions are function types used to derive solutions using the structure of the visualization express to define the result set.

TABLEAU BOOKMARK (.TBM) Tableau bookmarks are a file type used to copy an individual worksheet visualization from one workbook to another.

TABLEAU DATA EXTRACT (.TDE) Tableau data extract files are files that contain data pulled from the host data source that has been stored in Tableau's proprietary data engine.

TABLEAU DESKTOP PERSONAL Refers to the desktop license for the personal edition. Personal edition connects to Tableau data extract files, Microsoft Access, Microsoft Excel, or text files.

TABLEAU DESKTOP PROFESSIONAL Refers to the desktop license for the professional edition, which connects to most commercially popular databases or to unsupported databases through ODBC.

TABLEAU ONLINE A version of Tableau Server that is a cloud-based service managed by Tableau Software. Unlike Tableau Server, Online does not have a minimum user count licensing requirement.

TABLEAU PACKAGED WORKBOOK (.TWBX) Tableau packaged workbooks refer to workbook files that have been saved in a way that bundles the source data with the Tableau visualizations so that both are contained in a single, compressed file. Packaged workbooks can be consumed by Tableau Reader.

TABLEAU PUBLIC A free version of Tableau desktop that is limited to connecting to datasets with no more than 10,000,000 records. Data sources are also limited to text files, Excel spreadsheets, or Access database files.

TABLEAU PUBLIC PREMIUM A deprecated paid version of Tableau public that offers access to unlimited data sizes, more data connection options, and more control over data security.

TABLEAU SERVER Tableau's data consumption environment for entities that want to share information securely within a defined group of users.

TABLEAU WORKBOOK (.TWB) Files in which the design of the data visualizations, the dashboards, and the data connections are saved, but not the data itself. As a result, these files are normally small because the data source keeps the data, not Tableau.

TDWI (THE DATA WAREHOUSE INSTITUTE) A database industry information service.

TEXT TABLE A filter type in Tableau that results in a temporary table being created that restricts the data in the sheet to include only the contents of the temporary file. Context filters can help or hurt performance. They help when the filter includes a small subset of data and when the filter is changed infrequently. They hurt performance if the filter replicates the entire data source contents because the filter is replicating the entire data source. Context filters are applied using the Add to Context menu option. Identify context filters by the gray field pill color.

TIME SERIES Time series charts are an information visualization type used to display data over time.

TOOLTIPS Tooltips are pop-up windows that display additional details regarding the mark selected within data visualizations in Tableau.

TRANSACTION SCHEMA Refers to databases designed to ensure the accuracy of an individual transaction in a database as opposed to a data warehouse schema used to chunk information for analysis.

TREND LINE Trend lines are invoked in Tableau by right-clicking within the worksheet to display a trend line of the data using linear, logarithmic, exponential, or polynomial regression.

TREND MODEL The trend model is used to describe the math employed to create a trend line in Tableau.

TRUSTED AUTHENTICATION A service used by Tableau Server when views are embedded into web pages to ensure that everyone visiting the page is a licensed Tableau Server user. It is a means for establishing a trusted relationship between Tableau Server and one or more web servers. See the Tableau Server Administrator Guide for details.

UNION A type of join clause used to join two different tables of identical structure, essentially adding them together. Union joins can be achieved only in Tableau by editing the connection script.

URL ACTION An action type that can be used to call web page URLs to make data from the data visualization interact with websites embedded within a dashboard, or called from a dashboard, and visualized within a web browser.

USER FORUM The Tableau user forum is a place to ask questions and have questions answered about Tableau.

VIEW A view in Tableau refers to an individual data visualization or a pane within a dashboard.

VIZ An abbreviated way of saying data visualization.

VIZERATI Anyone generally recognized as a member of the data visualization world.

WEB AUTHORING A Tableau Server capability that is controlled by the server administrator through permissions. When edit capability is enabled Tableau Server users can edit existing views or create new views.

WORKBOOK Shorthand for a Tableau workbook.

WORKSHEET An individual design page within Tableau Desktop.

YODA The nickname given by Giedra Aleknonyte (a member of the Vizerati) to the author of this book.

INDEX

Index

B

C

D

E

F

Q

R

S

U

V

W

X

Y

Z