INTRODUCTION TO USING TABLEAU

Introduction to using Tableau | 2 days

Introduction to using Tableau is a two-day, hands-on course that will provide attendees with a solid knowledge of the core features, benefits and uses of Tableau, and how to maximize use of the toolset. Special emphasis will be placed on the Tableau 10's newest features. Students will learn foundation-level skills, such as data preparation, and then extend those skills into building and customizing their own data visualizations and dashboards, essential for high-level visibility and effective data storytelling. Students will also explore best practices for trend analysis and forecasting using clustering and distribution models to inform your analytics.

Students will also learn how to share their Tableau visualizations, a critical skill and key to any organization that needs to make better informed data driven decisions based on easily viewable, accessible information.

Working in a hands-on learning environment, attendees will:

- Create stylish visualizations and dashboards that explain complexity with clarity
- Learn effective data storytelling to transform how your business uses ideas and makes decisions
- Explore all the new features in Tableau 10 and start to redefine what business analytics means to your organization
- Find out how to build effective visualizations and dashboards
- Prepare and clean your data so you can be sure Tableau is finding answers to your questions – not raising more problems
- Discover how to create advanced visualizations that explain complexity with clarity and style
- Dig deeper into your data with clustering and distribution models that allow you to analyze trends and make forecasts
- Learn how to use data storytelling to aid decision-making and strategy
- Share dashboards and visualizations to cultivate a culture where data is available and valued

Version

This course will be taught with the Version 10 course materials and software. Users of Tableau Version 9 will feel comfortable in the class because the user interface is very similar between versions 9 and 10. Instructors will also be able to highlight new features and differences between the two versions, as appropriate.

Audience

This two-day course is designed for the **beginner** Tableau user, new to the toolset and environment. This course will help attendees understand and use the basic techniques for creating visualizations and combining them in interactive dashboards.

Prerequisites

Before attending this course, students should have experience using Microsoft Excel.

Course Outline

1: Creating Your First Visualizations and Dashboard
Connecting to data
Foundations for building visualizations
Visualizing data
Creating bar charts
Creating line charts
Creating geographic visualizations
Using Show Me
Bringing everything together in a dashboard
Summary

2: Working with Data in Tableau
The Tableau paradigm
Connecting to data
Managing data source metadata
Working with extracts instead of live connections
Tableau file types
Joins and blends
Filtering data
Summary

3: Moving from Foundational to More Advanced Visualizations
Comparing values across different dimensions
Visualizing dates and times
Relating parts of the data to the whole
Visualizing distributions
Visualizing multiple axes to compare different measures
Summary

4: Using Row-Level, Aggregate, and Level of Detail Calculations
Creating and editing calculations
Overview of the three main types of calculations
Level of Detail calculations
Parameters
Practical examples of calculations and parameters
Ad hoc calculations
Performance considerations
Summary

5: Table Calculations
Overview of table calculations
Quick table calculations
Relative versus fixed
Scope and direction

Addressing and partitioning
Custom table calculations
Practical examples
Data densification
Summary

6: Formatting a Visualization to Look Great and Work Well
Formatting considerations
Understanding how formatting works in Tableau
Adding value to visualizations
Summary

7: Telling a Data Story with Dashboards
Dashboard objectives
Example - is least profitable always unprofitable?
Designing for different displays and devices
How actions work
Example - regional scorecard
Stories
Summary

8: Deeper Analysis – Trends, Clustering, Distributions, and Forecasting
Trending
Clustering
Distributions
Forecasting
Summary

9: Making Data Work for You
Structuring data for Tableau
Techniques for dealing with data structure issues
Overview of advanced fixes for data problems
Summary

10: Advanced Visualizations, Techniques, Tips, and Tricks
Advanced visualizations
Sheet swapping and dynamic dashboards
Advanced mapping techniques
Using background images
Animation
Summary

11: Sharing Your Data Story
Presenting, printing, and exporting

Sharing with users of Tableau Desktop and Tableau Reader
Sharing with users of Tableau Server, Tableau Online, and Tableau Public
Additional distribution options using Tableau Server
Summary

12. Overview of Tableau Server

User interactions with Tableau Server
Single-server installation
Content administration
Authorization and permissions
Users, groups, and sites
Data sources and extracts
Schedules, tasks, and subscriptions
Monitoring server status

PART 1: TABLEAU DESKTOP

Part 1: Table of Contents

1
Connecting to Data Sources

We will cover the following topics in this chapter:

- Connecting to text files
- Connecting to Excel files
- Connecting to Access databases
- Connecting to a SQL Server
- Pasting from a clipboard
- Connecting to other databases
- Connecting to Windows Azure Marketplace
- Understanding dimensions and measures
- Changing data types
- Applying filters
- Merging multiple data sources

Introduction

This chapter will cover the basics to get Tableau connected with various data sources, such as text files, Excel/Access files, SQL Server, ODBC sources, and the clipboard. We will cover simplistic versions of data files, where data is clean and ready-to-use. This chapter also covers how to apply filters to reduce the available data for analysis as well as merging two different data sources.

Connecting to text files

When you open Tableau for the first time, you should see a screen similar to the one shown in the following screenshot. This image shows the various data sources available for analysis. Tableau provides you with two sample data sources, **Sample - Coffee Chain (Access)** and **Sample - Superstore Sales (Excel)**, as shown in the following screenshot:

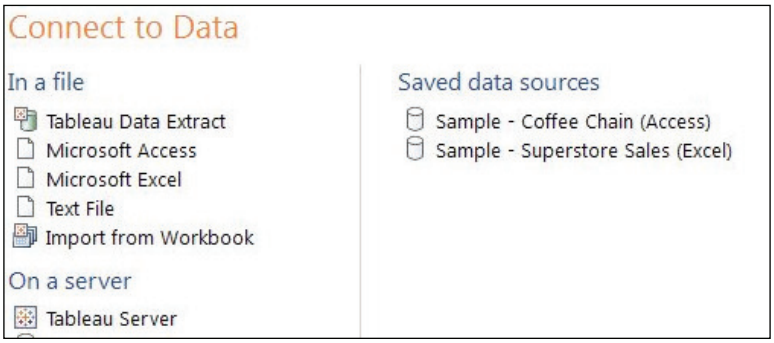
Getting ready

To prepare for the recipe, download and save `titanic.txt` from `http://biostat.mc.vanderbilt.edu/wiki/pub/Main/DataSets/titanic.txt` on your local hard drive. Remember this location, as we will use this file for this recipe. This file lists all the passengers (and their details) that boarded Titanic on its disastrous voyage.

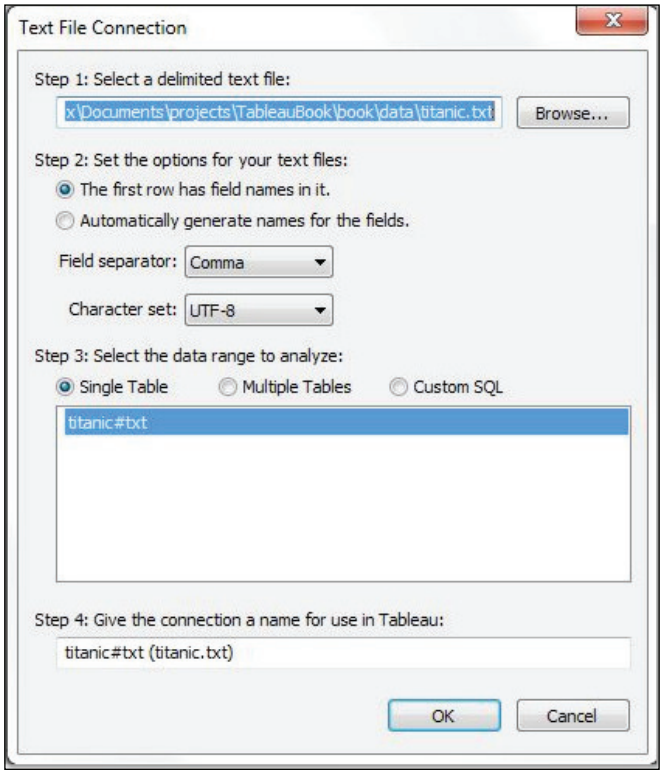
How to do it...

Once you have downloaded the text file, perform the following steps to get the data in Tableau:

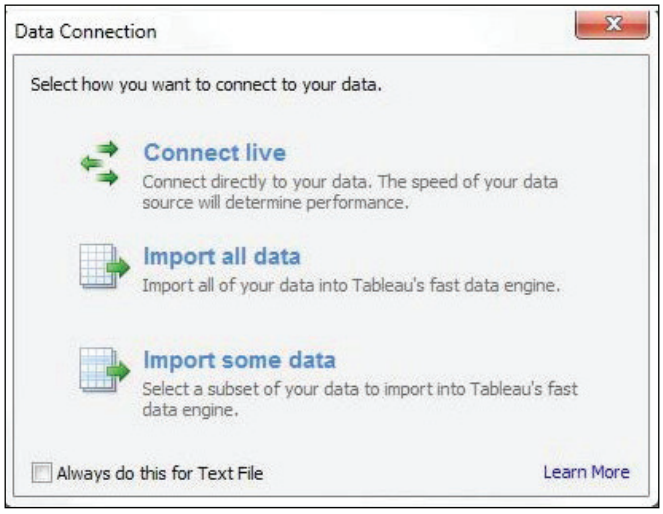
1. Click on the **Connect to data** link to expand that area as shown in the following screenshot:



2. Click on **Text File** under the **In a file** section to launch the following screen:



3. Find and select `titanic.txt` in the given **Open** dialog box.
4. As Tableau loads the data, it will prefill some of the options. For example, the field separator or delimiter and the header row. In this case, the field separator is a comma and the first row does have field names in it. So, hit **OK** on the dialog box as shown in the previous screenshot.
5. Tableau provides three options to allow you to interact with the data, which is a text file in this recipe. These three options are shown in the following screenshot. By using the **Connect live** option, we can use the file connection as it is, and by using the **Import all data** or **Import some data** option, we can speed up the analysis by importing the data in Tableau's own format. In this case, let's just use the **Connect Live** option to load all 1,313 rows in the `titanic.txt` file.



6. As you can see, Tableau determined the data types and put some fields from the text file in the **Dimensions** section and others in **Measures**. Tableau determines data types of various fields using the Microsoft Jet Database Engine driver. Due to the driver's limitations, however, some fields are at times misinterpreted as measures when they should be detected as dimensions and vice versa. Since the field **survived** shows up as a **Measures** section, but contains a binary value of zero and one (no and yes), it would make sense to convert that field to a **Dimensions** section. To do so, simply drag the field over to the **Dimensions** section or right-click on the field and click on **Convert to Dimension**.

How it works...

We used a text file as a data source and connected to it using Tableau's data source connection options. Although most of the time Tableau can determine data types accurately, sometimes you need to pay attention to changing the data types to reflect the actual data type. In this case, we converted a binary field (containing zero and one) from the **Measures** field to the **Dimensions** field.

There's more...

In its online Knowledge Base, Tableau discusses how to handle situations where data types are misinterpreted because of Microsoft's Jet Database Engine's limitations. You can find that article at `http://kb.tableausoftware.com/articles/knowledgebase/jet-incorrect-data-type-issues`.

Connecting to Excel files

Since Microsoft Excel is a very commonly used tool for analyzing data, Tableau makes it easy for the users to connect to Excel files.

Getting ready

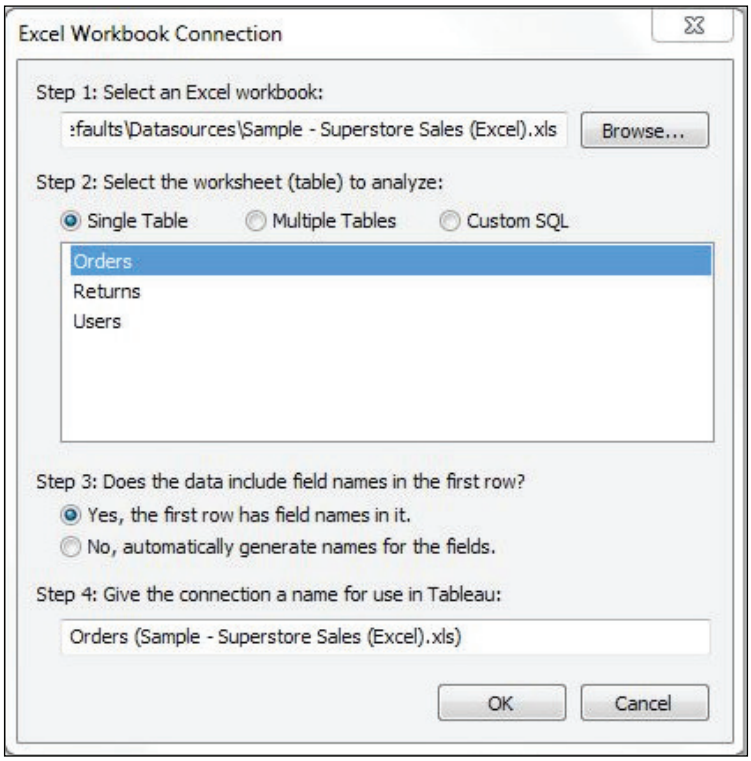
To use an Excel file as a data source, let's use the sample file that comes with the Tableau installation. Unless you have customized your Tableau installation, you should find the `Sample – Superstore Sales (Excel).xls` file when you navigate to **My Documents** | **My Tableau Repository** | **Datasources**.

How to do it...

Once you have identified the presence of the sample Excel file, perform the following steps to connect to the Excel file:

1. From Tableau's main screen, click on **Connect to Data** as shown in the following screenshot.
2. Under the **In a file** option, select **Microsoft Excel** as the connection option.
3. Browse and select the file `Sample - Superstore Sales (Excel).xls`.

4. Tableau will determine the number of sheets in the file and provide an option to import a single worksheet (also called tabs or sheets) or multiple worksheets.



5. Select the **Orders** sheet and hit **OK**.
6. Use the **Connect live** option to get the data loaded as is from the Excel file.
7. You'll see that Tableau determines the field types based on the available data.

How it works...

When Tableau connects to Excel files, it provides options for connecting to a single worksheet or multiple worksheets. Custom SQL commands can also be written to access data from multiple worksheets of an Excel file. In this recipe, we connected to a well-formatted worksheet from the sample Excel file. As was the case with plain text files, Tableau determines the data types of the fields using Microsoft's Jet Engine Driver.

There's more...

Although we used a well-formatted Excel file for this recipe, we know that analysts spend a lot time cleaning and manipulating data before any analysis. Before connecting to Tableau, we have to make sure the Excel files are formatted according to what Tableau is expecting as a data source. The *Preparing Excel Files for Analysis* article in the **Knowledge Base** section at the following link provides more information on how to prepare Excel files to be used in Tableau for analysis:

`http://kb.tableausoftware.com/articles/knowledgebase/preparing-excel-files-analysis`

Connecting to Access databases

Microsoft Access is a good tool to store smaller datasets in a relational database format without purchasing and installing complete data storage solutions, such as Microsoft SQL Server, Oracle, or MySQL. Tableau provides an option to connect to Access databases.

Getting ready

Let's use the Access database (`Sample - Coffee Chain.mdb`) that came with the standard installation. As with the Excel file used in the previous recipe, unless you made any customization during installation, the database file should be found by navigating to **Documents** | **My Tableau Repository** | **Datasources**.

How to do it...

Once you have located the Access database file, perform the following steps to connect to the sample Access database file:

1. From Tableau's main screen, click on **Connect to Data**.
2. Under the **In a file** option, select **Microsoft Access** as the connection option.
3. Browse and select the file `Sample - Coffee Chain.mdb`.
4. Tableau will determine and list tables found in the database and will ask whether to analyze a single table or a query. Select **CoffeeChain Query** and hit **OK**.
5. Use the **Connect live** option to connect to the data from the Access database.
6. You'll see that Tableau loaded the query from the Access database and also determined the data types using Microsoft's Jet Engine Driver.

How it works...

Using Microsoft's Jet Engine Driver, Tableau connects to Microsoft Access and determines the data types of the fields of a table. Just like the connection to Excel files, Tableau allows the user to select a single table, multiple tables, or write custom SQL commands.

Connecting to a SQL Server

Although it is pretty easy to connect to Access, Excel, and other flat files, data is frequently stored on some sort of relational database on a server, such as on the SQL Server or Oracle. Tableau offers connections to various data stores too. Here, we'll focus on Microsoft SQL Server.

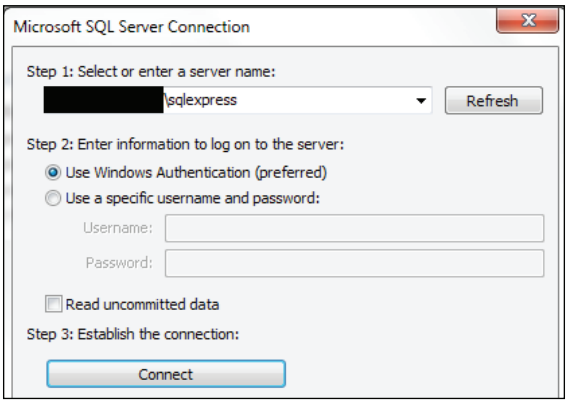
Getting ready

Security roles, server connections, authentication issues, port and firewall details, and other factors can create problems while trying to access data from a server. The solutions to these problems are out of the scope of this book but you should make sure that you are able to access the server database from the same computer on which Tableau is installed.

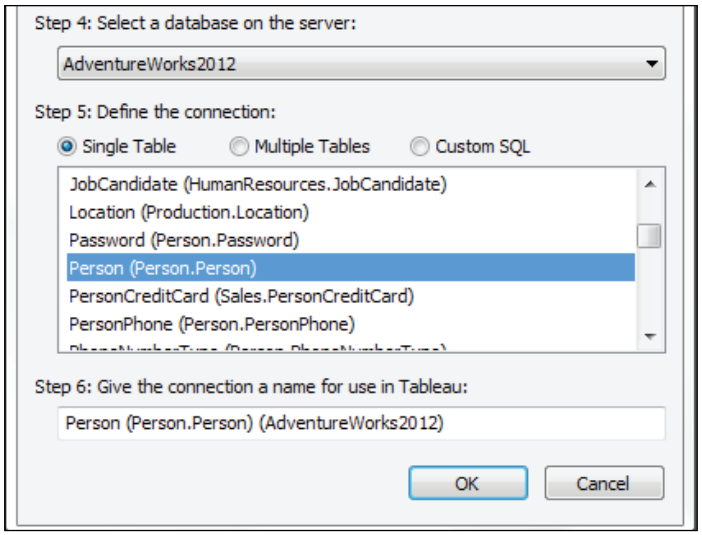
How to do it...

Once you have made sure you have access to the database server and the database, perform the following steps to connect to a Microsoft SQL Server table:

1. From Tableau's main screen, click on **Connect to Data**.
2. Under the **On a server** option, click on **Microsoft SQL Server**.
3. In the **Microsoft SQL Server Connection** dialog box, enter the server name as shown in the following screenshot.
4. Click on the **Connect** button under the **Establish the connection** option.



5. Select **AdventureWorks2012** as a database on the server.
6. Under **Define the connection**, select **Single Table** and then **Person (Person.Person)**, as shown in the following screenshot:



7. Hit **OK**.
8. Select the **Connect Live** option to connect to the SQL Server database directly.

How it works...

Using the the **Connect Live** option in Tableau, we connected to a SQL Server database directly. This option allows users to create visualizations that will be refreshed as the underlying data changes. If connected live, Tableau will create results based on the SQL Server's settings, which are usually set to maximize performance.

Pasting from a clipboard

Sometimes it is easier to just paste data from the clipboard than pasting it to Excel or CSV files and then importing them again in Tableau. Tableau does provide a quick import method from the clipboard.

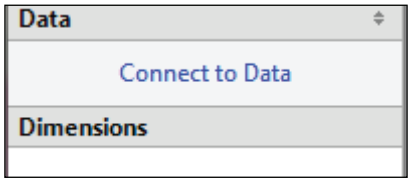
Getting ready

Let's use the Titanic dataset that we used in the *Connecting to the text file* recipe. The file is at `http://biostat.mc.vanderbilt.edu/wiki/pub/Main/DataSets/titanic.txt`.

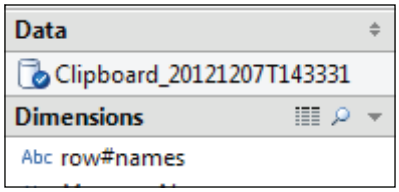
How to do it...

Once you have determined that the Titanic dataset file is on your local drive, perform the following steps to copy the data to Tableau using a clipboard:

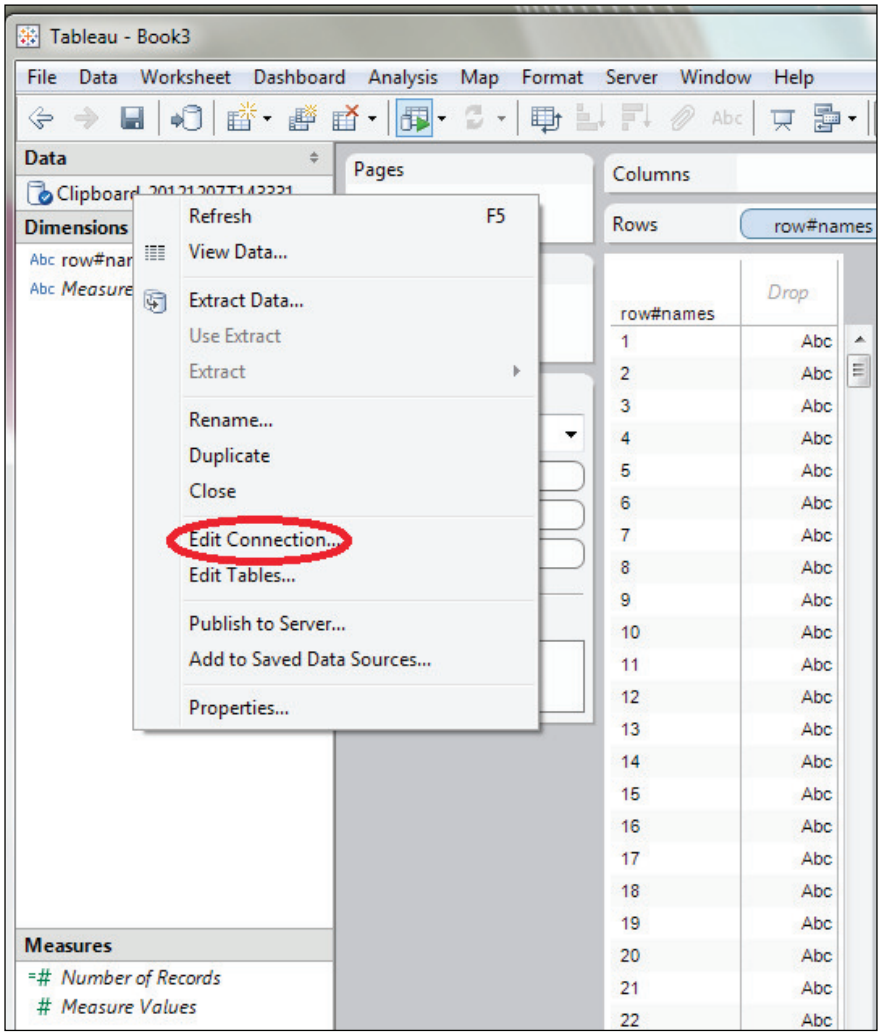
1. Open the file in Notepad.
2. Select all the data and copy everything from the file to the clipboard (*Ctrl* + *A* and then *Ctrl* + *C* on Windows), and perform the following steps:
    1. In Tableau, navigate to **File** | **New** to open a new blank workbook.



    2. Click on any open area on the workbook and paste the copied data by going to **Data** | **Paste Data** (*Ctrl* + *V* on Windows). Note the top-left area of the workbook, where the data connections are shown. It should show text such as **Clipboard_timestamp**, where **timestamp** is the time and date when the paste occurred.



3. As you can see from the workbook, the data was improperly imported. To fix this problem, right-click on the data connection named as **Clipboard_timestamp** under **Data** and click on **Edit Connection**.



4. From the **Field** separator drop-down menu, select **Comma**; we do this because the original file was separated by commas and Tableau used tab as the separator.
5. After making your selection, hit **OK** and you'll see that all the fields from the Titanic data text file are shown in **Dimensions** and **Measures**.

How it works...

While copying data from the clipboard, Tableau uses tab as the default separator of data. This causes improper import of data when the data is separated by other delimiters such as a comma. You can easily correct this problem by editing the connection to the clipboard file saved in Windows' temporary folder.

There's more...

If you created visualizations based on the data from the clipboard and you need to regularly update this visualization, you will find the Tableau online article, *Editing Pasted Data*, in the **Knowledge Base** section at the following link:

`http://kb.tableausoftware.com/articles/knowledgebase/editing-pasted-data`

This article explains how to save the data source of the clipboard and modify the data source.

Connecting to other databases

Connecting to most of the databases on a server is straightforward. Providing the server name and authentication details is usually sufficient. There are at times, however, when either Tableau does not provide a direct connection to that database server or you want to use an ODBC connection.

Getting ready

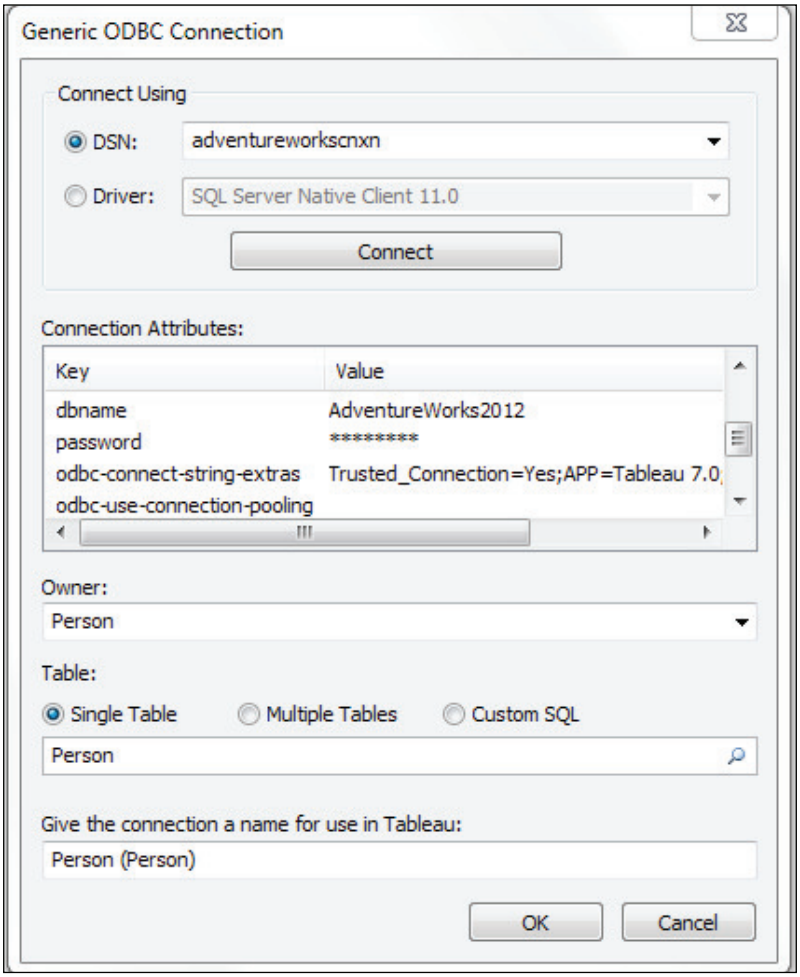
The easiest way to make a connection to a database server is using an ODBC connection. To use this type of connection, we must first set up a **Data Source Name** (**DSN**). For this recipe, we will set up a DSN to connect to a database server.

In **Control Panel**, click on **System and Security** and then on **Administrative Tools**. Double-click on **Data Source (ODBC)**. You'll see a **Data Source Administrator** window. Under **User DSN**, click on **Add** and follow the steps to create a DSN for your database. If you have a SQL Server instance installed and the `Adventure Works` database populated, select **SQL Server Native Client** and hit **Finish**. In the **Name** field, enter a name that you'll remember easily; remember that it cannot contain spaces. I chose `adventureworkscnxn`. Under **Server**, either enter the database server name or select **Local**. Continue with the default selections until you see a **Finish** button.

How to do it...

Once the DSN is set up, open a new worksheet in Tableau and perform the following steps to connect to a database using ODBC:

1. Click on **Connect to data** and select **Other Databases (ODBC)**.
2. In the **DSN** dropdown, select the DSN that you created earlier.
3. Click on **Connect** to test the data connection.
4. Under **Owner**, select **Person**.
5. Among the **Table** selection radio buttons, select **Single Table**; search for a table name by clicking on the magnifying glass icon.
6. Under the **Give the connection a name ...** textbox, enter a name and hit **OK**.



7. If you can see the **Connect live**, **Import all data**, and **Import some data** options in the **Data Connection** page, you were able to successfully connect to the SQL Server using ODBC and DSN.

How it works...

With ODBC, Tableau provides an option to connect to the data sources that otherwise do not have native support in Tableau. This option provides flexibility to connect any data source that has an ODBC drive. Although SQL Server is supported directly by Tableau, in this recipe, we saw how easy it is to create a data connection using an ODBC driver and DSN.

There's more...

Microsoft's online support provides an excellent overview of ODBC connectivity at `http://msdn.microsoft.com/en-us/library/windows/desktop/ms710252%28v=vs.85%29.aspx`.

Connecting to Windows Azure Marketplace

Microsoft created an online platform called Windows Azure Marketplace for trading **Software as a Service** applications and data. Users can choose to buy and sell various datasets, and that makes it a great place to use datasets hosted on the cloud with Tableau.

Getting ready

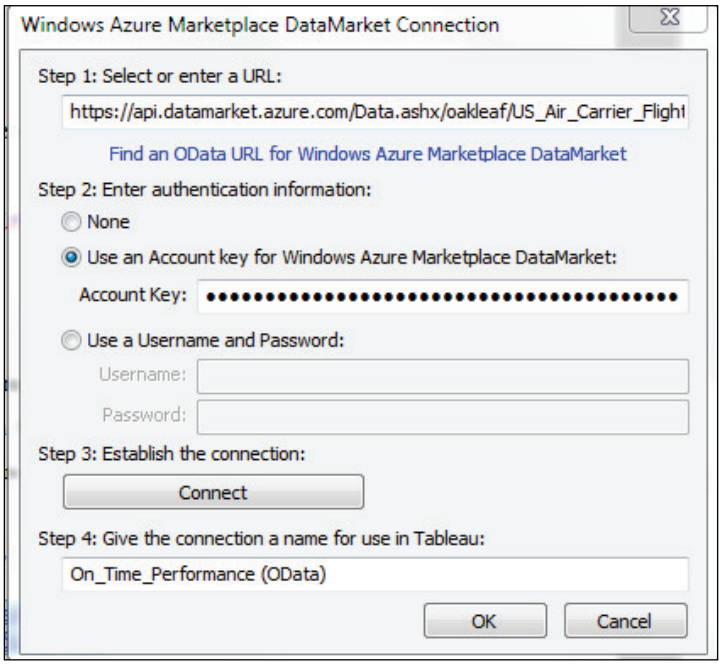
To access the Azure Marketplace datasets, you'll need to create or already have an account with Microsoft (which was earlier called Windows Live ID). Once you are logged in with your Live ID, you'll also need to complete the registration process.

How to do it...

Once you have completed the registration process for the Azure Marketplace, perform the following steps to get the data from the Marketplace to Tableau:

1. Log in to Windows Azure Marketplace (`datamarket.azure.com`) with your account credentials.
2. Click on **Data** on the top navigation menu.
3. Click on the **US Air Carrier Flight Delays** dataset.
4. Click on **Sign-up** to subscribe to the dataset, and perform all the required steps.
5. Once the subscription process is complete, click on the **Explore this Dataset** link.

6. In the **Build Query** area, select **On_Time_Performance** from the **Query** field and hit **Run Query**.
7. You should see a URL for the currently expressed query, which would look something like `https://api.datamarket.azure.com/Data.ashx/oakleaf/US_Air_Carrier_Flight_Delays_Incr/v1/On_Time_Performance?$top=100`.
8. Copy this URL.
9. In Tableau, go to the **Connect to Data** options screen.
10. Click on **Windows Azure Marketplace DataMarket**.
11. In the **OData Connection** pop-up box, shown in the following image, under the **Step 1** input box, enter the copied URL:



12. In **Step 2** of this process, either select the **Account key** or **Username** option and enter the credentials (you'll find the **Account key** value below the OData URL information; you need to click on **Show** to make the key visible).
13. In **Step 3**, click on the **Connect** button. If you do not see any message, you will be able to connect to the data file.
14. In **Step 4**, enter a name for the connection and hit **OK**.

How it works....

Microsoft Azure Marketplace offers a data market for users to explore datasets, which are usually scattered everywhere on the Web. This marketplace creates a central repository of datasets, and with Tableau's integration of this marketplace, it is very easy to analyze various datasets.

There's more...

Microsoft's online support provides detailed information for users, who could be consumers or publishers of the data, as well as developers of Azure applications and services. You can read it at `http://msdn.microsoft.com/en-us/library/windowsazure/gg315539.aspx`.

Understanding dimensions and measures

Tableau divides the data in two main types: dimensions and measures. Dimensions are usually those fields that cannot be aggregated; measures, as its name suggests, are those fields that can be *measured*, aggregated, or used for mathematical operations. Dimension fields are usually used for row or column headings; measures are usually used for plotting or giving values to the sizes of markers.

When you import the data for the first time, Tableau determines whether to consider a field as a dimension or a measure. This determination involves considering fields with all text (nominal or other text) values and fields with numeric values. Depending on the data source, Tableau also uses Microsoft's Jet Engine Driver to classify fields into dimensions and measures.

Tableau visualizations are heavily dependent on the structure of dimensions and measures. Thus, organizing data properly into dimensions and measures is important, and if Tableau's determinations are wrong about the field data types, it is easy to convert these fields to the other category. Simply dragging the field to the pane works just like right-clicking on the field and clicking on **Convert to Dimension** or **Convert to Measure**.

Changing data types

Depending on the data source and connection, Tableau tries to determine the field data type. Most often, the field data types are identified correctly; sometimes, however, changing data types becomes necessary.

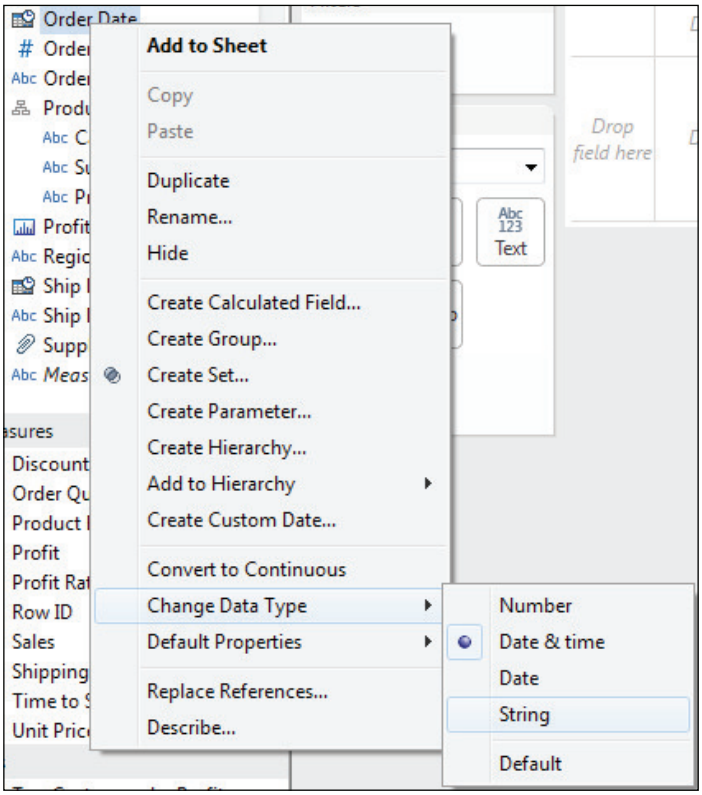
Getting ready

We will use the sample superstore sales saved data source for this exercise. Open a new worksheet and connect to the **Sample – Superstore Sales (Excel)** data source.

How to do it....

Once the sample file is loaded on the worksheet, perform the following steps to convert data types:

1. In the **Dimensions** pane, right-click on **Order Date**.



2. You'll notice some data types in Tableau: **Number**, **String**, and **Date**. **Date & time** is also a type, which is suited for data with a timestamp.
3. Select **String** as the data type for this field. Next to this field name, you'll notice a symbol with letters (**Abc**); this symbol indicates that this field contains data of type String.
4. Drag the **Order Date** field from the **Dimensions** to the **Measures** pane.
5. You'll notice in the **Measures** pane that the field **Order Date** has an aggregation of **Count**.

6. Right-click on **Order Date (Count)** in the **Measures** pane and select **Change Data Type**. Select **Date** as the new data type.
7. You'll notice that the **Order Date** field is back in the **Dimensions** pane.

How it works...

Since the data type and role of a field (dimension or measure) determines how the data will be used in the visualizations, it is critical to have the right data type for fields in the data. You will notice that, if you convert a field in the **Measures** pane to a **Date** type, that field will be moved to **Dimensions**. If a field from the **Dimensions** pane is converted to **Number**, it will stay in the **Dimensions** pane. If a field from the **Measures** pane is converted to **String**, the default aggregation changes to **Count**.

Applying filters

If you want to reduce the amount of data available for visualizations or restrict the data for a particular field value, applying filters is a very good solution. This recipe will provide a basic overview of filters, and later in the book you'll see some other uses of filters.

Getting ready

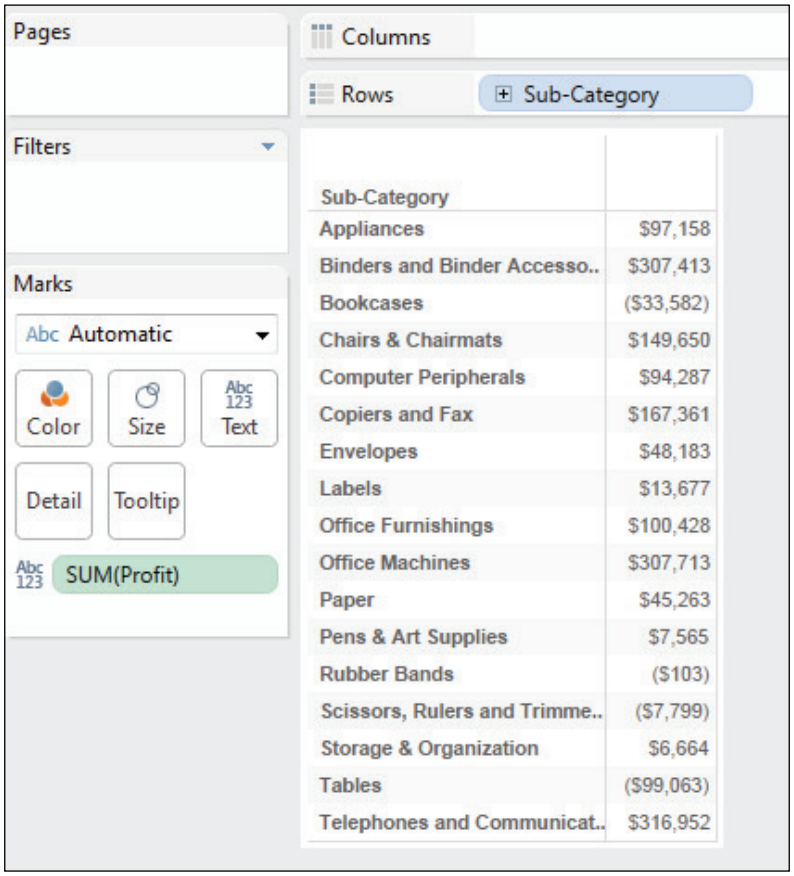
We will use the sample superstore sales saved data source for this exercise. Open a new worksheet and connect to the **Sample – Superstore Sales (Excel**) data source.

How to do it...

Once the sample file is loaded on the worksheet, perform the following steps to explore the **Filters** feature in Tableau:

1. Drag-and-drop **Sub-category** from **Dimensions** into the **Rows** shelf.

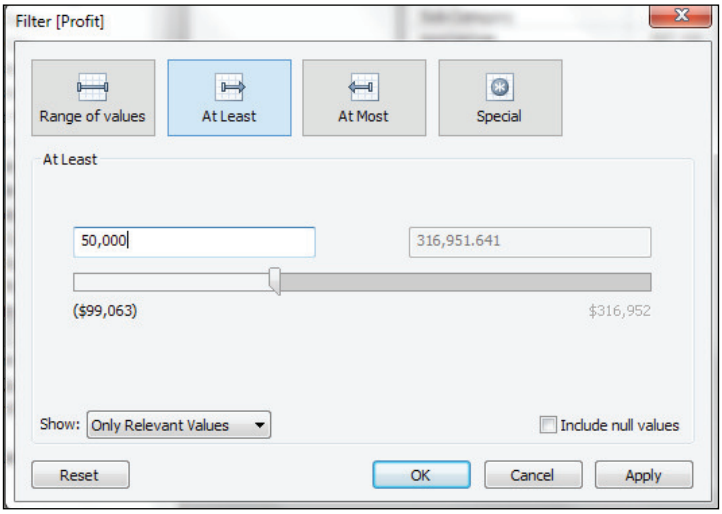
2. Then drag-and-drop **Profit** from **Measures** into the **Text Marks** box. You can also right-click on **Profit** and click on **Add to Sheet**. Your worksheet should look like the following screenshot:



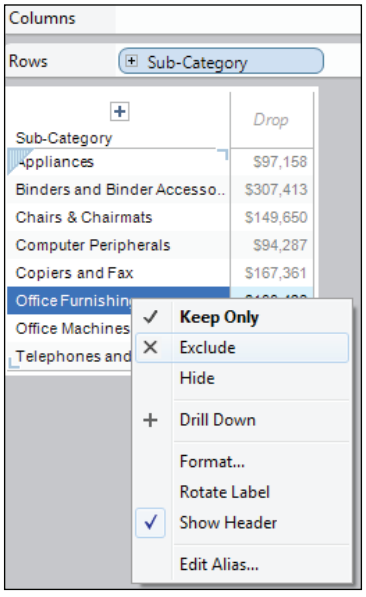
Sub-Category	
Appliances	$97,158
Binders and Binder Accesso..	$307,413
Bookcases	($33,582)
Chairs & Chairmats	$149,650
Computer Peripherals	$94,287
Copiers and Fax	$167,361
Envelopes	$48,183
Labels	$13,677
Office Furnishings	$100,428
Office Machines	$307,713
Paper	$45,263
Pens & Art Supplies	$7,565
Rubber Bands	($103)
Scissors, Rulers and Trimme..	($7,799)
Storage & Organization	$6,664
Tables	($99,063)
Telephones and Communicat..	$316,952

3. If we want to see the subcategories that generated profit of more than $50,000, right-click on the **Text** box from the **Marks** pane box where it says **SUM(Profit)**, and click on **Filter**

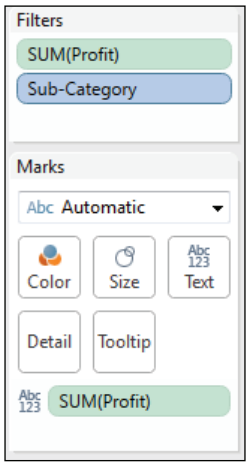
4. In the **Filter** dialog box, click on the **At Least** option and either drag the slide to **50,000** or type `50,000` in the input box and hit **OK**.



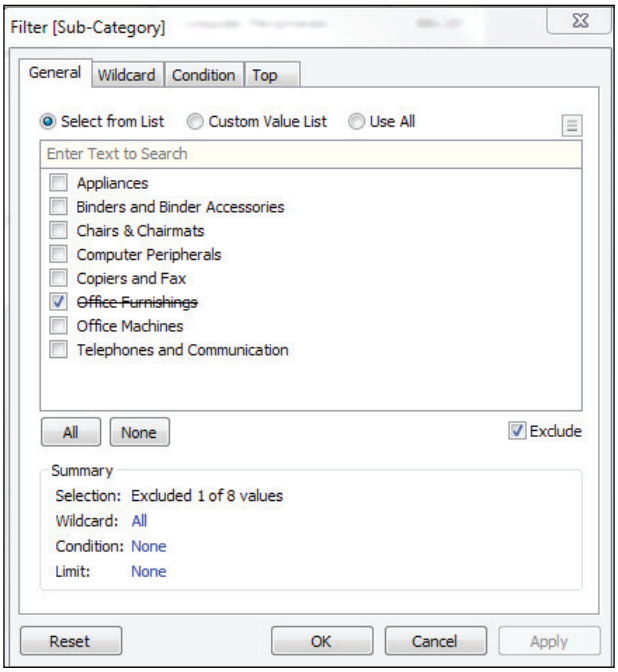
5. Once the filter is applied, you'll see eight subcategories that generated profit of more than $50,000.
6. We can continue to filter this information further. For example, to remove **Office Furnishings** as a subcategory, right-click on **Office Furnishings** and click on **Exclude**. Now only seven rows of subcategories are visible.



7. Both the filters will now show up in the **Filters** pane, as shown in the following screenshot:



8. To remove the filters, right-click on the **Filters** pane and select **Clear Shelf**.
9. To change a filter, right-click on the **Sub-Category** filter from the **Filters** pane and select **Filter**.
10. In the **Filter** window, you'll see tabs such as **General**, **Wildcard**, **Condition**, and **Top**.



11. In the **General** tab, you can type or select a value from the field. In the **Wildcard** tab, you can enter approximate string values to match certain patterns. In the **Condition** tab, you can enter conditions by the **Fields** or **Formula** values. In the **Top** tab, you can select the top *n* or bottom *n* items by a field or using a formula.

How it works...

Filters are a great way to manipulate the data on a worksheet. Depending on the field data type, various types of filters can be applied to a field. These filters can be numeric conditions to limit a numeric field or text patterns to limit a string field.

Merging multiple data sources

Often, our data is stored in different formats or different files. In relational databases, if two different tables have a common field, we can join these two tables with this field and pull the data in one single query. Tableau supports joins within a single data source connection; however, to merge multiple data source connections, Tableau uses a concept called **data blending**. In this recipe, we will look at how to blend two different data sources.

Getting ready

Download the following Google Spreadsheet, which contains the U.S. population by states, after signing in:

```
http://bit.ly/12rUIh3
```

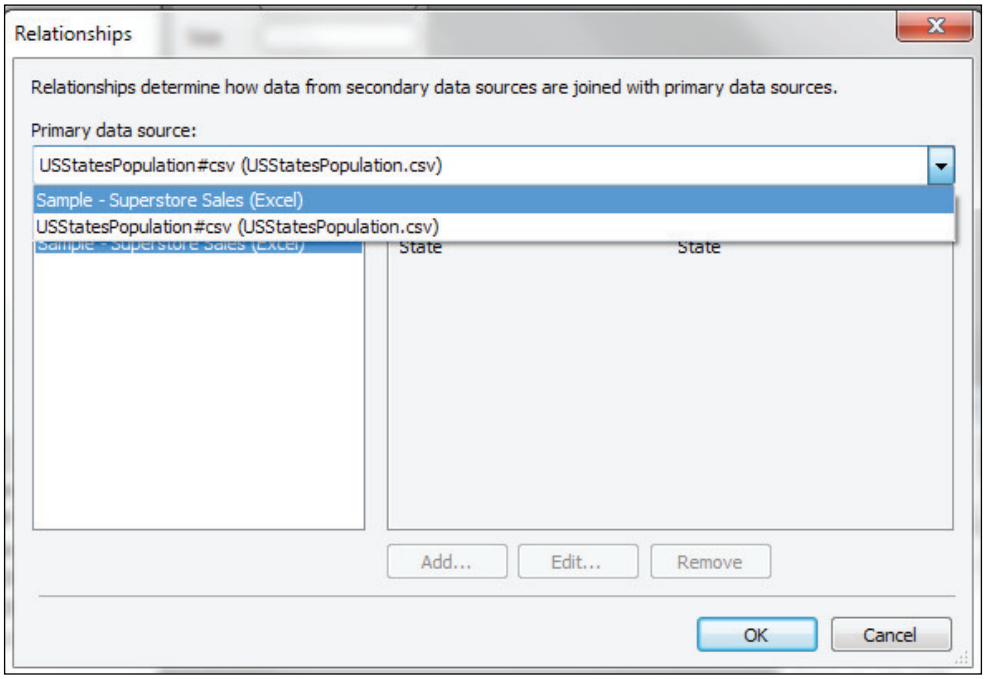
Download it as a CSV on your local hard drive and name it `USStatesPopulation.csv`.

How to do it...

Once you have downloaded the CSV file, create a new worksheet in Tableau and perform the following steps to merge the CSV file and an Excel file:

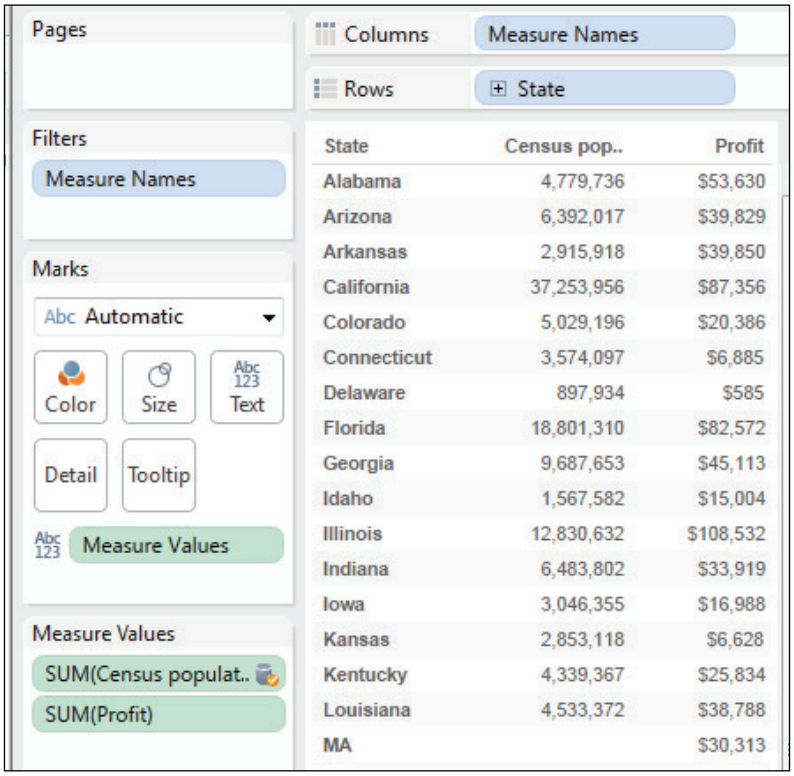
1. In a new workbook, connect to the **Sample – Superstore Sales (Excel)** data source.
2. Once the data is loaded and you can see **Dimensions** and **Measures** populated, click on **Connect to Data** in **Data** and select the text file `USStatesPopulation.csv`.
3. Accept all the defaults in the **Text File Connection** dialog box and hit **OK**.
4. Choose the **Connect Live** option in the next dialog box.
5. Tableau will match field names, and if it finds the same field names in both the data sources, it will create relationships between those common fields. To manually create relationships, click on **Data** and select **Edit Relationships**.

6. In the **Relationships** dialog box, select **Sample – Superstore Sales (Excel)** as the **Primary** data source. Tableau will make `USStatesPopulation.csv` a secondary data source file.



7. Click on the **Custom** radio button and select **State** from the left-hand side column and **State** from the right-hand side column and hit **OK**.
8. To see profit by state, drag-and-drop the **State** value from the **Sample – Superstore Sales (Excel)** data source into the **Rows** shelf and the **Profit** measure into the **Text Marks** box.
9. Click on the **USStatesPopulation#csv** data source in the **Data** pane, and right-click on **Census population_April 1, 2010** from the **Measures** pane and select **Add to Sheet**.

10. As shown in the following screenshot, you should see three measure values in the **Measure Values** pane, **Measure Names** in the **Columns** shelf and **Census population_April 1, 2010** and **Profit** in the datasheet:



How it works...

Tableau can merge two or more different data sources in the same worksheet by creating relationships among common fields of these data sources. You can customize the blending operation by specifying the common fields in the data sources in the relationships. You should also note that this blending is different from joining two tables, because when we join tables, we create row-level joins and we can add fields from both the tables. Whereas, in blending, we merely show different fields from different data sources in a single visualization.

There's more...

Since the blending or merging of multiple data sources can prove challenging, it might be easier to understand this concept better by watching somebody actually do it. A YouTube user named *James Wright* uploaded a video of blending data at `http://youtu.be/-G0lIz7y6y0`.

2
Creating Univariate Charts

The recipes covered in this chapter are:

- Creating tables
- Creating bar graphs
- Creating pie charts
- Sorting the graphs
- Creating histograms
- Creating line charts
- Using the Show Me toolbar
- Creating stacked bar graphs
- Creating box plots
- Showing aggregate measures
- Showing the top 10 items

Introduction

An analysis involving one measure is called **univariate analysis**. In this chapter, we will cover various univariate charts. Note that charts are also referred to as graphs or plots, and you will find these terms used interchangeably throughout this book.

Creating tables

Sometimes the best way to present the data is using a table. Tables use very little space and pack a lot of information in a very small area without losing any detail. Please note that, in Tableau, we refer to tables as cross tabs as well.

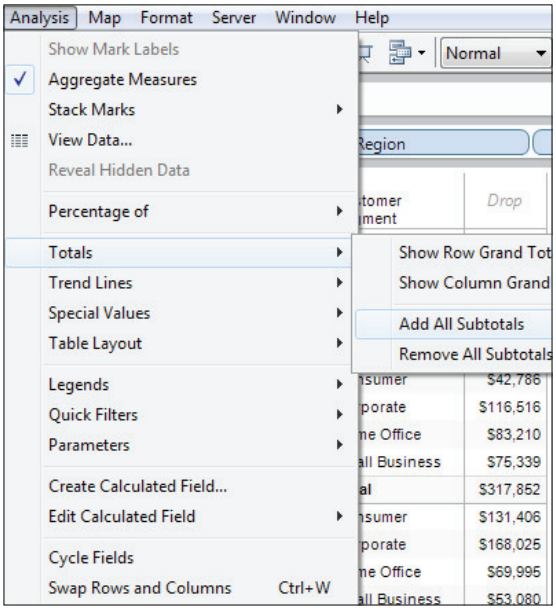
Getting ready

For this recipe, let's use the sample file `Sample - Superstore Sales (Excel)`. Open a new worksheet and select **Sample – Superstore Sales (Excel)** as the data source.

How to do it...

Although we can create complicated tables, in this recipe let's create a simple table with two dimensions and one measure.

1. Drag-and-drop **Region** into the **Rows** shelf (you can also right-click and select **Add to Sheet**).
2. Drag-and-drop **Customer Segment** into the **Rows** shelf.
3. Drag-and-drop **Profit** into the **Text** marks box. Now, you should see the total of profits as per **Region** and **Customer Segment**.
4. To add totals, click on **Analysis** from the toolbar and select **Totals** and click on **Add All Subtotals**, as shown in the following screenshot:



How it works...

To a graph designer, using tables instead of charts is a strong alternative as tables can provide very minute details of the data to the reader. Tables used with other graphs can create a compelling narrative for the reader, and Tableau makes it easy to create and combine tables and graphs by creating dashboards, which are covered later in the book.

Creating bar graphs

As bar graphs are very easy to understand, they are the most common type of graphs. The graphs that have a horizontal orientation are called **bar graphs** and the graphs that have a vertical orientation are called **column graphs**. The length of the bar represents the quantity of a particular measure. They are best used with categorical information, such as gender, state, regions, countries, business types, and others. One very important thing to note with the column bar charts is that this type of chart's y axis must always start at zero, otherwise it is very difficult and misleading to encode the length of the bar to a measure.

Getting ready

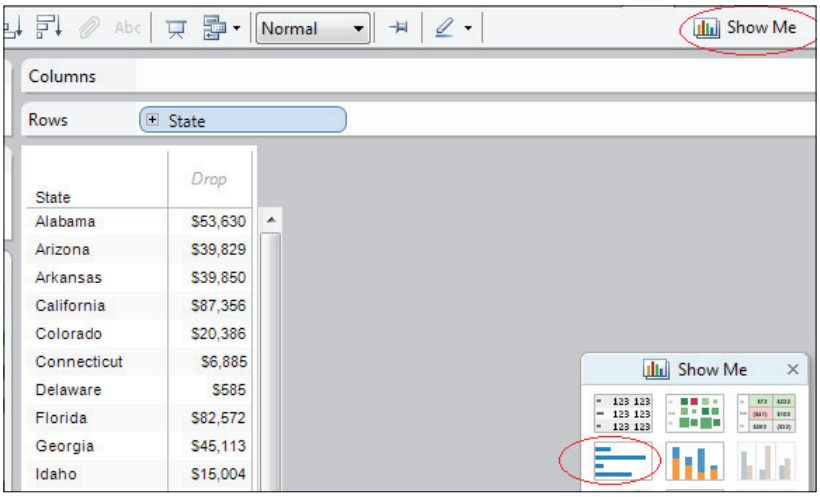
Let's use the sample file, `Sample - Superstore Sales (Excel)`. Open a new worksheet and select **Sample – Superstore Sales (Excel)** as the data source.

How to do it...

Once the data is loaded, perform the following steps to create a simple bar graph:

1. Drag-and-drop **State** into the **Rows** shelf.
2. Drag-and-drop **Profit** into the **Columns** shelf.
3. You should see the **Profit** totals by **State** in the **Columns** area in plain numbers.
4. To change the text representation to a bar graph, click on the **Show Me** button on the toolbar (shown in the following screenshot).

5. From the **Show Me** toolbar, select **horizontal bars** as shown in the following screenshot:



How it works...

By changing the default behavior of showing text tables to showing bar charts, we can make it easier for the reader to compare various measures by comparing the lengths of the different bars. In addition, since bar charts are common in various publications, the reader is adept at understanding bar charts.

Creating pie charts

Pie charts and their variations are one of the most controversial types of charts. Many experts in the information graphics and information visualization fields have warned against the use of pie charts but they are still quite common in business presentations as well as reports. There are a few key things to consider while creating such a graphic:

1. Limit the number of *slices* to three to four. In addition, slices must be large enough for easy differentiation.
2. Limit the use of color (if there are only three to four slices to show, it is easy to use one color with different hues). Colors are better used only for differentiating one item from another and not for decorating. If many colors are used, the reader faces the difficulty of distinguishing items by color.
3. Start the largest slice at 12 o'clock and move to the right; next to it on the left must be the next biggest slice. The smallest slice should be close to the bottom. This helps the reader see the bigger slices and make comparisons easily.

4. Do not use any 3D pie charts as they make matters worse. Tableau saves the users from this trouble as 3D pie charts are not supported.
5. Do not allow a sliced pie section to be sliced further, as it distracts the reader. Again, Tableau does not support *exploding* a slice.
6. Make sure that the pie totals up to 100 percent of the measure; any other total will render the pie meaningless. For example, if you want to plot the profit by store location, the pie should total to the total profit by *all* stores; that is, 100 percent of the profit.
7. Avoid excessive labels on the slices. Although labels will remove any guessing by the reader, if you label all the slices, then perhaps it is better if you replace the chart with a table.

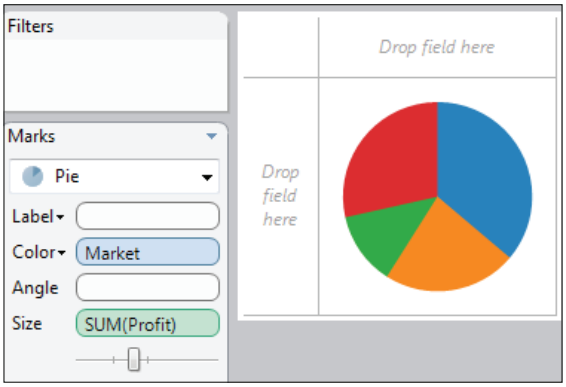
Getting ready

Let's use the sample file `Sample - Coffee Chain (Access)`. Open a new worksheet and select **Sample – Coffee Chain (Access)** as the data source.

How to do it...

Once the data is loaded, perform the following steps to create a simple pie chart:

1. Drag-and-drop **Profit** into the **Size** box in the **Marks** pane.
2. Drag-and-drop **Market** into the **Rows** shelf.
3. Click on the **Show Me** button to make the toolbar visible.
4. Click on the pie chart icon to create the pie chart.
5. Right-click on **SUM(Profit)** in the **Angle** box in the **Marks** pane and hit **Remove**.
6. To make the chart bigger, click on the zoom slider that can be seen after clicking the **Size** mark box.



How it works...

Tableau's settings for default colors for pie charts as well as the setting for the placement of the various slices creates good-looking and effective pie charts; however, by tweaking colors and adjusting sizes, we can make the charts even better. The topic of changing color palettes is covered later in this book.

Sorting the graphs

Although Tableau generates default graphs using the best practices in information visualization, often they need modification for meeting the business needs and sometimes for better representation. Tableau provides various ways to adjust and modify various aspects of the graph. Sorting is useful to display the most or least influential number or category at the top or bottom.

Getting ready

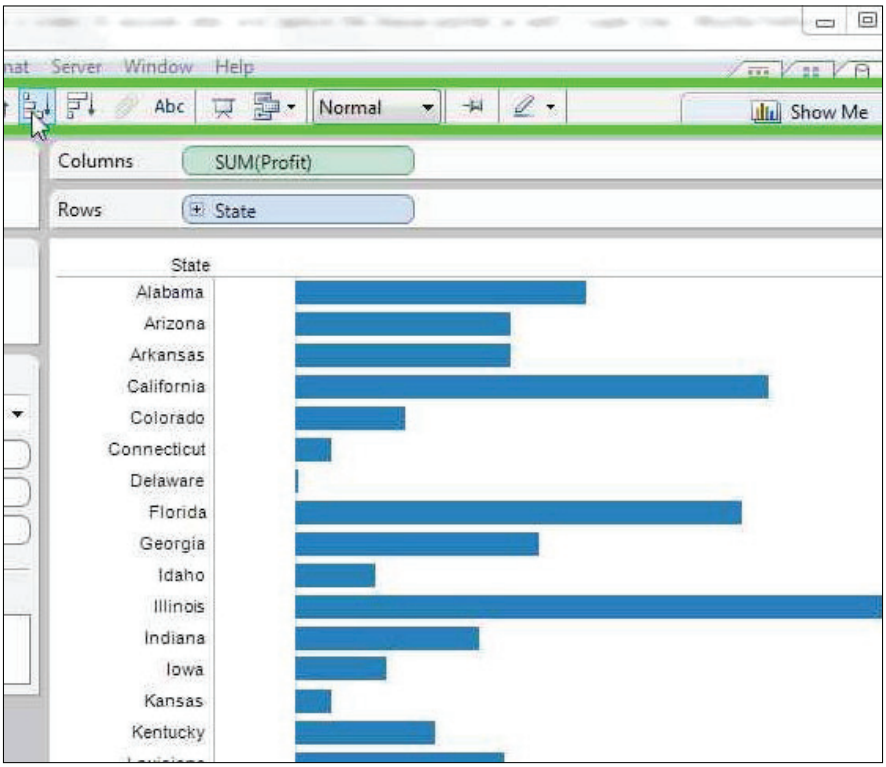
Let's use the sample file `Sample – Superstore Sales (Excel)`. Open a new worksheet and select **Sample – Superstore Sales (Excel)** as the data source.

How to do it...

Once the data is loaded, perform the following steps to create and sort a simple bar graph:

1. Drag-and-drop **State** into the **Rows** shelf.
2. Drag-and-drop **Profit** into the **Columns** shelf.
3. You should see the **Profit** totals by **State** in the **Columns** area in plain numbers.
4. To change the text representation to a bar graph, click on the **Show Me** button on the toolbar (shown in the following screenshot).
5. From the **Show Me** toolbar, select **horizontal bars** (also shown in the following screenshot).
6. Once the bar chart is created, click on the **Sort** button, which has the shortest bar on the top, on the toolbar (again, shown in the following screenshot) to show the least profitable **State** value up at the top.
7. You'll see that **Montana** is the least profitable state with **$9,127** in profit. To see the most profitable **State** value, click on the **Sort** button (this button has the longest bar at the top) to the right.

8. You'll see that **Illinois** is the most profitable state with **$108,532** in **Profit**.



How it works....

Since sorted bar charts do not require the additional tasks of identifying the longest bar and comparing lengths of various bars, they are better and more effective than unsorted bar charts.

Creating histograms

Histograms show counts or density of a measure, which is then discretized (binned) to make counting meaningful. They are best used to observe the distribution of the measure. They are sometimes confused with plain bar charts, which can be modified to show counts but usually encode the measure value as the length of the bar.

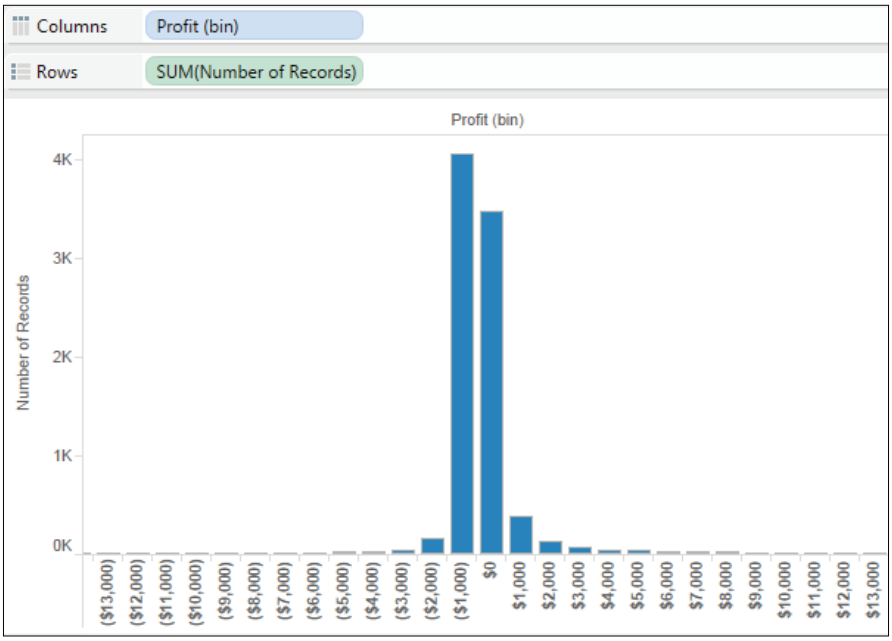
Getting ready

Let's use the sample file, `Sample - Superstore Sales (Excel)`. Open a new worksheet and select **Sample – Superstore Sales (Excel)** as the data source.

How to do it...

Once the data is loaded, perform the following steps to create a histogram:

1. Drag-and-drop **Profit (bin)** from the **Dimensions** pane into the **Columns** shelf.
2. Drag-and-drop **Number of Records** from the **Measures** pane into the **Rows** shelf.
3. You should see **SUM(Number of Records)** in the **Rows** shelf now and also a histogram with a very narrow distribution, as shown in the following screenshot:



How it works...

Histograms are very effective charts in observing the distribution of the measure of interest; however, sometimes the distribution is quite skewed or centered at one range or position, and in such cases rebinning (creating different bins is covered later in *Chapter 6*, *Calculating User-defined Field* , in the *Discretizing data* recipe) assists the reader in observing the detailed distribution or uncovering some patterns.

Creating line charts

Although line charts are best used for time-series data to observe trends by various time units, such as day, week, month, quarter, and year, they could be used for other types of data as well; however, the ups and downs in the lines themselves are less important in such cases.

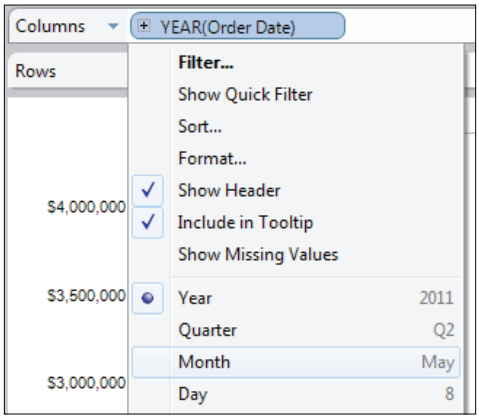
Getting ready

Let's use the sample file, `Sample – Superstore Sales (Excel)`. Open a new worksheet and select **Sample – Superstore Sales (Excel)** as the data source.

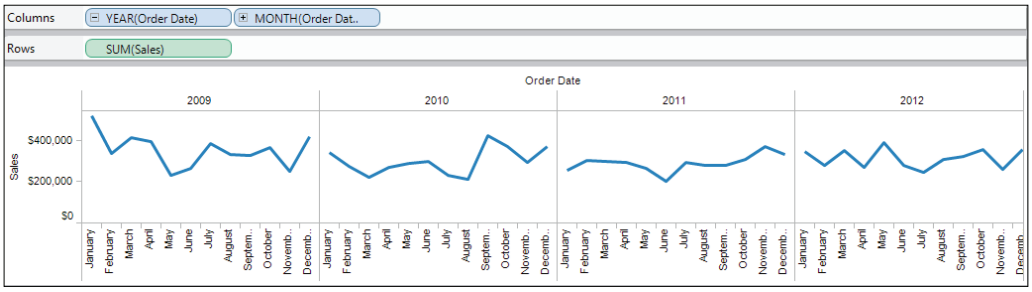
How to do it...

Once the data is loaded, perform the following steps to create a line chart:

1. Drag-and-drop **Order Date** into the **Columns** shelf. Tableau, by default, will show **YEAR(Order Date)**.
2. Drag-and-drop **Sales** into the **Rows** shelf.
3. To observe the trends by month of orders, click on **YEAR(Order Date)**, which is shown in the **Columns** shelf, and click on **Month**, as shown in the following screenshot:



4. Drag-and-drop **Order Date** into the **Columns** shelf again, but this time in front of the **MONTH(Order Date)** field. This change will show the trend by month of every year, as shown in this following screenshot:



How it works...

Line charts are quite effective in representing trends over time. These trends, however, could be misrepresented if improper zoom level, axis scale units, or aspect ratios are used. For example, if a reader is observing a line chart that has hour as the unit, the reader may think that the observed measure fluctuates quite often. But if the axis unit is changed to months, the lines will show trends over a longer time period and will not show major fluctuations, as shown in the chart with hour as the axis unit. The chart designer should carefully select the unit of time for the x axis.

Using the Show Me toolbar

The **Show Me** toolbar is one of the most powerful features of Tableau. It removes many steps required to create a graphic and automatically determines the axis location of the variables used.

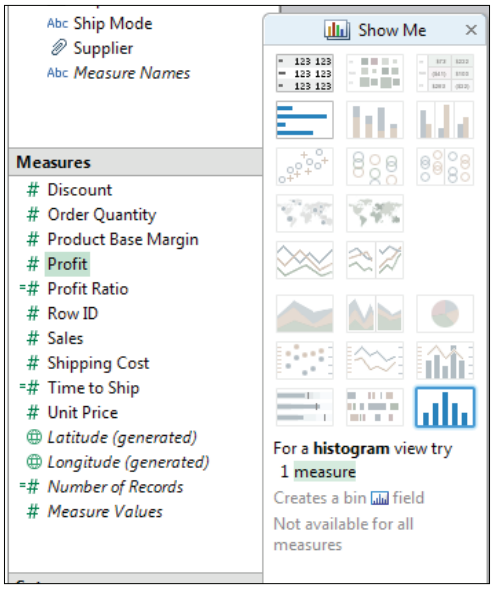
Getting ready

Let's recreate the histogram from an earlier recipe using the **Show Me** toolbar. We will use the same sample file, `Sample – Superstore Sales (Excel)`. Open a new worksheet and select **Sample – Superstore Sales (Excel)** as the data source.

How to do it...

Once the data is loaded, perform the following steps to reproduce the histogram from an earlier recipe using the **Show Me** toolbar:

1. If the **Show Me** toolbar is not visible, click on the **Show Me** button to make the toolbar visible.
2. Click on **Profit** from **Measures**.
3. While **Profit** from **Measures** is still highlighted, click on the histogram graph button from the **Show Me** toolbar as shown in the following screenshot:



How it works....

As you can see by comparing the number of steps listed in this recipe with the number of steps listed in the earlier histogram recipe, the **Show Me** toolbar reduces the effort required to produce the same graphic. Similarly, you can generate various types of graphs in two clicks. The **Show Me** toolbar also lists the number of measures and the number of dimensions required to create a graphic. If a graphic cannot be drawn with the selected measures or dimensions, the graph button is shaded gray (disabled) to indicate unavailability of the graphic.

Creating stacked bar graphs

In stacked bar graphs, various categories of the same field are plotted on top of each other. One of the biggest problems with the stacked bar graphs is that the length of the bars is hard to measure, except for the bottom bar in the stack. Some people argue that it is good at showing the proportion or comparison of two or more categories; however, if comparison is the objective, there are much better alternatives, such as facets and small multiples, which have higher efficacy in comparing data. If you must use a stacked bar graph, limit the number of stacks to two to three categories and avoid very disproportionate stacks, such as 99 percent and 1 percent stacks.

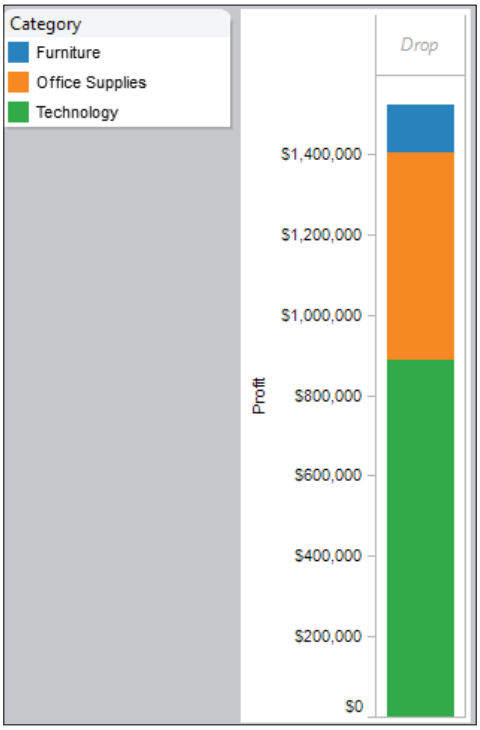
Getting ready

Let's use the sample file, `Sample – Superstore Sales (Excel)`. Open a new worksheet and select **Sample – Superstore Sales (Excel)** as the data source.

How to do it...

Once the data is loaded, perform the following steps to create a simple, one-variable stacked bar graph:

1. Drag-and-drop **Profit** from **Measures** into the **Rows** shelf.
2. Drag-and-drop **Category** under **Products** from **Dimensions** into the **Color** box from the **Marks** pane.
3. Change the **Marks** type to **Automatic**. The resulting graph is shown in the following screenshot:



How it works...

Since we use **Color** to encode the **Category** variable, Tableau automatically assigns the default colors to various **Category** types and, in effect, produces the stacked bar graph. You could also select **Category** from **Dimensions** and **Profit** from **Measures** and click on the stacked bars graph button on the **Show Me** toolbar to create the same graph. As you can see from the graph, the reader has to look at the legend colors to distinguish various **Category** types, thus increasing the difficulty in understanding the information presented. An alternative to this type of chart is creating multiple charts or facets. To create such facets, in this recipe, we can drag-and-drop **Category** into the **Rows** shelf before **Profit**.

Creating box plots

Box and whisker plots, also known as box plots, show the distribution of the observed measure. This distribution includes the 25th, 50th, and 75th percentile as well as the minimum and the maximum values of the measure. A box surrounds the interquartile values of the 25th, 50th, and 75th percentile, and whiskers represent the minimum and the maximum values. Since Tableau does not support creating a box plot directly, this recipe is a workaround to create a box plot in Tableau.

Getting ready

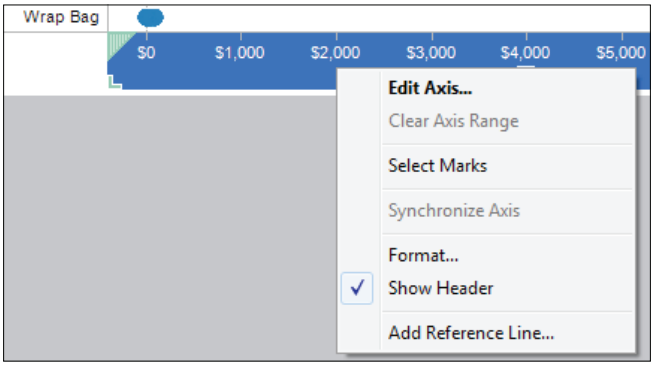
Let's use the sample file, `Sample - Superstore Sales (Excel)`. Open a new worksheet and select **Sample – Superstore Sales (Excel)** as the data source.

How to do it...

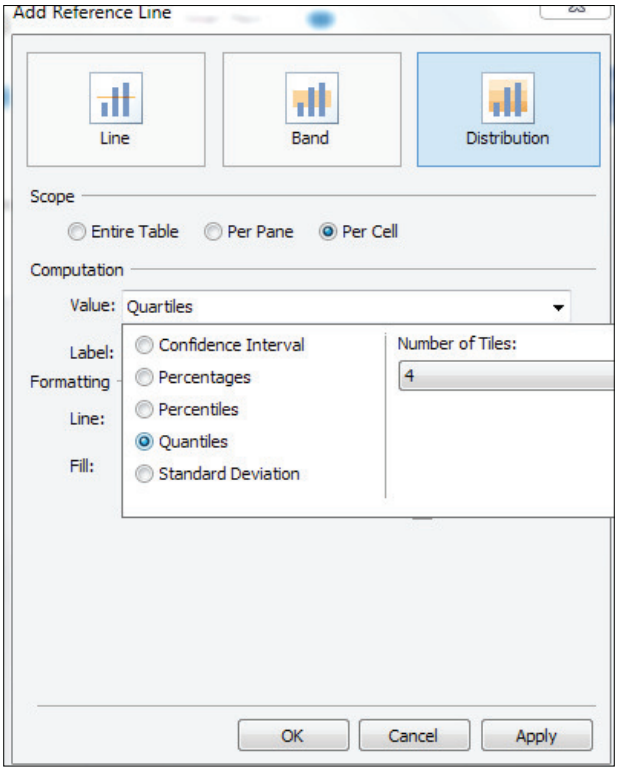
Once the data is loaded, perform the following steps to create a box plot of shipping cost by container types:

1. Drag-and-drop **Container** from **Dimensions** into the **Rows** shelf.
2. Drag-and-drop **Unit Price** from **Measures** into the **Columns** shelf.
3. Click on the **Analysis** menu option from the top toolbar and uncheck **Aggregate Measures** to remove aggregation.

4. Right-click on the x axis and click on **Add Reference Line**, as shown in the following screenshot:

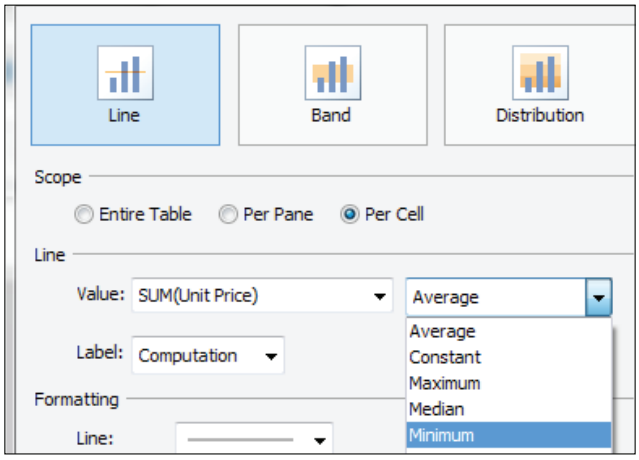


5. Click on the **Distribution** pane in the **Add Reference Line** options box.
6. Select the **Per Cell** button under **Scope**.
7. Under **Computation**, in the **Value** drop-down selection, select **Quantiles** and keep the **Number of Tiles** value to **4**, as shown in the following screenshot:



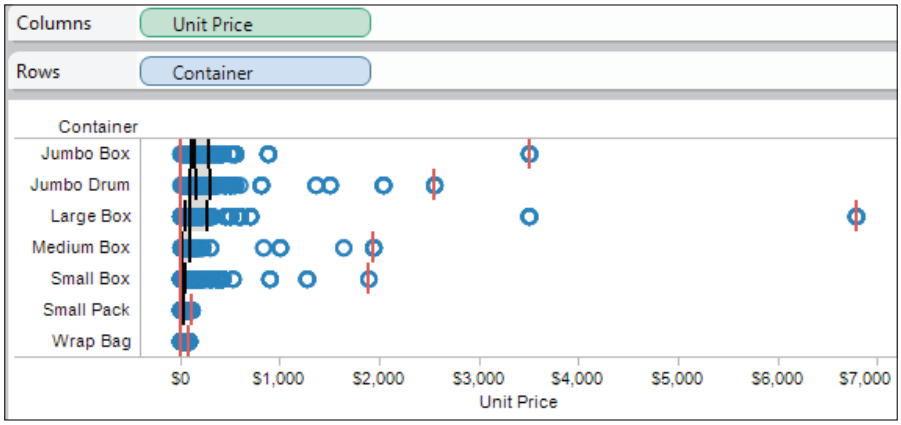
8. Under **Computation**, in the **Label** drop-down selection, select **None**.
9. Under **Formatting**, in the **Line** drop-down selection, select the first thick and solid line.
10. Under **Formatting**, check the **Symmetric** formatting box as shown in the following screenshot:

11. Under **Formatting**, in the **Fill** drop-down selection, keep the default gray color, which is in the first column and fourth from the top.
12. Hit **OK**.
13. To add whiskers for the minimum values, right-click on the x axis and click on **Add Reference Line**.
14. Keep the **Line** pane selected.
15. Select the **Per Cell** option under **Scope**.
16. Under **Line**, in the **Value** dropdowns, select **Minimum** as shown in the following screenshot:



17. Under **Line**, in the **Label** dropdown, select **None**.
18. Under **Formatting**, in the **Line** dropdown, change the line color to red.
19. Hit **OK**.

20. Follow steps 13-19 to add the maximum whiskers, and instead of selecting **Minimum**, select **Maximum** in step 16. The final box plot should look like the one in the following screenshot:



How it works...

Although Tableau does not provide a quick way to create box plots, adding reference lines is a very powerful feature that can be used to create box plots. Adding reference lines can be very useful for the reader to observe trends, distributions, and variance. In the case of box plots, we added reference distributions using quantiles (also known as quartiles) and added minimum-maximum lines.

Showing aggregate measures

Tableau, by default, aggregates measure values, and this behavior can be changed to show all individual values of the measures by clicking on the **Analysis** menu option from the top toolbar and unchecking **Aggregate Measures** to remove aggregation. It is also possible to change the aggregation type, such as total, average, variance, and others.

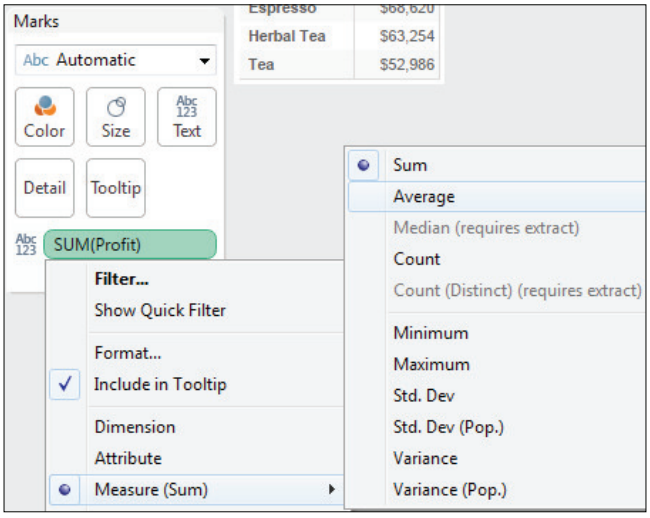
Getting ready

Let's use the sample file `Sample - Coffee Chain (Access)`. Open a new worksheet and select **Sample – Coffee Chain (Access)** as the data source.

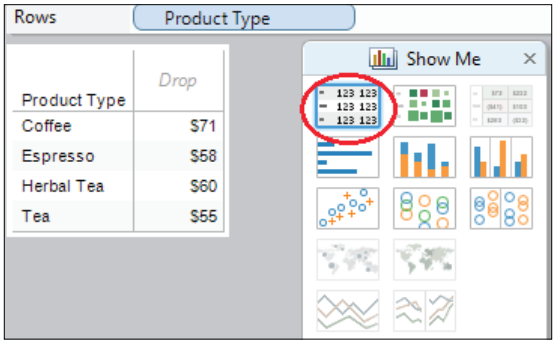
How to do it...

Once the data is loaded, perform the following steps to change and add various aggregate measures:

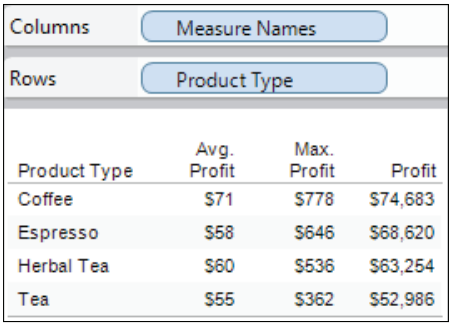
1. Make sure the **Aggregate Measures** option is checked under the **Analysis** menu option on the top toolbar.
2. Drag-and-drop **Product Type** from **Dimensions** into the **Rows** shelf.
3. Drag-and-drop **Profit** from **Measures** into the **Text** input box under **Marks**.
4. To view the average profit by **Product Type**, click on **SUM(Profit)** in the **Text** input box, expand **Measure (Sum)**, and select **Average**, as shown in the following screenshot:



5. Click on the **Show Me** button to bring the **Show Me** toolbar on the screen.
6. To add total profit by **Product Type**, click on **Profit** from **Measures** and click on the text tables icon on the **Show Me** toolbar, as shown in the following screenshot:



7. To add the maximum profit value by **Product Type**, click on **SUM(Profit)** under **Measure Values**, expand **Measure (Sum)**, and select **Maximum**.
8. To add the total profit by **Product Type** again, drag-and-drop **Profit** from **Measures** into the **Measure Values** pane. Once all the aggregate values are added, the table should look like the one in the following screenshot:



Product Type	Avg. Profit	Max. Profit	Profit
Coffee	$71	$778	$74,683
Espresso	$58	$646	$68,620
Herbal Tea	$60	$536	$63,254
Tea	$55	$362	$52,986

How it works...

Adding various aggregate measures of the same measure is somewhat counterintuitive and is hardly straightforward. Since Tableau allows one type of aggregation only once, users must change the aggregation type of the already displayed aggregation and add the measure again. You could also create duplicate copies of the measure and add the new measure with a different aggregation, but you will have to rename the measures.

Showing the top 10 items

At times, it is just easier to view the top 10 items by a certain measure, such as the top 10 most profitable customers or the top 10 least expensive vendors, rather than viewing all the items of a field. This approach, although easier, must be used with caution since anomalous items or patterns could be missed by viewing only the top *n* or bottom *n* items.

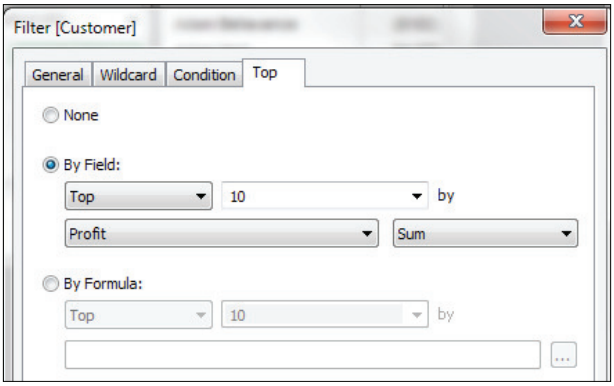
Getting ready

Let's use the sample file `Sample – Superstore Sales (Excel)`. Open a new worksheet and select **Sample – Superstore Sales (Excel)** as the data source.

How to do it...

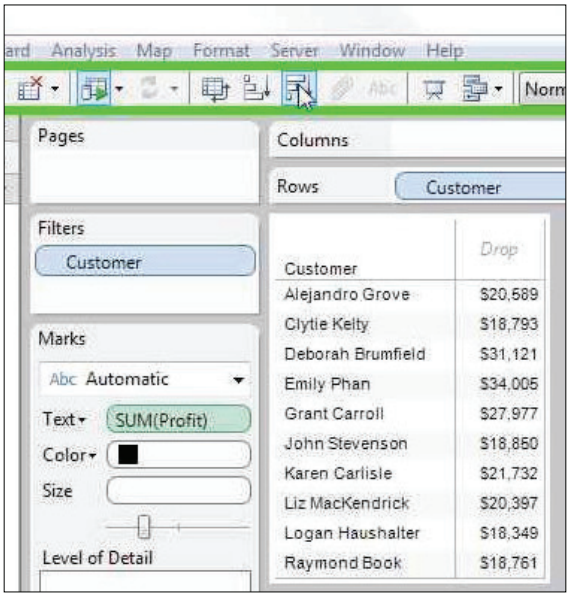
Once the data is loaded, perform the following steps to view a table with the top 10 customers by total profit:

1. Drag-and-drop **Customer** from **Dimensions** into the **Rows** shelf.
2. Drag-and-drop **Profit** from **Measures** into the **Text** input box under **Marks**.
3. Drag-and-drop **Customer** from **Dimensions** into the **Filters** pane.
4. Click on the **Top** tab in the **Filter [Customer]** options box.
5. Check the **By Field** option.
6. Make sure your options (top 10 by **Profit** and aggregation is **Sum**) look like the one in the following screenshot:



7. Click on **OK**.

8. To sort the customer list in descending order of profit (that is, the most profitable customer up at the top), click on the **Sort** button that has the longest bar up at the top and a down arrow to its right, as shown in the following screenshot:



How it works...

Tableau not only provides filtering by item names, but also provides filtering by aggregate measures such as limiting to top or bottom items by sum, average, and other aggregations.

3
Creating Bivariate Charts

An analysis involving two measures is called **bivariate analysis**, and in this chapter we will cover various bivariate charts.

We will be covering the following topics:

- Creating tables
- Creating scatter plots
- Swapping rows and columns
- Adding trend lines
- Selecting color palettes
- Using dates

Introduction

This chapter provides recipes for generating visualizations when using two measures. Such visualizations can help a user with formulating questions that can be answered using data. There are other recipes that manipulate existing data to generate alternative visualizations, such as swapping rows and columns and using color palettes. This chapter also explains how to add trend lines to existing visualizations to extend the effectiveness of a chart.

Creating tables

If you want to present any data with all the details, tables often are a good choice as they retain all the information and reduce the chances of misrepresentation of data. Tables are also effective in presenting data with precision. For example, a reader might get confused by a value of 100.8 in a chart to be 100.5 or 101, but in a table, all values are presented accurately and there is no scope for misinterpretation. Tables are great for smaller number of columns or rows but charts are better suited for complex information.

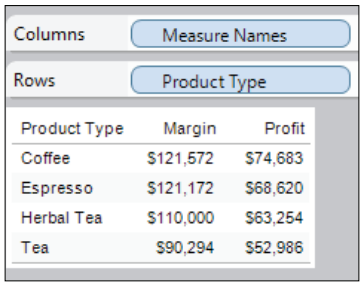
Getting ready

Let's use the sample file `Sample – Coffee Chain (Access)`. Open a new worksheet and select **Sample – Coffee Chain (Access)** as the data source.

How to do it....

Once the data is loaded, perform the following steps to create a table with one dimension and two measures:

1. Drag-and-drop **Product Type** from **Dimensions** into the **Rows** shelf.
2. Click on **Margin** and **Profit** from **Measures**.
3. Click on the **Show Me** button to bring the **Show Me** toolbar on the screen.
4. Click on the text tables icon in the **Show Me** toolbar. Your table should look like the one in the following screenshot:



Product Type	Margin	Profit
Coffee	$121,572	$74,683
Espresso	$121,172	$68,620
Herbal Tea	$110,000	$63,254
Tea	$90,294	$52,986

How it works....

As we click on the two measures and then click on the text tables icon in the **Show Me** toolbar, Tableau will automatically create filters on **Measure Names** to limit the measures to **Margin** and **Profit**, and it will also put the **Measure** names in the **Columns** shelf. By default, Tableau will total the measure value and this can be changed by clicking on the **Measure Values** shelf and changing the aggregate measure to view individual values.

There's more...

You can read up on good arguments when to use tables and when not to use them on the University of Leicester's page: `http://www2.le.ac.uk/offices/ld/resources/numeracy/numerical-data`

Creating scatter plots

Scatter plots are often used to identify any correlation or observe relationships between two variables. By looking at these plots, the reader can quickly observe any trends, if present. A scatter plot is a very useful tool in any analyst's toolbox.

Getting ready

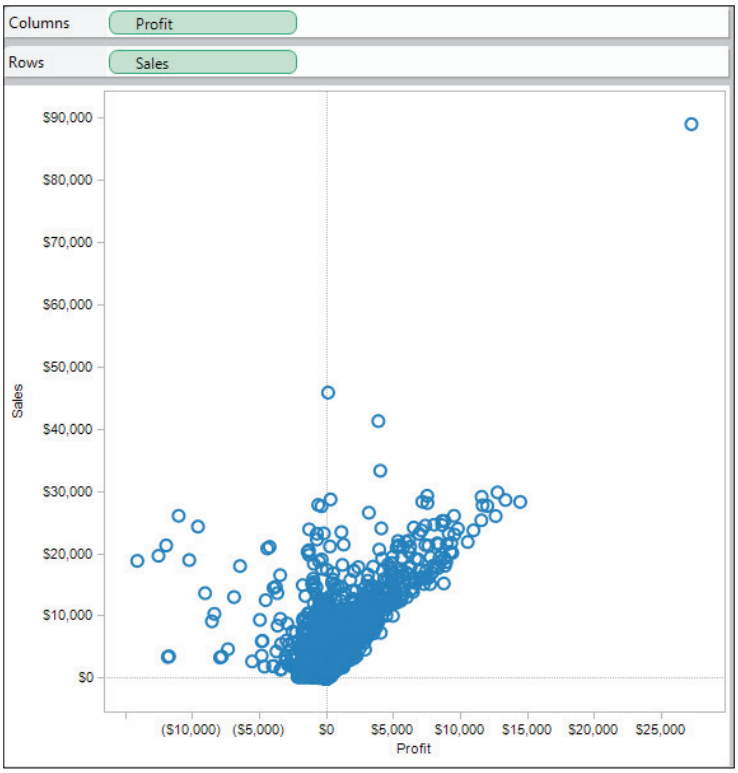
For this recipe, let's use the sample file `Sample – Superstore Sales (Excel)`. Open a new worksheet and select **Sample – Superstore Sales (Excel)** as the data source.

How to do it....

Once the data is loaded, perform the following steps to create a scatter plot of two measures:

1. From the top toolbar, under **Analysis**, uncheck **Aggregate Measures**.
2. Drag-and-drop **Profit** into the **Columns** shelf.

3. Drag-and-drop **Sales** into the **Rows** shelf. The generated scatter plot should look like the one in the following screenshot:



How it works...

By default, Tableau will aggregate measures to show only the aggregated values. In traditional statistics, however, to observe any trends or correlation between two variables, individual data points are plotted across both axes. Therefore, we removed the aggregation for this recipe, but please note that some applications of scatter plots may warrant aggregation.

There's more...

Scatter plots are one of the most common techniques to observe the relationship between two variables. It is important to note, however, that a plot may suggest a correlation between two variables but cannot conclusively prove a causal relationship. You can read more about scatter plots on the **National Institute of Standards and Technology** (**NIST**) exploratory data analysis handbook at `http://www.itl.nist.gov/div898/handbook/eda/section3/scatterp.htm`.

Swapping rows and columns

Sometimes the data points are located in undesirable locations, which makes looking at the numbers slightly challenging. By swapping rows with columns, we can offer a different point of view to the reader.

Getting ready

For this recipe, let's use the same sample file, `Sample - Superstore Sales (Excel)`. Open a new worksheet and select **Sample – Superstore Sales (Excel)** as the data source.

How to do it...

Once the data is loaded, we will follow the steps from the previous recipe to reproduce the scatter plot and then swap the rows with columns using the following steps:

1. From the top toolbar, under **Analysis**, uncheck **Aggregate Measures**.
2. Drag-and-drop **Profit** into the **Columns** shelf.
3. Drag-and-drop **Sales** into the **Rows** shelf.
4. Click on the swap button to place **Sales** in the **Columns** shelf and **Profit** in the **Rows** shelf. The swap button is shown in the following screenshot:

How it works...

Swapping of rows with columns and columns with rows works with almost any type of chart and it is a very useful tool when we want to quickly change the orientation or position of the visualization. Tableau makes it very easy to make such a change.

Adding trend lines

Trend lines are very useful in observing the relationship between two variables as well as predicting future values. Trend lines are frequently used in simple linear regression to observe the relationship between two variables. The shape of the trend line explains the type of the relationship between the variables. For example, in the case of simple linear regression, the trend line is a straight line, which is represented by the mathematical equation of a straight line: $y = mx + c$.

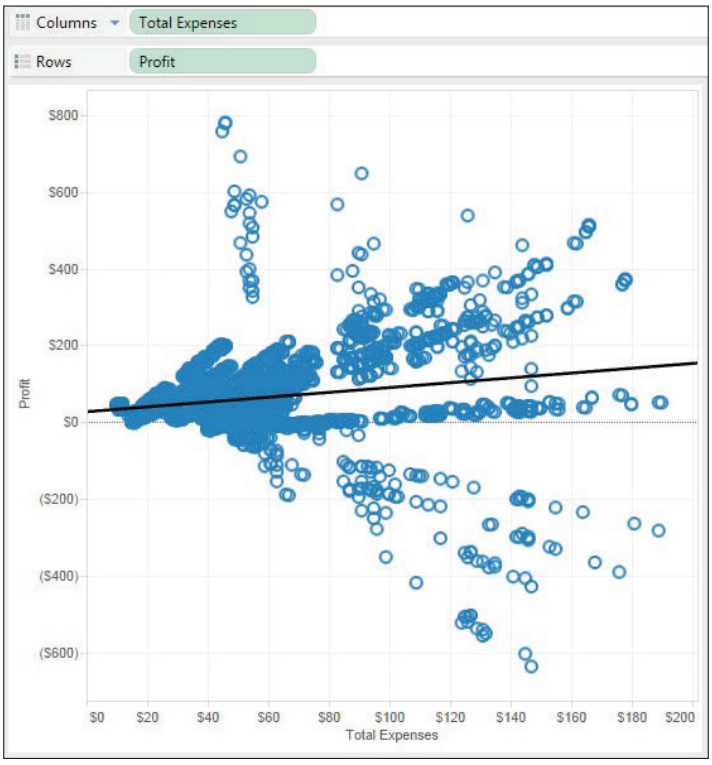
Getting ready

Let's use the sample file `Sample - Coffee Chain (Access)`. Open a new worksheet and select **Sample – Coffee Chain (Access)** as the data source.

How to do it...

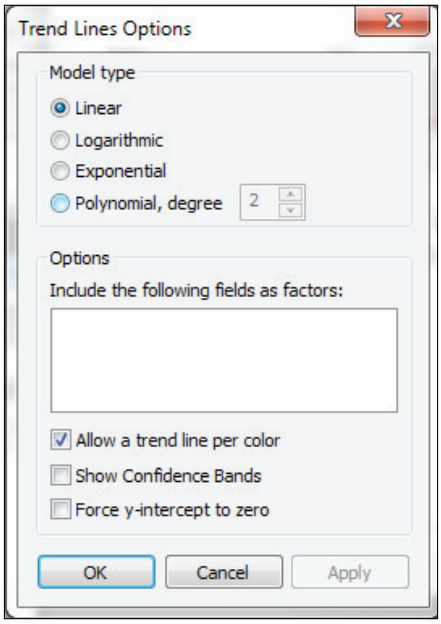
Once the data is loaded, perform the following steps to add a trend line to a plot:

1. Click on the **Show Me** button to bring the **Show Me** toolbar on the screen.
2. Select **Profit** and **Total Expenses** from **Measures by pressing the** *Ctrl* **key and clicking on both the fields.**
3. Click on the scatter plots button from the **Show Me** toolbar.
4. Click on the **Analysis** menu option and uncheck the **Aggregate Measures** option.
5. Right-click on any data marker or anywhere in the plot area and click on the **Show Trend Lines** option in **Trend Lines** to see a plot with a linear trend line, as shown in the following screenshot:



How it works....

The trend line that is added by default is a linear trend line, which is the simplest type of trend line explaining the relationship between two variables, and as we would expect in this recipe, the relationship is quite linear; that is, the profits are generally higher for expensive products. This trend line is mathematically represented as *Profit = 0.628675* otal Expenses + 27.1093*. This trend line can be edited to observe complex relationships, such as logarithmic, exponential, and polynomial. To change the trend line, right-click on the trend line and click on **Edit Trend Line**. You'll see a dialog box with various options, as shown in the following screenshot. From this box, select options from **Model type** to observe which trend line fits the data better:



There's more...

When we fit a linear model trend line on the data, we are essentially performing linear regression to fit a straight line to the data. If you are unfamiliar with linear regression, you will find the following resources helpful:

- The Khan academy video at `http://youtu.be/OhUkMQtBGmE`
- A reference from a Psychology course at Illinois State University: `http://psychology.illinoisstate.edu/jccutti/psych340/fall02/oldlecturefiles/regression.html`

Selecting color palettes

One of the biggest strengths of Tableau is its color selection for various visualizations. This color selection is based on best practices and the concepts of information visualization. Sometimes, however, we may want to change the default color settings.

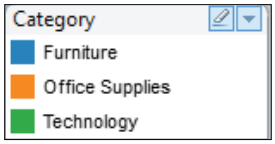
Getting ready

For this recipe, let's use the sample file `Sample – Superstore Sales (Excel)`. Open a new worksheet and select **Sample – Superstore Sales (Excel)** as the data source.

How to do it...

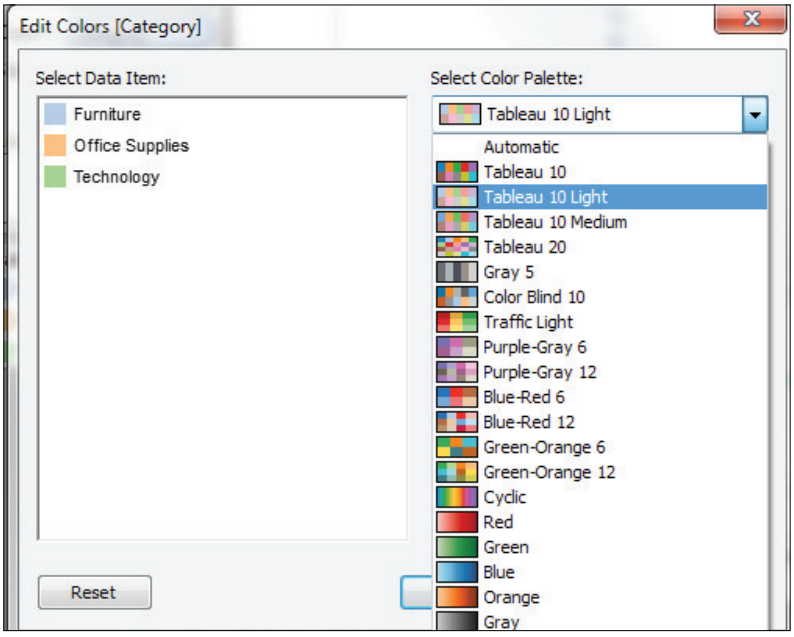
Once the data is loaded, perform the following steps to select different types of color palettes:

1. Click on the **Show Me** button to bring the **Show Me** toolbar on the screen.
2. Select **Discount** and **Profit** from **Measures by pressing the** *Ctrl* **key and clicking on both the fields.**
3. Click on the **Scatter Plots** button from the **Show Me** toolbar.
4. Click on the **Analysis** menu option and uncheck the **Aggregate Measures** option.
5. Drag-and-drop **Category** from **Dimensions** into the **Color** box under the **Marks** pane.
6. Click on the small dropdown, which becomes visible after hovering on the **Category** legend pane, as shown in the following screenshot, or double-click on any legend key:



7. Click on the **Edit Colors** option.

8. In the **Edit Colors [Category]** properties box, under **Select Color Palette**, expand the dropdown and select the **Tableau 10 Light** color palette as shown in the following screenshot:



9. Click on the **Assign Palette** button to assign the colors from this palette to various categories.
10. Click on **OK**.

How it works...

Tableau provides many color palettes to choose from, and these palettes are designed to maximize the effectiveness of colors in the visualizations. It is also possible to change the color of a single legend by selecting a value from the legends and then selecting a color from the palette and clicking on the **Assign Palette** button.

There's more...

If you are interested in learning more about the theory behind using proper colors, you will find the slides on color from the Information Visualization Stat 120 class by *Ross Ihaka* at the University of Auckland very insightful; go to the following link:

`https://www.stat.auckland.ac.nz/~ihaka/120/Lectures/lecture14.pdf`

Using dates

Tableau provides various options to analyze data fields that are of date type. Some options include grouping by quarters, years, or months.

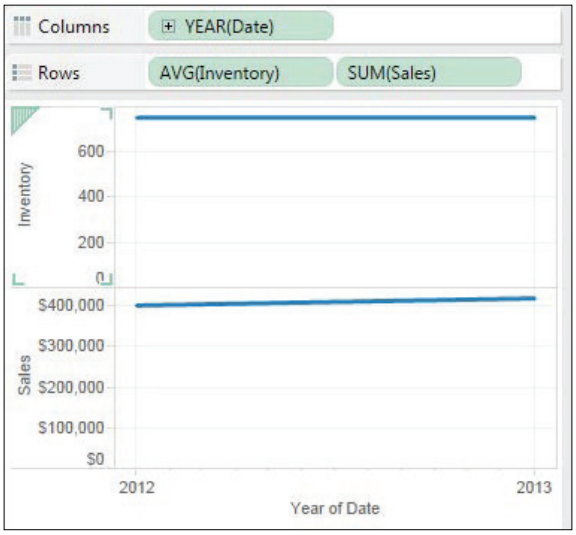
Getting ready

Let's use the sample file `Sample - Coffee Chain (Access)`. Open a new worksheet and select **Sample – Coffee Chain (Access)** as the data source.

How to do it....

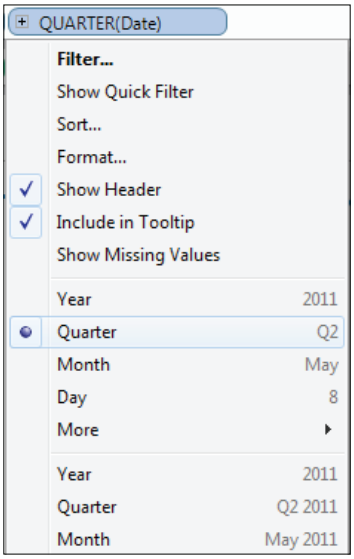
Once the data is loaded, perform the following steps to use dates in your analysis:

1. Click on the **Show Me** button to bring the **Show Me** toolbar on the screen.
2. Select **Inventory** and **Sales** from **Measures** and **Date** from **Dimensions** by pressing the *Ctrl* key and clicking on these fields.
3. Select the **lines (continuous)** chart type from the **Show Me** toolbar to generate a visualization as shown in the following screenshot. The following chart shows us the average inventory size and total sales for the years 2012 and 2013:

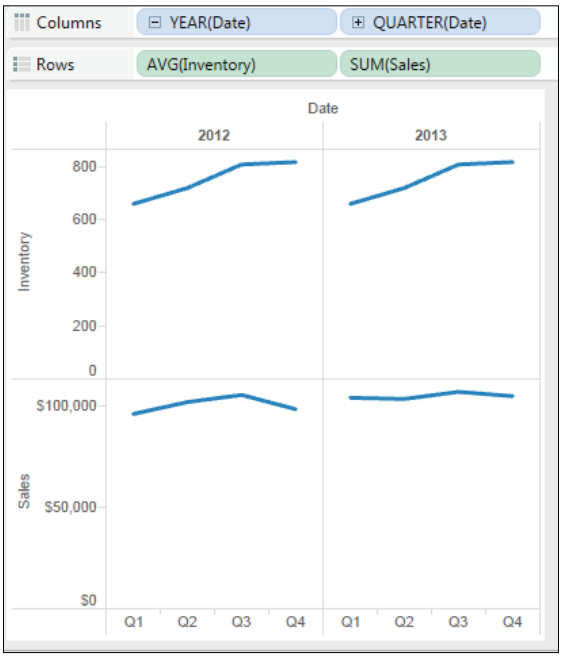


4. To observe the time-series data by the quarters of a year, we need to add the **Date** field again. Drag-and-drop **Date** from **Dimensions** into the **Columns** shelf.
5. In the **Columns** shelf, you'll now see two **Date** fields grouped by year. To show the time-series by quarter, click on the second **YEAR(Date)** value.

6. From the drop-down menu, select the **Quarter** value that has the format **Q1** or **Q2**, as shown in the following screenshot:



7. You can now observe the time-series trend by the quarters of 2012 and 2013, as shown in the following screenshot:



How it works...

Tableau will automatically group a date type of field by year, but it does provide various grouping options, such as quarter, month, day, weekday, and even by hour, minute, and second. To add multiple groupings, we need to add the date field multiple times and change the grouping option. This allows the user to generate data for various types of analyses that generate trends by different date combinations, thus assisting the reader to observe micro or macro trends. We saw the different time-series trends; when the data was grouped only by year, we observed that the time-series lines were almost flat. However, when we added the quarters, we observed that the average inventory went up in the third quarter of 2012 but the total sales went down in the same time period.

4

Creating Multivariate Charts

Multivariate analysis involves analyzing multiple measures. In this chapter, we will create graphs that can effectively visualize multiple measures.

We will cover the following topics:

- Creating facets
- Creating area charts
- Creating bullet graphs
- Creating dual axes charts
- Creating Gantt charts
- Creating heat maps

Introduction

With increasing number of variables, any analysis can become challenging and any observations harder; however, Tableau simplifies the process for the designer and uses effective layouts for the reader even in multivariate analysis. Using various combinations of colors and charts, we can create compelling graphics that generate critical insights from our data. Among the charts covered in this chapter, facets and area charts are easier to understand and easier to create compared to bullet graphs and dual axes charts.

Creating facets

Facets are one of the powerful features in Tableau. *Edward Tufte*, a pioneer in the field of information graphics, championed these types of charts, also called **grid** or **panel charts**; he called them small multiples. These charts show the same measure(s) across various values of one or two variables for easier comparison.

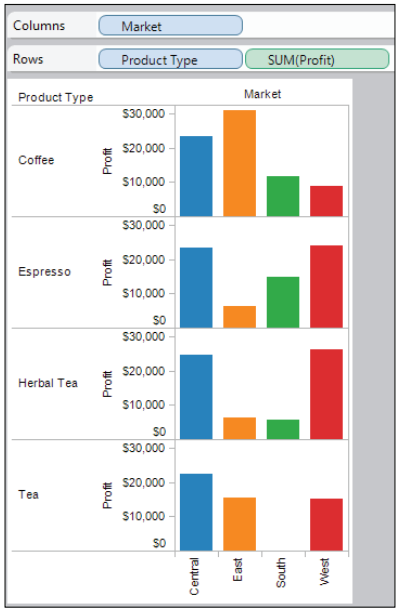
Getting ready

Let's use the sample file `Sample - Coffee Chain (Access)`. Open a new worksheet and select **Sample – Coffee Chain (Access)** as the data source.

How to do it...

Once the data file is loaded on the new worksheet, perform the following steps to create a simple faceted chart:

1. Drag-and-drop **Market** from **Dimensions** into the **Columns** shelf.
2. Drag-and-drop **Product Type** from **Dimensions** into the **Rows** shelf.
3. Drag-and-drop **Profit** from **Measures** into the **Rows** shelf next to **Product Type**.
4. Optionally, you can drag-and-drop **Market** into the **Color Marks** box to give color to the four bars of different **Market** areas. The chart should look like the one in the following screenshot:



How it works...

When there is one dimension on one of the shelves, either **Columns** or **Rows**, and one measure on the other shelf, Tableau creates a univariate bar chart, but when we drop additional dimensions along with the measure, Tableau creates small charts or facets and displays univariate charts broken down by a dimension.

There's more...

A company named Juice Analytics has a great blog article on the topic of small multiples. This article lists the benefits of using small multiples as well as some examples of small multiples in practice. Find this blog at `http://www.juiceanalytics.com/writing/better-know-visualization-small-multiples/`.

Creating area charts

An area chart is an extension of a line chart. The area chart shows the line of the measure but fills the area below the line to emphasize on the value of the measure. A special case of area chart is a stacked area chart, which shows a line per measure and the area between the lines is filled. Tableau's implementation of area charts uses one date variable and one or more measures.

Getting ready

Let's use the sample file `Sample – Superstore Sales (Excel)`. Open a new worksheet and select **Sample – Superstore Sales (Excel)** as the data source.

How to do it...

Once the data is loaded on the new worksheet, perform the following steps to create an area chart:

1. Click on the **Show Me** button to bring the **Show Me** toolbar on the screen.
2. Select **Order Date** from **Dimensions** and **Order Quantity** from **Measures** by clicking and holding the *Ctrl* key.
3. Click on **Area charts (continuous)** from the **Show Me** toolbar.
4. Drag-and-drop **Order Date** into the **Columns** shelf next to **YEAR(Order Date)**.
5. Expand **YEAR(Order Date)**, seen on the right-hand side, by clicking on the plus sign.

6. Drag-and-drop **Region** from **Dimensions** into the the **Rows** shelf to the left of **SUM(Order Quantity)**. The chart should look like the one in the following screenshot:

How it works...

When we added **Order Date** for the first time, Tableau, by default, aggregated the date field by year; therefore, we added **Order Date** again to create aggregation by quarter of the **Order Date**. We also added **Region** to create facets on the regions that provide trends of order quantity over time.

There's more...

A blog post by `visual.ly`, an information graphics company, discusses the key differences between line charts and area charts. You can find this post at `http://blog.visual.ly/line-vs-area-charts/`.

Creating bullet graphs

Stephen Few, an information visualization consultant and author, designed this chart to solve some of the problems that the gauges and meters type of charts poses. Gauges, although simple to understand, take a lot of space to show only one measure. Bullet graphs are a combination of the bar graph and thermometer types of charts, and they show a measure of interest in the form of a bar graph (which is the bullet) and target variables.

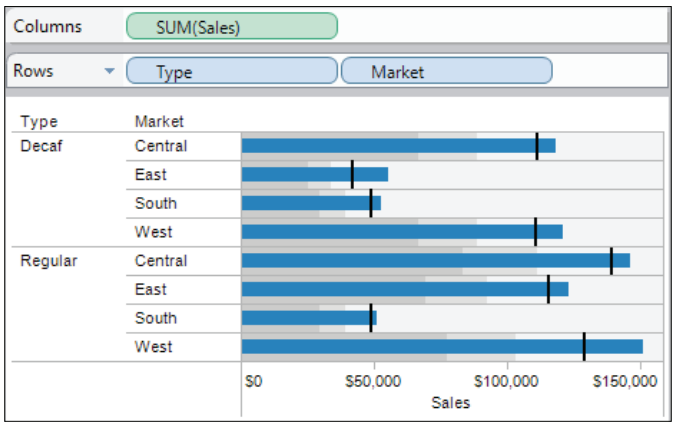
Getting ready

Let's use the sample file `Sample - Coffee Chain (Access)`. Open a new worksheet and select **Sample – Coffee Chain (Access)** as the data source.

How to do it...

Once the data is loaded on the sheet, perform the following steps to create a bullet graph:

1. Click on the **Show Me** button to bring the **Show Me** toolbar on the screen.
2. While holding the *Ctrl* key, click on **Type** and **Market** from **Dimensions** and **Budget Sales** and **Sales** from **Measures**.
3. Click on the bullet graphs icon on the **Show Me** toolbar.
4. Right-click on the x axis (the **Budget Sales** axis) and click on **Swap Reference Line Fields**. The final chart should look like the one in the following screenshot:



How it works...

Although bullet graphs maximize the available space to show relevant information, readers require detailed explanation as to what all the components of the graphic are encoding. In this recipe, since we want to compare the budgeted sales with the actual sales, we had to swap the reference line from **Sales** to **Budget Sales**. The black bar on the graphic shows the budgeted sales and the blue bar shows the actual sales. The dark gray background color shows 60 percent of the actual sales and the lighter gray shows 80 percent of the actual sales. As we can see in this chart, blue bars crossed all the black lines, and that tells us that both the coffee types and all market regions exceeded the budgeted sales.

There's more...

A blog post by Data Pig Technologies discusses some of the problems with the bullet graph. The main problem is intuitive understanding of this chart. You can read about this problem and the reply by *Stephen Few* at `http://datapigtechnologies.com/blog/index.php/the-good-and-bad-of-bullet-graphs/`.

Creating dual axes charts

Dual axes charts are useful to compare two similar types of measures that may have different types of measurement units, such as pounds and dollars. In this recipe, we will look at the dual axes chart.

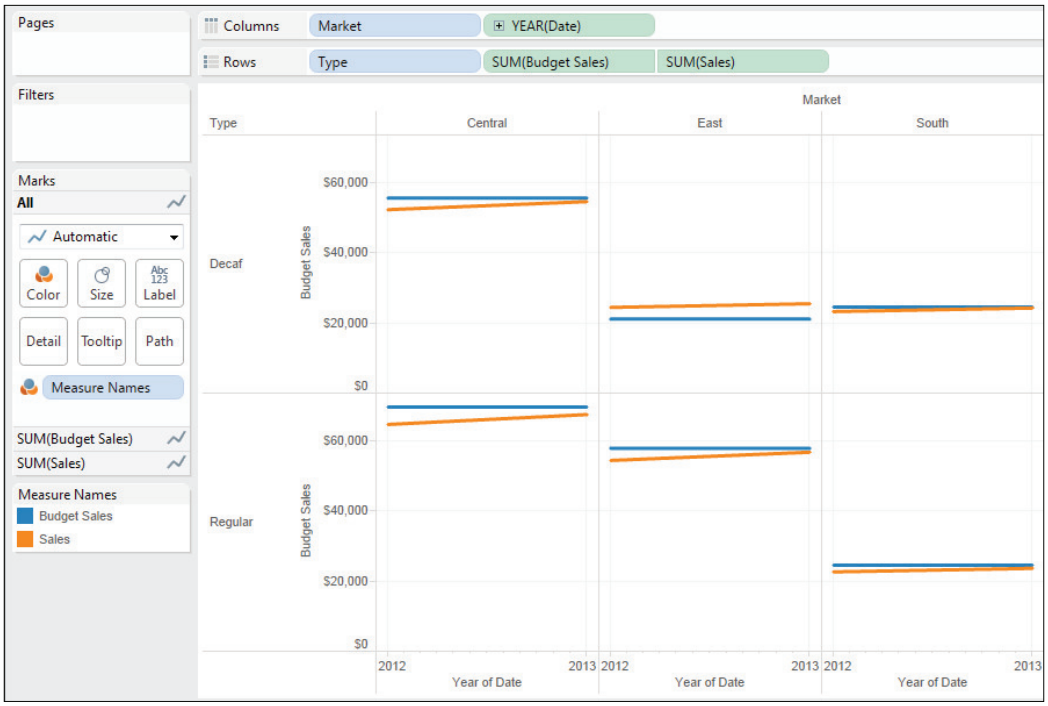
Getting ready

Let's use the same sample file, `Sample – Coffee Chain (Access)`. Open a new worksheet and select **Sample – Coffee Chain (Access)** as the data source.

How to do it...

Once the data is loaded on the sheet, perform the following steps to create a dual axes chart:

1. Click on the **Show Me** button to bring the **Show Me** toolbar on the screen.
2. While holding the *Ctrl* key, click on **Date**, **Type**, and **Market** from **Dimensions** and **Sales** and **Budget Sales** from **Measures**.
3. Click on the dual line graph icon on the **Show Me** toolbar.
4. Click-and-drag **Market** from the **Rows** shelf into the **Columns** shelf.
5. Right-click on the **Sales** vertical axis and click on **Synchronize Axis**. The chart should look like the one shown in the following screenshot:



How it works...

Tableau will create two vertical axes and automatically place **Sales** on one dual axes charts vertical axis and **Budget Sales** on the other. The scales on both the vertical axes are different, however. By synchronizing the axes, we get the same scales on both axes for better comparison and accurate representation of the patterns.

Creating Gantt charts

Gantt charts are most commonly used in project management as these charts show various activities and tasks with the time required to complete those tasks. Gantt charts are even more useful when they show dependencies among various tasks. This type of chart is very helpful when the number of activities is low (around 20-30), otherwise the chart becomes too big to be understood easily.

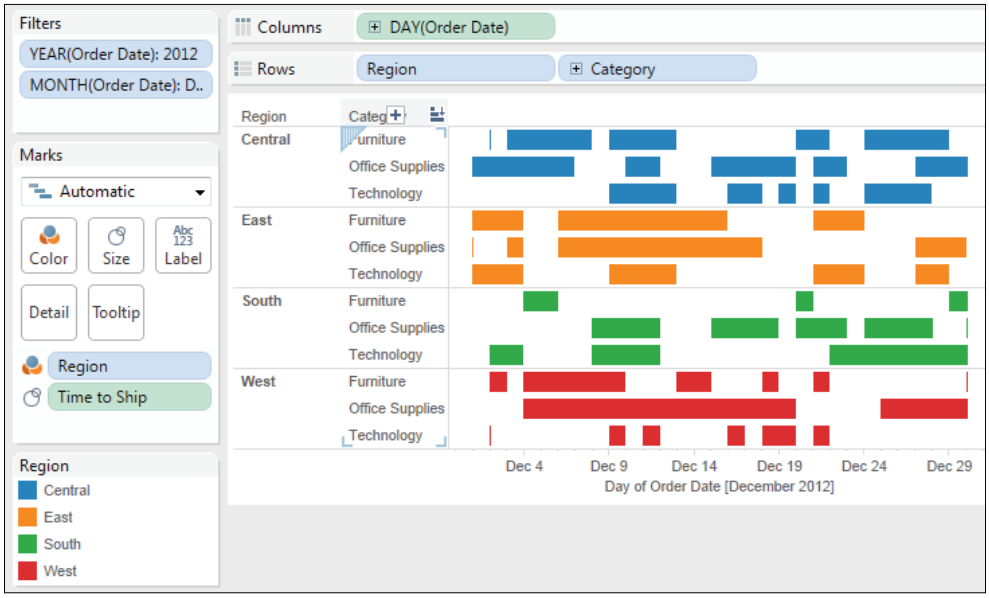
Getting ready

Let's use the sample file `Sample - Superstore Sales (Excel)`. Open a new worksheet and select **Sample – Superstore Sales (Excel)** as the data source.

How to do it...

Once the data is loaded, perform the following steps to create a Gantt chart:

1. Click on **Analysis** from the top menu toolbar, and if **Aggregate Measures** is checked, click on it again to uncheck that option.
2. Click on the **Show Me** button to bring the **Show Me** toolbar on the screen.
3. While holding the *Ctrl* key, click on **Order Date** and **Category** (under **Products**) from **Dimensions** and **Time to Ship** from **Measures**.
4. Click on the Gantt chart icon on the **Show Me** toolbar.
5. Drag-and-drop **Order Date** into the **Filters** pane.
6. Select **Years** from the **Filter Field [Order Date]** options dialog box and hit **Next**.
7. Check **2012** from the list and hit **OK**.
8. Right-click on **YEAR(Order Date)** on the **Columns** shelf and select the **Day May 8, 2011** option.
9. Drag-and-drop **Order Date** into the **Filters** pane.
10. Select **Months** from the **Filter Field [Order Date]** options dialog box and hit **Next**.
11. Check **December** from the list and hit **OK**.
12. Drag-and-drop **Region** from **Dimensions** into the **Color Marks** input box.
13. Drag-and-drop **Region** from **Dimensions** into the **Rows** shelf before **Category**. The generated Gantt chart should look like the one in the following screenshot:



How it works...

Representing time this way helps the reader to discern which activity took the longest amount of time. We added the **Order Date** field two times in the **Filters** pane to first filter for the year 2012 and then for the month of December. In this recipe, out of all the products shipped in December of 2012, we can easily see the red bars for the **West** region in the **Office Supplies** category is longer, suggesting that these products took the longest amount of time to ship.

There's more...

Andy Kriebel, a Tableau data visualization expert, has a great example of Gantt charts using US presidential data. The following link shows the lengths of terms in office of Presidents from various parties:

`http://vizwiz.blogspot.com/2010/09/tableau-tip-creating-waterfall-chart.html`

Creating heat maps

A heat map is a visual representation of numbers in a table or a grid such that the bigger numbers are encoded by darker colors or bigger sizes and the smaller numbers by lighter colors or smaller sizes. This type of representation makes the reader's pattern detection from the data easier.

Getting ready

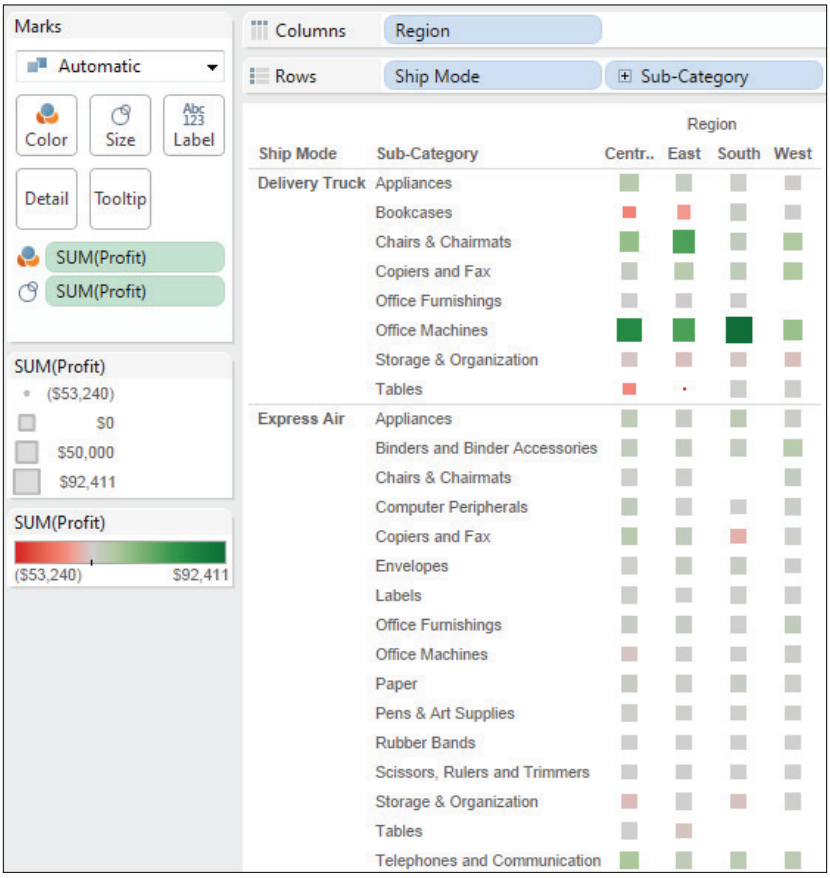
Let's use the same sample file, `Sample – Superstore Sales (Excel)`. Open a new worksheet and select **Sample – Superstore Sales (Excel)** as the data source.

How to do it...

Once the data is loaded, perform the following steps to create a heat map chart:

1. Click on the **Show Me** button to bring the **Show Me** toolbar on the screen.
2. While holding the *Ctrl* key, click on **Sub-Category** (under **Products**), **Region**, and **Ship Mode** from **Dimensions** and **Profit** from **Measures**.
3. Click on the heat maps chart icon on the **Show Me** toolbar.

4. Drag-and-drop **Profit** from **Measures** into the **Color Marks** box. The generated chart should look like the one in the following screenshot:



How it works...

When we created the chart for the first time, Tableau assigned various sizes to the square boxes, but when we placed **Profit** as a color mark, red was used for low amounts of profit and green was used for higher amounts of profit. This made spotting of patterns very easy. **Binders and Binder Accessories**, shipped by **Regular Air** in the **Central** region, generated very high amounts of profit and **Tables**, shipped by **Delivery Trucks** in the **East** region, generated very low amounts of profit (it actually created losses for the company).

5
Creating Maps

In this chapter, we will cover the following recipes:

- Setting geographic roles
- Placing marks on a map
- Overlaying demographic data
- Creating choropleth maps
- Using polygon shapes
- Customizing maps

Introduction

Overlaying information on top of maps allows the readers to understand and observe data by various regions and geographic boundaries. In some other software, creating such maps would be a time-consuming task; in Tableau, however, it is very straightforward. Although seeing dense data in a map could confuse readers, Tableau provides a couple of options to create insightful maps with the use of colors, shapes, and sizes.

Setting geographic roles

Once the data is loaded, Tableau will determine geographic fields using the field names, such as city, state, and zip code, and will generate latitude/longitude data for those fields. Tableau will denote the geographic fields by placing a globe symbol next to the field name. If Tableau misses the detection of any field, usually due to variations in field names, we can manually set these fields as geographic fields.

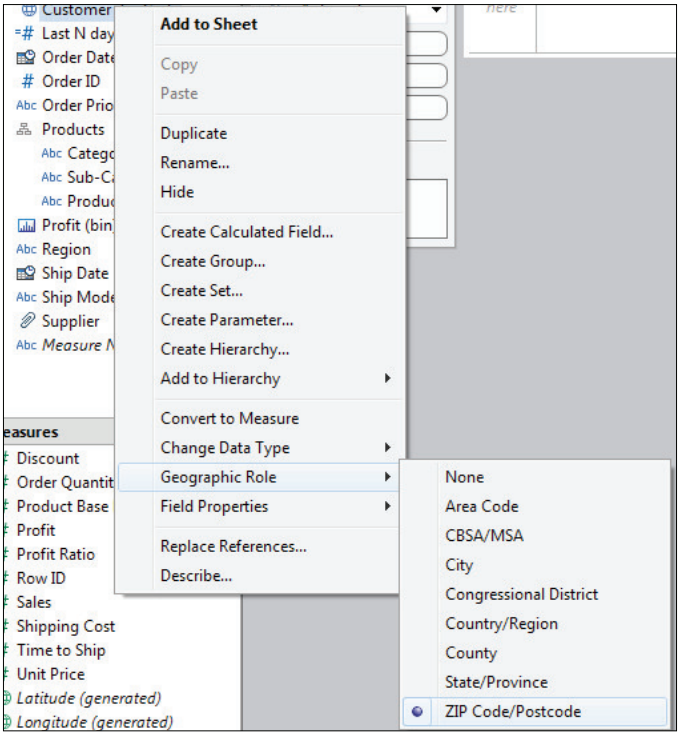
Getting ready

Let's use the sample file `Sample - Superstore Sales (Excel)`. Open a new worksheet and select **Sample – Superstore Sales (Excel)** as the data source.

How to do it...

Once the data is loaded in Tableau, perform the following steps to set a few fields as geographic fields:

1. Right-click on **Customer Zip Code** from **Dimensions**.
2. Expand **Geographic Role** from the dropdown.
3. Select **Zip code/Postcode** as shown in the following screenshot.
4. Right-click on **State** under **Customer City** from **Dimensions**.
5. Expand **Geographic Role** from the dropdown.
6. Select **State/Province** from the options.



How it works...

When the field names are different from conventional names, Tableau will not know that these fields are geographic fields. We can manually assign various fields as geographic fields, which generate latitude and longitude to be used in the maps. Sometimes Tableau cannot match a field to its internal data; for example, if there's a state called UH in our data, but Tableau does not have that value in the list of states of the US. In such cases, Tableau will prompt for mapping of the values from the data.

There's more...

In Tableau's documentation on geographic roles, you can find out which fields Tableau can geocode automatically; browse to `http://onlinehelp.tableausoftware.com/v8.0/pro/online/en-us/maps_geographicroles.html`. Geocoding of fields with information on area code, CBSA/MSA, congressional district, and county are limited to the US only.

Placing marks on a map

One of the ways to encode information on a map is placing a mark for each geographic value and adjusting the size/color of that mark based on some measure. This is the most common type of a map with information used in businesses and media.

Getting ready

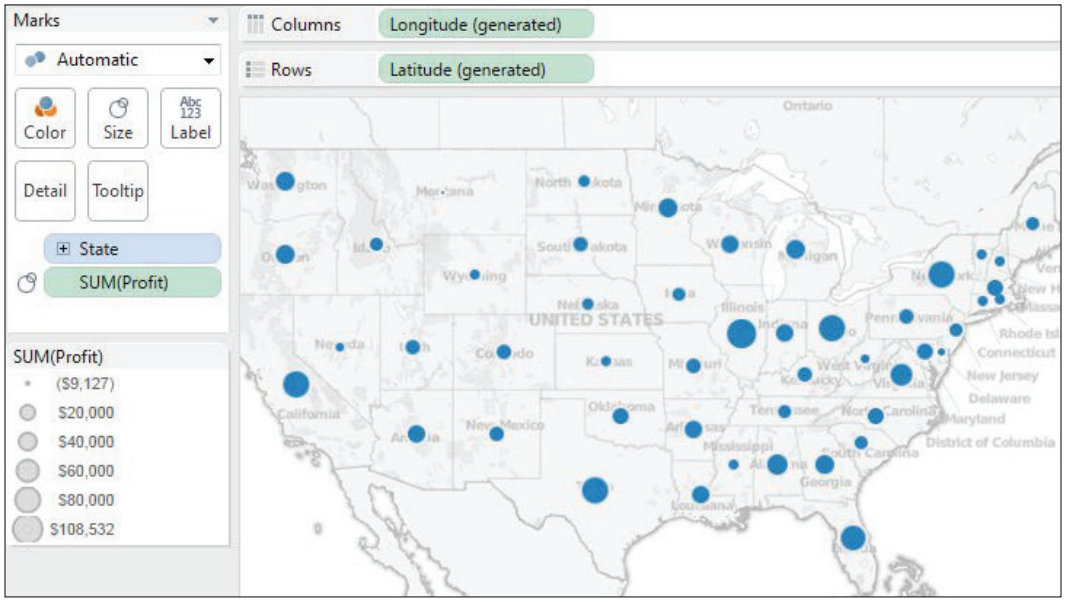
Let's use the sample file, `Sample – Superstore Sales (Excel)`. Open a new worksheet and select **Sample – Superstore Sales (Excel)** as the data source.

How to do it...

Once the data is loaded, perform the following steps to create a map with markers encoding information of a measure:

1. Click on the **Show Me** button to bring the **Show Me** toolbar on the screen.
2. While holding the *Ctrl* key, click on **State** (under **Customer City**) from **Dimensions** and **Profit** from **Measures**.

3. Click on the symbol of maps on the **Show Me** toolbar, and you will see the map as shown in the following screenshot:



How it works...

Once we loaded the data, Tableau assigned geographic roles to **State**, **City**, and **Customer Zip Code** using field names. Tableau also generated **Latitude** and **Longitude** for these geographic roles. When we clicked on the symbol of maps, Tableau automatically added **State** to the **Level of Detail** pane and placed marks for every state that was present in the data. If you add **Customer Zip Code** to the **Detail box**, Tableau will generate markers by zip code and adjust the mark size for the **Profit** value of that zip code.

Overlaying demographic data

Using data from various service providers, Tableau provides a powerful feature of overlaying the US census information, such as median household income, population, race, and others. This allows the reader to compare the measure of interest with some demographic information.

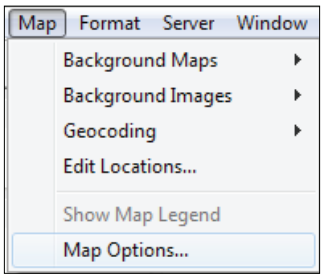
Getting ready

Let's use the sample file, `Sample – Superstore Sales (Excel)`. Open a new worksheet and select **Sample – Superstore Sales (Excel)** as the data source.

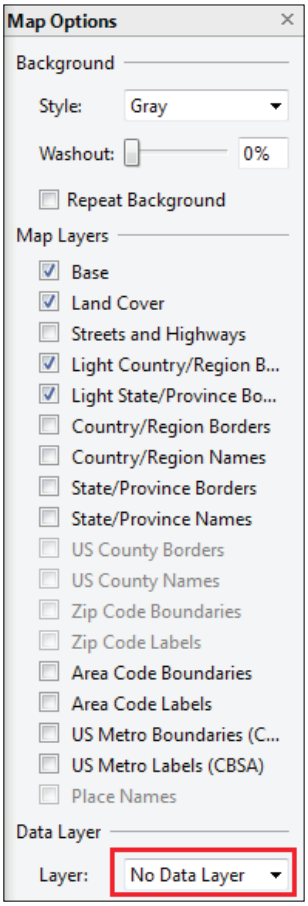
How to do it...

Once the data is loaded on the worksheet, perform the following steps to overlay the demographic data:

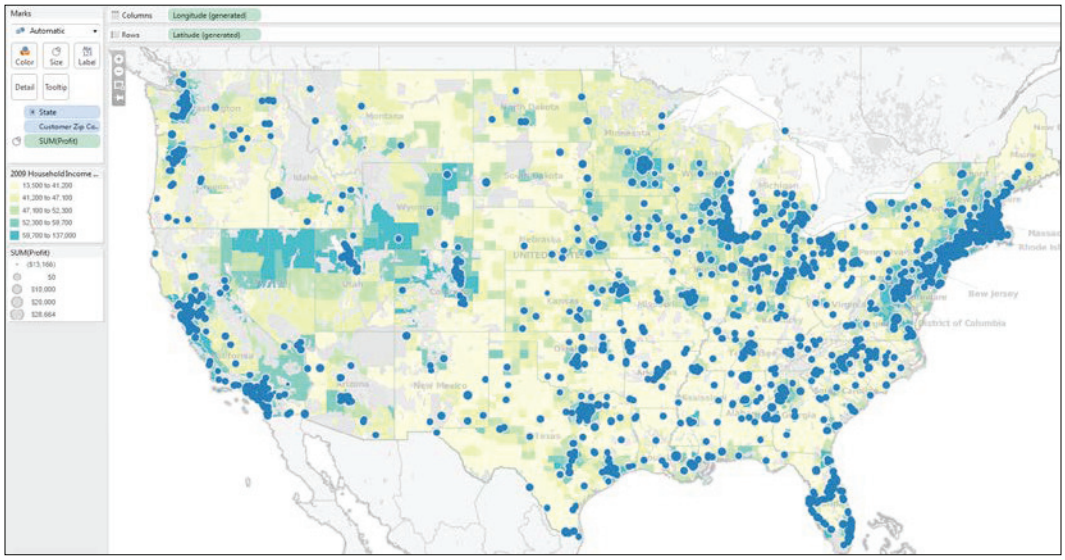
1. Click on the **Show Me** button to bring the **Show Me** toolbar on the screen.
2. While holding the *Ctrl* key, click on **State and Customer Zip Code** from **Dimensions** and **Profit** from **Measures**.
3. Click on the symbol of maps on the **Show Me** toolbar.
4. Drag-and-drop **State** and **Customer Zip Code** from **Dimensions** into the **Level of Detail** box.
5. From the main menu toolbar, click on **Map** and then on **Map Options**, as shown in the following screenshot:



6. From the **Map Options** (shown in the following screenshot) box, expand the dropdown **No Data Layer**.



7. Select **Household Income (median)** under **US Households**.
8. From the **By:** drop-down option, select **County** to generate a map similar to the one shown in the following screenshot:



How it works...

Based on our choice of aggregation level for overlaying demographic information, Tableau fills the level (that is, county, state, zip code, and the block group) with the selected demographic information. In our recipe, we are comparing the median household income of counties with the profits generated by each zip code in the data. We can see that there are many counties in **Nevada** with high median household income, but hardly any profit-generating zip codes. By such comparisons, the reader can identify areas of growth.

There's more...

Tableau provides these powerful features using various data providers listed on Tableau's website: `http://www.tableausoftware.com/mapdata`. If these maps do not meet your needs, Tableau also has the option of using an open technology called **Web Map Services** (**WMS**) to get the map source. You can read about using a WMS server at `http://onlinehelp.tableausoftware.com/current/pro/online/en-us/maps_mapsources_wms.html`.

Creating choropleth maps

A choropleth map, known as a **filled map** in Tableau, is a modification of a traditional marks map, in that study areas (regions, states, and counties) are filled with the measure of interest and colors are used with different hues or diverging progression to assist the reader in identifying areas of poor or good performance.

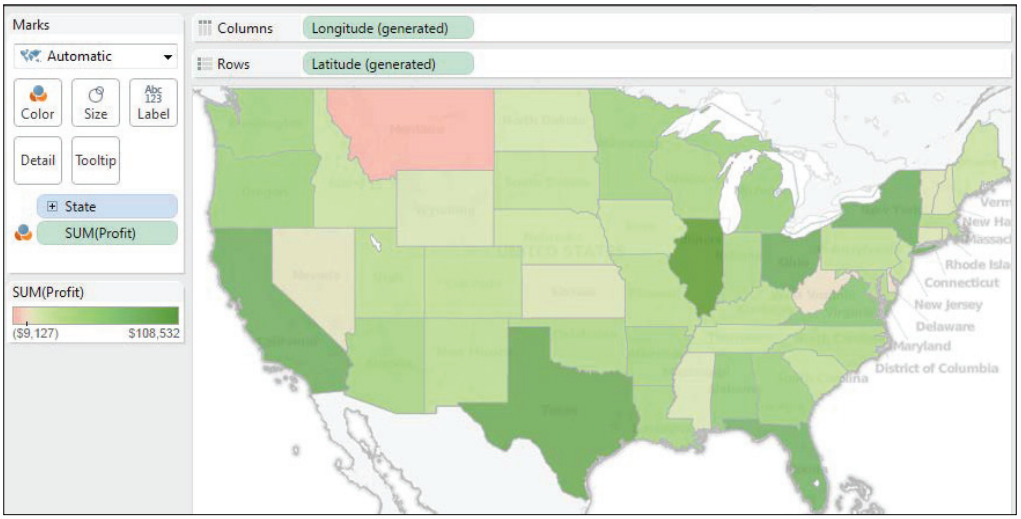
Getting ready

Let's use the sample file, `Sample - Superstore Sales (Excel)`. Open a new worksheet and select **Sample – Superstore Sales (Excel)** as the data source.

How to do it...

Once the data is loaded on the worksheet, perform the following steps to create a choropleth or a filled map:

1. Click on the **Show Me** button to bring the **Show Me** toolbar on the screen.
2. While holding the *Ctrl* key, click on **Profit** from **Measures** and **State** from **Dimensions**.
3. Click on **filled maps** on the **Show Me** toolbar to create a choropleth map as shown in the following screenshot:



How it works...

We selected **State** as the level of detail for this example of choropleth maps (known as filled maps in Tableau), and Tableau created a color range to encode **Profit**; that is, dark green for higher profit and pink for losses. This allows the reader to quickly identify **Montana** as a state with losses and **West Virginia** and **Nevada**, among others, as states with lower profit margins. It is slightly challenging, however, to identify which states grossed the highest profit, as hues of the green color at a higher profit level look very similar, and that is one of the main disadvantages of using this type of map. An alternative to overcome this problem will be creating a sorted bar chart, which, if plotted correctly, will help the reader identify similar states without any guesswork.

Using polygon shapes

Tableau 7 and above have a functionality to create filled maps using fill map marks. These marks are useful when your levels of detail (or the shape of the filled area) are limited to the US counties or county/state combinations. If you want to create custom-shaded maps using geographical boundaries, such as districts of India, you have to use the following steps. You can create such filled maps using a polygon file, which consists of latitude and longitude of various points on the boundaries or shapes of the custom region. An example of such a file is shown in the following screenshot:

```
"long","lat","group","order","region","subregion"
-90.4127349853516,46.5585517883301,1,1,"michigan","north"
-90.3783645629883,46.5642776489258,1,2,"michigan","north"
-90.3153381347656,46.5929260253906,1,3,"michigan","north"
-90.2809600830078,46.6158447265625,1,4,"michigan","north"
-90.2007522583008,46.6330337524414,1,5,"michigan","north"
-90.1377182006836,46.6444969177246,1,6,"michigan","north"
-90.0746994018555,46.6559524536133,1,7,"michigan","north"
-90.0231323242188,46.6788711547852,1,8,"michigan","north"
-89.9543762207031,46.713249206543,1,9,"michigan","north"
-89.9028091430664,46.7418975830078,1,10,"michigan","north"
-89.8741607666016,46.7648162841797,1,11,"michigan","north"
-89.8397827148438,46.7934608459473,1,12,"michigan","north"
```

Getting ready

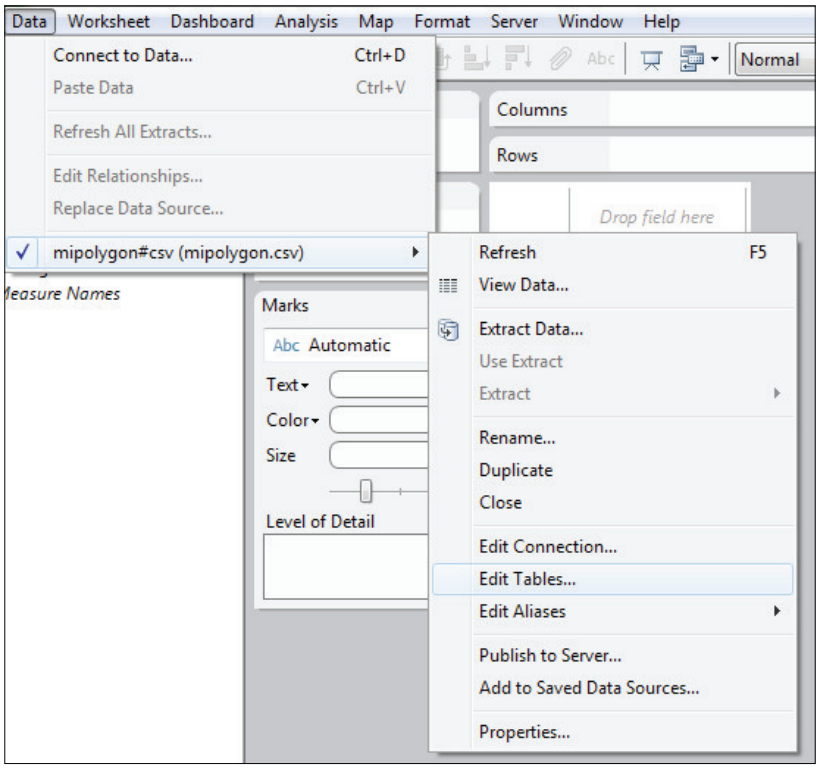
We'll plot the sales figures for the **Upper Peninsula** (**UP**) and **Lower Peninsula** (**LP**) of Michigan. To do so, we'll need two files: a file that contains latitude, longitude, and boundary groupings for Michigan (`mipolygon.csv`) and a file that contains sales numbers for the UP and LP (`misales.csv`). These files are provided with the downloadable code accompanying this book. In the following steps, we will join these two files to get all the required data and use the `mipolygon`.csv file to form the boundaries and the `misales`.csv file to fill the map.

How to do it...

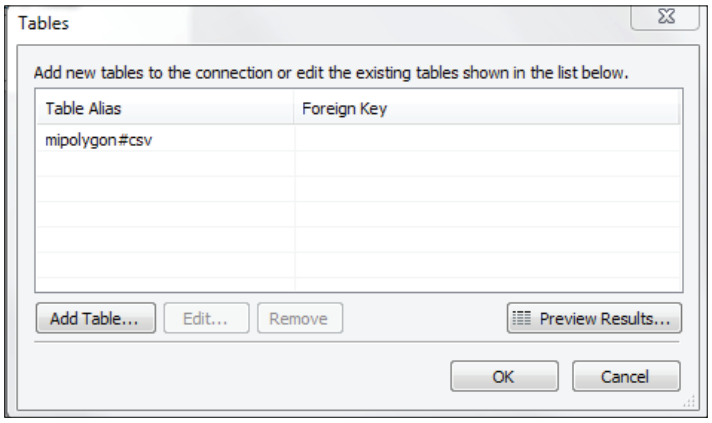
After you download and save `mipolygon.csv` and `misales.csv` locally, perform the following steps to create a map of Michigan with the UP and LP regions filled with the sales figures:

1. Open a new workbook by clicking on **New** under the **File** menu.
2. Click on **Connect to Data** to select the data file.
3. Under the **In a file** options, click on **Text File**.
4. Select `mipolygon.csv` from your downloaded file's location and hit **Open**.
5. Maintain all the default options and hit **OK**.
6. If you see a **Data Connection** option dialog box, select **Connect live**.

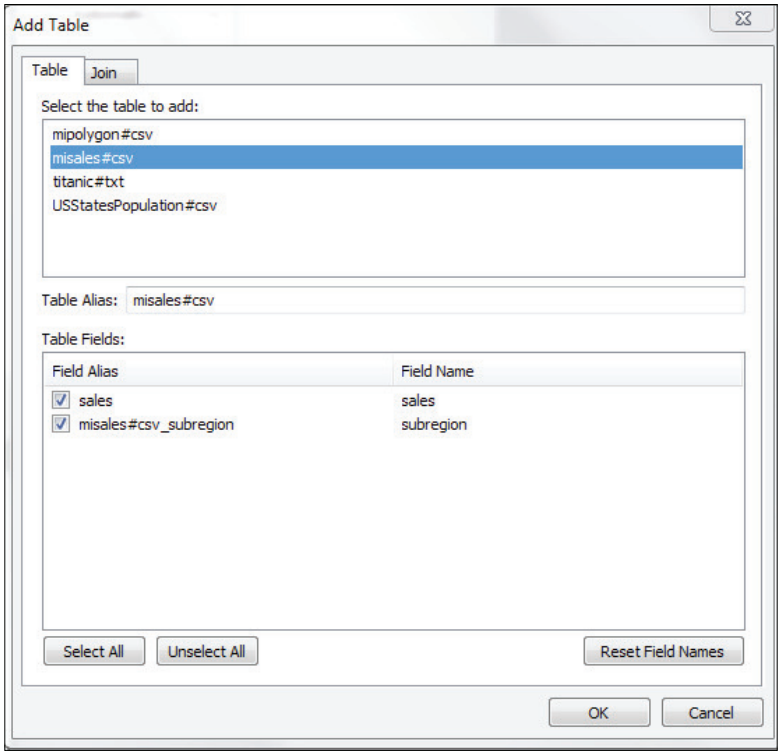
7. While holding the *Ctrl* key, click on **Group** and **Order** under **Measures** and then right-click on **Convert to Dimension**. Alternatively, you can drag-and-drop these fields into the **Dimensions** pane.
8. You'll notice that Tableau recognizes the measure **lat** as a geographic field as the field name matches Tableau's internal naming convention for latitude. However, the field **long** is not recognized. To set the geographic role, right-click on **long** under **Measures** and, under **Geographic Role**, select **Longitude**.
9. Click on **Data** from the top menu and expand **mipolygon#csv (mipolygon.csv)** and click on **Edit Tables**, as shown in the following screenshot:



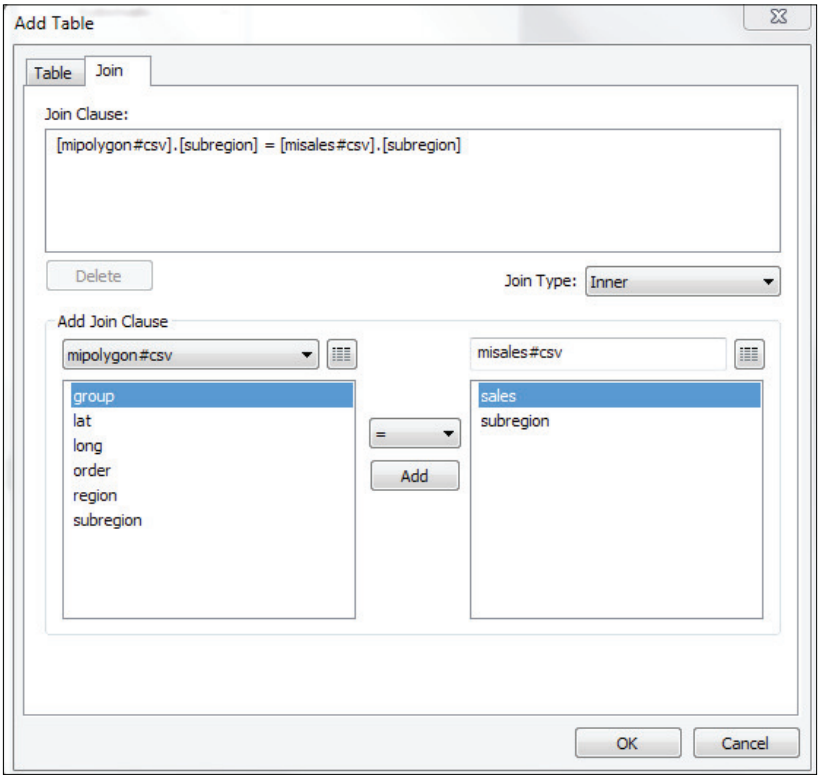
10. In the **Tables** dialog box, click on the **Add Table** button.



11. Select **misales#csv** under the **Table** tab.

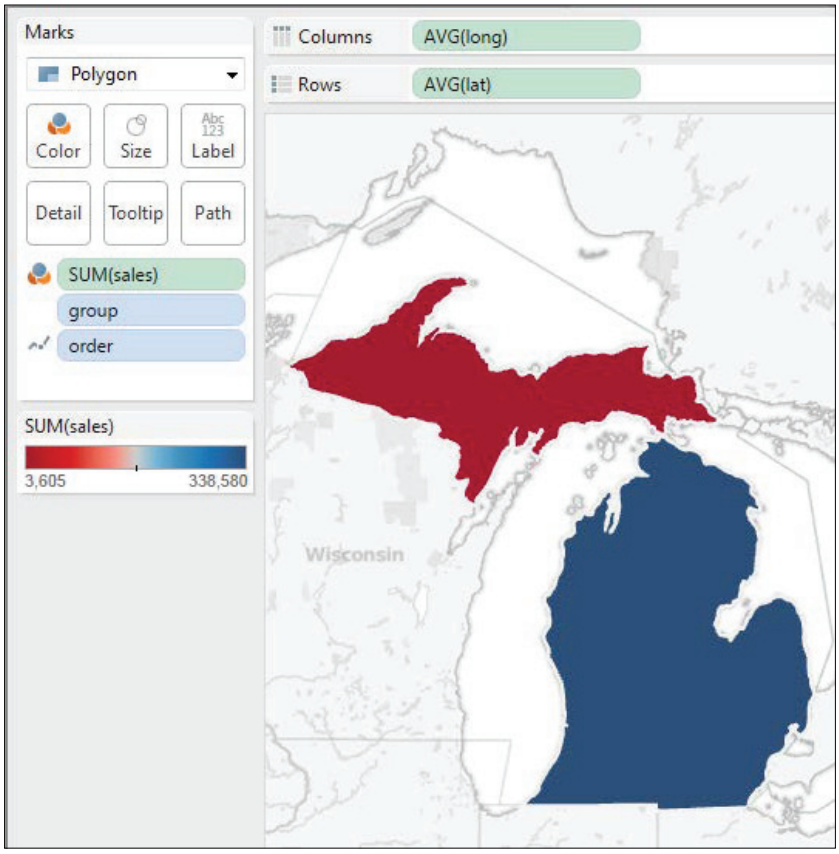


12. Click on the **Join** tab and make sure that the **Join Clause** pane shows that the **subregion** value from the `mipolygon.csv` file is joined to the **subregion** option of the `misales.csv` file, as shown in the following screenshot. We do so to make sure that all the records from both files where the subregions match are returned. This is called an **inner join**.



13. Hit **OK** on the **Edit Tables** options dialog box.
14. Hit **OK** on the **Tables** dialog box.
15. Drag-and-drop **lat** under **Measures** into the **Rows** shelf.
16. Drag-and-drop **long** under **Measures** into the **Columns** shelf.
17. Expand the **Marks** dropdown and select **Polygon**.

18. Drag-and-drop **order** from **Dimensions** into the **Path** box in the **Marks** pane.
19. Drag-and-drop **group** from **Dimensions** into the **Detail** box.
20. Drag-and-drop **sales** from **Measures** into the **Color** box.
21. Click on the drop-down arrow on the **SUM(sales)** legend and click on **Edit Colors**.
22. Select the **Red-Blue Diverging** color from the **Palette** dropdown and hit **OK** to generate a custom polygon-filled map similar to the one shown in this following screenshot:



How it works...

Although we can create filled maps with Tableau 7.0 or higher, we sometimes need custom-filled maps that may not be contained in Tableau's internal data, such as regions within a state. To create such maps, we need a polygon file that has coordinates (latitude and longitude) for the area, an **Order** field to indicate the order of the outline of the polygon, and a grouping variable to indicate the boundaries. In this recipe, we stored two areas (north and south) and also the group field, whose coordinates fell under those two areas. You will need software such as ArcGIS to generate polygon files, which are also called **shape files**. Quantum GIS, an open source geographic information system, can also be used to create such a shape file. The polygon file used in this example was generated using the R language and the `map_data` function of the `ggplot2` package. The code to generate this file is as follows:

```
install.packages('maps')
install.packages('ggplot2')
library(maps)
library(ggplot2)
mimap <- map_data(map = "state", region = "michigan")
write.csv(mimap, file = "mipolygon.csv", row.names = F)
```

Customizing maps

Tableau provides quite a few options to change the format of a generated map. Some of the options include washout, to make the map transparent, and removing borders. By customizing maps this way, we improve the readability as well as increase the efficacy of the maps.

Getting ready

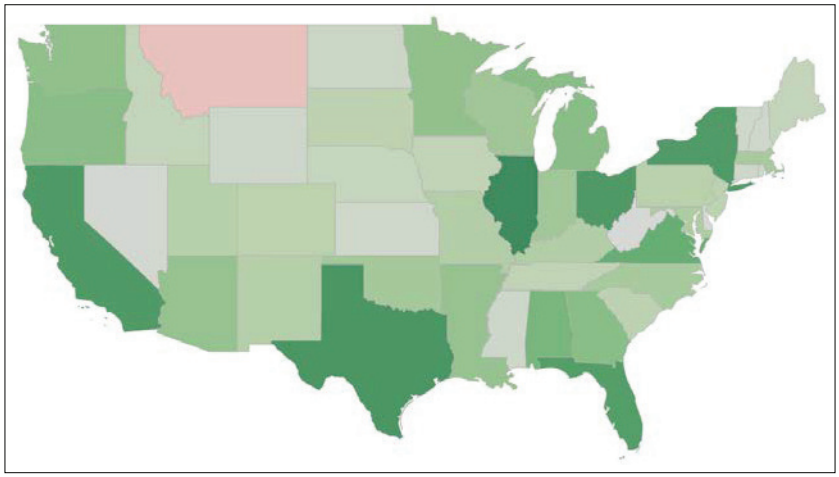
Let's use the sample file `Sample – Superstore Sales (Excel)`. Open a new worksheet and select **Sample – Superstore Sales (Excel)** as the data source.

How to do it...

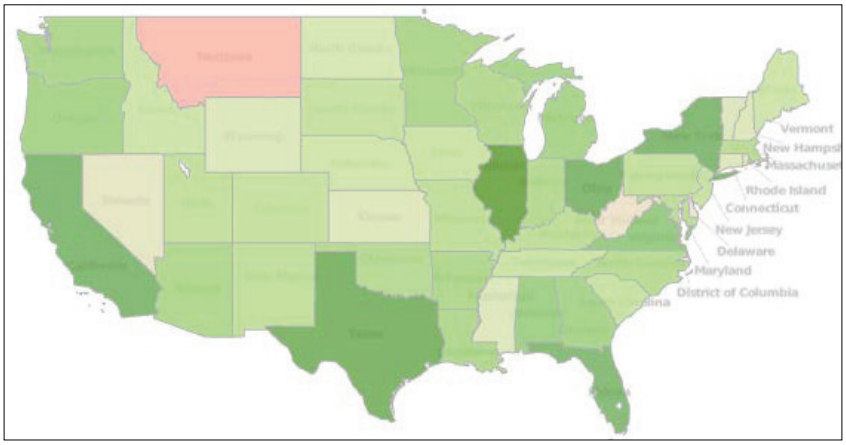
Once the data is loaded, perform the following steps to customize the generated map:

1. Click on the **Show Me** button to bring the **Show Me** toolbar on the screen.
2. While holding the *Ctrl* key, click on **Profit** from **Measures** and **State** from **Dimensions**.
3. Click on **filled maps** on the **Show Me** toolbar to create a choropleth map.
4. From the main menu toolbar, click on **Map** and then on **Map Options**.
5. Note the difference in the background color of the map by changing the **Style** value from **Gray** to **Normal**.
6. Change the **Style** value back to **Gray** again.

7. Uncheck the box in front of **Base** to make the map look more clean and aesthetically pleasing.
8. Uncheck the box in front of **Light State Border & Names** to make the map look even more clean, since the map is already grouping the states.
9. Drag the **Washout** slider to 100 percent if you want to see only the filled map with no other additional information, as shown in this following screenshot:



10. Experiment with the **Washout** slider and check the box **State/Province Names** to create an informative yet good-looking map. For example, by changing the **Washout** slider to 40 percent and checking the box **State/Province Names**, we have minimized the distractions but still show the state names, as shown in the following screenshot:



How it works...

Tableau automatically selects the options that will work in most cases, but we can customize the options even further to make the maps aesthetically pleasing. This is achieved by changing the map layers, which are drawn from an online map provider or Tableau's offline maps. It is very similar to painting a layer of color on top of another on a canvas, with one big difference—our ability to add and remove layers as we please.

6
Calculating User-defined Fields

In this chapter, we will cover the following recipes:

- Using predefined functions
- Calculating percentages
- Applying the If-Then logic
- Applying logical functions
- Showing totals
- Showing the percentage of totals
- Discretizing data
- Manipulating text
- Aggregating data

Introduction

Many a times we need to manipulate data in a certain way to generate the desired visualization or text. Tableau provides ways to calculate and create new fields, which could be used to enhance our visualization.

Using predefined functions

Tableau provides many predefined functions that help us manipulate data in a certain way. These functions are divided by the type of manipulation, such as functions of numeric and string data types, or aggregate operations.

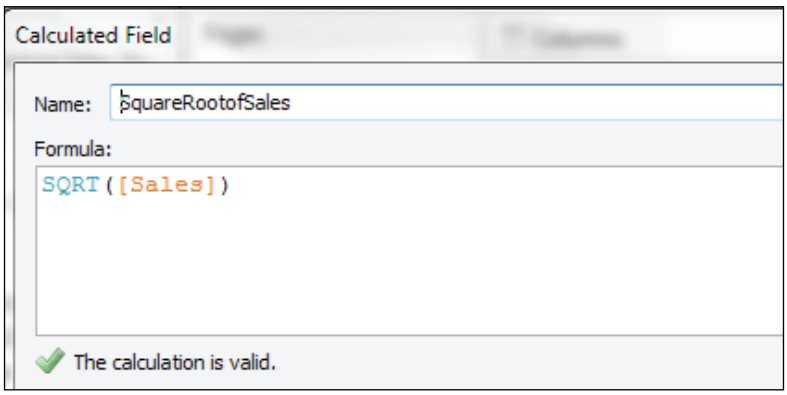
Getting ready

Let's use the sample file `Sample – Superstore Sales (Excel)`. Open a new worksheet and select **Sample – Superstore Sales (Excel)** as the data source.

How to do it...

Once the data is loaded on the worksheet, perform the following steps to create a new calculated field:

1. To calculate the square root of a number, right-click on the **Measures** pane and select **Create Calculated Field**.
2. In the **Name** box, enter `SquareRootofSales` as shown in the following screenshot:



3. In the **Formula** box, enter `SQRT([Sales])` and hit **OK**.
4. To compare the square root of sales with the original sales, create a scatter plot of **Unit Price**, **Sales**, and **SquareRootofSales** by **Region**. From the main menu, click on **Analysis** and uncheck **Aggregate Measures**.
5. Drag-and-drop **SquareRootofSales** and **Sales** in the **Columns** shelf.
6. Drag-and-drop **Region** and **Unit Price** in the **Rows** shelf.

7. Change the **Mark** type to **Circle** to create a chart similar to the one in the following screenshot:

How it works...

Using a predefined function, we created a field that houses the square root values of the **Sales** field. This newly created field can be used similarly for other existing fields, as shown in the created chart. Taking the square root of numeric values is a common data-transformation technique used to better observe the distribution of values, including outliers. You can see from the previous chart that the **SquareRootOfSales** values are spread more than the original sales values, which are more clustered around certain areas.

There's more...

You can learn more about data transformation objectives and techniques from the lecture slides of *Regression III: Advanced Methods Workshop* by *William Jacoby* at `http://polisci.msu.edu/jacoby/icpsr/regress3/lectures/week1/4.Transformations.pdf`.

Calculating percentages

One of the most common type of measures is the percentage of a value within a population. Tableau provides options for converting values in a row or column to fractions of row or column totals; however, we can also create new fields with some calculated values presented as percentages.

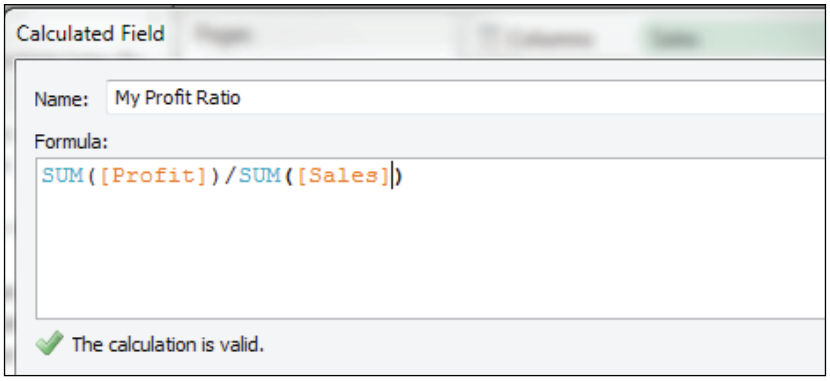
Getting ready

Let's use the sample file `Sample – Superstore Sales (Excel)`. Open a new worksheet and select **Sample – Superstore Sales (Excel)** as the data source.

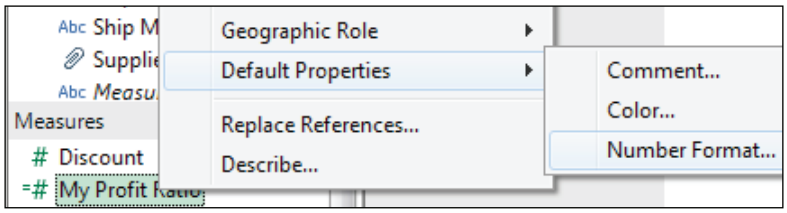
How to do it...

Once the data is loaded on the worksheet, perform the following steps to create a new calculated field:

1. To calculate the profit to sales ratio, right-click on the **Measures** pane and select **Create Calculated Field**.
2. In the **Name** box, enter `My Profit Ratio`.
3. In the **Formula** box, enter `sum([Profit])/sum([Sales])` as shown in the following screenshot:



4. Right-click on **My Profit Ratio** from the **Measures** pane, expand **Default Properties**, and select **Number Format** as shown in the following screenshot:



5. In the **Number Format** options box, select **Percentage** and hit **OK**.

How it works...

In the calculated field, we aggregated the **Profit** and **Sales** fields before dividing them. By calculating this way, we summed the **Profit** and **Sales** fields individually and then performed the division. This is different from dividing profit and sales first and then summing those values—this type of operation is a row-level operation and is suitable for many occasions. However, we have to remember to aggregate fields to avoid unexpected results.

Applying the If-Then logic

At times, it becomes necessary to report values in a certain way; for example, displaying blank values as dashes or categorizing some values into buckets. Since these modifications are based on logic, they are created using logical functions.

Getting ready

Let's use the sample file `Sample - Superstore Sales (Excel)`. Open a new worksheet and select **Sample – Superstore Sales (Excel)** as the data source.

How to do it...

Once the data is loaded on the worksheet, perform the following steps to create a new calculated field:

1. To categorize shipping modes into air and ground, right-click anywhere on the **Dimensions** pane and select **Create Calculated Field**.
2. In the **Name** box, enter `Ship Type`.
3. In the **Formula** box, enter `IF [Ship Mode] = 'Delivery Truck' then 'Ground' Else 'Air' End` and hit **OK**.

How it works...

Tableau provides seven logical functions to test logical conditions and return some values depending on the result of the condition. In the previous recipe, we checked whether the `Ship Mode` attribute was `Delivery Truck`, and if it were, we returned the `Ground` value as `Ship Type`. Since we know there are only three types of the `Ship Mode` attribute (`Delivery Truck`, `Express Air`, and `Regular Air`), we don't need to check for other shipping modes as they both are of the `Air` type.

Applying logical functions

Tableau provides various logical functions, such as `CASE`, `IF`, and `IIF`, to create calculated fields based on some conditions. In this recipe, we will create and see the use of some of these logical functions.

Getting ready

Let's use the sample file `Sample – Coffee Chain (Access)`. Open a new worksheet and select **Sample – Coffee Chain (Access)** as the data source.

How to do it...

Once the data is loaded on the worksheet, perform the following steps to create calculated fields based on conditions:

1. Right-click on **Product Type** from **Dimensions** and click on **Create Calculated Field**.
2. In the **Name** box, enter `Coffee or Tea`.
3. In the **Formula** box, enter the `CASE [Product Type] WHEN 'Coffee' THEN 'Coffee' WHEN 'Espresso' THEN 'Coffee' WHEN 'Herbal Tea' THEN 'Tea' ELSE 'Tea' END` formula and hit **OK**.
4. To use `0` instead of missing values of **Sales**, right-click on **Sales** from **Measures** and click on **Create Calculated Field**.
5. In the **Name** box, enter `Non-missing Sales`.
6. In the **Formula** box, enter `ZN([Sales])` and hit **OK**.
7. To see **Total Expenses** in some categories, right-click on **Total Expenses** from **Measures** and click on **Create Calculated Field**.

8. In the **Name** box, enter `Expensive Type`.
9. In the **Formula** box, enter the `IF [Total Expenses] <= 49.99 THEN 'Cheap' ELSEIF [Total Expenses] >= 50 and [Total Expenses] < 100 THEN 'Somewhat Expensive' ELSEIF [Total Expenses] >= 100 and [Total Expenses] < 150 THEN 'Slightly Expensive' ELSE 'Very Expensive' END` formula.

How it works...

The `CASE` and `IF` functions are similar in that they both allow testing of an expression and returning values on various conditions. The `CASE` function is usually easier to read and is usually the preferred way of testing expressions. The `IF` function allows us to test on numeric conditions whereas the `CASE` function doesn't allow that; for example, we cannot write a `CASE [Profit] < 100` condition, but we can write `IF [Profit] < 100`. The `IFNULL` function is very useful when we want to return any value (numbers in case of numeric expressions and a string in case of string expressions) if the expression is null, and the `ZN` function is useful when we want to return `0` if the expression is null.

Showing totals

Although it is useful to show the breakdown of measures by various dimensions, readers value seeing grand totals for rows and columns. If at least one **Columns** or **Rows** value is present, it is very easy to show grand totals.

Getting ready

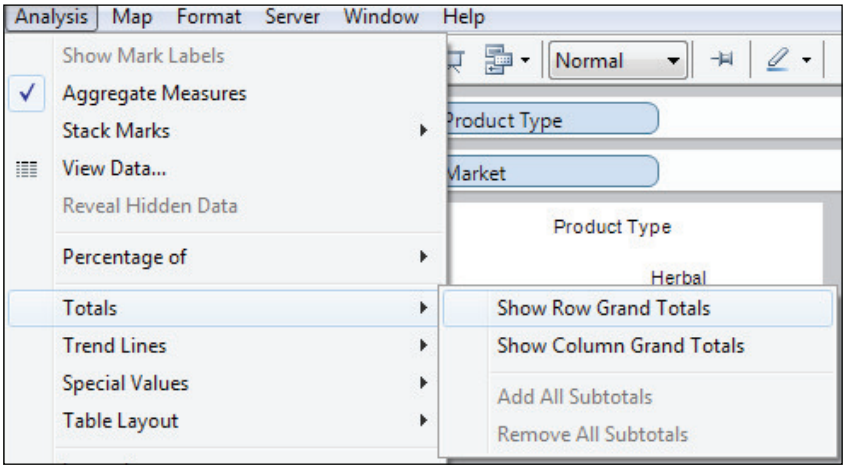
Let's use the sample file `Sample – Coffee Chain (Access)`. Open a new worksheet and select **Sample – Coffee Chain (Access)** as the data source.

How to do it...

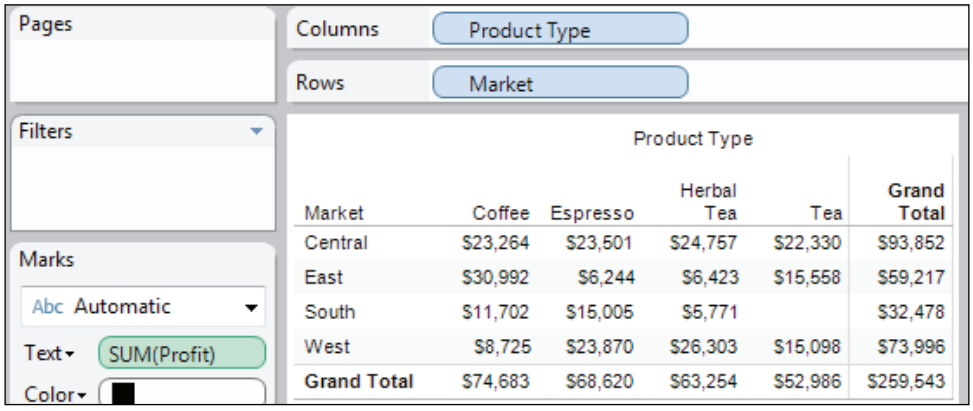
Once the data is loaded on the worksheet, perform the following steps to show grand totals:

1. Drag-and-drop **Product Type** from **Dimensions** in the **Columns** shelf.
2. Drag-and-drop **Market** from **Dimensions** in the **Rows** shelf.
3. Drag-and-drop **Profit** from **Measures** in the **Text** box under the **Marks** pane.

4. From the top menu bar, click on **Analysis**, expand **Totals**, and click on **Show Row Grand Totals** as shown in the following screenshot:



5. Again, click on **Analysis**, expand **Totals**, and click on **Show Column Grand Totals** to see both the column and row totals as shown in the following screenshot:



Product Type

Market	Coffee	Espresso	Herbal Tea	Tea	Grand Total
Central	$23,264	$23,501	$24,757	$22,330	$93,852
East	$30,992	$6,244	$6,423	$15,558	$59,217
South	$11,702	$15,005	$5,771		$32,478
West	$8,725	$23,870	$26,303	$15,098	$73,996
Grand Total	$74,683	$68,620	$63,254	$52,986	$259,543

Showing the percentage of totals

Seeing the percentage of each group as compared to the total of all groups is as useful as seeing totals and breakdowns. This gives the reader an idea about the magnitude of every value compared to that of the totals. Tableau offers various options to see values as percentages of the totals. These options include seeing percentages of the row totals, the column totals, or the grand total.

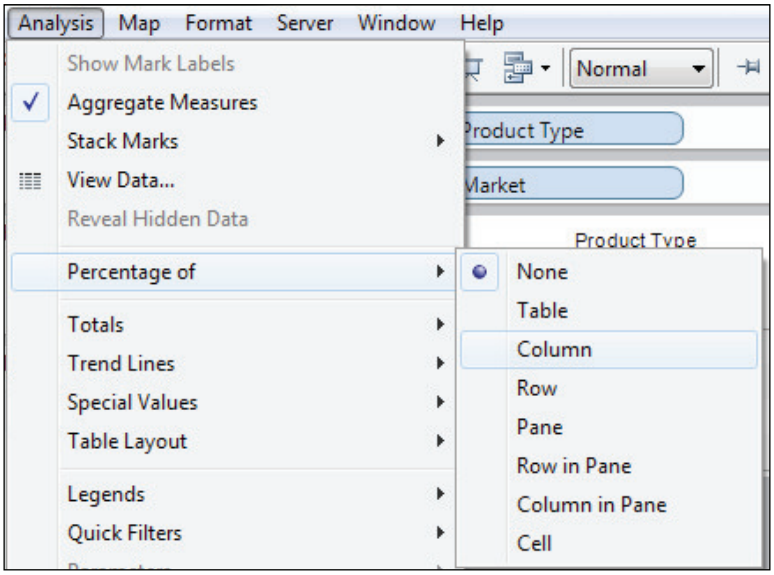
Getting ready

Let's use the sample file `Sample - Coffee Chain (Access)`. Open a new worksheet and select **Sample – Coffee Chain (Access)** as the data source.

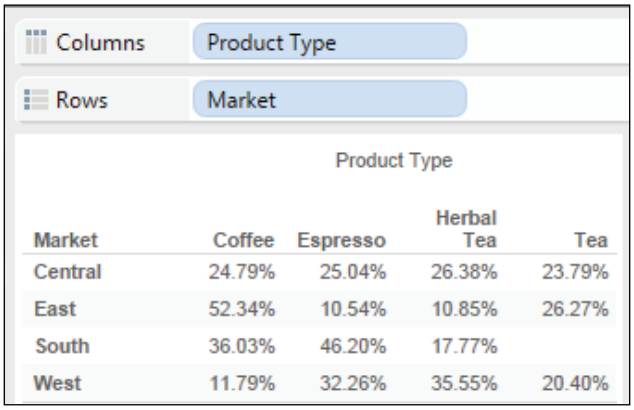
How to do it...

Once the data is loaded on the worksheet, perform the following steps to see percentage values:

1. Drag-and-drop **Product Type** from **Dimensions** in the **Columns** shelf.
2. Drag-and-drop **Market** from **Dimensions** in the **Rows** shelf.
3. Drag-and-drop **Profit** from **Measures** in the **Text** box under the **Marks** pane.
4. To see percentage profit by every **Product Type** in all the **Market** types, expand the **Analysis** menu option from the main menu toolbar. Then, expand the **Percentage of** option and select **Column** as shown in the following screenshot:

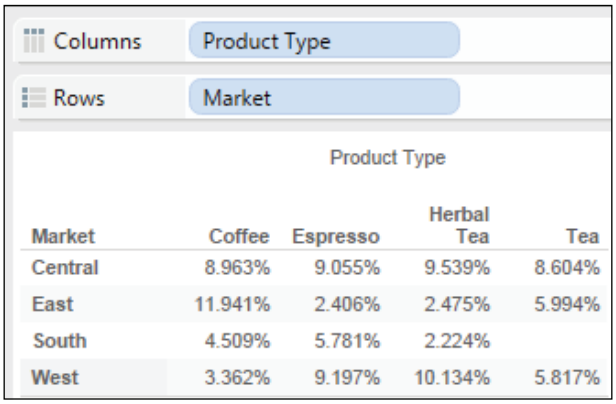


5. To see the percentage profit by every **Market** type in all the **Product Type** values, expand the **Analysis** menu option from the main menu toolbar, followed by expanding the **Percentage of** option, and select **Row** to generate a table as shown in the following screenshot:



	Product Type			
Market	Coffee	Espresso	Herbal Tea	Tea
Central	24.79%	25.04%	26.38%	23.79%
East	52.34%	10.54%	10.85%	26.27%
South	36.03%	46.20%	17.77%	
West	11.79%	32.26%	35.55%	20.40%

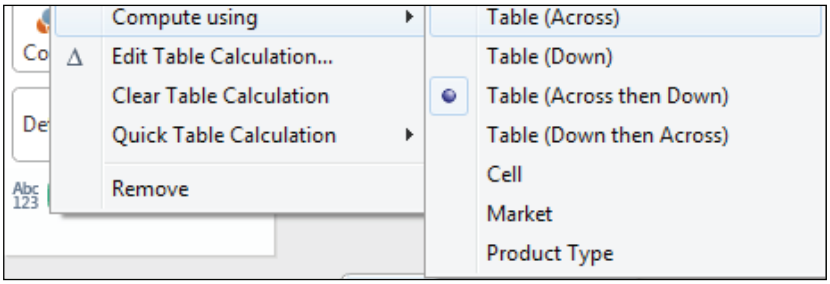
6. To see profit by every **Market** and **Product Type** values as a fraction of the total **Profit** value, expand the **Analysis** menu option from the main menu toolbar, followed by the **Percentage of** option, and select **Table** to generate a percentage table as shown in the following screenshot:



	Product Type			
Market	Coffee	Espresso	Herbal Tea	Tea
Central	8.963%	9.055%	9.539%	8.604%
East	11.941%	2.406%	2.475%	5.994%
South	4.509%	5.781%	2.224%	
West	3.362%	9.197%	10.134%	5.817%

There's more...

Tableau provides options to change how percentages are calculated, such as across, down, by cell, or by other fields. These options can be changed by clicking on the aggregated measure and expanding options under **Compute using**, as shown in the following screenshot:



Discretizing data

Sometimes we require discretizing (or binning) of numeric data for pretty labeling or meeting some format guidelines; for example, you may need to report the sales amount in thousands, and thus you will need to create a field that will put every sales amount in various bins, for example, 0-1000, 1000-2000, and so on.

Getting ready

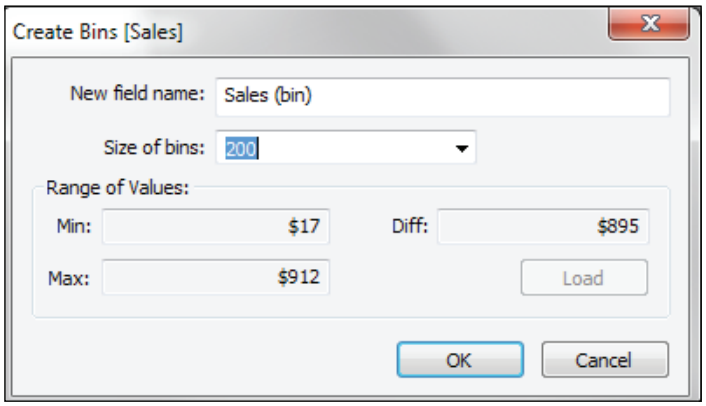
Let's use the sample file `Sample - Coffee Chain (Access)`. Open a new worksheet and select **Sample – Coffee Chain (Access)** as the data source.

How to do it...

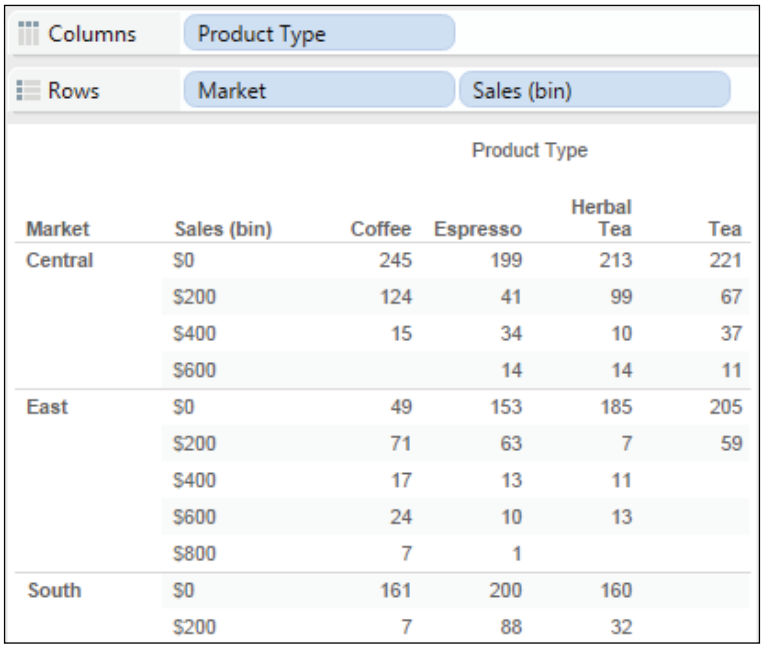
Once the data is loaded on the worksheet, perform the following steps to discretize a numeric value or create bins:

1. Right-click on **Sales** from **Measures** and select the **Create Bins** option.
2. Hit the **Load** button to see the distribution of the **Sales** amount.

3. In the **Size of bins** box, enter `200`, as shown in the following screenshot, and hit **OK**:



4. Drag-and-drop **Sales (bin)** from **Dimensions** in the **Rows** shelf.
5. Drag-and-drop **Product Type** from **Dimensions** in the **Columns** shelf.
6. Drag-and-drop **Market** from **Dimensions** in the **Rows** shelf, but place it before **Sales (bin)**.
7. Drag-and-drop **Number of Records** from **Measures** in the **Text** box under the **Marks** pane to create a table similar to the one shown in the following screenshot:



		Product Type			
Market	Sales (bin)	Coffee	Espresso	Herbal Tea	Tea
Central	$0	245	199	213	221
	$200	124	41	99	67
	$400	15	34	10	37
	$600		14	14	11
East	$0	49	153	185	205
	$200	71	63	7	59
	$400	17	13	11	
	$600	24	10	13	
	$800	7	1		
South	$0	161	200	160	
	$200	7	88	32	

How it works...

When we hit the **Load** button on the **Create Bins [Sales]** dialog box, Tableau loads the distribution; that is, minimum, maximum, and the difference between the minimum and maximum value of the underlying measure. By looking at those values, we can decide the appropriate number of bins. Once we enter the number of bins, Tableau puts all the individual values of the underlying measure into bins, which start with zero and end with the highest possible value of the range that doesn't exceed the maximum value. For this recipe, we had the maximum value of 895 dollars and the maximum value of the bins was 800 dollars, because the next bin value would be 1,000 dollars and there are no values that are over 1,000 dollars.

Manipulating text

At times, we are required to parse or manipulate text variables to get something meaningful out of those variables; for example, a **Full Name** field may contain both the first name and last name of a sales representative, but our reporting standards may require us to show two different columns for the first and last names. With Tableau's string operators, we can easily manipulate the text to meet our requirements.

Getting ready

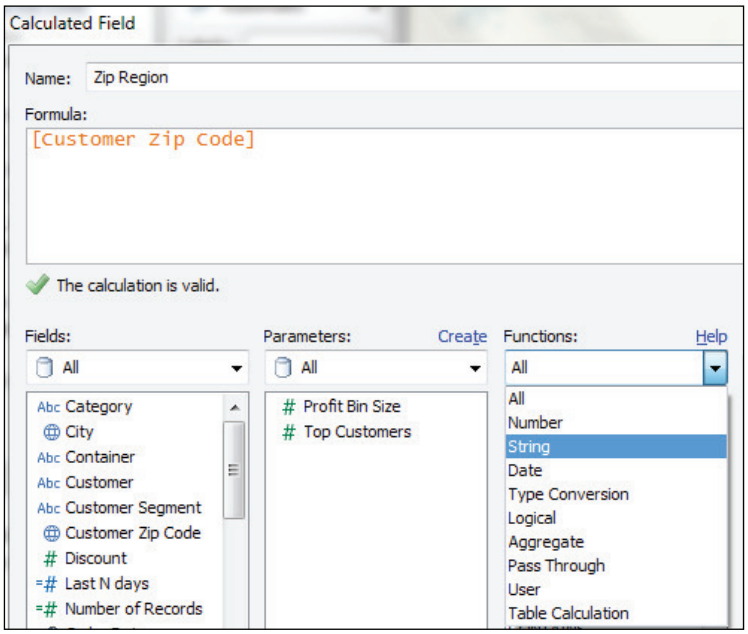
Let's use the sample file `Sample – Superstore Sales (Excel)`. Open a new worksheet and select **Sample – Superstore Sales (Excel)** as the data source.

How to do it...

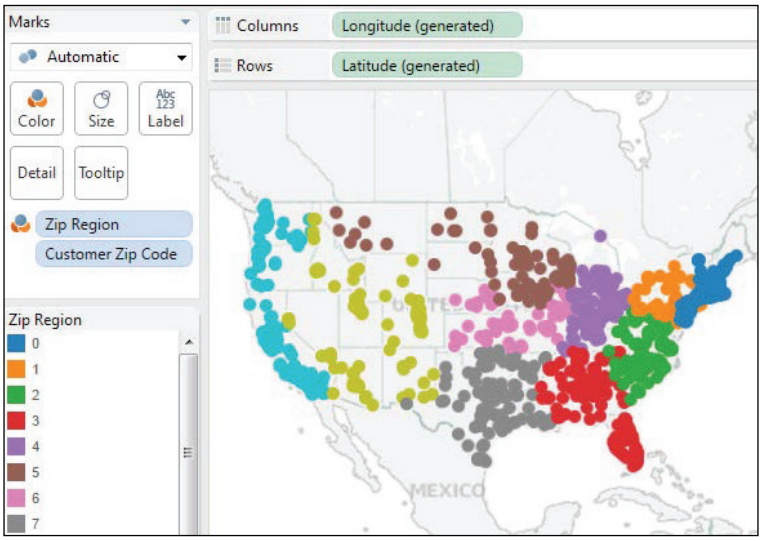
Once the data is loaded on the worksheet, perform the following steps to create new string fields based on existing text variables:

1. Right-click on **Customer Zip Code** from **Dimensions**, and select **Create Calculated Field**.
2. In the **Name** box, type `Zip Region`.

3. From the **Functions** dropdown, select **String**, as shown in the following screenshot:



4. Find the `LEFT` function and double-click on it.
5. Adjust the formula in the **Formula** box to `LEFT([Customer Zip Code], 1)` and hit **OK**. We can use this newly generated field to create a map as shown in the following screenshot:



6. To extract the customer's last name, right-click on **Customer** from **Dimensions** and select **Create Calculated Field**.
7. In the **Name** box, enter `Customer Last Name`.
8. In the **Formula** box, enter `RIGHT([Customer], LEN([Customer]) - FIND([Customer], " "))` and hit **OK**.

How it works...

The `LEFT` function extracts the specified number of characters from the start of the given string variable. In our recipe, we extracted the first character of the **Customer Zip Code** value. The `RIGHT` function works similarly except that it extracts characters from the end of the given string variable. The `FIND` function returns the position of the searched string within a string variable. To extract the customer's last name, we first found the position of the space between the customer's first and last names in the **Customer** field. Then we computed the number of characters between the space and the end of the string by subtracting the position of the space from the total number (found using the `LEN` function) of characters in the **Customer** field.

Aggregating data

Although the type of aggregation of a measure can be changed from the **Marks** pane, it is sometimes necessary to show different aggregations of the same measure, and we can do this by creating multiple aggregate fields. We can also add the same **Measure** field multiple times to the **Rows** or **Columns** shelf and then change the aggregation type.

Getting ready

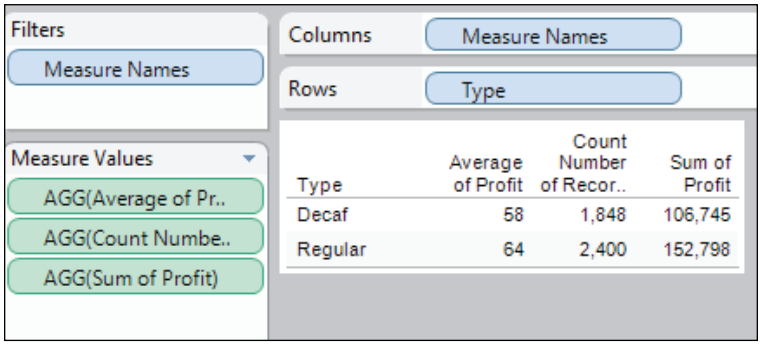
Let's use the sample file `Sample - Coffee Chain (Access)`. Open a new worksheet and select **Sample – Coffee Chain (Access)** as the data source.

How to do it...

Once the data is loaded on the worksheet, perform the following steps to create calculated fields with different aggregations:

1. Right-click on **Profit** from **Measures**, and click on **Create Calculated Field**.
2. In the **Name** box, enter `Sum of Profit`.
3. In the **Formula** box, enter the `SUM([Profit])` formula and hit **OK**.
4. Right-click on **Profit** from **Measures** and click on **Create Calculated Field**.
5. In the **Name** box, enter `Average of Profit`.
6. In the **Formula** box, enter the `AVG([Profit])` formula and hit **OK**.

7. Right-click on **Number of Records** from **Measures**, and click on **Create Calculated Field**.
8. In the **Name** box, enter `Count Number of Records`.
9. In the **Formula** box, enter the `COUNT([Number of Records])` formula and hit **OK**.
10. Click on the **Show Me** button to display the **Show Me** toolbar on the screen.
11. Select **Sum of Profit**, **Average of Profit**, and **Count Number of Records** from **Measures** and **Type** from **Dimensions**.
12. Click on **text tables** on the **Show Me** toolbar to create a table as shown in the following screenshot:



7
Customizing and Saving

In this chapter, we will cover the following recipes to customize and save files:

- Adding title and caption
- Modifying font sizes and colors
- Applying various marks
- Adding colors
- Adding labels
- Changing marks sizes
- Adding reference lines
- Printing to PDF
- Saving packaged workbooks
- Creating a workbook data extract

Introduction

Once you have generated insightful graphics, you would want to customize it first and share it with others. Tableau offers various customization options including modifying font sizes and colors, applying various marks, adding labels, and others.

Adding title and caption

Adding as much information as is possible to the title and caption that describe or summarize some of the important points of the visualization helps readers understand the visualization better. If you have applied filters, Tableau will automatically describe the filters in the caption area, but as a designer, you will have to describe any other important aspects of the visualization.

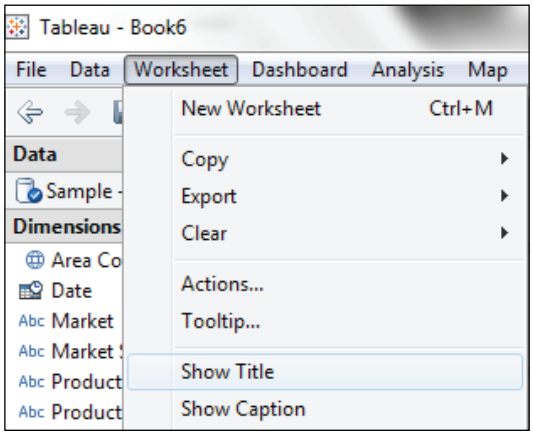
Getting ready

Let's use the sample file `Sample – Coffee Chain (Access)`. Open a new worksheet and select **Sample – Coffee Chain (Access)** as the data source.

How to do it...

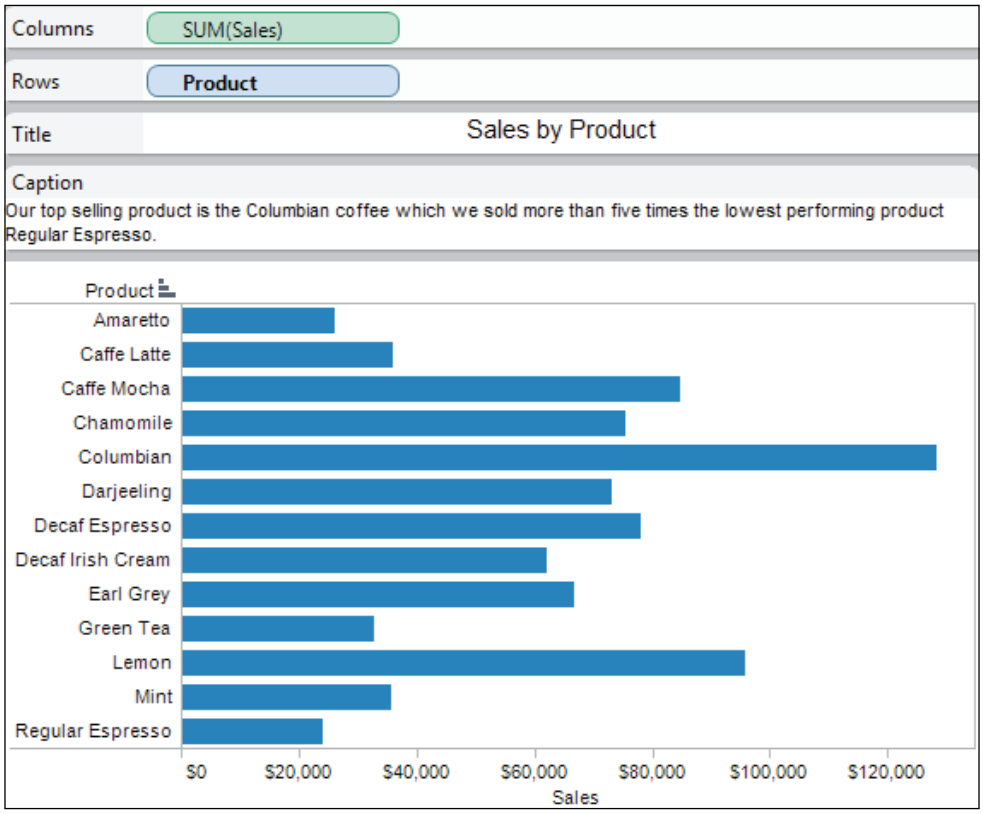
Once the data is loaded on the worksheet, perform the following steps to add a tittle and a caption:

1. Drag-and-drop **Sales** from **Measures** into the **Columns** shelf.
2. Drag-and-drop **Product** from **Dimensions** into the **Rows** shelf.
3. From the main **Menu** toolbar, select **Show Title**, as shown in the following screenshot:



4. From the main menu toolbar, select **Show Caption**.
5. Double-click anywhere in the area next to **Title** and in the **Edit Title** box enter `Sales by Product`, and hit **OK**.

6. Double-click anywhere in the area next to **Caption** and in the **Edit Caption** box enter `Our top selling product is the Columbian coffee which we sold more than five times the lowest performing product Regular Espresso`, and hit **OK**.
7. Move the **Caption** box below the **Title** box and the final visualization should look like the following screenshot:



Modifying font sizes and colors

Tableau provides options to modify font sizes and colors for the whole worksheet or individual components of the worksheet, such as the pane, headers, tooltip, and grand total. Although the default scheme is good enough to be used in production-quality material, there might be instances where you would want to customize these options.

Getting ready

To customize the font and color for this recipe, repeat the *Adding title and caption* recipe.

How to do it...

Once you have recreated the graphic, perform the following steps to customize the font and color:

1. From the main menu toolbar, click on **Format** and then select **Font**.
2. Make sure that the **Format Font** button, which has the letter A in its icon, is highlighted.
3. To modify all the fonts on the worksheet, select a different font size and font from the **Worksheet** dropdown.
4. To modify the font color of the header, select a different font color from the **Header** dropdown.

Applying various marks

Tableau provides various ways to encode data using different marks including **Square**, **Circle**, and **Shape**. The **Square** and **Circle** marks will show data points using a square or circle shape. By using the **Shape** mark, however, we can assign some attribute values to various shapes and help the reader distinguish data points by those shapes.

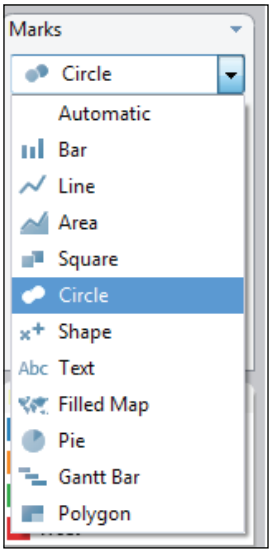
Getting ready

Let's use the sample file `Sample – Coffee Chain (Access)`. Open a new worksheet and select **Sample – Coffee Chain (Access)** as the data source.

How to do it...

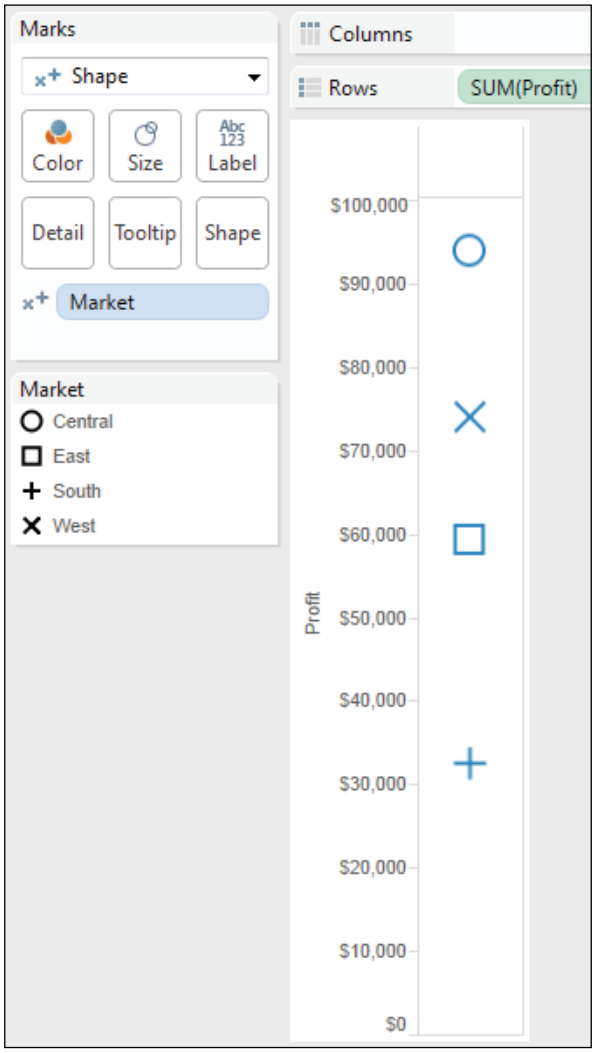
Once the data is loaded on the worksheet, perform the following steps to use various marks to denote the data points:

1. While holding the *Ctrl* key, click on **Market** from **Dimensions** and **Profit** from **Measures**.
2. Click on the **Show Me** button to bring the **Show Me** toolbar on the screen.
3. Click on **circle views** on the **Show Me** toolbar.
4. To change the shape, from the **Marks** pane, select **Square** or **Circle** from the dropdown as shown in the following screenshot:



5. To use a different shape for every **Market** value, select **Shape** from the dropdown in the **Marks** pane.

6. Drag-and-drop **Market** from **Color** into the **Shape** box to generate a chart shown in the following screenshot:



How it works...

Although the **Square**, **Circle**, and **Shape** options may appear similar, they provide different ways to view and distinguish various data points. That is especially true in the case of the **Shape** option: as every attribute value is given a shape, it becomes easy to identify and note the different data points. In the case of many data points or many categories, however, identification and distinction of data points even with different shapes is challenging. To assist the readers, you should limit shapes to three to four attribute values.

Adding colors

Depending on the type of chart you created, Tableau may or may not color code any data. Using the **Color** box in the **Marks** pane, however, you could easily add colors to your graphs.

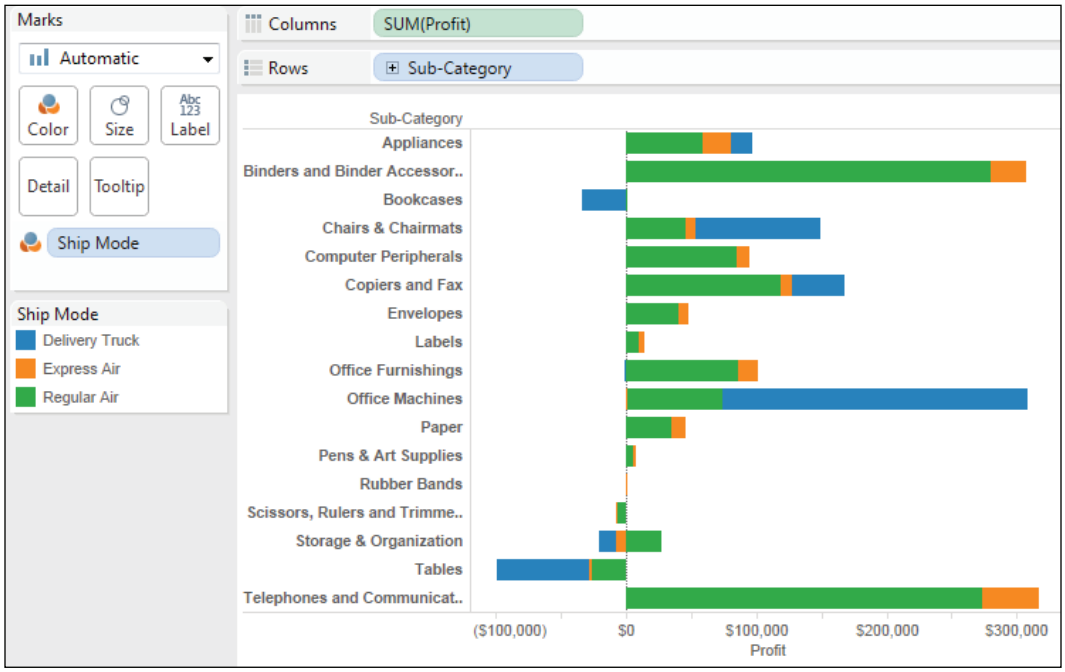
Getting ready

Let's use the sample file `Sample – Superstore Sales (Excel)`. Open a new worksheet and select **Sample – Superstore Sales (Excel)** as the data source.

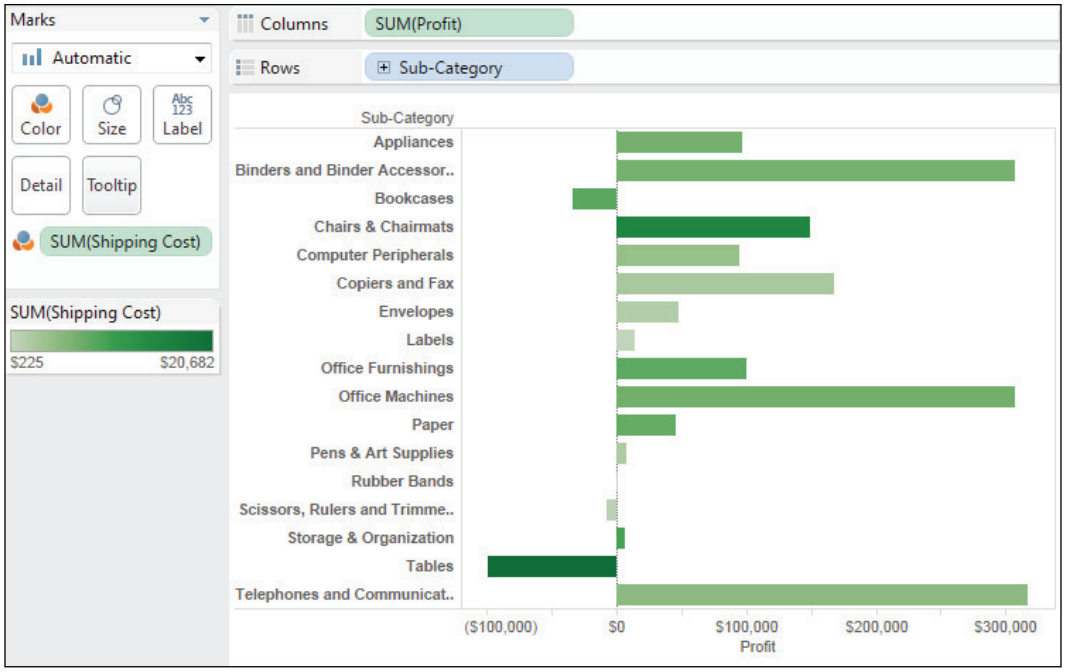
How to do it...

Once the data is loaded on the worksheet, perform the following steps to add colors to encode the data:

1. Drag-and-drop **Profit** from **Measures** into the **Columns** shelf.
2. Drag-and-drop **Sub-Category** from **Dimensions** into the **Rows** shelf.
3. To see the **Profit** values by **Ship Mode**, drag-and-drop **Ship Mode** in the **Color** box in the **Marks** pane to generate a chart shown in the following screenshot:



4. To see the **Profit** values by **Customer Segment**, drag-and-drop **Customer Segment** in the **Color** box in the **Marks** pane.
5. To compare **Profit** and **Shipping Cost** values, drag-and-drop **Shipping Cost** in the **Color** box in the **Marks** pane to generate a chart shown in the following screenshot. You can see that tables generated a loss and had the highest shipping costs, whereas chairs and chair mats were profitable, though this category incurred the second-highest shipping costs:



How it works...

When you drag a dimension to the **Color** box, Tableau automatically selects a color palette to draw colors from, and these colors are very distinctive and use different hues. When you drag a measure to the **Color** box, however, Tableau selects a gradient of a single color, and bigger data points (larger values) are encoded by a darker gradient, and smaller data points (smaller values) are encoded by a lighter gradient. It is important to note, however, that the underlying value of the dimension or measure also dictates what type of color palette would be chosen: a dimension of a continuous type of data can generate a gradient palette, and a discrete measure can generate a discrete color palette.

Adding labels

Although adding labels to data points is sometimes redundant, Tableau makes it very easy to add labels to your graphs. With the help of data labels, readers of the graph are able to read the exact value of the data point instead of speculating about the values by gauging the heights of bars or sizes of shapes. However, as a designer of a visualization, you must ask this question to yourself: if data labels are important to your graph, can you replace the graph with a simple table to provide all the details?

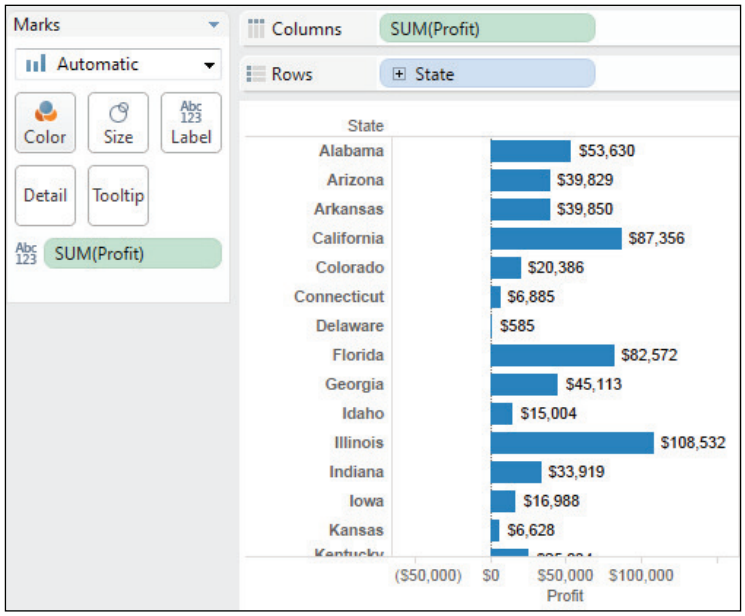
Getting ready

Let's use the sample file `Sample - Superstore Sales (Excel)`. Open a new worksheet and select **Sample – Superstore Sales (Excel)** as the data source.

How to do it...

Once the data is loaded on the worksheet, perform the following steps to add the data labels to your graph:

1. Drag-and-drop **State** from **Dimensions** into the **Rows** shelf.
2. Drag-and-drop **Profit** from **Measures** into the Columns shelf
3. Drag-and-drop **Profit** from **Measures** again into the **Label** box in the **Marks** pane to display data labels next to the bars, as shown in the following screenshot:



How it works...

In the simple cases, as we have seen in this recipe, Tableau places the value next to the marks. In overlapping data points cases, Tableau will hide some of the labels to increase the clarity of the graphic. Instead of repeating the value encoded in a mark, it is possible to show a completely different measure as a label. This could be misleading, however, and could confuse readers.

Changing marks sizes

Since we can tell differences in sizes easily compared to differences in colors, encoding data in various sizes of marks will increase the effectiveness of a graph. If the differences in data points are hard to observe, then a different type of visualization might be needed.

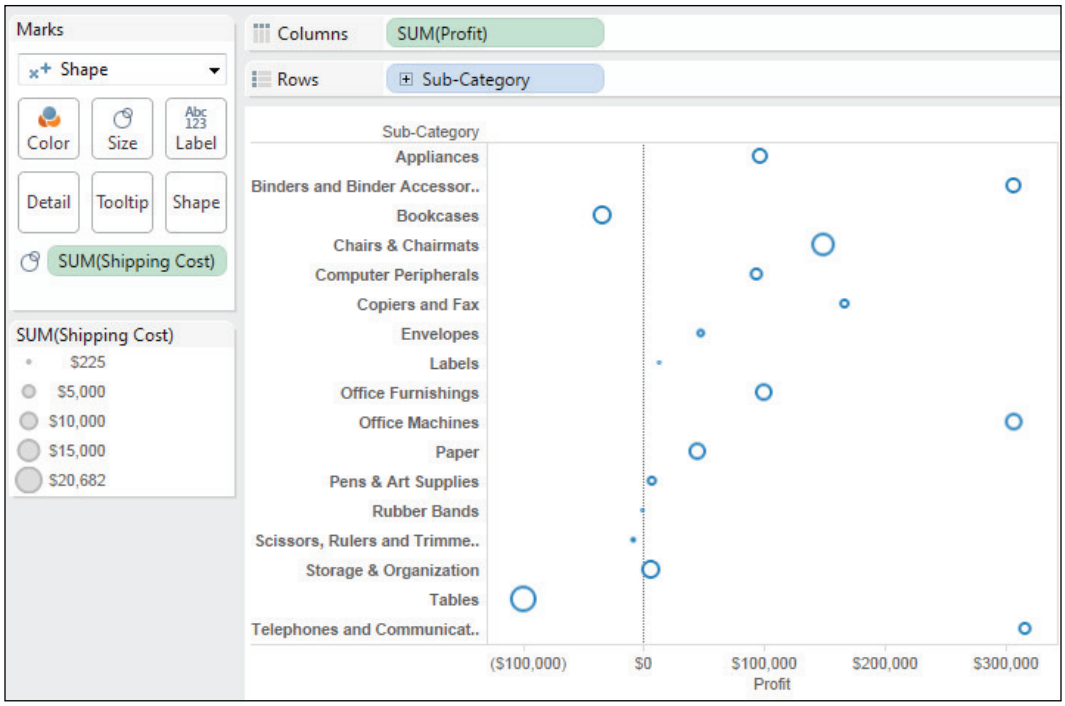
Getting ready

Let's use the sample file `Sample – Superstore Sales (Excel)`. Open a new worksheet and select **Sample – Superstore Sales (Excel)** as the data source.

How to do it...

Once the data is loaded, perform the following steps to show marks sizes encoded by the values of an attribute:

1. Drag-and-drop **Sub-Category** from **Dimensions** into the **Rows** shelf.
2. Drag-and-drop **Profit** from **Measures** into the **Columns** shelf.
3. Drag-and-drop **Shipping Cost** from **Measures** into the **Size** box in the **Marks** pane.
4. Change the mark type to **Shape** to see the mark size vary by **Shipping Cost**.
5. Click on the **Size** box to increase the size of shapes by dragging the visible slider. The final chart should look like the one in the following screenshot:



Adding reference lines

Among many of Tableau's features, adding reference lines to graphs is one of them. By adding reference lines, we can compare data points with either any constant value or any statistical computation such as average of the measure values.

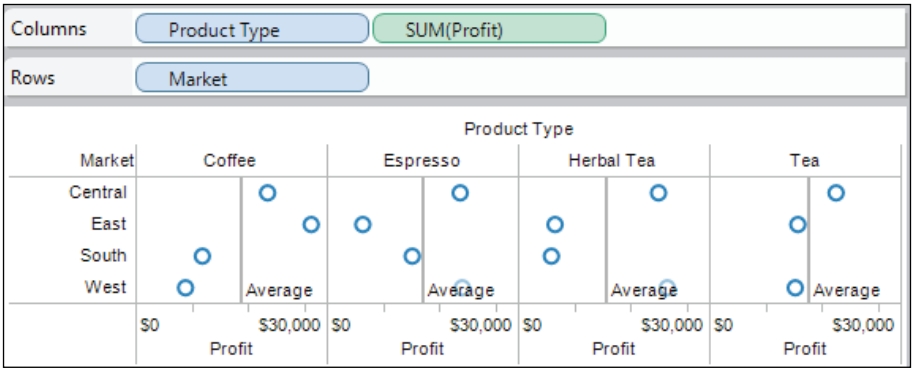
Getting ready

Let's use the sample file `Sample - Coffee Chain (Access)`. Open a new worksheet and select **Sample – Coffee Chain (Access)** as the data source.

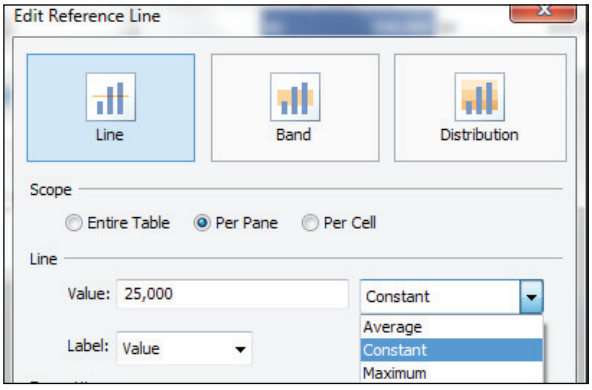
How to do it...

Once the data is loaded on the worksheet, perform the following steps to add various reference lines:

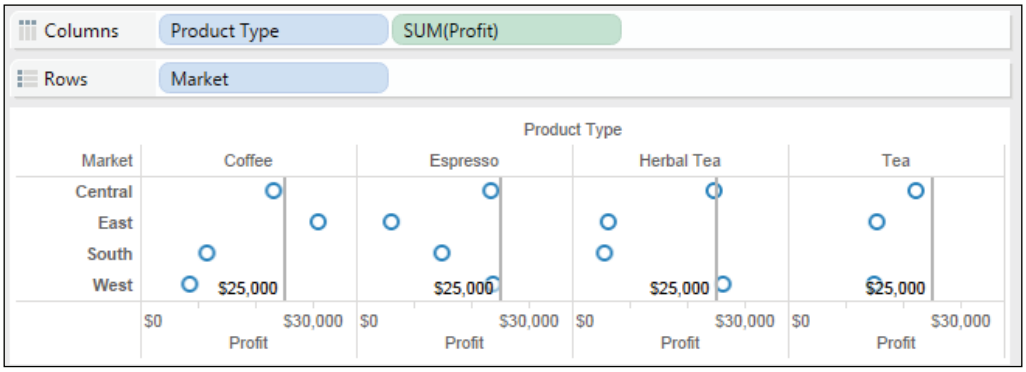
1. Drag-and-drop **Product Type** from **Dimensions** into the **Columns** shelf.
2. Drag-and-drop **Profit** from **Measures** into the **Columns** shelf.
3. Drag-and-drop **Market** from **Measures** into the **Rows** shelf.
4. Change the mark type to **Shape**.
5. To compare **Profit** values for each **Product Type** and **Market** value with the average **Profit** value for all markets and product types, right-click on the **Profit** axis, select **Add Reference Line**, keep the **Scope** option value to **Per Pane**, accept all the default values, and hit **OK**. The graph should look like the one in the following screenshot:



6. To compare the **Profit** values to a fixed value, right-click on the **Profit** axis, select **Edit Reference Line**, keep the **Scope** value at **Per Pane**, under the **Line** selections, change the drop-down value to **Constant** from **Average** (as shown in the following screenshot), in the **Value** box enter `25,000`, and hit **OK**:



7. The chart with a constant reference line should look similar to the one shown in the following screenshot:



Printing to PDF

One of the easiest ways to share your Tableau graphs is to save them as PDF files. Printing graphs to PDF files is built into Tableau, and you do not require any additional software to print the PDF files.

Getting ready

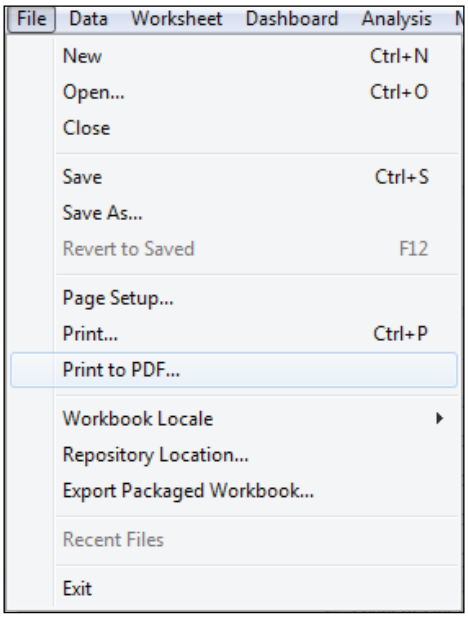
To create a nice looking PDF, repeat the *Adding title and caption* recipe.

How to do it...

Once you have a graph with a title and caption, perform the following steps to print it to PDF:

1. From the main menu toolbar, select **File**.

2. Under **File**, select the **Print to PDF** option as shown in the following screenshot:



3. Select the **Active Sheet** button in the **Range** options box.
4. Select the **Landscape** button in the **Paper Size** box and hit **OK**.
5. Select a folder to save the file in and enter a filename in the **Save As** dialog box, and hit **Save**.

Saving packaged workbooks

When you use local data sources, such as using sample files, Excel, the Access files, or text files, to create a Tableau workbook, sharing could become a challenge when your users do not have access to those data sources. We could overcome such a problem by saving packaged workbooks, which have the workbook as well as the local data.

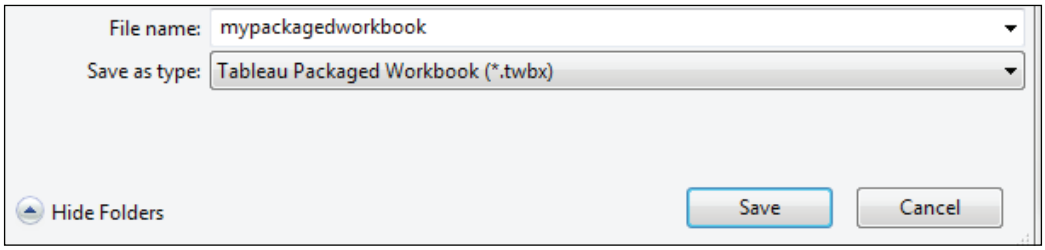
Getting ready

Follow the *Adding labels* recipe to create a workbook using an Excel data source.

How to do it...

Once you have created the workbook with the graph and labels, perform the following steps to save the workbook as a packaged workbook:

1. From the main menu bar, click on **File** and then on **Save As**.
2. Enter a name for the workbook in the **File name** box
3. Select **Tableau Packaged Workbook (*.twbx)** from the **Save as type** dropdown as shown in the following screenshot:



4. Hit **Save**.

Creating a workbook data extract

If your data is coming from some connected data sources, extracting such data from a workbook will let you work on that data even if you are disconnected. The extracts are also useful when dealing with large data files as you can apply filters to select only a few rows (based on conditions).

Getting ready

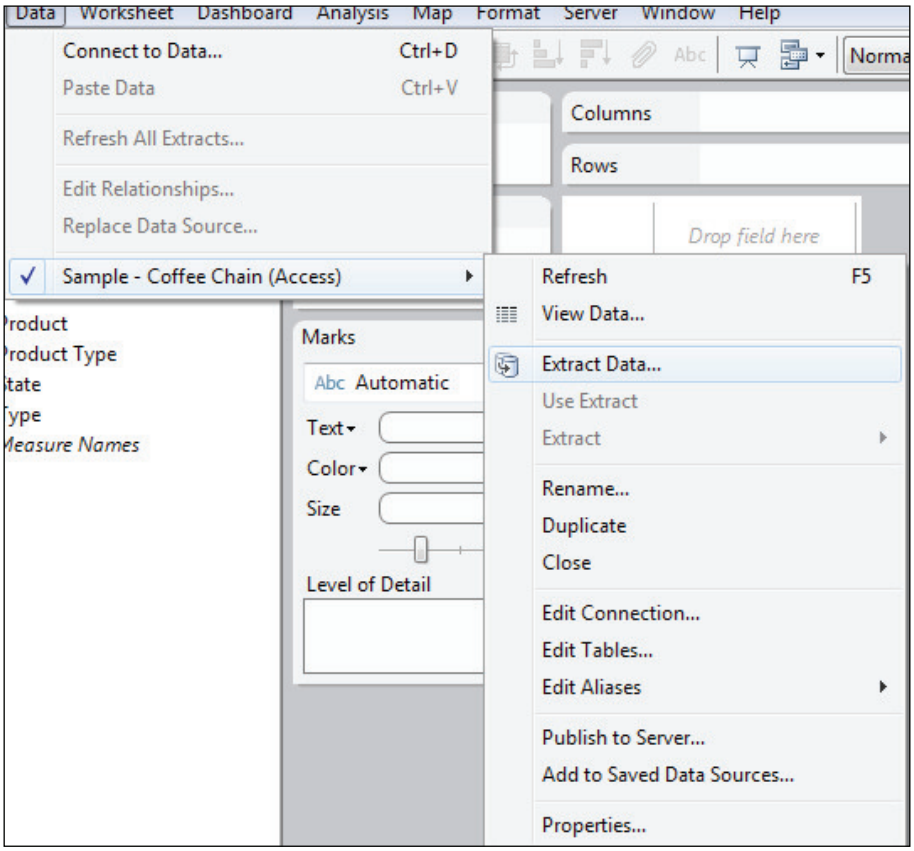
Let's use the sample file `Sample - Coffee Chain (Access)`. Open a new worksheet and select **Sample – Coffee Chain (Access)** as the data source.

How to do it...

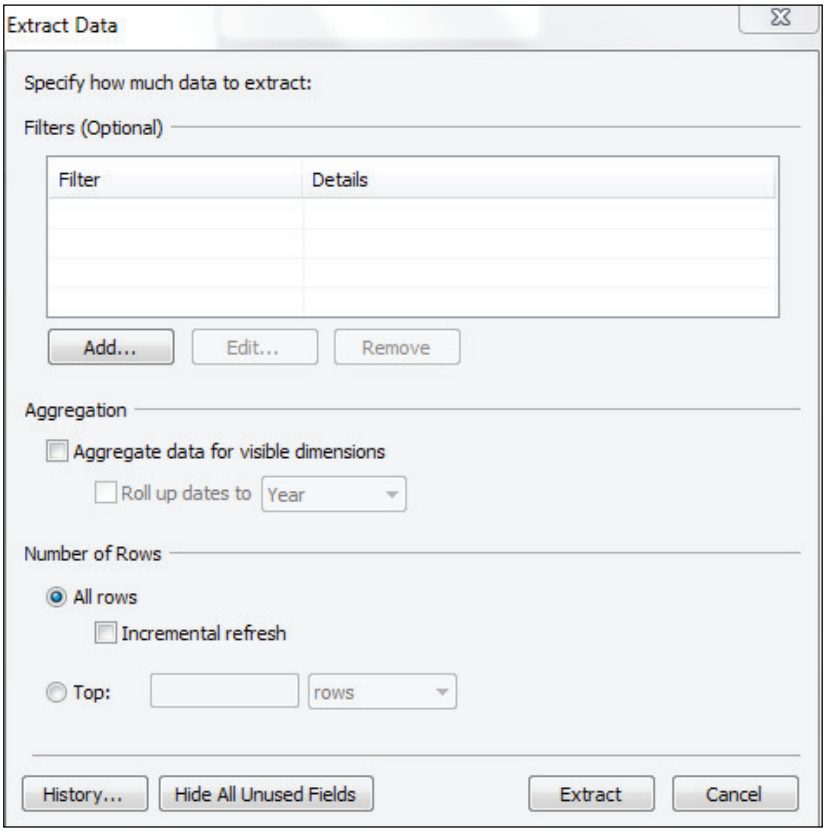
Once the data is loaded on the worksheet, perform the following steps to extract the data:

1. From the main menu toolbar, select **Data**.

2. Under **Data**, expand **Sample - Coffee Chain (Access)**, and select **Extract Data** as shown in the following screenshot:



3. To select all rows, keep the **All rows** option selected as shown in the following screenshot:



4. Hit **Extract**.
5. In the **Save As** dialog box, select a desired location to save the extract.
6. Enter the filename of the Tableau data extract file in the **File name** box in the **Save As** dialog box.
7. Hit **Save**.

There's more...

Robin Kennedy from the The Information Lab Team, a Tableau consulting firm, has written a great insightful blog article about the reasons for a Tableau data extract. You can find this blog post at `http://www.theinformationlab.co.uk/2011/01/20/tableau-extracts-what-why-how-etc/`.

8
Exporting and Sharing

In this chapter, we will see the following recipes:

- Saving a workbook on a Tableau server
- Sharing a workbook on the Web
- Exporting images
- Exporting data

Introduction

Tableau, apart from a workbook, extracts and prints to PDFs and offers customized options to export and share workbooks and data.

Saving a workbook on a Tableau server

Sharing visualizations on a Tableau server is one of the best ways to ensure that the readers are seeing the latest, and sometimes, live information. With the Tableau 8 server, the readers are also able to interact with the visualization by customizing it to their liking.

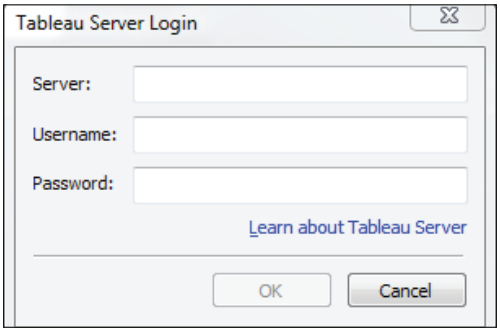
Getting ready

Let's use the sample file `Sample – Coffee Chain (Access)`. Open a new worksheet and select **Sample – Coffee Chain (Access)** as the data source. You would also need access to a Tableau server to complete this example.

How to do it...

Once the data is loaded on the worksheet, perform the following steps to save the workbook on a Tableau server:

1. Drag-and-drop **Market** from **Dimensions** into the **Rows** shelf.
2. Drag-and-drop **Sales** from **Measures** into the **Text input** section in the **Marks** pane.
3. From the main menu toolbar, click on **Server** and then click on **Publish Workbook**.
4. In the **Tableau Server Login** window (shown in the following screenshot), enter the **Server** name or the full path and the **Username** and **Password** values, and hit **OK**:



5. In the **Publish Workbook** field in the **Tableau Server** dialog box, keep **Sheet1** selected in the **Views to Share** pane.
6. Click on **Publish**.

Saving a workbook on the Web

Tableau Public is a great free product offered by Tableau to share any workbook with anyone on the Web. If you do not mind sharing your raw data with the users, then this is a great way of sharing your visualizations as the Tableau server can be very expensive.

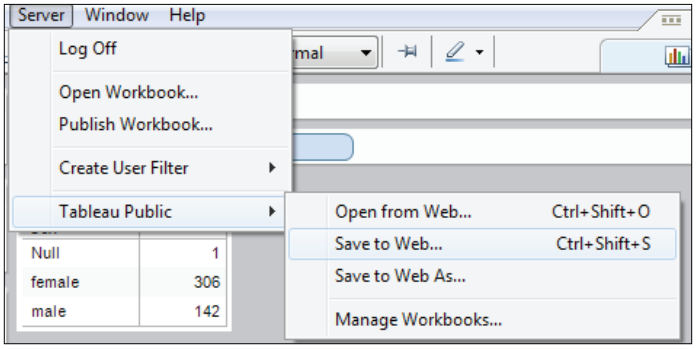
Getting ready

Download and save `titanic.txt` from `http://biostat.mc.vanderbilt.edu/wiki/pub/Main/DataSets/titanic.txt` on your local hard drive. Remember this location, as we will use this file for this recipe. This file lists all the passengers (and their details) that boarded the Titanic on its disastrous voyage. In addition, you'll need to create a Tableau Public free account from `https://public.tableausoftware.com/auth/signup`.

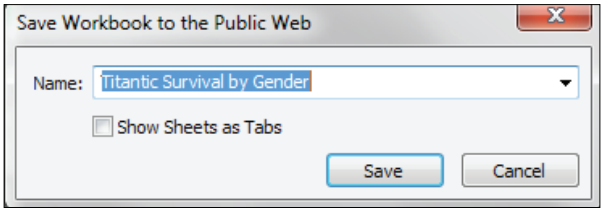
How to do it...

Create a new worksheet and perform the following steps to create a simple graphic and save the workbook on the Web:

1. Click on **Connect to data** to expand that area.
2. Click on **Text File** under the **In a file** section.
3. Find and select `titanic.txt` using the **Browse** button.
4. Keep all the options as is and hit **OK**.
5. In the **Data Connection** dialog box, click on the **Import all data** option.
6. Save the **Tableau Data Extract** file in a familiar location and hit **Save**.
7. Drag-and-drop **sex** from **Dimensions** into the **Rows** shelf.
8. Drag-and-drop **survived** from **Measures** into the **Text** box in the **Marks** pane.
9. From the main menu toolbar, select **Server**, expand **Tableau Public**, and select the **Save to Web** option as shown in the following screenshot:



10. In the **Tableau Public Login** window, enter your e-mail address and password that you used to create your Tableau Public account.
11. In the **Save Workbook to the Public Web** box, enter `Titantic Survival by Gender` and hit **Save** as shown in the following screenshot:



Exporting images

At times, it is quicker or necessary to share visualizations in image formats. Tableau provides an option to export visualizations in JPEG, PNG, BMP, and EMF formats.

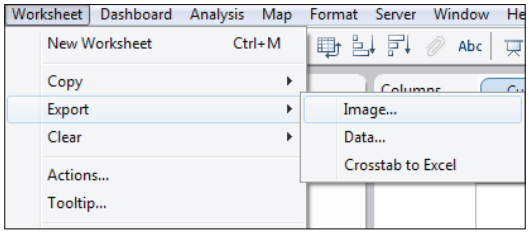
Getting ready

Let's use the sample file `Sample - Superstore Sales (Excel)`. Open a new worksheet and select **Sample – Superstore Sales (Excel)** as the data source.

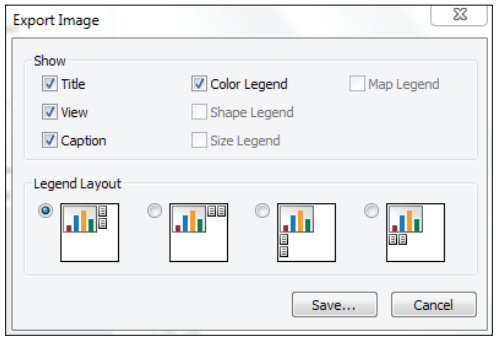
How to do it...

Once the data is loaded on the worksheet, perform the following steps to export a simple visualization in an image format:

1. Click on the **Show Me** button to display the **Show Me** toolbar on the screen.
2. Select **Customer Segment** and **Category** from **Dimensions** as well as **Profit** from **Measures**.
3. Click on the circle views icon on the **Show Me** toolbar.
4. From the main menu toolbar, select **Worksheet**, expand **Export**, and select **Image** as shown in the following screenshot:



5. In the **Export Image** dialog box, keep all options selected and hit **Save**:



6. In the **Save Image** file box, select a folder where you would like to save the file, give a **File name** value, and hit **Save**.

Exporting data

Raw data is helpful and is needed either for tables or for presentations, and Tableau provides an option of exporting the data behind the visualizations in an MS Access database format. This allows the user to get data from a different source, aggregate/manipulate the data, and export it to a database for further use.

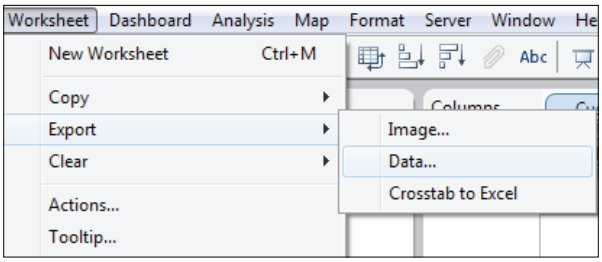
Getting ready

Let's use the sample file `Sample - Superstore Sales (Excel)`. Open a new worksheet and select **Sample – Superstore Sales (Excel)** as the data source.

How to do it...

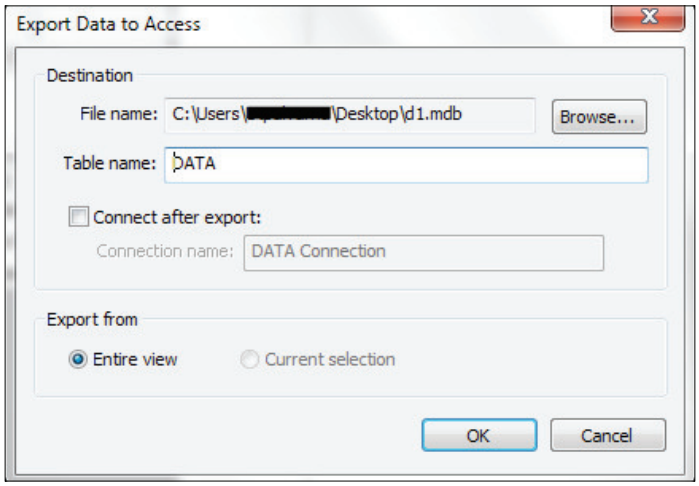
Once the data is loaded on the worksheet, perform the following steps to export a simple visualization in an image format:

1. Click on the **Show Me** button to bring the **Show Me** toolbar on the screen.
2. While holding the *Ctrl* key, click on **Customer Segment** and **Category** from **Dimensions** as well as **Profit** from **Measures**.
3. Click on the circle views icon on the **Show Me** toolbar.
4. From the main menu toolbar, select **Worksheet**, expand **Export**, and select **Data** as shown in the following screenshot:



5. In the **Export Data to Access** save file dialog box, select a location to save the Access database and enter a **File name** value, and hit **Save**.

6. In the next **Export Data to Access** dialog box, keep the default values selected (as shown in the following screenshot) and hit **OK**:



How it works...

If all the default options are selected, Tableau will export the underlying data supporting the visualization to an MS Access database. If any filters are applied in the visualization, Tableau will export only the data that was not excluded because of filters, that is, the exact same data that is supporting the current visualization. You can expect the exported table `DATA` to look similar to the one in the following screenshot:

Category	Customer Segment	Profit (SUM)
Technology	Consumer	156699.3999
Office Supplies	Consumer	88532.3490000001
Furniture	Consumer	42728.234879
Technology	Corporate	374700.5542
Office Supplies	Corporate	203037.286
Furniture	Corporate	22008.083958
Technology	Home Office	173229.137
Office Supplies	Home Office	121145.782
Furniture	Home Office	23979.176491
Technology	Small Business	181684.4242
Office Supplies	Small Business	105306.041
Furniture	Small Business	28717.490953

9
Exploring Advanced Features

The recipes in this chapter are as follows:

- Viewing data
- Changing the mark size
- Using the presentation mode
- Adding annotations
- Excluding data on the fly
- Customizing mark shapes
- Adding drop-down selectors
- Adding search box selectors
- Adding slider selectors
- Creating dashboards
- Creating animated visualizations
- Creating parameters

Introduction

This chapter will cover some advanced features and capabilities of Tableau.

Viewing data

Tableau, by default, aggregates the data and shows the aggregation on the visualization, but it is useful to see the underlying data that is used for that visualization. Using this feature, the user can view data behind the visualization.

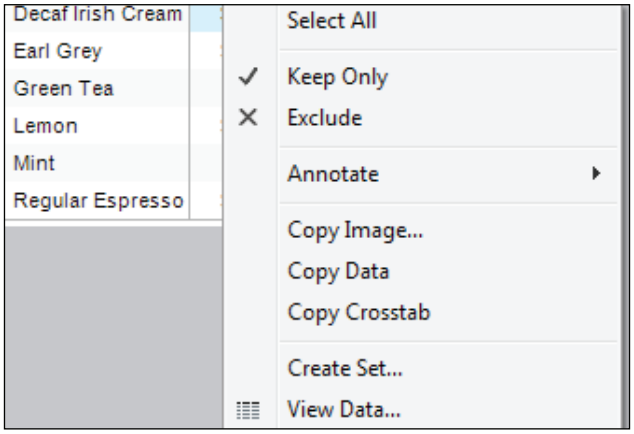
Getting ready

Let's use the sample file `Sample - Coffee Chain (Access)`. Open a new worksheet and select **Sample – Coffee Chain (Access)** as the data source.

How to do it...

Once the data is loaded on the worksheet, follow these steps to create a simple table, and then view the data populating that table.

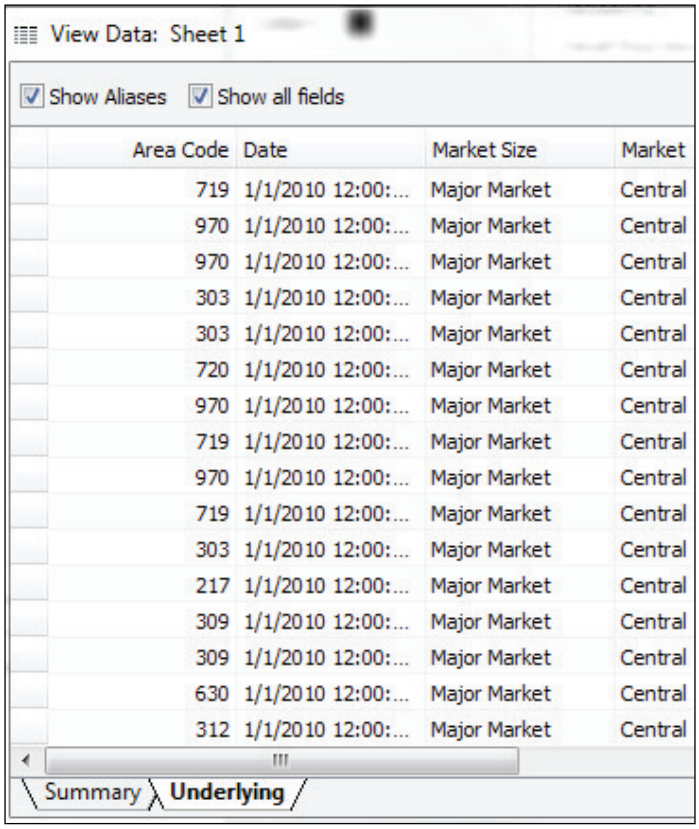
1. Drag-and-drop **Product** from **Dimensions** into the **Rows** shelf.
2. Drag-and-drop **Profit** from **Measures** into the **Text** box under the **Marks** pane.
3. Hit *Ctrl* + *A* on the keyboard to select all the data or the totaled **Profit** value for a single **Product** value.
4. Right-click anywhere on the selected area in the data table, and click on **View Data** as shown in the following screenshot:



5. You can see the summarized data as shown in the following screenshot:

Product	Profit
Regular Espresso	$10,065
Mint	$6,154
Lemon	$29,869
Darjeeling	$29,053
Green Tea	($231)
Earl Grey	$24,164
Decaf Espresso	$29,502
Decaf Irish Cream	$13,989
Columbian	$55,804
Chamomile	$27,231
Caffe Mocha	$17,678
Caffe Latte	$11,375
Amaretto	$4,890

6. Click on the **Underlying** tab to see the complete and raw data supporting the summary view as shown in the following screenshot:



Area Code	Date	Market Size	Market
719	1/1/2010 12:00:...	Major Market	Central
970	1/1/2010 12:00:...	Major Market	Central
970	1/1/2010 12:00:...	Major Market	Central
303	1/1/2010 12:00:...	Major Market	Central
303	1/1/2010 12:00:...	Major Market	Central
720	1/1/2010 12:00:...	Major Market	Central
970	1/1/2010 12:00:...	Major Market	Central
719	1/1/2010 12:00:...	Major Market	Central
970	1/1/2010 12:00:...	Major Market	Central
719	1/1/2010 12:00:...	Major Market	Central
303	1/1/2010 12:00:...	Major Market	Central
217	1/1/2010 12:00:...	Major Market	Central
309	1/1/2010 12:00:...	Major Market	Central
309	1/1/2010 12:00:...	Major Market	Central
630	1/1/2010 12:00:...	Major Market	Central
312	1/1/2010 12:00:...	Major Market	Central

7. To view fields shown on the worksheet, check the **Show all fields** option.

Changing the mark size

Although Tableau's default mark sizes are good, the option of changing (increasing or decreasing) those mark sizes is also useful. This can help the reader see the data better and reduce the confusion caused by overlapping or small marks.

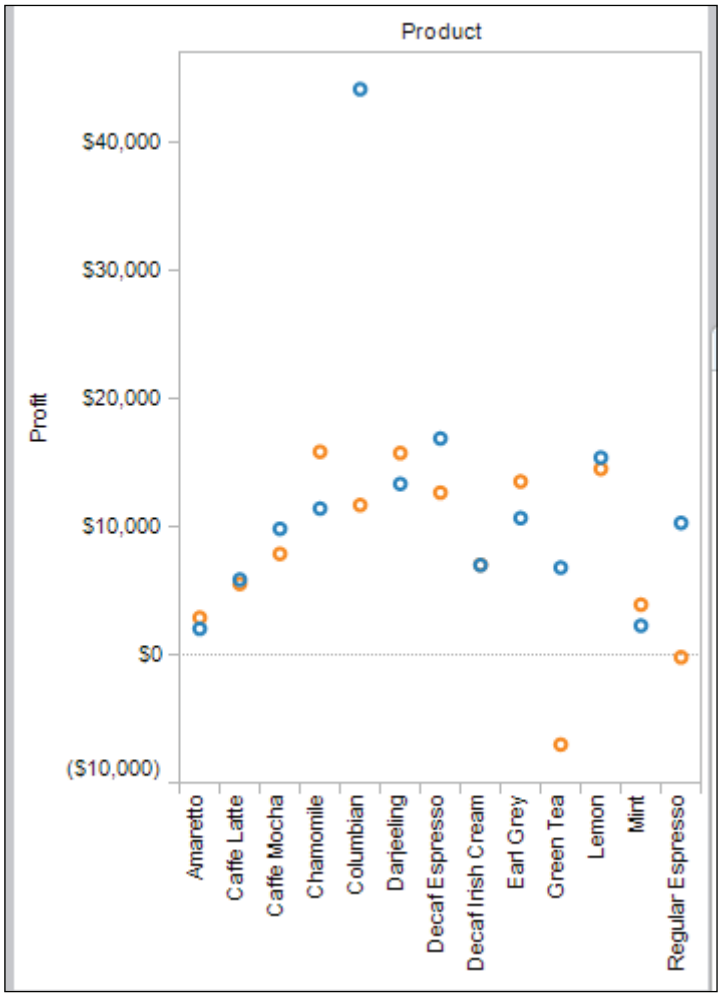
Getting ready

Let's use the sample file `Sample - Coffee Chain (Access)`. Open a new worksheet and select **Sample – Coffee Chain (Access)** as the data source.

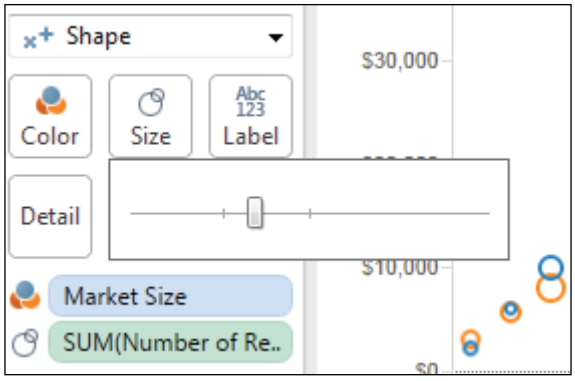
How to do it...

Once the data is loaded on the worksheet, perform the following steps to increase or decrease the mark size:

1. Click on the **Show Me** button to bring the **Show Me** toolbar on the screen.
2. While holding the *Ctrl* key, click on **Product** and **Market Size** from **Dimensions** and **Profit** from **Measures**.
3. Click on the circle view icon on the **Show Me** toolbar to generate a visualization as shown in the following screenshot:



4. To vary the mark size by the number of records in each **Product** and **Market Size** values, drag-and-drop **Number of Records** from **Measures** into the **Size** box under the **Marks** pane.
5. To increase the size of the marks, click on the **Size** box to show the slider and drag the slider to the right as shown in the following screenshot:



6. To decrease the size of the marks, drag the slider to the left.

Using the presentation mode

While editing a Tableau workbook, a user sees many options and a lot of screen real estate is used towards showing such information. Tableau offers a useful feature of displaying the visualization in a presentation mode, removing any clutter or unnecessary options for exploring the visualization.

Getting ready

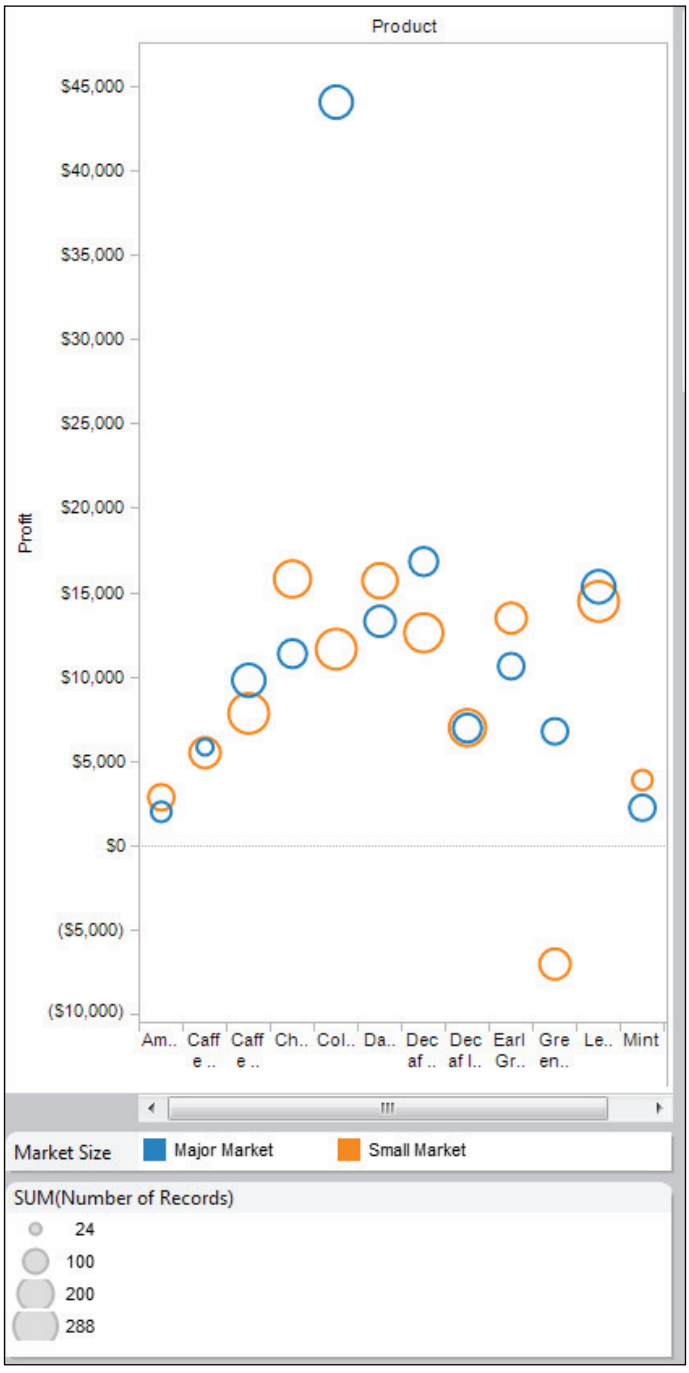
Follow the *Changing the mark size* recipe to create a simple visualization. We will view this visualization in the presentation mode.

How to do it...

Once you have created the simple visualization, perform the following steps to view it in the presentation mode:

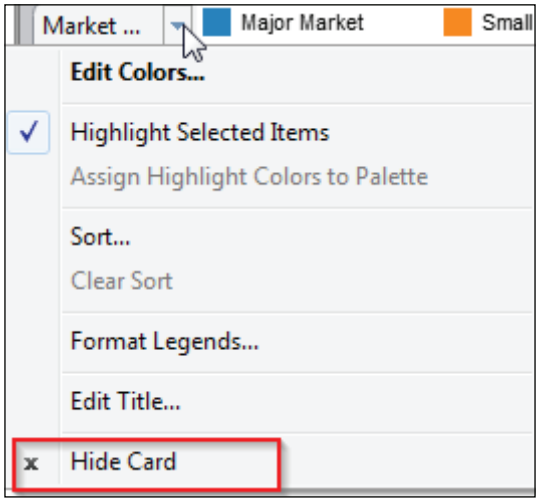
1. Since you cannot change the positions of legends in the presentation mode, move the **Market Size** legend and the **SUM(Number of Records)** legend below the chart.

2. Adjust the height and width of the legend boxes as shown in the following screenshot:



3. Once you are satisfied with the positioning and the dimensions of the legends, click on the **Presentation Mode** button shown in the following screenshot:

4. To hide a legend card during the presentation mode, click on the small dropdown, seen by hovering the mouse over the legend title, and click on **Hide Card** as shown in the following screenshot:



Adding annotations

Although many graphs are self explanatory, annotations on visualizations help the reader understand the graphic better and note any important trends or characteristics of the data. Tableau provides three main types of annotations: **Point**, **Area**, and **Mark**. The **Point** annotation creates annotations for a data point in the visualization. The **Mark** annotation creates annotations for a specific, selected mark whereas the **Area** pane covers an area consisting of many other points of interest.

Getting ready

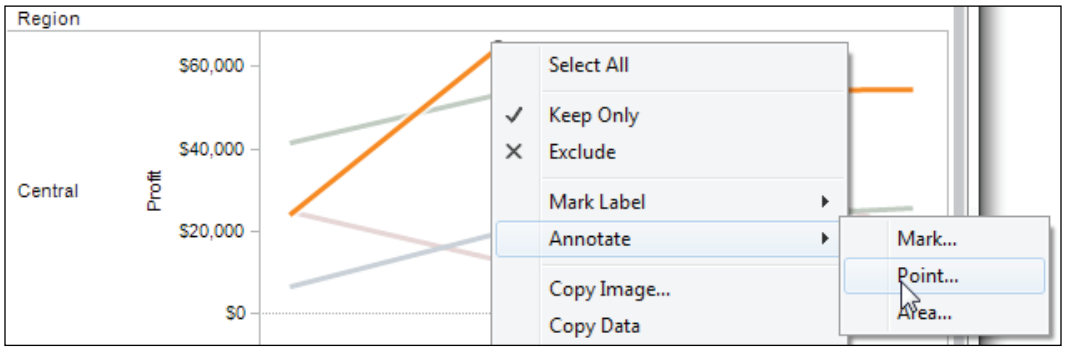
Let's use the sample file `Sample – Superstore Sales (Excel)`. Open a new worksheet and select **Sample – Superstore Sales (Excel)** as the data source.

How to do it...

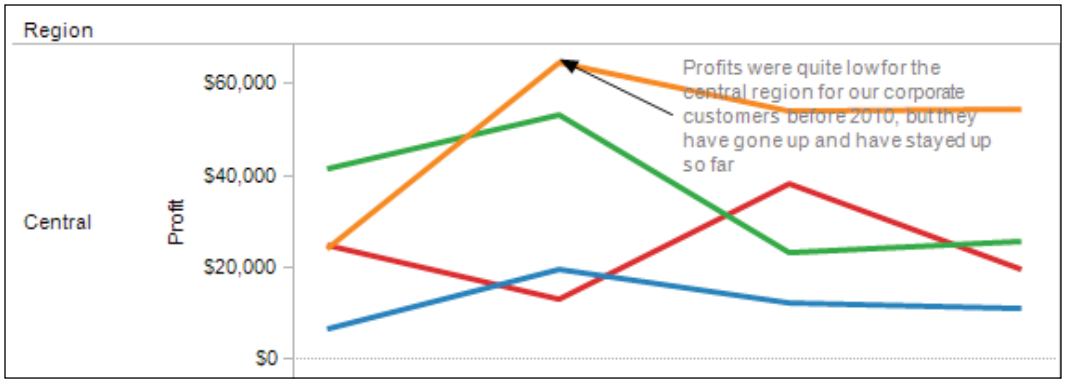
Once the data is loaded on the worksheet, perform the following steps to add annotations to a visualization:

1. Click on the **Show Me** button to bring the **Show Me** toolbar on the screen.
2. While holding the *Ctrl* key, click on **Customer Segment**, **Order Date**, and **Region** from **Dimensions** and **Profit** from **Measures**.
3. Click on **lines (continuous)** on the **Show Me** toolbar to create a visualization as shown in the following screenshot:

4. Right-click on the **Central** region's **Profit** value for the year **2010**, expand the menu for **Annotate**, and select **Point...** as shown in the following screenshot:

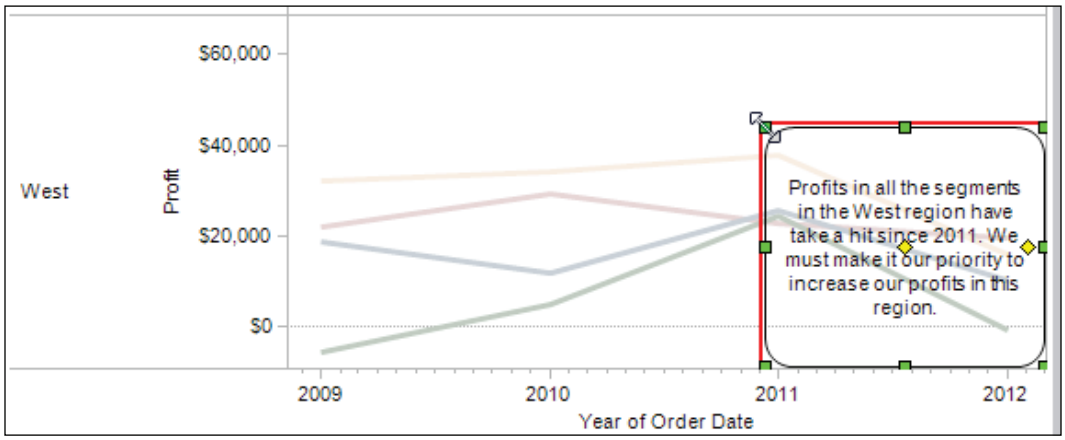


5. In the **Edit Annotation** box, enter `Profits were quite low for the central region for our corporate customers before 2010, but they have gone up and have stayed up so far` and select **OK**.
6. You can see the annotation for the year **2010** in the **Central** region pane as shown in the following screenshot:



7. Right-click on the **Home Office Customer Segment** point for the year **2012** in the **West** region, expand the menu for **Annotate**, and select **Area**.
8. In the **Edit Annotation** box, enter `Profits in all the segments in the West region have taken a hit since 2011. We must make it our priority to increase our profits in this region.` and hit **OK**.

9. Click on the **Annotation** box and drag the top-left corner to cover points from the year **2011** as shown in the following screenshot:



Excluding data on the fly

We can easily set multiple filters on a visualization; however, while exploring the data, the user has a very common need to exclude certain data without explicitly creating a filter beforehand. Tableau offers this feature, and exclusion happens on the selected marks and a filter is created.

Getting ready

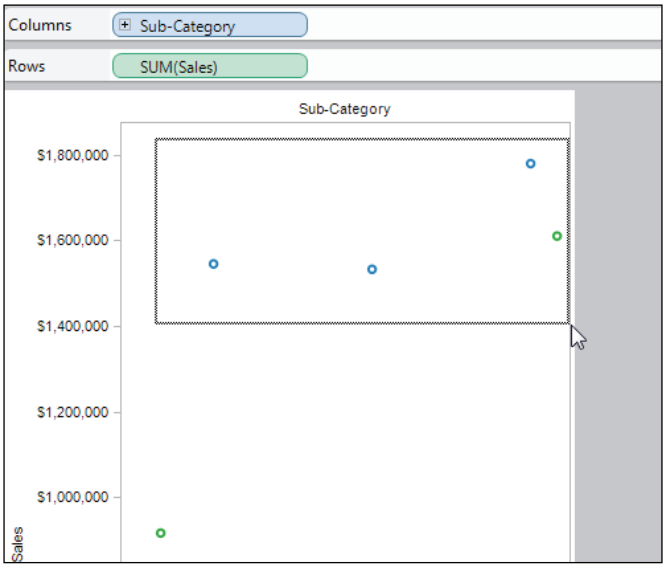
Let's use the sample file `Sample - Superstore Sales (Excel)`. Open a new worksheet and select **Sample – Superstore Sales (Excel)** as the data source.

How to do it...

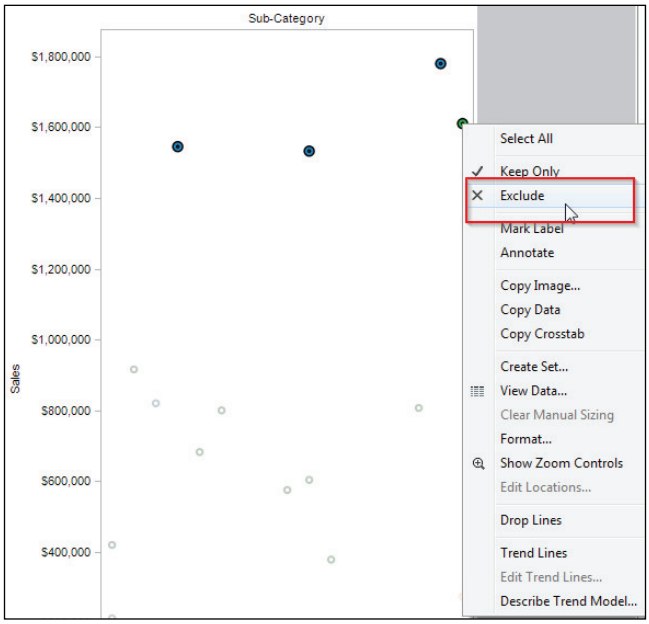
Once the data is loaded on the worksheet, perform the following steps to create a visualization and exclude or filter some data from the visualization:

1. Click on the **Show Me** button to bring the **Show Me** toolbar on the screen.
2. While holding the *Ctrl* key, click on **Ship Mode** and **Sub-Category** from **Dimensions** and **Sales** from **Measures**.
3. Click on the circle view icon on the **Show Me** toolbar.

4. To exclude every product with a **Sales** value of more than 1,000,000 dollars, select all the four marks above **$1,000,000** as shown in the following screenshot:



5. The selected marks will become darker in color, and all the other marks will dim down. Right-click on any selected mark, and click-on **Exclude** as shown in the following screenshot:



Customizing mark shapes

From branding your company to distinguishing the data, custom mark shapes can help in customizing the visualization. Once you have the desired shape of files, you can easily use them instead of using Tableau's standard mark shapes.

Getting ready

This recipe requires modifying the custom shape image files, which could be of the JPG, BMP, and GIF formats. Specifically, it makes the background transparent while keeps the original size of 32 by 32 pixels. The tutorial given at `http://www.interworks.com/blogs/iwbiteam/2012/01/27/using-custom-shapes-tableau` explains in detail how to modify the shape properties. For this recipe, you can choose to create and modify your own shapes or download the ones used in this recipe.

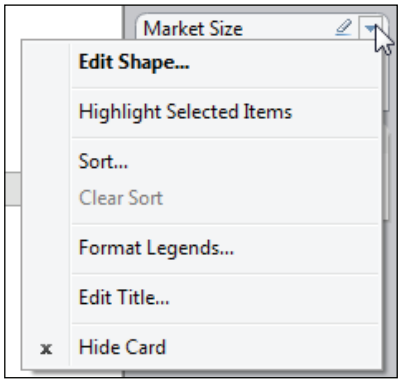
Let's use the sample file `Sample – Coffee Chain (Access)`. Open a new worksheet and select **Sample – Coffee Chain (Access)** as the data source.

How to do it...

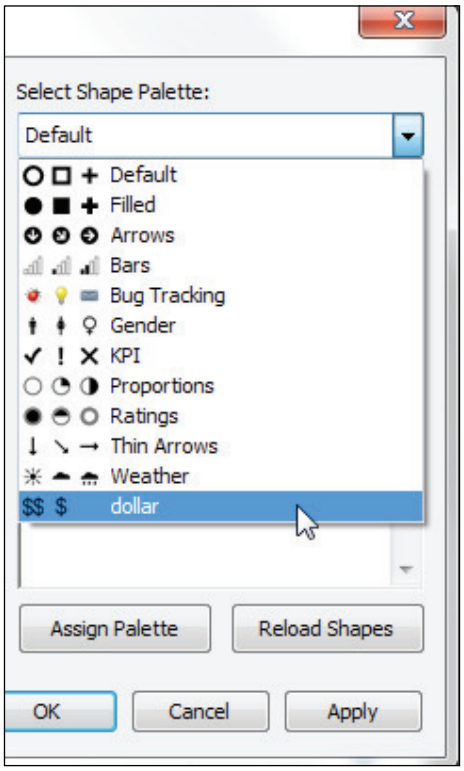
Once the data is loaded on the worksheet, perform the following steps to use the custom shape marks:

1. Create a folder called `dollar` in the `Shapes` folder in `My Tableau Repository`.
2. Copy the `single.png` and `double.png` images in the `dollar` folder.
3. Drag-and-drop **Product** from **Dimensions** into the **Columns** shelf.
4. Drag-and-drop **Sales** from **Measures** into the **Rows** shelf.
5. From the **Marks** dropdown, select **Shape**.
6. Drag-and-drop **Market Size** from **Dimensions** into the **Shape** box.
7. Drag-and-drop **Market Size** from **Dimensions** into the **Color** box.

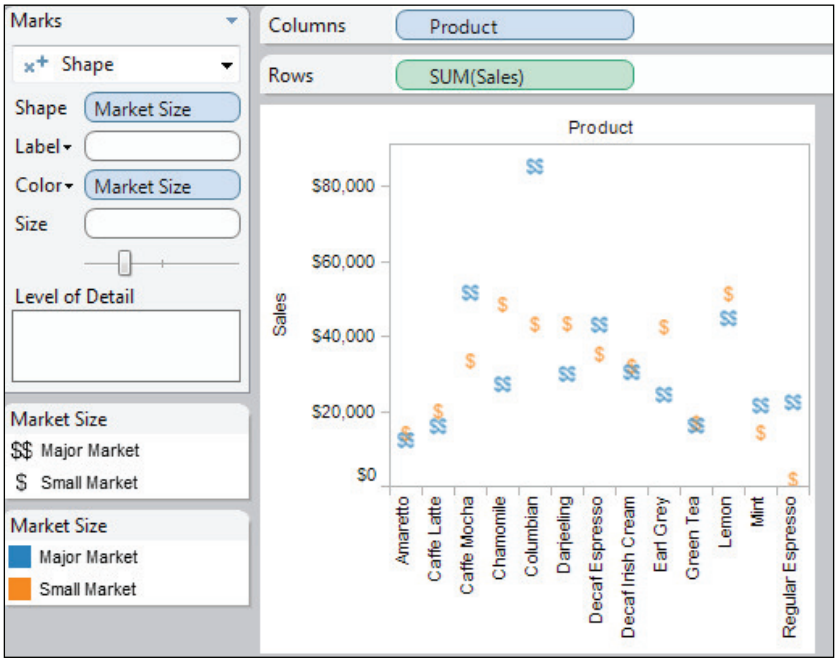
8. Hover your mouse over the **Market Size** legend shelf, click on the small drop-drown arrow, and select **Edit Shape...** as shown in the following screenshot:



9. In the **Edit Shape [Market Size]** dialog box, hit the **Reload Shapes** button.
10. In the **Select Shape Palette** dropdrown, select **dollar** as shown as in the following screenshot:



11. Select **Major Market** under **Select Data Item**, and select the double dollar sign.
12. Select **Small Market** under **Select Data Item**, and select the single dollar sign.
13. Select **OK** and the final visualization should look like the following screenshot:



Adding drop-down selectors

It is useful to give the readers some control over the visualization, and using quick filters is a great way of doing so. Drop-drown selectors are good if you want the reader to select one value without the risk of misspelling.

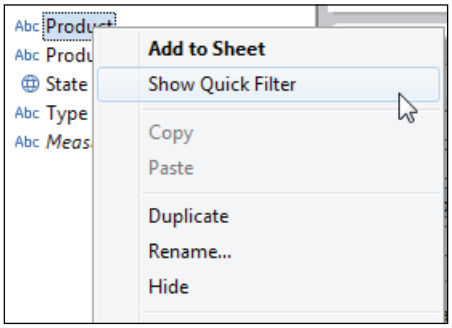
Getting ready

Let's use the sample file `Sample – Coffee Chain (Access)`. Open a new worksheet and select **Sample – Coffee Chain (Access)** as the data source.

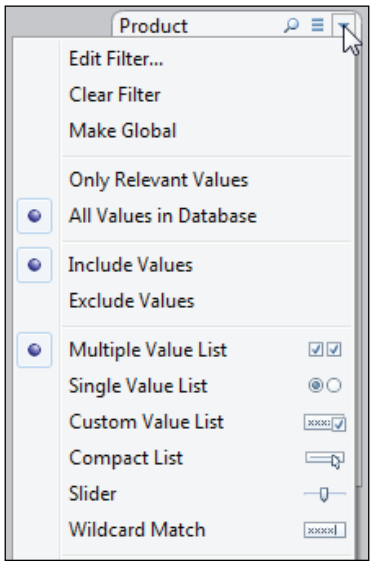
How to do it...

Once the data is loaded on the worksheet, perform the following steps to create a drop-down selector on your visualization:

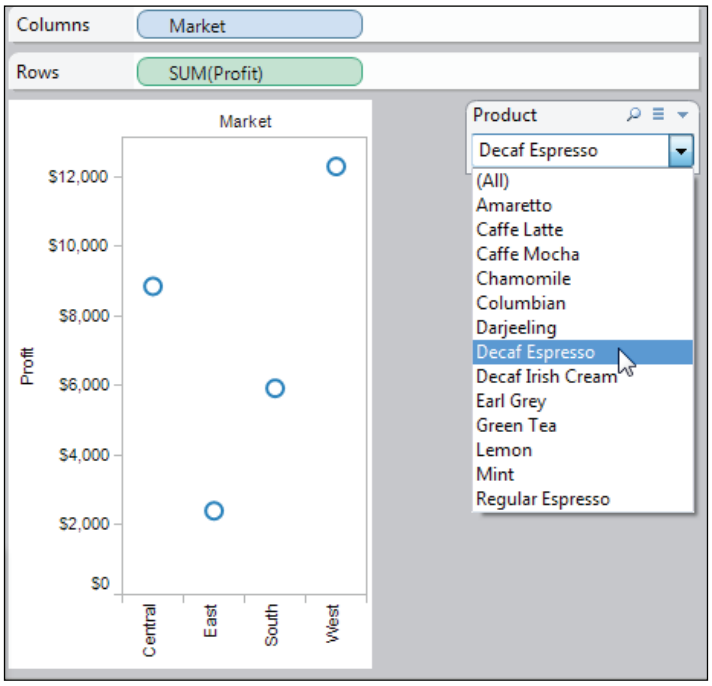
1. Click on the **Show Me** button to bring the **Show Me** toolbar on the screen.
2. While holding the *Ctrl* key, click on **Market** and **Type** from **Dimensions** and **Profit** from **Measures**.
3. Click on the circle views icon on the **Show Me** toolbar.
4. Right-click on **Product** from **Dimensions**, and select **Show Quick Filter** as shown in the following screenshot:



5. On the **Product** filter box, click on the small drop-down arrow to open the filter properties menu as shown in the following screenshot:



6. From this menu, select **Compact List** to create a drop-down selector box. The final visualization will look similar to the one shown in the following screenshot:



Adding search box selectors

Search box selectors are useful when you want the readers to type a part of the value and yet be able to filter the data. This does assume that the readers know the underlying values of the filtered fields.

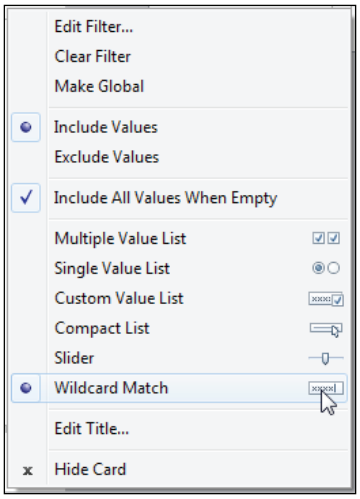
Getting ready

Let's use the sample file `Sample - Coffee Chain (Access)`. Open a new worksheet and select **Sample – Coffee Chain (Access)** as the data source. Follow all the steps, except for the last step, given in the recipe *Adding drop-down selectors*.

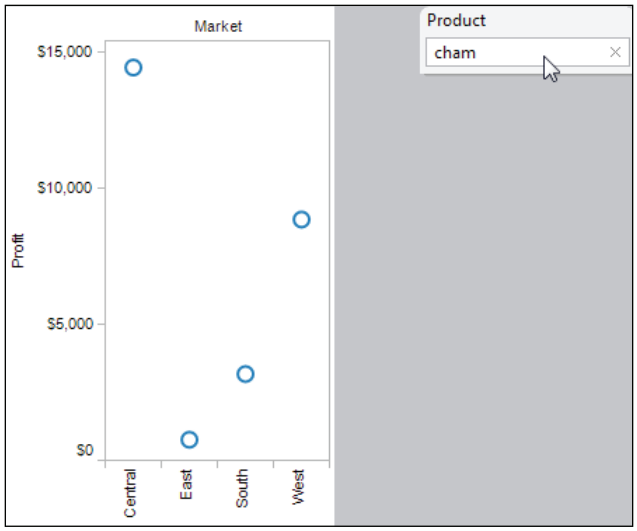
How to do it...

Once the data is loaded on the worksheet and you have placed a quick filter, perform the following steps to add a search box selector to your visualization:

1. From the quick filter drop-down menu, select **Wildcard Match** as shown in the following screenshot:



2. To see this search box in action, enter `cham` in the search box and hit *Enter*. You'll see a visualization for the **Product** values that contain `cham` in their names as shown in the following screenshot:



Adding slider selectors

Slider selectors add more flexibility for filtering the numeric or date type of data, though this selector can be used on any type of data.

Getting ready

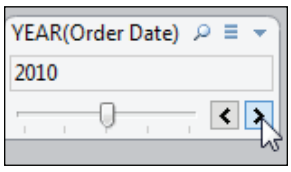
Let's use the sample file `Sample - Superstore Sales (Excel)`. Open a new worksheet and select **Sample – Superstore Sales (Excel)** as the data source.

How to do it...

Once the data is loaded on the worksheet, perform the following steps to add slider selector to your visualization:

1. Click on the **Show Me** button to bring the **Show Me** toolbar on the screen.
2. While holding the *Ctrl* key, click on **Customer Segment** and **Category** from **Dimensions** and **Sales** from **Measures**.
3. Click on the circle views icon on the **Show Me** toolbar.
4. Right-click on **Order Date** from **Dimensions** and select **Show Quick Filter** to show a multiple checkbox selector as shown in the following screenshot:

5. On the **YEAR(Order Date)** quick filter box, click on the small drop-down arrow to modify the filter properties.
6. Select **Slider** from this box to add a slider filter as shown in the following screenshot:



Creating dashboards

Dashboards in Tableau are very powerful as they are a compilation of individual visualizations on different sheets. This provides the reader with a lot of information on one single view with all the filters, parameters, and legends of individual visualizations. Complex types of dashboards can be created, such as those allowing drill-through aggregate information and viewing the details.

Getting ready

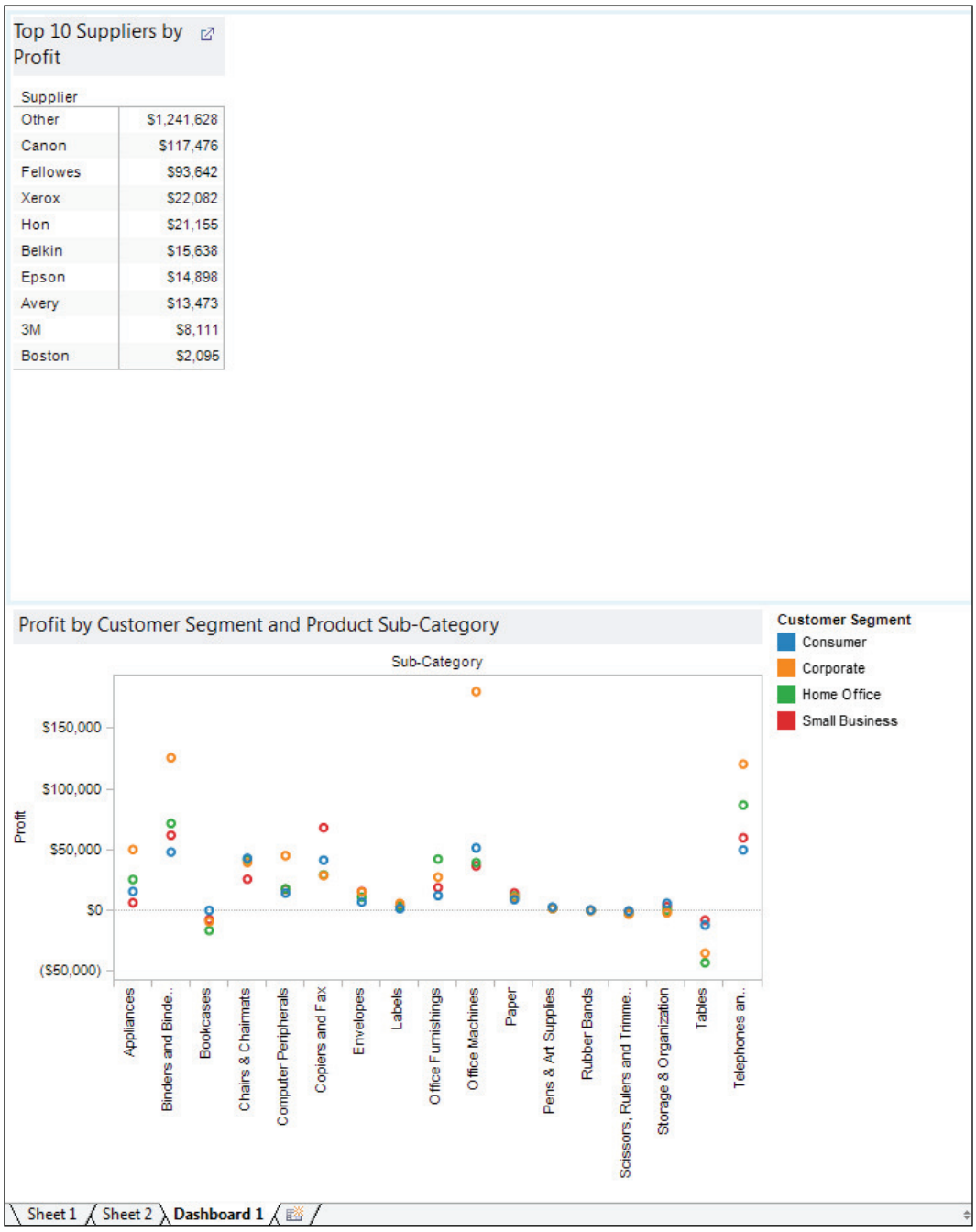
Let's use the sample file `Sample – Superstore Sales (Excel)`. Open a new worksheet and select **Sample – Superstore Sales (Excel)** as the data source.

How to do it...

Once the data is loaded on the worksheet, perform the following steps to create a simple dashboard:

1. Click on the **Show Me** button to bring the **Show Me** toolbar on the screen.
2. While holding the *Ctrl* key, click on **Customer Segment** and **Sub-Category** from **Dimensions** and **Profit** from **Measures**.
3. Click on the circle views icon on the **Show Me** toolbar.
4. From the main menu toolbar, select **Show Title** under **Worksheet**.
5. Double-click on the **Title** shelf and change the title to `Profit by Customer Segment and Product Sub-Category`.
6. From the main menu toolbar, select **New Worksheet** under **Worksheet**.
7. Drag-and-drop **Supplier** into the **Rows** shelf.
8. Drag-and- drop **Profit** into the **Text** shelf under the **Marks** pane.
9. Drag-and-drop **Supplier** into the **Filters** shelf.
10. In the **Filters [Supplier]** dialog box, select the **Top** tab and select the **By Field** radio button.
11. Click on **OK**.
12. From the main menu toolbar, select **Show Title** under **Worksheet**.
13. Double-click on the **Title** shelf and change the title to `Top 10 Suppliers by Profit`.
14. From the main menu toolbar, select **New Dashboard** under **Dashboard**.
15. In the **Dashboard** view, drag-and-drop **Sheet 1**.

16. Drag-and-drop **Sheet 2** on top of **Sheet 1** as shown in the following screenshot:



Creating animated visualizations

Animated visualizations are useful for spotting a measure in seasonal trends or simply observing measures over a period of time.

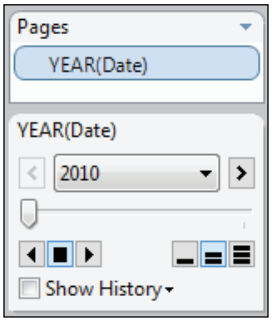
Getting ready

Let's use the sample file `Sample - Coffee Chain (Access)`. Open a new worksheet and select **Sample – Coffee Chain (Access)** as the data source.

How to do it...

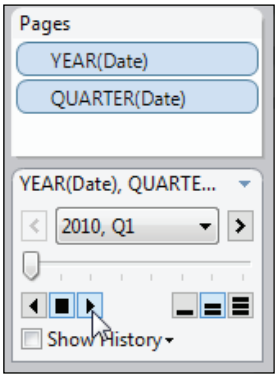
Once the data is loaded on the worksheet, perform the following steps to create an animated visualization:

1. Click on the **Show Me** button to bring the **Show Me** toolbar on the screen.
2. While holding the *Ctrl* key, click on **Product** from **Dimensions** and **Sales** from **Measures**.
3. Click on the horizontal bars icon on the **Show Me** toolbar.
4. Drag-and-drop **Date** from **Dimensions** into the **Pages** shelf as shown in the following screenshot:



5. To show the graphs by quarters, drag-and-drop **Date** from **Dimensions** into the **Pages** shelf again.

6. To play the animation, click on the play button from the **Pages** shelf as shown in the following screenshot:



Creating parameters

Parameters allow more interaction with the reader by allowing him/her to change certain values and see how this impacts other measures. By creating parameters, the reader can be put in charge of evaluating various what-if scenarios or given options to choose the number of items to view.

Getting ready

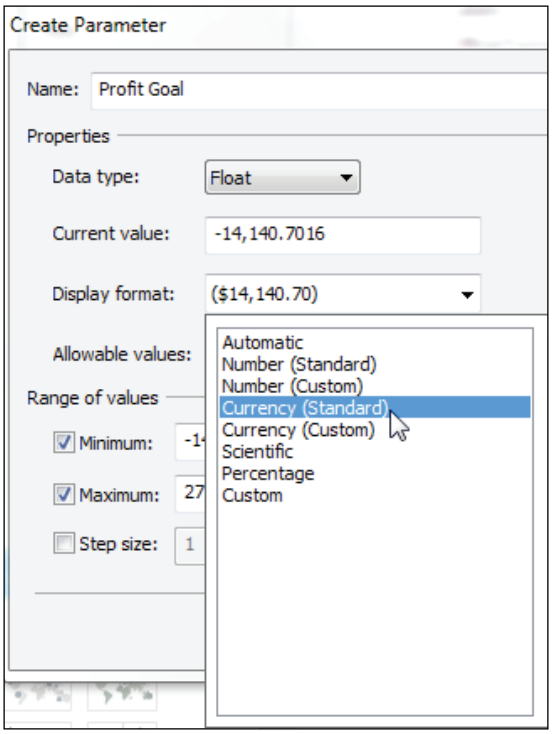
Let's use the sample file `Sample - Superstore Sales (Excel)`. Open a new worksheet and select **Sample – Superstore Sales (Excel)** as the data source.

How to do it...

Once the data is loaded on the worksheet, perform the following steps to create a parameter, and use it to dynamically adjust the visualization:

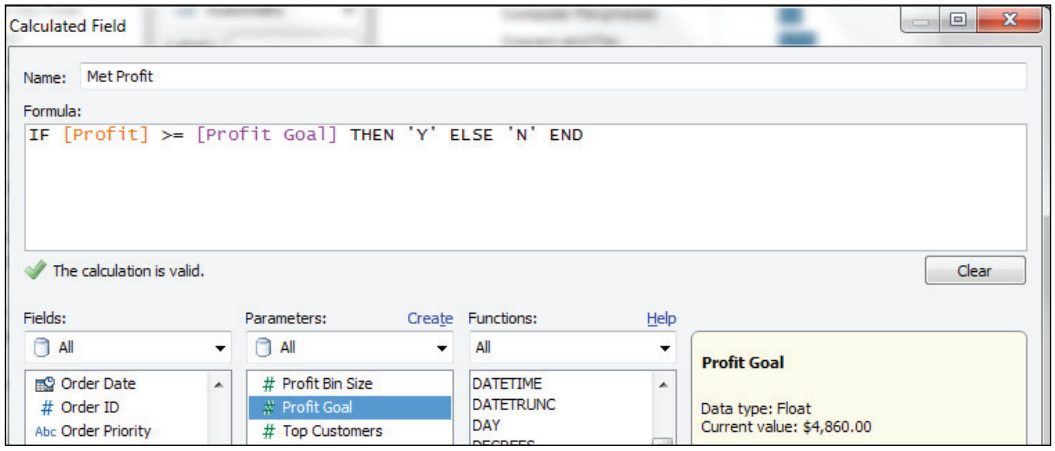
1. Click on the **Show Me** button to bring the **Show Me** toolbar on the screen.
2. While holding the *Ctrl* key, click on **Sub-Category** and **Region** from **Dimensions**, and **Profit** from **Measures**.
3. Click on the horizontal bars icon on the **Show Me** toolbar.
4. Right-click on **Profit** from **Measures** and select **Create Parameter**.
5. In the **Create Parameter** dialog box, enter `Profit Goal` in the **Name** textbox.

6. Change the **Display** format to **Currency (Standard)** as shown in the following screenshot:

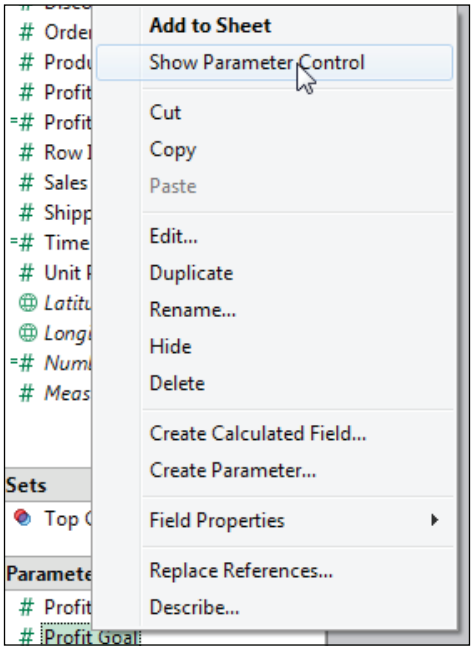


7. Check the **Step size** checkbox and enter `1000`.
8. Select **OK**.
9. Right-click on the newly created parameter **Profit Goal**, and select **Create Calculated Field**.
10. In the **Calculated Field** dialog box, in the **Name** textbox, enter `Met Profit`.

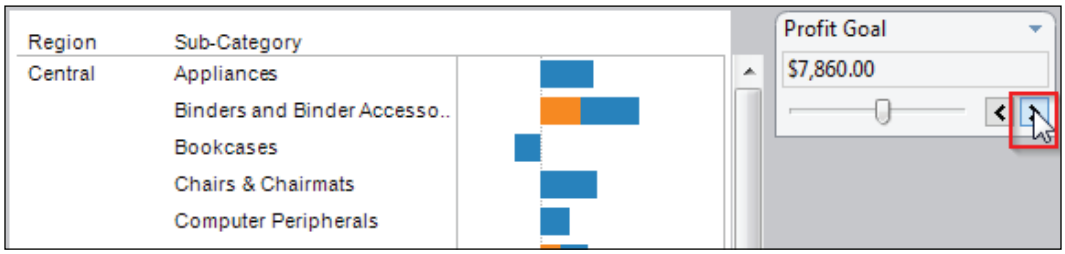
11. In the **Formula** box, enter the `IF [Profit] >= [Profit Goal] THEN 'Y' ELSE 'N' END` formula as shown in the following screenshot:



12. Select **OK**.
13. Drag-and-drop the newly created calculated field **Met Profit Goal** in the **Color** box into the **Marks** pane.
14. Right-click on the **Profit Goal** parameter and select **Show Parameter Control** as shown in the following screenshot:



15. Change the values of **Profit Goal** using the slider or left and right arrow buttons as shown in the following screenshot:



Part 2: Table of Contents

Chapter 10. Interacting with Tableau Server

Not too long ago I had a Tableau engagement in Houston, Texas with a relatively small company in the oil and gas industry. This company was not ready to purchase a copy of Tableau Server. I spent quite a bit of time considering how best to create and deploy a centralized repository to share packaged workbooks and data sources that could be easily shared among Tableau authors and also downloaded by end users for display on Tableau Reader. The security concerns, maintenance difficulties and functionality limitations caused me to really begin to appreciate Tableau Server. Although Tableau Server is not necessary for every environment in which Tableau Desktop is used, it is necessary for many environments, and even when not strictly required can be very helpful.

This chapter does not assume any knowledge of Tableau Server. Thus, when Tableau Server is the focus of a given section, basic information is provided. When Tableau Desktop is the focus, however, proficiency is assumed.

Given the preceding paragraph, let's begin with a brief explanation of what Tableau Server is. Let's paraphrase the help documentation:

Note

Tableau Server is an online solution for sharing, distributing, and collaborating on content created in Tableau Desktop. Benefits include providing an environment where end users can securely view and explore data visualizations that are constantly updated from underlying data sources so that content is always fresh.

This chapter does assume access to Tableau Server with sufficient privileges for publishing data sources and editing in the web authoring environment. If you do not have access to Tableau Server but would like to work through the exercises in this chapter, consider downloading a trial version which, at the time of writing, is fully functional for two weeks. If you would like a longer trial, consider joining the Tableau Beta program at http://www.tableau.com/getbeta, which will give you two weeks of access for each beta release in which you participate.

The scope of this chapter is limited to the Tableau Desktop author's interaction with Tableau Server. Topics such as installation and upgrades, authentication and access, security configuration, and command line utilities are not directly related to the Tableau Desktop author's interaction with Tableau Server and are thus are not included in this chapter. At the time writing I am unaware of any books focused exclusively on Tableau Server; however, the help documentation is quite good. If you have questions related to the topics listed or other Tableau Server centric topics, be sure to visit online help at http://onlinehelp.tableau.com/current/server/en-us/help.htm.

This chapter will explore the following topics:

- Tableau file types
- Tableau Server architecture
- Tableau Server web authoring environment
- Tableau Server revision history
- User filters
- Accessing the Tableau Server Performance Recording dashboard

Tableau file types

We will begin our discussion of Tableau Server by considering the various Tableau file types. This may seem a surprising place to begin, but as you read you will discover that a clear understanding of file types provides the Tableau Desktop author with foundational knowledge for efficiently and effectively interacting with Tableau Server.

The file types discussed in the following section that are relevant for understanding how to interact with Tableau Server are considered in some depth. The file types that are not relevant for understanding Tableau Server are considered only briefly. Some file types (for example, those associated with license activation) are not considered.

Tableau Data Source (.tds)

- **File format type**: XML.
- **What it contains**: Metadata.
- **Why it is useful**: The `.tds` file is important because it allows the Tableau author to define default formatting and aggregation, calculated fields, data types, field types, and more. Furthermore, the `.tds` file can be published to Tableau Server and thus accessed by other authors in the environment. This effectively makes a `.tds` file a playbook ensuring consistency across the organization. This important feature will be explored more fully in the section on *Tableau Server architecture*.
- **How it's generated**: A `.tds` file can be generated by right-clicking on a data source in the **Data** pane and selecting **Add to Saved Data Sources...** followed by selecting **Tableau Data Source** in the resulting dialog box. A `.tds` file can also be generated when publishing to Tableau Server via **Server** | **Publish Data Source** | **[data source]**. Later in this chapter, an exercise is included that demonstrates how to publish a `.tds` file and also a `.tdsx` file.
- **How to access it**: The `.tds` file type is usually accessed in one of two places. First it can be stored in **My Tableau Repository** | **Datasources**. When stored in this directory, a .tds file will be displayed in the left-hand portion of the **Start Page** under the section entitled **Saved Data Sources**. The second place a `.tds` file is often stored is on Tableau Server. Selecting **Data** | **New Data Source** and choosing **Tableau Server** allows the Tableau author to point to the `.tds` and `.tdsx` files that have been published to Tableau Server.

Tableau Packaged Data Source (.tdsx)

- **File format type**: Compressed.
- **What it contains**: Metadata and a data extract.
- **Why it's useful**: The `.tdsx` file is useful because it can be accessed for both metadata and data. Tableau authors can access a `.tdsx` file located on Tableau Server as a data source thus eliminating the need for a workbook to connect directly to an external data source. A published `.tdsx` file can be placed on a schedule so that it is regularly updated from the underlying data source.
- **How it's generated**: A `.tdsx` file can be generated by right-clicking on a data source in the **Data** pane and selecting **Add to Saved Data Sources...** followed by selecting **Tableau Packaged Data Source** in the resulting dialog box. Like the `.tds` file, the `.tdsx` file can also be generated when publishing to Tableau server via **Server** | **Publish Data Source** | **[data source]**. See the following exercise for more details.
- **How to access it**: A `.tdsx` file is accessed in the same way a `.tds` file is. First, it can be stored in **My Tableau Repository** | **Datasources**. When stored in this directory a `.tdsx` file will display in the left-hand portion of the **Start Page** under the section entitled **Saved Data Sources**. The second place a `.tdsx` file is often stored is on Tableau Server. Selecting **Data** | **New Data Source** and choosing **Tableau Server** allows the Tableau author to point to the `.tds` and `.tdsx` files that have been published to a given instance of Tableau Server.

Exercise - publish a data source to Tableau Server

1. Navigate to https://public.tableau.com/profile/david.baldwin#!/ to locate and download the workbook associated with this chapter.
2. Navigate to the worksheet entitled `Publish`.
3. Reference the **Caption** in the worksheet to download the content necessary to complete the exercise.
4. Select **Data** | **New Data Source** and connect to `My Superstore.xls`, which was downloaded in the previous step.
5. Connect to the **Orders** table.
6. Name the data source `My Superstore`.
7. Return to the `Publish` worksheet.
8. If you have not already done so, log in to an instance of Tableau Server.
9. In the **Data** pane, select the `My Superstore` data source.
10. Select **Server** | **Publish Data Source** | **My Superstore**.
11. In the resulting dialog box, choose your desired settings:

Note

Note the bottom of the dialog box where you will see an option entitled **Include external files**. Selecting this box will cause Tableau to create and upload a package data source, that is, a `.tdsx` file. Deselecting this box will cause Tableau to generate and upload a `.tds` file. Of course, in this case, since the data source is an Excel file, deselecting **Include external files** would cause Tableau to create a data source that would only be accessible via your local

machine.

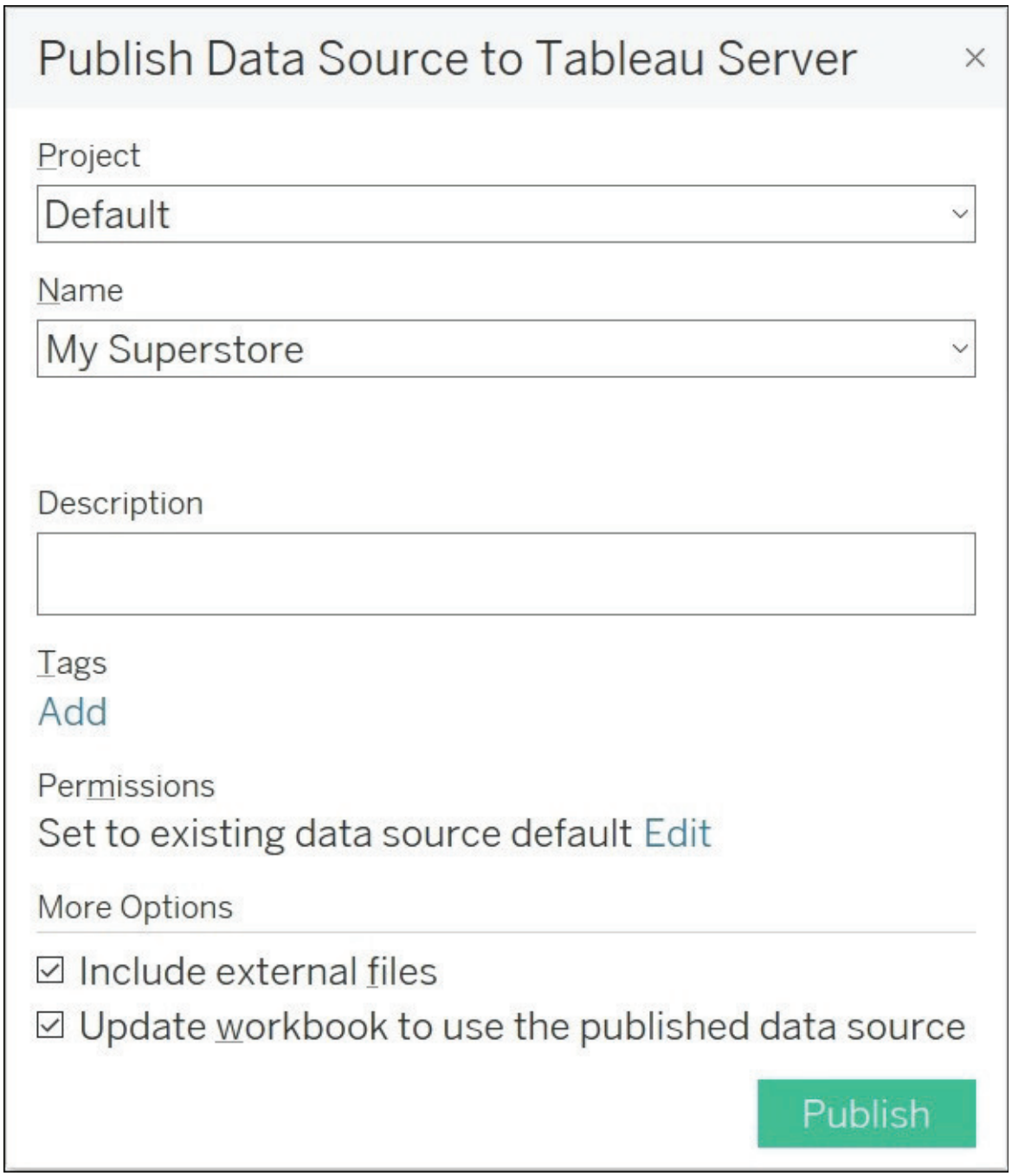

Publish Data Source to Tableau Server
Project
Default
Name
My Superstore
Description
Tags
Add
Permissions
Set to existing data source default Edit
More Options
Include external files
Update workbook to use the published data source
Publish

Tableau Workbook (.twb)

- **File format type**: XML.
- **What it contains**: Metadata and schema. The schema defines the visualizations in the workbook. Note that schema, in this context, refers to the XML that defines the visual components of the workbook, including the visualizations displayed on worksheets as well as the layout of dashboards and stories.
- **Why it's useful**: The `.twb` file type is the file type most often used by the Tableau author. It is necessary for creating visualizations that point to a live dataset. Thus, real-time solutions will utilize this file type.
- **How it's generated**: A `.twb` file is created via **File** | **Save As** and selecting the `.twb` file type in the resulting dialog box.
- **How it's accessed**: A `.twb` file can be opened via Tableau Desktop or accessed via a browser pointing to an instance of Tableau Server. Since a `.twb` file is XML, it can be opened, viewed, and updated via a text editor.

Tableau Packaged Workbook (.twbx)

- **File format type**: Compressed.
- **What it contains**: Metadata, schema, and optionally, one or more data extracts.
- **Why it's useful**: The `.twbx` file type is required for use with Tableau Reader. It can also be effectively used with Tableau Server when accessing data sources to which Tableau Server does not directly connect such as flat files, Excel, and Access. Drawbacks to `.twbx` files will be discussed in the following section.
- **How it's generated**: A `.twbx` file is created via **File** | **Save As** and selecting the `.twbx` file type in the resulting dialog box.
- **How it's accessed**: A `.twbx` file can be opened via Tableau Desktop or accessed via a browser pointing to an instance of Tableau Server. Since a `.twbx` file is a compressed file it can also be unzipped via a compression utility such as WinZip or 7-Zip.

Other file types

The remaining file types the Tableau author should be familiar with are not particularly relevant for Tableau Server and will thus only be briefly discussed here:

- **Tableau Data Extract (`.tde`)**: The `.tde` file can be generated via the following: If a `.twb` file is opened in Tableau Desktop, a `.tde` file can be created by right-clicking on a data source in the **Data** pane and selecting **Extract Data**. After selecting the desired options, Tableau will provide a dialog box for the author to save the `.tde` file in a given location. The `.tde` file can be used to create a local snapshot of a data source for a quicker authoring experience. That same local snapshot is also portable and can thus be used offline. Often the data extracts compressed inside `.twbx` and `.tdsx` files are in the `.tde` format.

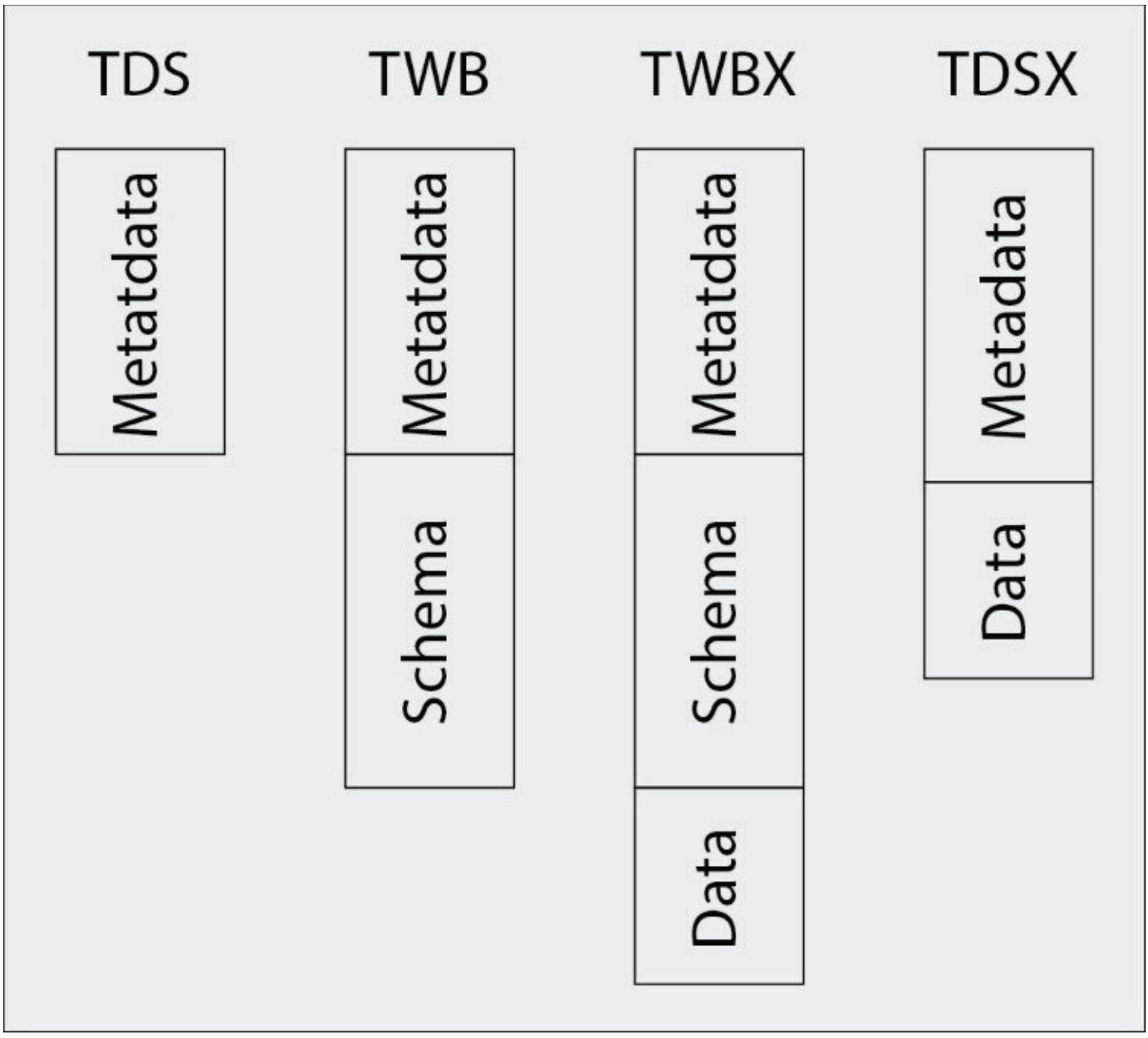
  ### Note

  An important clarification point should be made here. Often Tableau authors will refer to publishing an extract to Tableau Server. The extract that is published is not a `.tde` file. Rather, it is a `.tdsx` file. See Chapter 11, *Improving Performance* for additional information about the `.tde` file format.

- **Tableau Bookmark (`.tbm`)**: The `.tbm` file can be generated via **Window** | **Bookmark** | **Create Bookmark**. It can be useful for duplicating worksheets across multiple workbooks and also for sharing formatting across multiple workbooks.
- **Tableau Map Source (`.tms`)**: The `.tms` file is discussed in detail in Chapter 8, *Mapping*.
- **Tableau Preferences Source (`.tps`)**: The `.tps` file can be used to create custom color palettes. This can be helpful when an organization wishes to use its color scheme within Tableau workbooks. The `.tps` file that Tableau utilizes is called `Preferences.tps` and is located in the `My Tableau Repository`. Since it's in XML format, it can be altered via a text editor. Matt Francis has posted a helpful blog at http://wannabeadatarockstar.blogspot.in/ that clearly communicates how to adjust this file. Also reference Tableau help.

Tableau Server architecture

Now that we have reviewed the various Tableau file types, we can use that information to understand various ways to architect a Tableau Server environment. Since this is not a book dedicated to Tableau Server, this architecture discussion is presented at a high level. The intent is to help the Tableau Desktop author understand how to interact with Tableau Server so that workbooks best serve the end user:



The preceding diagram visually represents the contents of the four file types that are most relevant when considering how the Tableau Desktop author should interface with Tableau Server. To be clear, the previous section of this chapter provided descriptions of the various

file types. Each of these descriptions included the line **What it contains**. **What it contains** is visually represented in the preceding diagram.

Each of the four file types listed in the diagram includes metadata; two include data and two include schema.

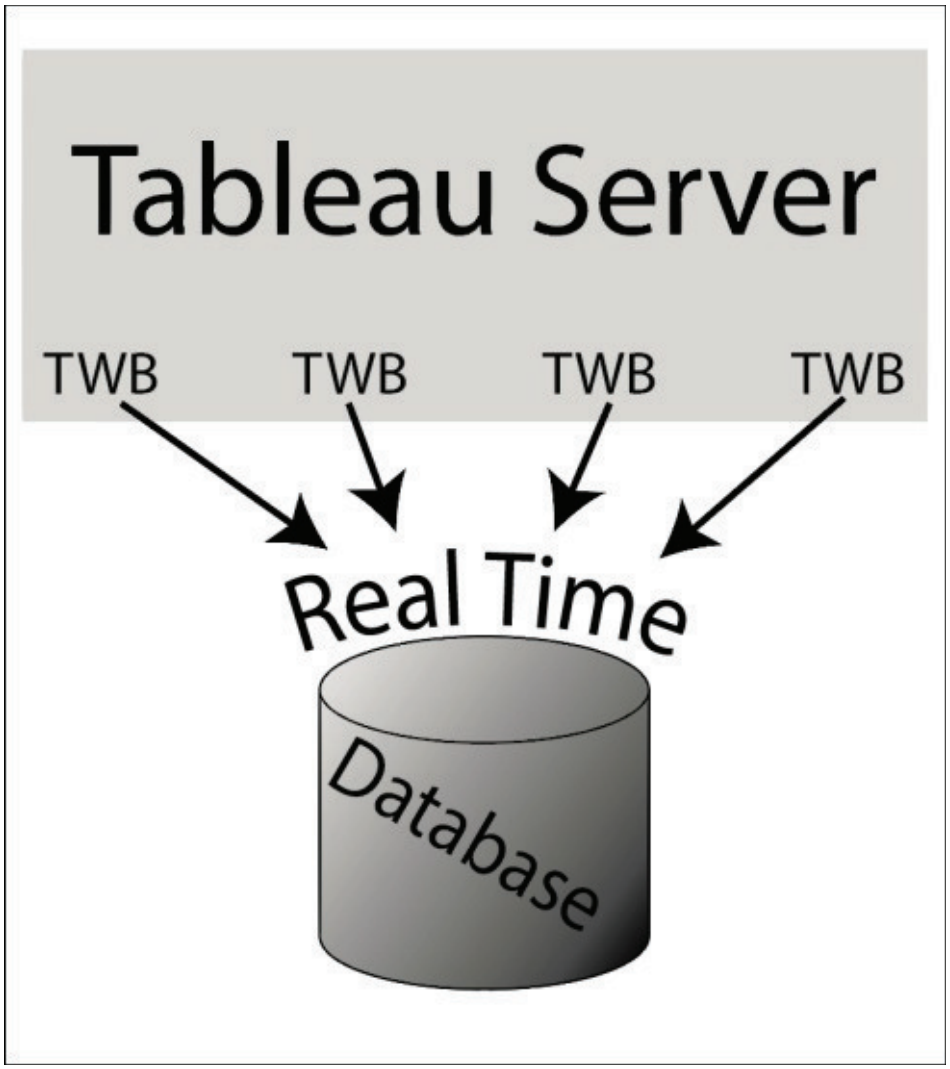
Let's consider four approaches to Tableau Server architecture and how each presents advantages and disadvantages the Tableau Desktop author should be aware of. The first two approaches presented should generally be avoided. The second two approaches should generally be adopted. Of course, the four approaches listed do not encompass every possible approach. They do, however, provide a basic framework.

Tableau Server architecture approaches to avoid

If the following Tableau Server architecture approaches should be avoided, why mention them at all? Because they are often utilized! An administrator who hasn't had the opportunity to go to Tableau Server training, may default to these approaches and may even build out a large infrastructure before realizing that it's difficult to maintain and scale.

Tableau Server architecture - TWB Centric

The following diagram shows `.twb` files that have been published to Tableau Server. The diagram also communicates that, since the `.twb` files do not include any data, each workbook must access an external data source in order to display a visualization. Furthermore, this access will cause Tableau to return data in real time; that is, the latest data available in the external data source:



The TWB Centric approach to Tableau Server architecture results in various advantages and disadvantages:

- TWB Centric advantages:
  - Small footprint
  - Easy revision history
  - Real-time

- TWB Centric disadvantages:
  - Difficult maintenance
  - Potentially poor performance

The *Small footprint* advantage stems from small file size; `.twb` files are rarely larger than a few MB. Small file size leads to fewer issues for revision history. This is in contrast with `.twbx` files which can become quite large and thus unexpectedly overload a hard drive when storing many copies via revision history.

The Real-time advantage is as a result of a `.twb` file always pointing to an external data source. As the data source updates, the workbook that is based on a `.twb` file updates. Of course, Real-time in this case should not be mistaken with stock ticker; that is, updated results are not displayed in a worksheet unless the end user performs a manual refresh. (Note that even when manually refreshed, a worksheet may be regenerated from cache as opposed to making a call to the data source. This will depend on the settings in Tableau Server.)

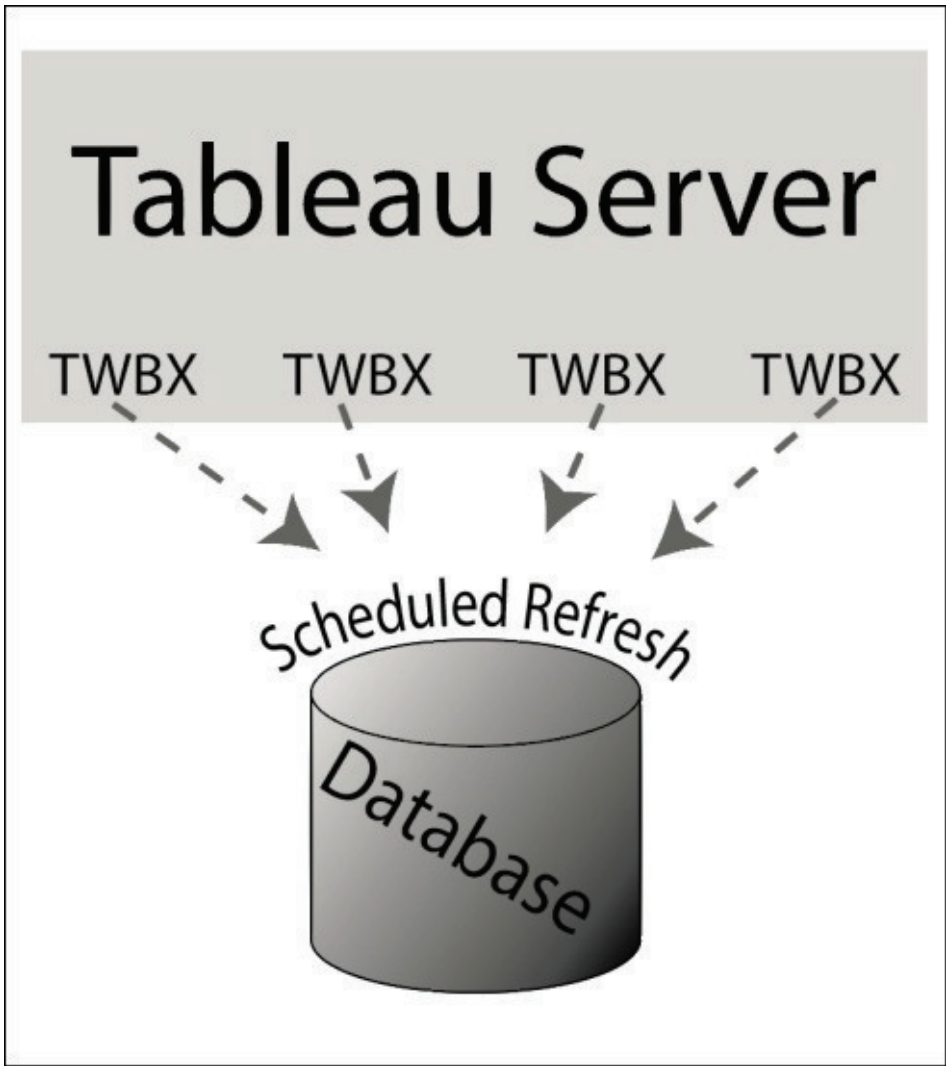
Note the phrase *Difficult maintenance* in the preceding list. From one perspective, maintenance is fairly easy; that is, a `.twb` file can be quickly downloaded, edited, and then reupload. From another perspective, the TWB Centric approach can be quite a chore. Consider a change to a single foreign key in a data source that breaks every workbook in production. Editing dozens of workbooks (or more) to rectify the issue would not be trivial.

Another disadvantage of TWB Centric architecture is *Potentially poor performance*. This is because it requires `.twb` files to point to external data sources. Network latency and slow database servers will negatively impact workbook performance. It should be noted, however, that some data source engines (such as the **Massively Parallel Processing** (**MPP**) systems discussed in Chapter 4, *All about Data – Data Densification, Cubes, and Big Data*) can potentially outperform the architecture options discussed in the following section.

Verdict: Avoid TWB centric architecture. A TDS centric architecture maintains all the advantages of a TWB centric architecture and mitigates the maintenance difficulties discussed previously.

Tableau Server architecture - TWBX Centric

The following image shows .twbx files that have been published to Tableau Server. Assuming that the `.twbx` files contain extracts for each required data source, no call is necessary to external data to display a visualization:



The TWBX Centric approach to Tableau Server architecture has a strong advantage and various disadvantages:

- TWBX Centric advantage:
  - Typically performant

- TWBX Centric disadvantages:
  - Large footprint
  - Very difficult maintenance
  - Potential problems with revision history
  - Not real-time

The TWBX Centric approach has at least one advantage: performance. Since a `.twbx` file can include data extracts, no calls to external data sources are required. This circumvents problems with network latency and slow database servers, thus enabling quick performance. Note that a `.twbx` file can be scheduled for refreshes, thus ensuring that the data is never stale.

Unfortunately, the TWBX Centric approach has major drawbacks. The *Large footprint*

disadvantage listed previously can occur as a result of large `.twbx` files. These files can be as large as several GB. Such large files lead to *Very difficult maintenance* and can lead to *Potential problems with revision history*. Large `.twbx` files can be difficult and time consuming to download, update, and reupload. Also, as mentioned previously, revision history on large `.twbx` files may unexpectedly overload a hard drive. Furthermore, the TWBX Centric solution is *Not real-time*.

For most Tableau Server implementations, the TWBX Centric solution should be avoided. The Tableau Server administrator who observes a `.twbx` file in excess of 500 MB should probably contact the author who uploaded the file in order to seek a better solution. This is not to say that `.twbx` files should never be used on Tableau Server. If a Tableau author uses a local spreadsheet as a data source, then a `.twbx` file will almost certainly be used in order for the workbook to function on Tableau Server. However, this will typically not lead to large `.twbx` files. Thus the disadvantages listed previously would not apply.

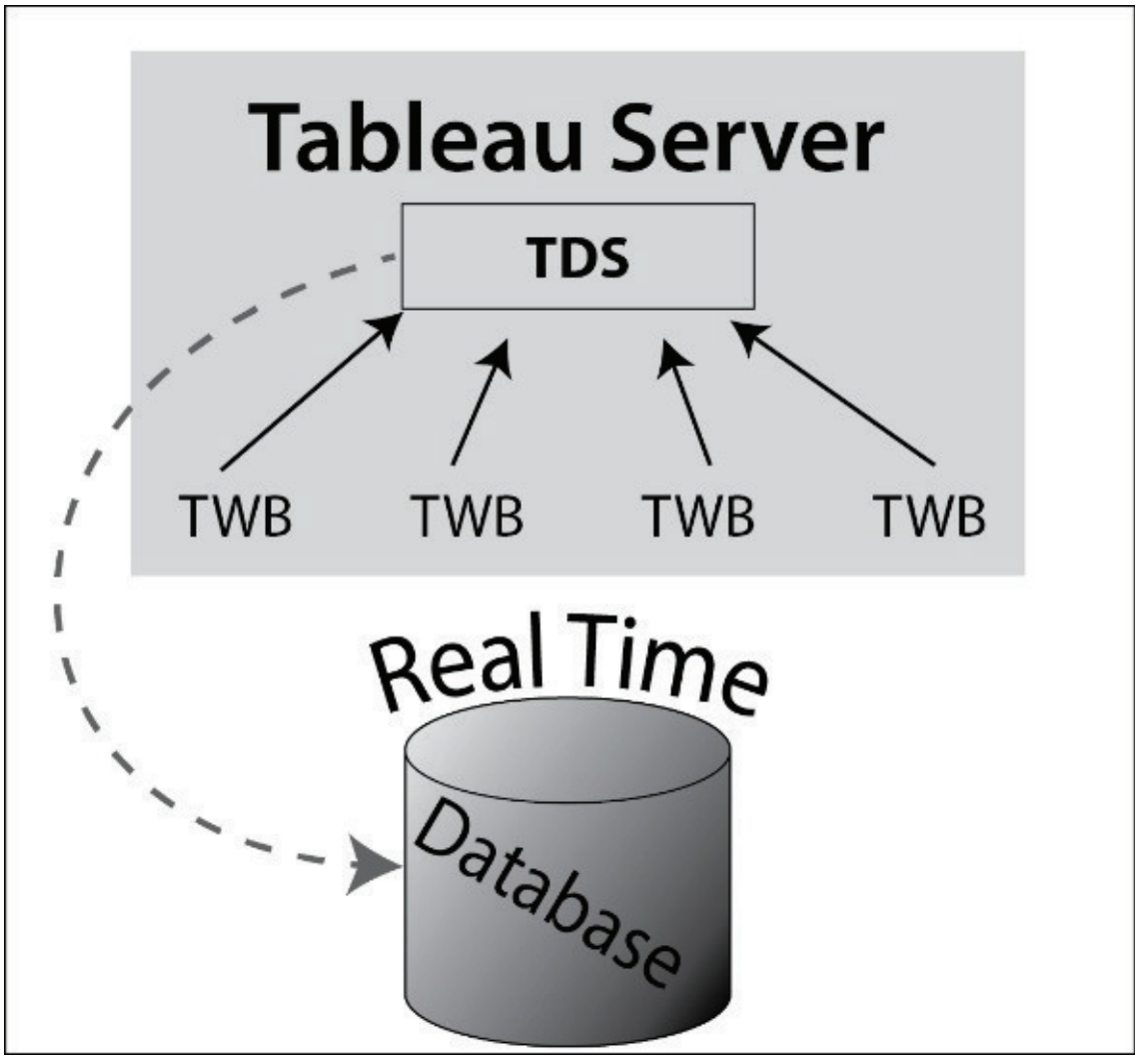
Verdict: Avoid TWBX centric architecture. A TDSX centric architecture maintains all the advantages of a TWBX centric architecture and mitigates most of the difficulties discussed previously.

Tableau Server architecture approaches to adopt

In the previous sections, we considered two approaches to Tableau Server architecture to avoid. Now let's consider two approaches to adopt.

Tableau Server architecture - TDS Centric

The following image shows .twbx files that have been published to Tableau Server. Assuming that the `.twbx` files contain extracts for each required data source, no call is necessary to external data to display a visualization:



The TDS Centric approach to Tableau Server architecture results in various advantages and at least one disadvantage:

- TWB Centric advantages
  - Small footprint
  - Easy revision history
  - Easy maintenance

    - Real-time

- TWB Centric disadvantage
    - Potentially poor performance

Like the TWB Centric approach discussed previously, the TDS Centric approach has the advantage of a *Small footprint*, which stems from the small size of both `.tds` and `.twb` files. These small file sizes result in fewer issues for revision history.

By using `.twb` files with their corresponding small footprints, maintenance is relatively easy since `.twb` files can be quickly downloaded, updated and then reuploaded. Furthermore, pointing to a `.tds` file has an additional maintenance advantage. If changes are made to the metadata in the `.tds` file (for example, a calculated field is updated), those changes will trickle down to every `.twb` file that points to the `.tds` file thus allowing for an update in a single location to impact multiple workbooks. Previously, we considered a scenario in which a change to a single foreign key broke every workbook in production. By utilizing the TDS Centric approach, updating the metadata in a `.tds` file to account for the change to the foreign key could instantly fix the problem for every `.twb` file pointing to the `.tds` file.

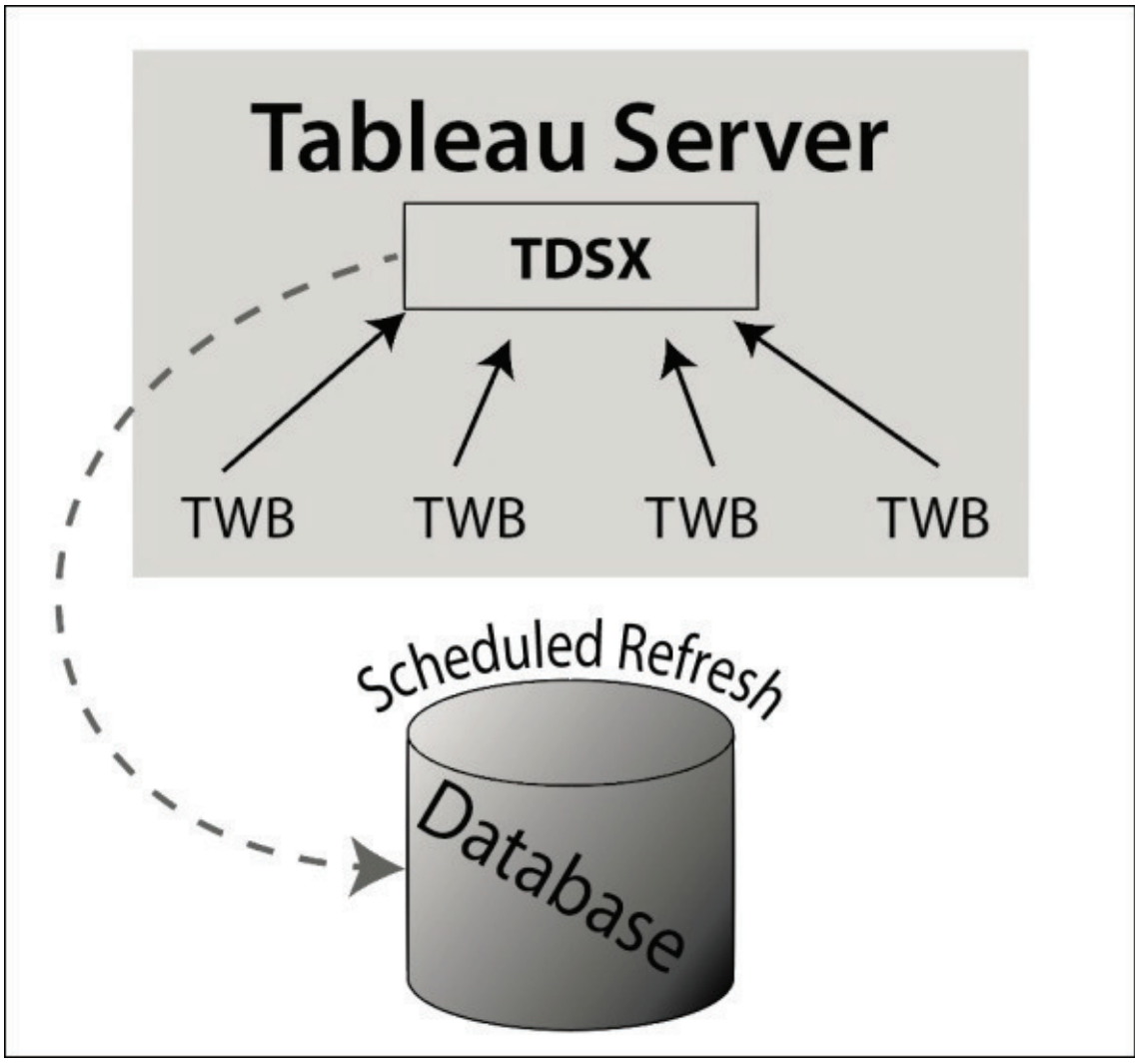
As in the TWB Centric approach discussed earlier, the TDS Centric approach provides a *Real-time* advantage.

Lastly, the TDS Centric architecture has a disadvantage, that is, *Potentially poor performance*. This is because a `.tds` file must point to external data sources that could, in turn, introduce network latency and slow database engines that negatively impact workbook performance.

Verdict: Consider adopting the TDS Centric approach, especially when a real-time solution is required. The TDS centric architecture maintains all the advantages of a TWB centric architecture while providing easier maintenance.

Tableau Server architecture - TDSX Centric

The following image shows .twbx files that have been published to Tableau Server. Assuming that the `.twbx` files contain extracts for each required data source, no call is necessary to external data to display a visualization:



The TDSX Centric approach to Tableau Server architecture results in various advantages and at least one disadvantage:

- TDSX Centric advantages:
  - Typically quick performance
  - Smallish footprint
  - Easy maintenance
  - Revision history friendly

- TDSX Centric disadvantage:
  - Not real-time

The TDSX Centric approach allows for *Typically quick performance*. Since the `.twb` files point to a `.tdsx` file that resides on Tableau Server, problems with network latency and slow database servers are circumvented.

By using `.twb` files with their corresponding small footprints, maintenance is relatively easy since `.twb` files can be quickly downloaded, updated, and then reuploaded. Furthermore,

pointing to a `.tdsx` file has an additional maintenance advantage. If changes are made to the metadata in the `.tdsx` file (for example, a calculated field is updated), those changes will trickle down to every `.twb` file that points to the `.tdsx` file, thus allowing an update in a single location to impact multiple workbooks. The term *Small footprint* is adopted in the list since `.tdsx` files can become quite large even though the `.twb` files remain small.

Verdict: In most environments a TDSX Centric architecture should be the approach most frequently used. Even the one disadvantage listed previously, *Not real-time*, can be mitigated via frequent updates. That said, a TDS Centric approach should be used when a real-time solution is required or when accessing a data source engine that outperforms Tableau extracts.

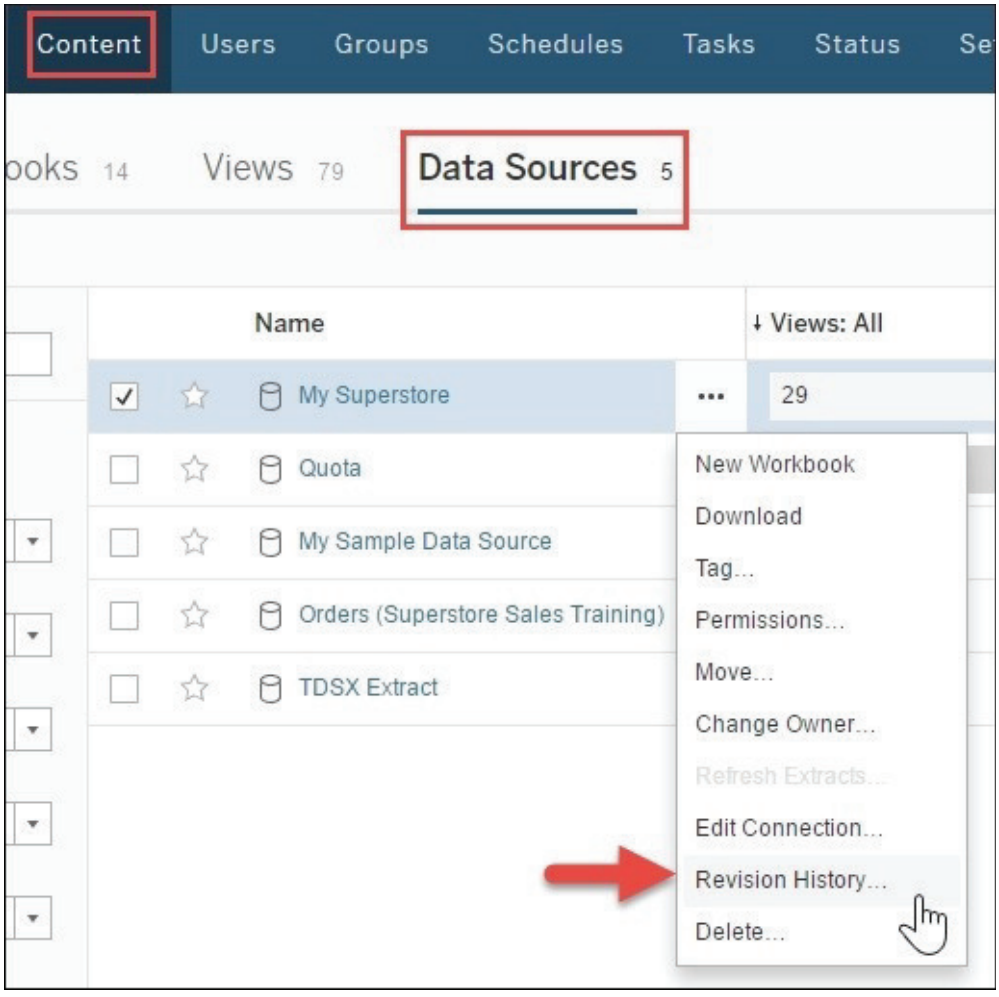
As previously stated, the preceding discussion of Tableau Server architecture is at a high level. Different environments require different approaches. Thus some combination of the previous two *to be adopted* approaches may often be appropriate. Also, there may be special cases that would utilize one of the *to be avoided* previous approaches. For example, it may be advantageous in some environments to programmatically generate a `.twbx` file for different end users thus allowing those end users to download `.twbx` files containing only their data.

Tableau Server revision history

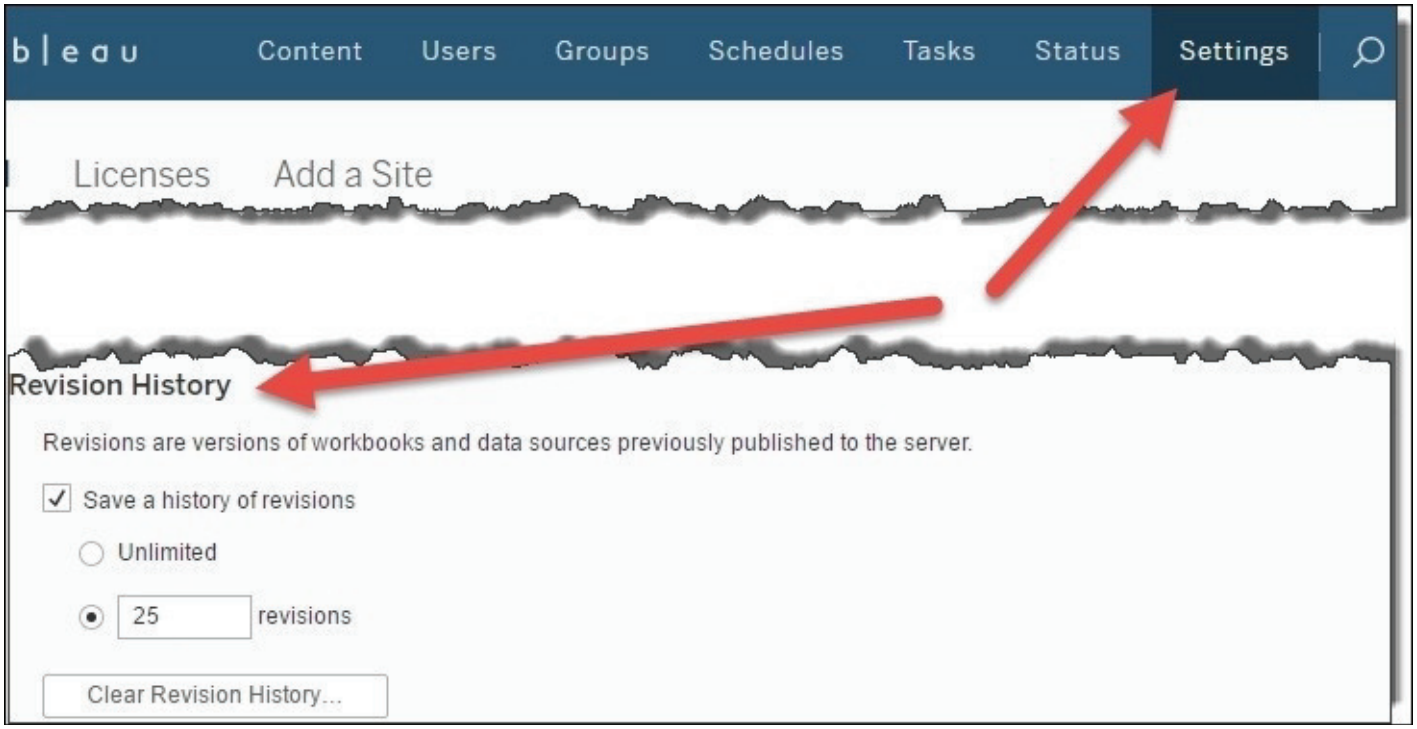
In Tableau 9.3, revision history was introduced for workbooks, including `.twb` and `.twbx` files. In Tableau 10.0 revision history was extended to data sources, including `.tds` and `.tdsx` files. By default Tableau Server's revision history is set to 25. In other words, the past 25 versions of each workbook and data source are retrievable. Previous to Tableau 9.3, revision history was only available through third party tools such as InterWorks' Tableau Enterprise Deployment Tool.

In Tableau Desktop, if you attempt to upload a workbook or data source with the same name as a previously uploaded file, Tableau Desktop will display a warning such as *Data source is already in use*. Publishing will overwrite the existing data source. If you proceed with the upload, revision control will be activated and the previous version of the file will remain accessible as a revision.

To access revision history for individual workbooks and data sources in Tableau Server, simply select **Content** | **Data Sources** or **Content** | **Workbooks**, and click on the dropdown menu for the desired data source/workbook. In the drop-down menu, select **Revision History** to open a dialog box in order to choose which previous version to restore:



The Tableau Server browser interface provides easy access to adjust revision history settings and also access revisions of individual workbooks and data sources. To adjust revision history settings simply click on **Settings** and scroll to **Revision History**:



If you'd like to peek beneath the hood of Tableau Server to begin to understand how revision history works, do the following: on a computer with Tableau Server installed, go to the directory for revision history. By default, this is located at `C:\ProgramData\Tableau\Tableau Server\data\tabsvc\dataengine\revision`. In the revision directory there are files with alphanumeric names. Simply rename one of those files with the appropriate extension (for example, `.twbx`) to open the file in Tableau Desktop.

Tableau Server web authoring environment

Web authoring is a Tableau Server feature introduced in 8.0 that provides an interface for authoring that is similar to Tableau Desktop. Originally, the web authoring interface was quite limited but more features are introduced with each version. Thus the capability gap between the Tableau Server web authoring environment and Tableau Desktop have shrunk. To mention a couple of examples of continuing improvements, in Tableau 10 the web authoring environment provides robust capabilities for creating and applying table calculations and also provides capabilities for creating dashboards, though the dashboard functionality is still fairly rudimentary. At the time of writing, some features are still missing altogether in the Tableau Server web authoring environment. For instance there is no interface for creating parameters or for connecting to data sources that have not already been published to Tableau Server. A fairly detailed comparison of the Tableau Desktop and web authoring environment capabilities is included later in this chapter.

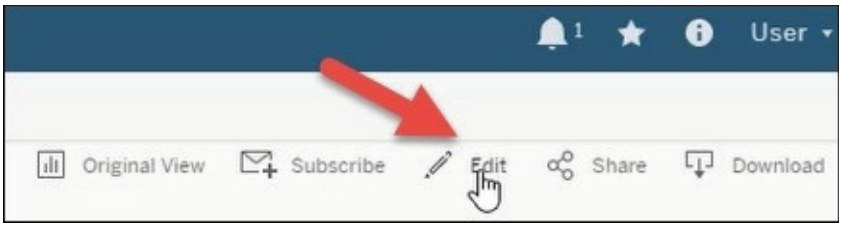
The question of what one can and cannot do in the Tableau Server web authoring environment is quite important. After all, why pay for additional copies of Tableau Desktop when the web authoring environment will suffice? The following exercises provide an opportunity to explore some of the limitations and workarounds of the web authoring environment.

Basic web authoring instructions

The two very brief exercises demonstrate how to access the authoring environment within Tableau Server.

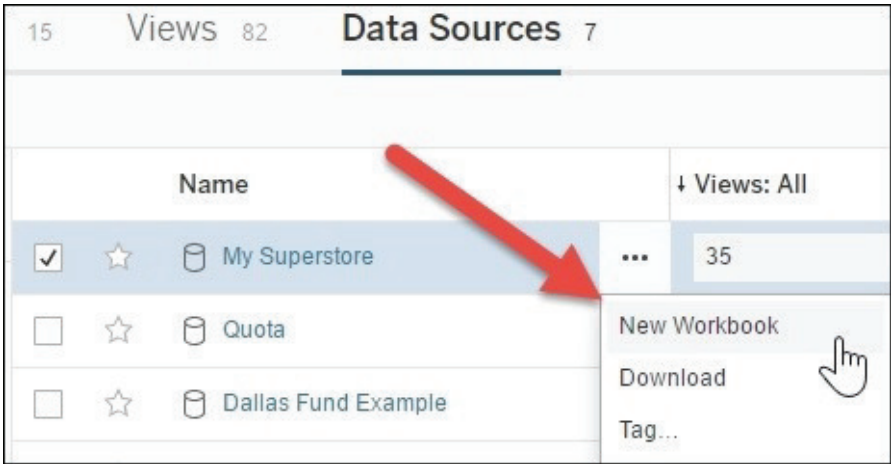
Exercise - edit an existing workbook on Tableau Server

1. Log in to an instance of Tableau Server.
2. Open an existing workbook within Tableau Server via **Content** | **Workbooks**.
3. Click on the **Edit** icon:



Exercise - create a new workbook on Tableau Server

1. Log in to an instance of Tableau Server.
2. Navigate to an existing data source via **Content** | **Data Sources**.
3. Click on the dropdown for the desired data source and choose to create a new workbook:



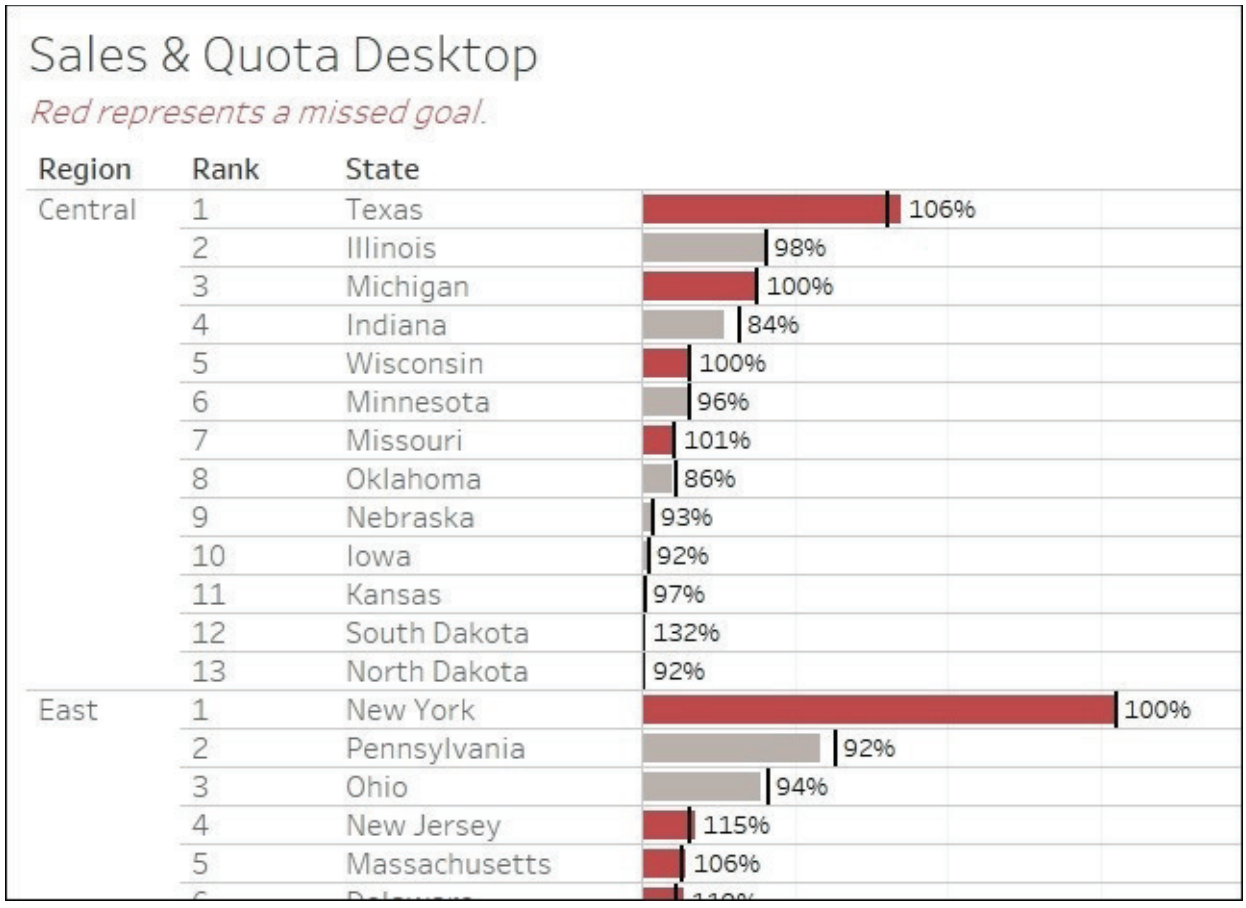
Exploring the capabilities and limitations of the Tableau Server web authoring environment

Although a comparison list is provided in the following section, a good understanding of the capabilities and limitations of the Tableau Server web authoring environment is most effectively explored by creating worksheets and dashboards in that environment. In that spirit, consider accessing the workbook associated with this chapter and referencing the worksheet entitled `Sales & Quota` in order to build out your own variation in Tableau Server. As you attempt to build the worksheet, you will discover the web authoring environment's exciting possibilities and sometimes frustrating limitations. Once you complete the build, compare your solution with the one included in the following section. This should help you understand which shortcomings can be overcome and which cannot.

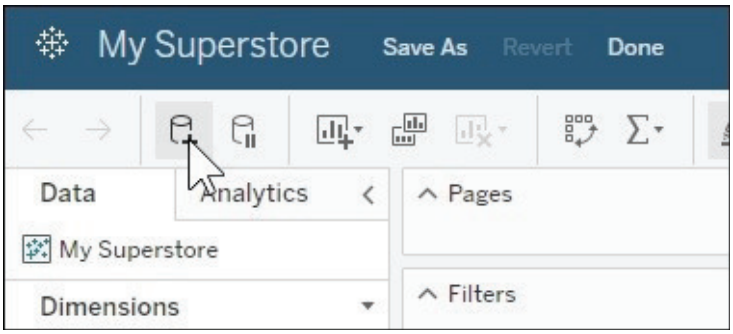
Exercise - the Tableau Server web authoring environment

If you do not have access to the Tableau Server web authoring environment, the following step-by-step instructions and the subsequent commentary should be of help. Carefully reading through the exercise should allow you to begin to understand what can and cannot be accomplished in the web authoring environment:

1. Access the workbook associated with this chapter and note the worksheet entitled `Sales & Quota`:



2. In the caption of the worksheet, you will note a Dropbox link with an Excel spreadsheet for **My Superstore** and **Quota**. Download and then publish both to an instance of Tableau Server. See the preceding, *Exercise – publish a data source to Tableau Server* for instructions.
3. Log in to Tableau Server and create a new workbook based on **My Superstore**. See the preceding, *Exercise – create a new workbook on Tableau Server* for instructions.
4. Click on the **Sheet 1** tab and rename the sheet `Sales & Quota`.
5. Add new data source via the **New Data Source** icon:

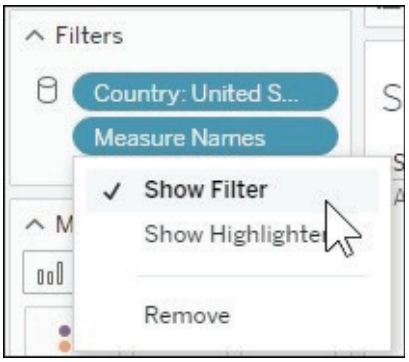


6. In the resulting window, choose the **Quota** data source.
7. Within the **My Superstore** dataset, create the following calculated fields. Calculated fields can be created in the web authoring interface by clicking on the dropdown next to the word **Dimensions** in the **Data** pane:

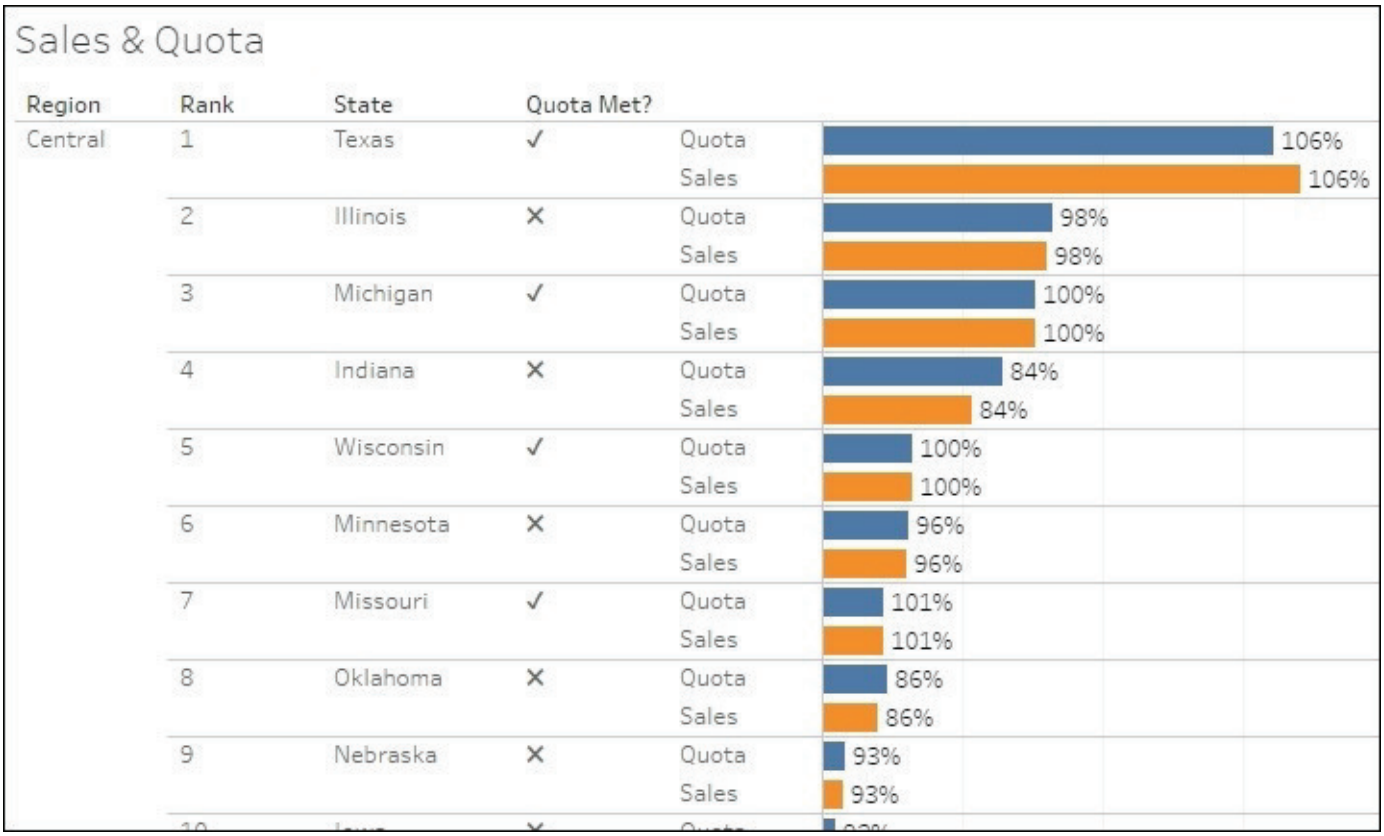
Quota	**% Achieved**	**Quota Met?**	**Rank**
`SUM([Quota].[Quota])`	`STR(ROUND((SUM([Sales]) / [Quota]*100) )) + '%'`	`IF SUM([Sales]) >= [Quota] THEN '✓' ELSE '✗' END`	`RANK(SUM(Sales))`

8. From the **Data** pane:
    1. Place **Measure Values** on the **Columns** shelf.
    2. Place **Region** on the **Rows** shelf.
    3. Place **Rank** on the **Detail** shelf.
    4. Click on the drop-down menu on **Rank** and select **Discrete**.
    5. Move **Rank** from the **Detail** shelf to the **Rows** shelf after **Region**.
    6. Place **State** and **Quota Met** on the **Rows** shelf after **Rank**.
    7. Place **% Achieved** on the **Label** shelf.
    8. Place **Measure Names** as the last dimension on the **Rows** shelf and also on the **Color** shelf.

9. On the **Filters** shelf, click the drop-down menu associated with **Measure Names** and select **Show Filter**:



10. In the **Measure Names** filter, check only **Quota** and **Sales**.
11. Click on the drop-down menu for **Rank** on the **Rows** shelf and select **Edit Table Calculation**.
12. In the resulting dialog box select **Specific Dimensions** and check only **State**.
13. Click on the **Color** shelf to adjust colors as desired:

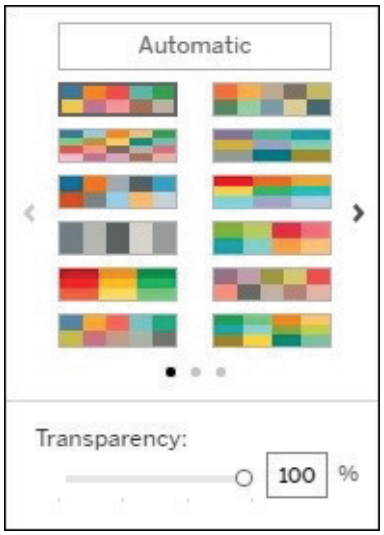


This worksheet looks rather different from the Tableau Desktop version. Let's consider some of the differences and similarities between the two environments that we discovered as a result of working through the exercise:

- Whereas Tableau Desktop provides connectivity to many data sources, the web authoring environment only provides access to data sources that have already been published to Tableau Server.
- The web authoring environment provides very limited formatting capabilities:
  - Consider the **% Achieved** calculated field. The STR function was included because the interface does not allow the number format to be changed.
  - Also note that label formatting is limited to show/hide.
- The table calculation functionality in the web authoring environment is quite robust.
- Data blending is functional.
- The shortcut for deploying **Measure Names** and **Measure Values** on a view utilizing an axis is not available:
  - To clarify, the Tableau Desktop shortcut referenced previously is as follows: an author can place a measure on the **Columns** or **Rows** shelf and then place a second measure on the resulting axis in the view to deploy **Measure Names** and **Measure Values** and thus view multiple measures on the same view type.
  - Note that in a new worksheet in the web authoring environment, placing a measure on the **Text** shelf and then double-clicking on a second measure will deploy **Measure Names** and **Measure Values**.
- Building a dual axis chart is not possible:
  - In the Tableau Desktop instance of the `Sales & Quota` worksheet, a dual axis is utilized.

- Reference lines are very limited:
  - Reference lines in Tableau Desktop are very flexible. Reference lines in the web authoring environment do not allow the modified bullet chart that was built in the Tableau Desktop worksheet example. Thus the previous exercise took a compromise approach.

- **Color** shelf functionality is different and more limited:
  - As you completed the last step in part II of the exercise, you probably noticed that the **Edit Color** dialog box is very different and somewhat more limited than in Tableau Desktop. There is no functionality for choosing specific colors for each dimension member, adding borders and halos, or defining individual RGB colors:



- Titles and captions can be hidden or displayed but not edited:
  - In Tableau Desktop, the title has been edited to display **Red represents a missed goal**. Also, the caption displays a dropbox link. This is not possible in the web authoring environment.

Comparing and contrasting Tableau Desktop with the Tableau Server web authoring environment

Sometimes it's unclear which individuals in an organization require a Tableau Desktop license and which only need a Tableau Server user license. For example, a technically-oriented executive with limited time to learn new software could go either way. On the one hand, she may want to delve into the details of the data but, on the other hand, she may not be able to invest the time necessary to learn enough about Tableau Desktop to warrant the price of an additional license. When the web authoring environment was released in Tableau Server 8.0, the features were fairly rudimentary. The busy executive may have been best served with a Tableau Desktop license. Since that time more and more capabilities have been introduced. Today, that same executive may be reasonably content with using the web authoring environment.

The following comparison and contrast list should help provide a quick way to decide who should receive which license type. Note that the list is primarily intended to communicate the major capabilities and limitations of Tableau Server's web authoring environment when compared to Tableau Desktop. Basic functionality capabilities such as the ability to place a field on a shelf are assumed and thus are not included:

Tableau Desktop feature	**Available in Tableau Server web authoring environment?**
Analytics Pane • Constant line • Average line • Median with quartiles • Box plot • Totals • Average/median with 95% CL • Trend line	✓ With limited or no customizability
• Forecast • Cluster • Custom Reference Line, Reference Band, Distribution Band, Box Plot	✗
Data Pane	

• Create extract • Replace Data Source • Connect/configure data sources outside of Tableau Server	✗
• Create folders • Edit Aliases • Find a Field • See Field Description	✗
• Duplicate, Show/Hide, Rename Fields • Convert to Measure/Dimension • Convert to Continuous/Discrete • Change Data Type • Change Geographic Role	✓
• Create Group • Create Set	✓ Groups and sets cannot be created via the Data Pane but can be created by selecting various marks on the view and interfacing with the tooltip command buttons.
• Create Bin	✓ Only by creating a histogram via 'Show Me'
• Create Hierarchy	✗
• Change default properties such as Comment, Color, Shape,	✗

Sort, Aggregation and Number format.	
• Create calculated fields including table calculations and LOD calculations	✓ With robust functionality
• Create Parameter	✗
• Group by Folder/Data Source Table	✓
View • Adjust Measure Aggregation • Create Basic Visualization Types such as Bar, Line, Area, Crosstab, Map, Treemap, Pie. • Create New Worksheet • Enable/Disable Sheet and Dashboard Highlighting • Fit Options (Standard, Fit Width, Fit Height, Entire View) • Show Mark Labels • Show/Hide Caption/Title • Utilize Measure Names/Measure Values	✓
• Create Intermediate to Advanced Chart Types such as Dual Axis, Bullet, Bar-In-Bar, Sankey, Control. • Edit Axis • Hide Legends	✗

• Map Layers and Format Windows • Map Options • Turn on/off Stack Marks	
• Adjust Color, Shape, and Size via Marks View Card • Sorting	✓ With Limited Functionality
• Formatting	✗ With Very Limited Functionality
Filters • Deploy Measure and Dimension Filters • Range - **At least** and **At most** • Only Relevant Values • All Values in Database • Display/Hide Filter • Include/Exclude • Choose filter type (Single Value, Multiple Values, List, Dropdown, Slider, and so on) • Search a filter for specific values • Apply to **All Using Related Data Sources** • Apply to **All Using This Data Source** • **Apply to Only This Worksheet**	✓
• Extract filter • Context filter	

• Special (Null, non-null values) • Row level filter • Data source filter • Wildcard filter • Conditional filter • Top/Bottom filter • Customize (Show **All** Value, Show Search Button, Show Apply Button, and so on.)	✗
Dashboard • Create New Dashboard • Add Horizontal, Vertical and Blank Objects • Add Objects as Tiled or Floating • Adjust Position of Floating Objects • Adjust Dashboard Size • Use a Worksheet as a Filter • Deploy Highlighters • Hide/Show Legend • Hide/Show Filter • Download crosstab, data, image, pdf, workbook • Save/Save As... • Fit Options (Standard, Fit Width, Fit Height, Entire View)	✓
• Add Image, Web Page and Text Objects • Create Custom Actions • Device Preview • Adjust Size of Floating Objects	✗
Story	✗

	No creation or editing capabilities

User filters

Students often ask me questions such as the following:

> *"I have sales managers over various territories. It's important that the sales managers see only their metrics, that is, not the metrics of the other sales managers. In order to accomplish this, do I have to create separate workbooks for each sales manager?"*

Fortunately, the answer is no. Tableau provides user filters which allow the Tableau author to make sure that each of those sales managers sees only the information for which they have clearance.

Exercise - deploying a view level user filter

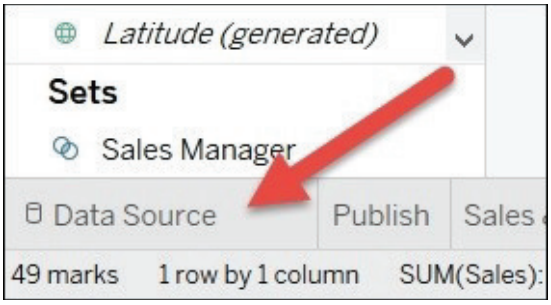
This exercise assumes a new, default install of Tableau Server. By default Tableau installs with a JaneDoe and a JohnSmith user. Both are used in this exercise:

1. Access the workbook associated with this chapter and navigate to the worksheet entitled `View Level User Filter`.
2. Note that the view is a field map of the USA with **State** on the **Detail** shelf, **Region** on the **Color** shelf, and **Sales** on the **Label** shelf.
3. Log in to an instance of Tableau Server via **Server** | **Sign In**.
4. Select **Server** | **Create User Filter** | **Region**.
5. Within the resulting **User Filter** dialog box, select **JaneDoe** and check the **Central** region. Also select **JohnSmith** and check the **East** region.
6. Name the user filter `Sales Manager` and click OK.

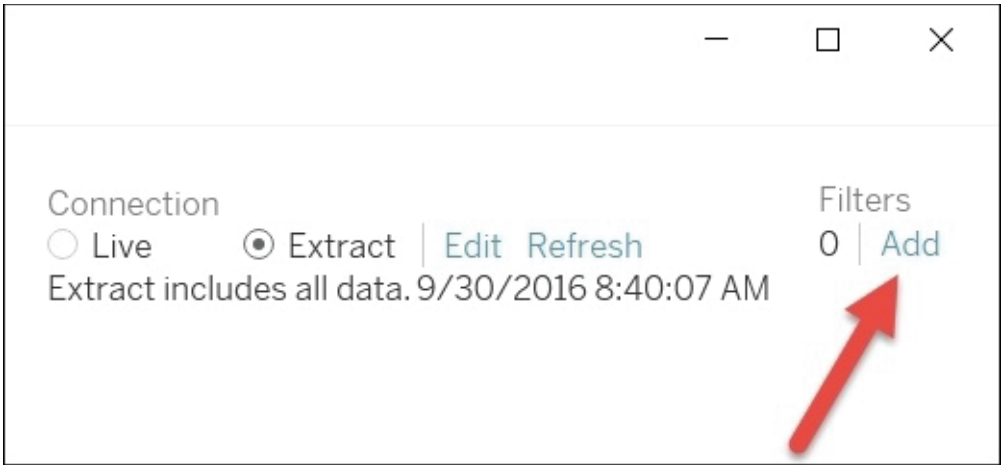
   ### Note

   Note that **Sales Manager** is actually added to the **Sets** portion of the **Data** pane.

7. Click on the **Data Source** tab to access the **Data Source** page:

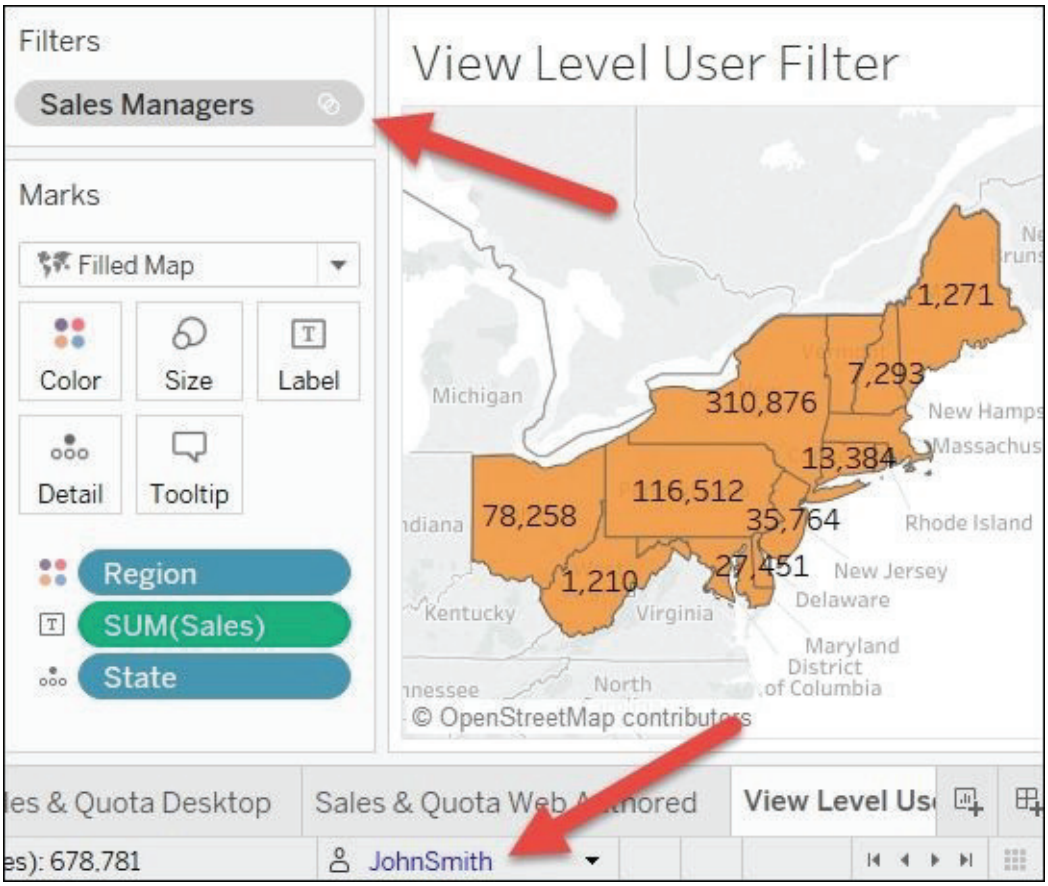


8. In the upper right-hand corner of the **Data Source** page, click **Add** to add a data source filter.



9. In the resulting dialog box, click the **Add** button to add a filter on **Sales Manager**.
10. After completing the data source filter, return to the worksheet and in the right-hand portion of the status bar click on the dropdown to access **Filter as User**:

11. Choose **JohnSmith** and note that the results display only the **East** region:



Once the workbook is published to Tableau Server, users who access the workbook will only see the information the filter allows.

You may have noticed in the preceding exercise that a data source filter was added as opposed to simply adding a dimension filter to the **Filters** shelf. This is important because any user with web authoring permission can simply remove a filter from the **Filters** shelf. In the case of this exercise, the user would then be able to see metrics for every **Region**. Data source filters, however, cannot be removed via the web authoring environment and are thus more secure. Furthermore, a data source filter is workbook wide which further secures the data.

Also note that user and group filters can be published as part of the data source. In other words, the `.tdsx` and `.tds` files discussed previously can include data source filters based on users and groups. This allows for centrally maintained security that is transparent even to those users with editing privileges.

Accessing the Tableau Server Performance Recording dashboard

In Chapter 11, *Improving Performance*, the Performance Recording dashboard was discussed in detail. Sometimes a workbook may perform satisfactorily on Tableau Desktop but, mysteriously, may perform poorly when published to Tableau Server. In such cases accessing the Performance Recording dashboard on Tableau Server can be very helpful. The following exercise provides step-by-step instructions for doing so.

Exercise - exploring performance recording on Tableau Server

1. Navigate to an instance of Tableau Server.
2. On the toolbar, click on **Settings**:

3. On the resulting page, locate the section entitled `Workbook Performance Metrics` and select **Record workbook performance metrics**.
4. Click **Save**.
5. Navigate to a workbook of your choosing and open a view. Note that the ending portion of the URL is `:iid=<n>`.
6. Type `:record_performance=yes&` immediately before `:iid=<n>`, for example, `http://localhost:8000/#/views/Finance/Taleof100Start-ups?:record_performance=yes&:iid=5`. Note that the toolbar now includes a **Performance** link:

7. Click the **Refresh** icon on the toolbar. It is located to the left of **Pause**. Click on the **Performance** link and observe the resulting Performance Recording dashboard that is displayed.

Summary

We began this chapter by considering the various Tableau file types; in particular, the `.tds`, `.tdsx`, `.twb`, and `.twbx` file types. This provided a foundation for understanding different ways to architect a Tableau Server deployment. We considered four basic architecture scenarios and the advantages and disadvantages of each.

Next, we considered Tableau Server revision history, where we learned that version control features beginning in Tableau 9.3 can provide a safety net against inadvertently overwriting files. This was followed by a section on the Tableau Server web authoring environment that compared and contrasted that environment with Tableau Desktop. The primary purpose of this section was to establish a knowledge base for determining which personnel should have Tableau Desktop licenses and those for whom the web authoring capabilities of Tableau Server should suffice.

Lastly, we considered user filters and the Performance Recording dashboard. User filters enable the Tableau author to ensure that users are only able to access data for which they have clearance. Although the Performance Recording dashboard was covered in the previous chapter, in this chapter we learned how to access it via Tableau Server.