EXPERT INSIGHT

Colored Version

Learning Tableau 2022

Create effective data visualizations, build interactive visual analytics, and improve your data storytelling capabilities

Forewords by:

Blair Hutchinson,
Senior Product Manager,
Tableau, a Salesforce Company

Mark Tossell,
Lead Solution Engineer,
Tableau, a Salesforce Company

Roberto Andreoli,
Senior Director Solution Engineering
Tableau, South EMEA, Salesforce

Fifth Edition

<packt>

Joshua N. Milligan

Preface

When Tableau was first introduced, it was a dramatic paradigm shift away from tabular reports and extensive cycles of data integration that produced results long after relevant decisions could have been made. Tableau disrupted the paradigm for visually interacting with data. And it continues to offer a unique paradigm — one that makes it easy and intuitive (and fun!) to be hands-on with the data, to receive instant visual feedback with every action, and to ask questions and uncover insights in a natural flow of thought and interaction. The result is an ever growing **#datafam**, a community that loves Tableau for its simplicity, beauty, and ability to make working with data fun!

Tableau continues to expand and evolve in ways that make seeing and understanding data easier and more powerful. The Tableau Data Model, new dashboard actions, ever-growing geospatial support, animations, and new dashboard objects expand what's possible and make it easier than ever to gain and share insights from data. New features such as dashboard extensions allow you to go beyond the built-in functionality and integrate Tableau fully into analytic workflows. Einstein Discovery brings the power of AI and powerful predictive models to your analysis.

The continued evolution of Tableau Prep brings the same intuitive instant feedback to data prep and cleansing that Tableau Desktop brought to data visualization, greatly extending the analytical platform of Tableau. We'll cover these new features (and more) in the chapters of this book!

We'll look at Tableau through the lens of understanding the underlying paradigm of how and why Tableau works, in the context of practical examples. Then we'll build on this solid foundation of understanding so that you will have the tools and skills to tackle even the toughest data challenges!

Who this audiobook is for

This book is for anyone who needs to see and understand their data! From the business user to the hardcore data analyst to the CEO, everyone needs to have the ability to ask and answer questions of data. Having a bit of background with data will definitely help, but you don't need to be confident with scripting, SQL, or database structures.

Whether you're new to Tableau, or have been using it for months or even years, with this book you'll gain a solid foundation of understanding Tableau, and the tools and skills to build toward advanced mastery of the tool.

What this audiobook covers

Chapter 1, *Taking Off with Tableau*, introduces the foundational principles of Tableau. We'll walk through multiple examples in a logical progression that will introduce everything from the interface to connecting to data, building your first visualization, and even building a dashboard. This chapter will give you a solid foundation for terminology and concepts that will be used throughout the book.

Chapter 2, *Connecting to Data in Tableau*, covers the foundational concepts involved in connecting to data, using several practical examples. It covers the various types of connections, file types, cloud-based and on-premises databases, and how to work with metadata.

Chapter 3, *Moving Beyond Basic Visualizations*, builds on the basic visualization principles covered in the first chapter to explore variations and extended possibilities. You will learn when and how to use a wide variety of visualizations to analyze and communicate data.

Chapter 4, *Starting an Adventure with Calculations and Parameters*, introduces calculations and parameters, giving an overview of the major types of calculation and then detailed examples of row-level and aggregate calculations. It combines conceptual knowledge with practical examples, and concludes with performance considerations.

Chapter 5, *Leveraging Level of Detail Calculations*, takes an in-depth look at the level of detail expressions and how to use them to solve complex data challenges. It not only gives an overview, but dives into examples of FIXED, INCLUDE, and EXCLUDE variations.

Chapter 6, *Diving Deep with Table Calculations*, gives you a strong foundation for understanding and using table calculations to solve a wide range of data challenges. It covers the concepts of scope and direction, addressing and partitioning, and walks through several in-depth practical examples.

Chapter 7, *Making Visualizations that Look Great and Work Well*, covers how to extend and alter the default formatting applied to visualizations by Tableau, to customize options such as font, color, lines, shading, annotations, and tooltips to effectively communicate a data story.

Chapter 8, *Telling a Data Story with Dashboards*, builds on concepts that were introduced in the first chapter and expanded on throughout. It walks through several practical examples of various kinds of dashboards to help you gain a solid understanding of what a dashboard is, how to build one and make it interactive, and how to use it to tell a compelling data story.

Chapter 9, *Visual Analytics: Trends, Clustering, Distributions, and Forecasting*, introduces the visual statistical analytics capabilities built into Tableau and supplies you with practical examples of how and when to leverage these capabilities. This includes adding and modifying trend models, leveraging clustering capabilities, using and modifying forecast models, and visualizing the distribution of data. This chapter also includes new details regarding the new features and interface present for Explain Data. You will not only understand how to employ statistical models but also evaluate their accuracy.

Chapter 10, *Advanced Visualizations*, builds upon the visualizations and techniques already covered, demonstrating how Tableau can be used to create any kind of visualization. A multitude of examples demonstrates a wide variety of advanced visualizations, from bump charts to Marimekko charts to animated visualizations.

Chapter 11, *Dynamic Dashboards*, builds your dashboard skills by demonstrating various techniques to show, hide, and swap content on a dashboard. The result is a truly dynamic user experience that enhances your ability to communicate data.

Chapter 12, *Exploring Mapping and Advanced Geospatial Features*, demonstrates everything about maps and geospatial visualization, from map basics to geospatial functions, custom territories, and plotting data on custom background images.

Chapter 13, *Integrating Advanced Features: Extensions, Scripts, and AI*, explores the incredible potential of extending Tableau's functionality using new features and integrations in Tableau. The chapter explores dashboard extensions that give you the ability to seamlessly integrate code, backend systems, and new user interfaces directly into your dashboards. You'll also see examples of leveraging scripts that could be used for anything from data cleansing to advanced predictive models. Finally, this chapter will explore the new integration of Einstein Discovery, which unleashes the full power of AI and predictive models and can be fully integrated into interactive dashboards.

Chapter 14, *Understanding the Tableau Data Model, Joins, and Blends*, explores the major ways of relating data in Tableau, including joins, blends, and relationships, using Tableau's Data Model. With practical examples and detailed descriptions, you will understand the difference between logical and physical layers and how to leverage relationships, joins, and blends to achieve great analytical results.

Chapter 15, *Structuring Messy Data to Work Well in Tableau*, acknowledges that real-world data is sometimes a mess, and gives you a foundation for understanding well-structured data and a toolset for dealing with data that isn't structured well in Tableau.

Chapter 16, *Taming Data with Tableau Prep*, explores the Tableau Prep Builder tool, including the overall paradigm and specific features. You will work through an extended practical example to understand how to leverage Tableau Prep's amazing ability to clean and structure data.

Chapter 17, *Sharing Your Data story*, concludes the book with a look at a wide range of options for sharing your story. From printing to sharing interactive dashboards to output PDFs and images, you'll be ready to share the stories contained in your data with those who need it most.

To get the most out of this audiobook

This book does not assume specific database knowledge, but it will definitely help to have some basic familiarity with data itself. We'll cover the foundational principles first, and while it may be tempting to skip the first chapter, please don't! We'll lay a foundation of terminology and explore the paradigm that will be used throughout the remainder of the book.

You'll be able to follow along with many of the examples in the book using Tableau Desktop and Tableau Prep Builder (in *Chapter 16*, *Taming Data with Tableau Prep*).

Most examples can be completed with almost any recent version of Tableau. New features and interfaces will require version 2022.1 or later.

You may download and install the most recent versions from Tableau using these links:

Tableau Desktop: https://www.tableau.com/products/desktop/download

Tableau Prep Builder: https://www.tableau.com/products/prep/download

Please speak to a Tableau representative for specific licensing information. In most cases, you may install a 14-day trial of each product if you do not currently have a license.

Depending on the terms of your license, Tableau also typically allows you to use your license on two machines. This means you might have Tableau installed at the office (perhaps even an older version) but can also install the latest version on your home machine. Check your licensing agreement and speak to a Tableau representative to verify details in your case.

Download the example code files

The code bundle for the book is hosted on GitHub at https://github.com/PacktPublishing/Learning-Tableau-2022. We also have other code bundles from our rich catalog of books and videos available at https://github.com/PacktPublishing/. Check them out!

Download the color images

We also provide a PDF file that has color images of the screenshots/diagrams used in this book. You can download it here: https://static.packt-cdn.com/downloads/9781801072328_ColorImages.pdf.

Conventions used

There are a number of text conventions used throughout this book.

CodeInText: Indicates code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles. For example: "Connect to `Hospital Visits.xlsx` and generate an extract."

A block of code is set as follows:

```
IF LEFT([Room], 1) = "1"
THEN "First Floor"
ELSEIF LEFT([Room], 1) = "2"
```

```
THEN "Second Floor"
END
```

Bold: Indicates a new term, an important word, or words that you see on the screen. For instance, words in menus or dialog boxes appear in the text like this. For example: "Select **Table Layout** | **Advanced** from the **Analysis** menu."

Chapter 1

Figures

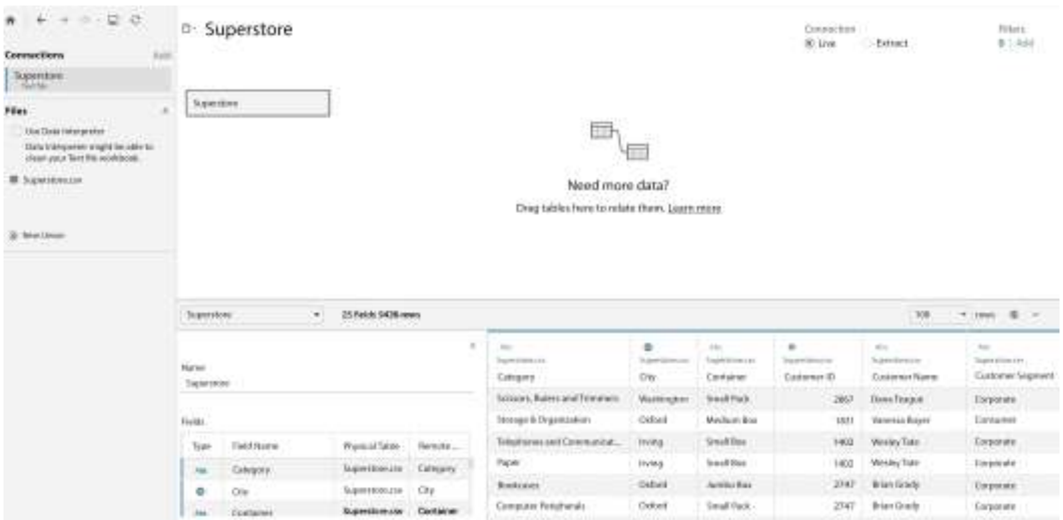


Figure 1.1: The data connection screen allows you to build a connection to your data

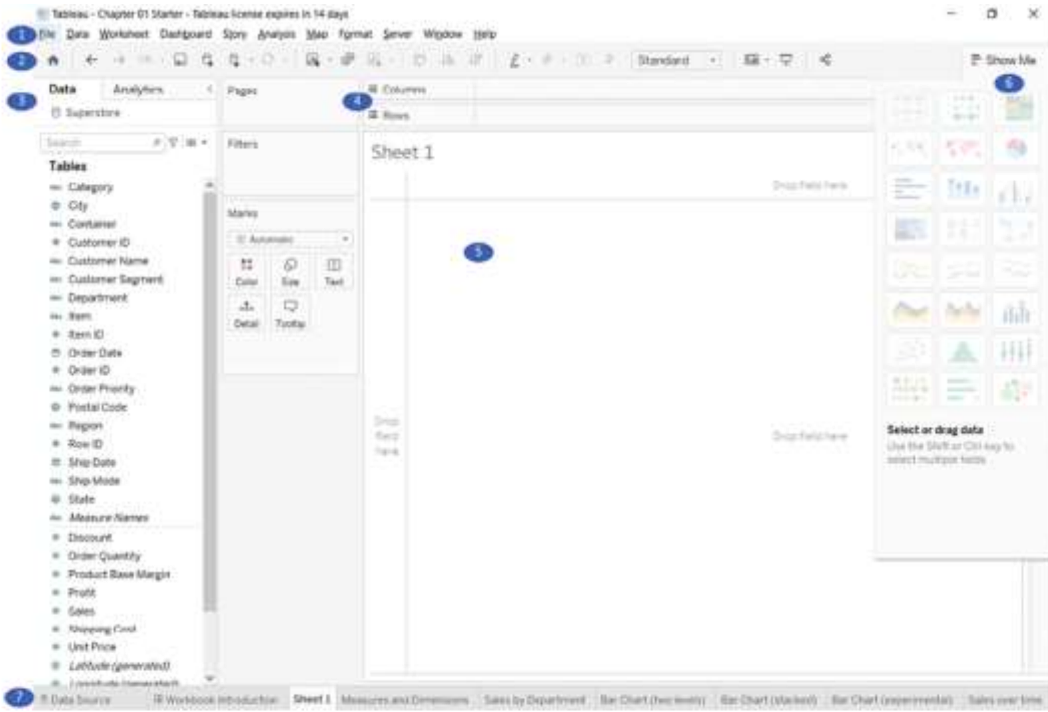


Figure 1.2: Elements of Tableau’s primary interface, numbered with descriptions below

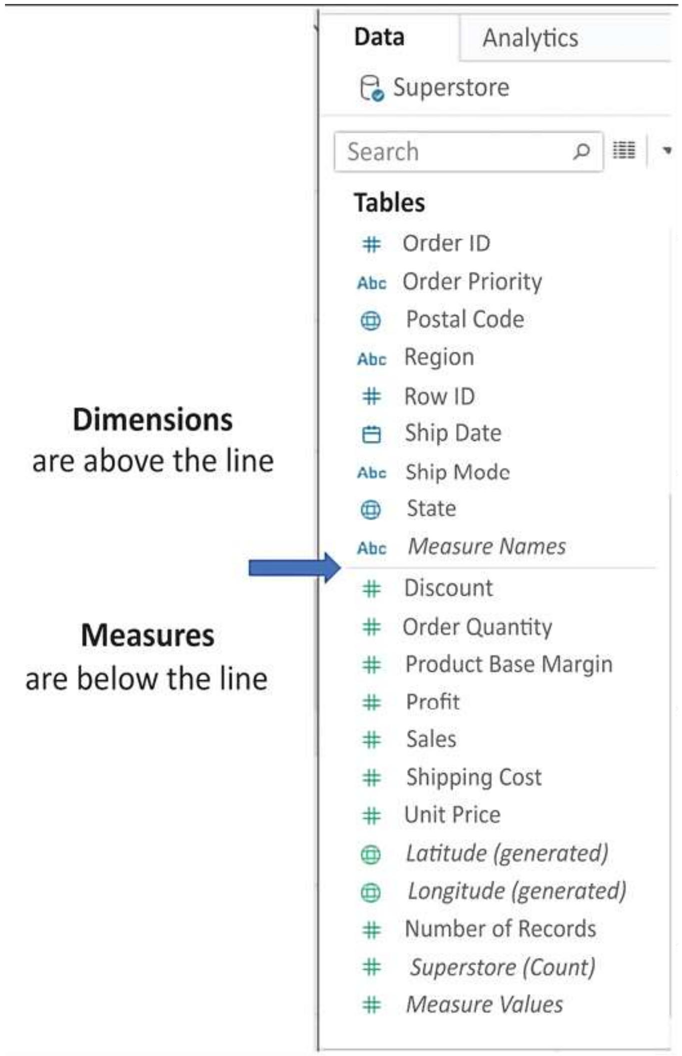


Figure 1.3: Each table (this data source only has one) has dimensions listed above the line and measures listed below the line

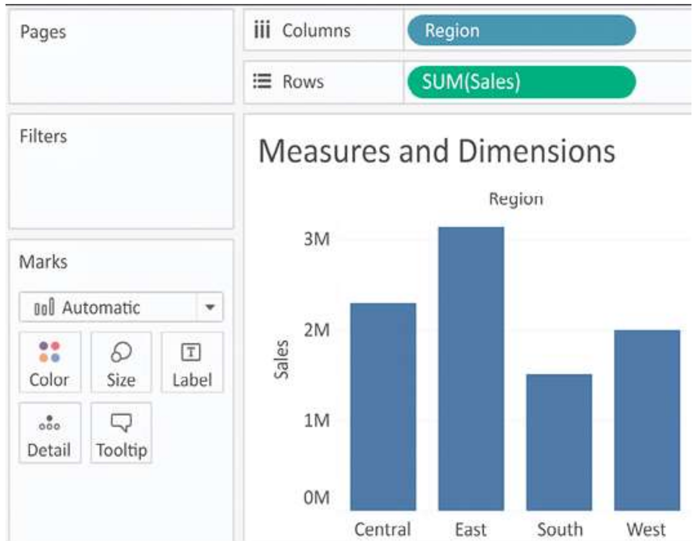


Figure 1.4: A bar chart demonstrating the use of Measures and Dimensions

Figure 1.5: The discrete field on Columns defines column headers

Figure 1.6: The discrete field on Rows defines row headers

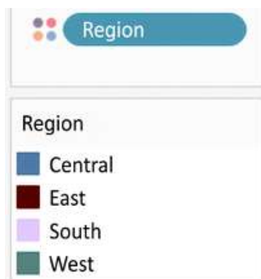


Figure 1.7: The discrete field on Color defines a discrete color palette

Figure 1.8: The continuous field on Columns (or Rows) defines an axis

Figure 1.9: The continuous field on Color defines a gradient color palette

Can be

A	Discrete	Continuous
Dimension	Yes	If **Numeric** or **Date**
Measure	Yes	Yes

Figure 1.10: Measures and dimensions can be discrete or continuous

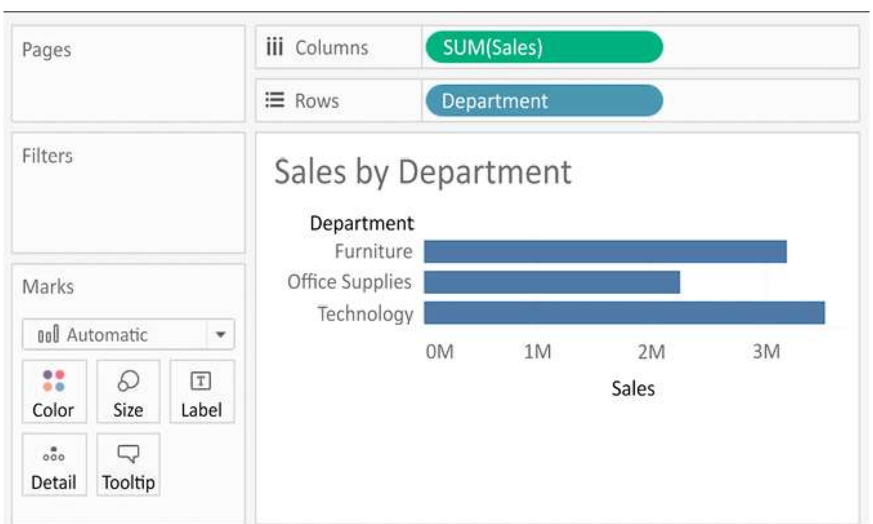


Figure 1.11: The view Sales by Department should look like this when you have completed the preceding steps

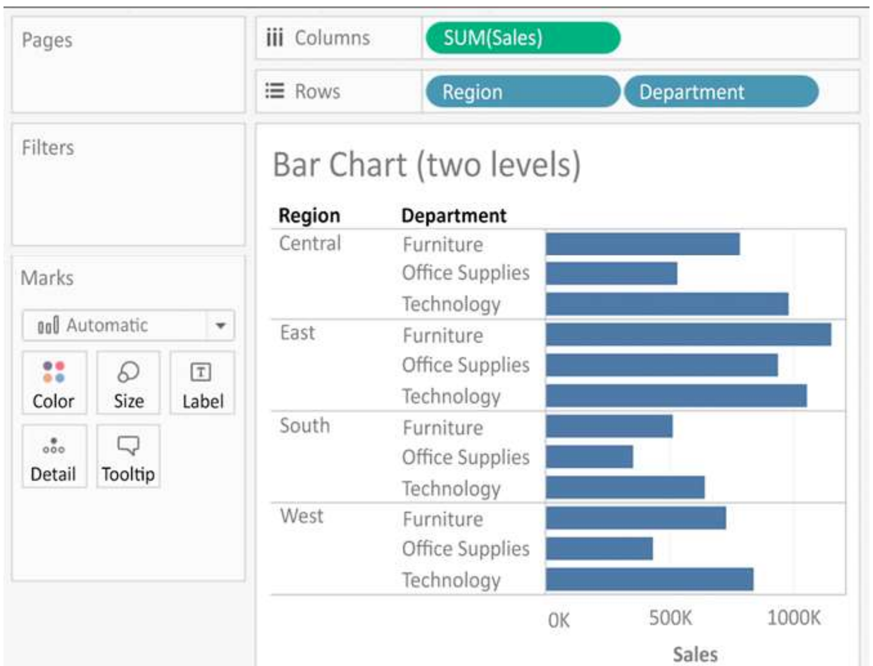


Figure 1.12: The view, Bar Chart (two levels), should look like this when you have completed the preceding steps

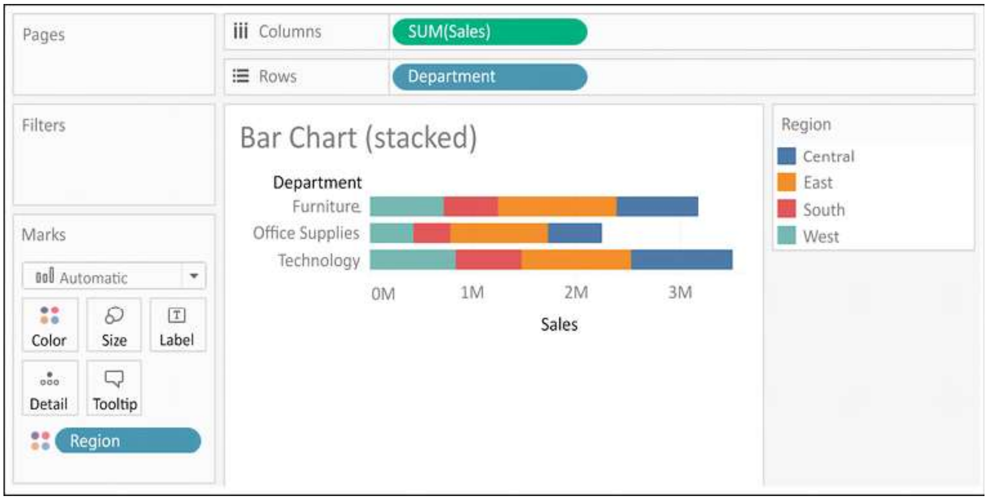


Figure 1.13: The Bar Chart (stacked) view should look like this

Figure 1.14: Swap Rows and Columns button

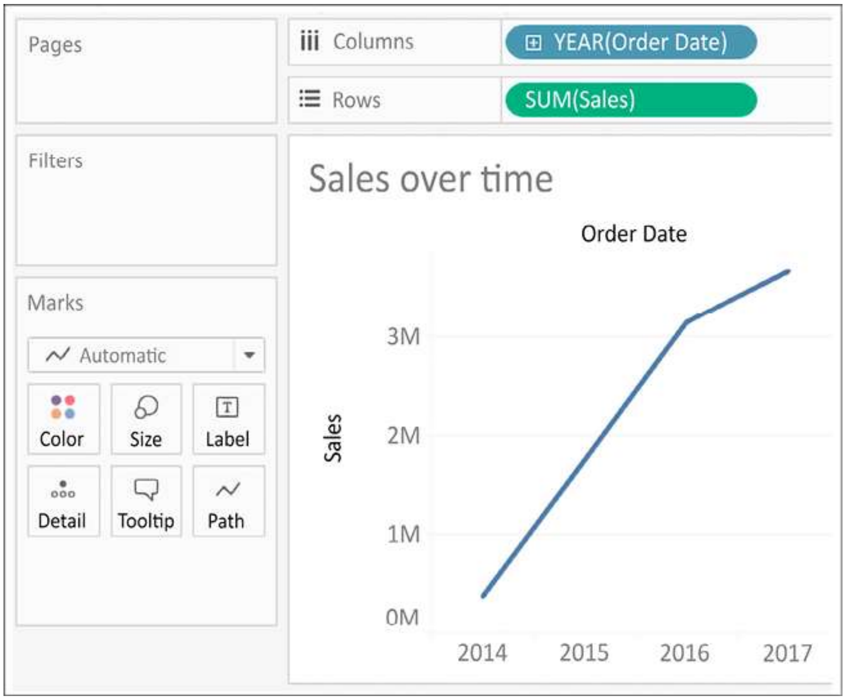


Figure 1.15: An interim step in creating the final line chart; this shows the sum of sales by year

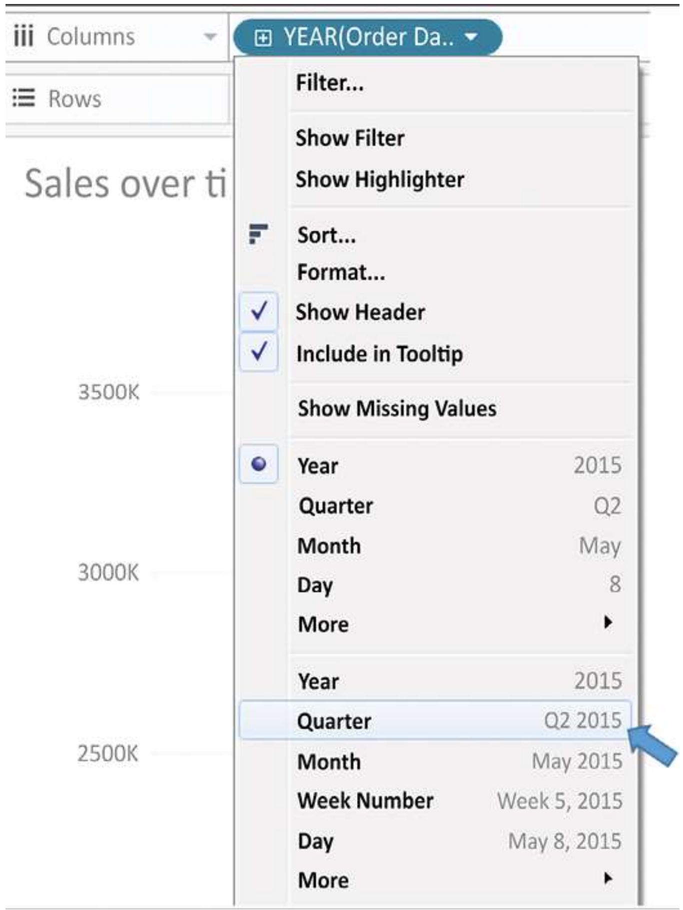


Figure 1.16: Select the second Quarter option in the drop-down menu.

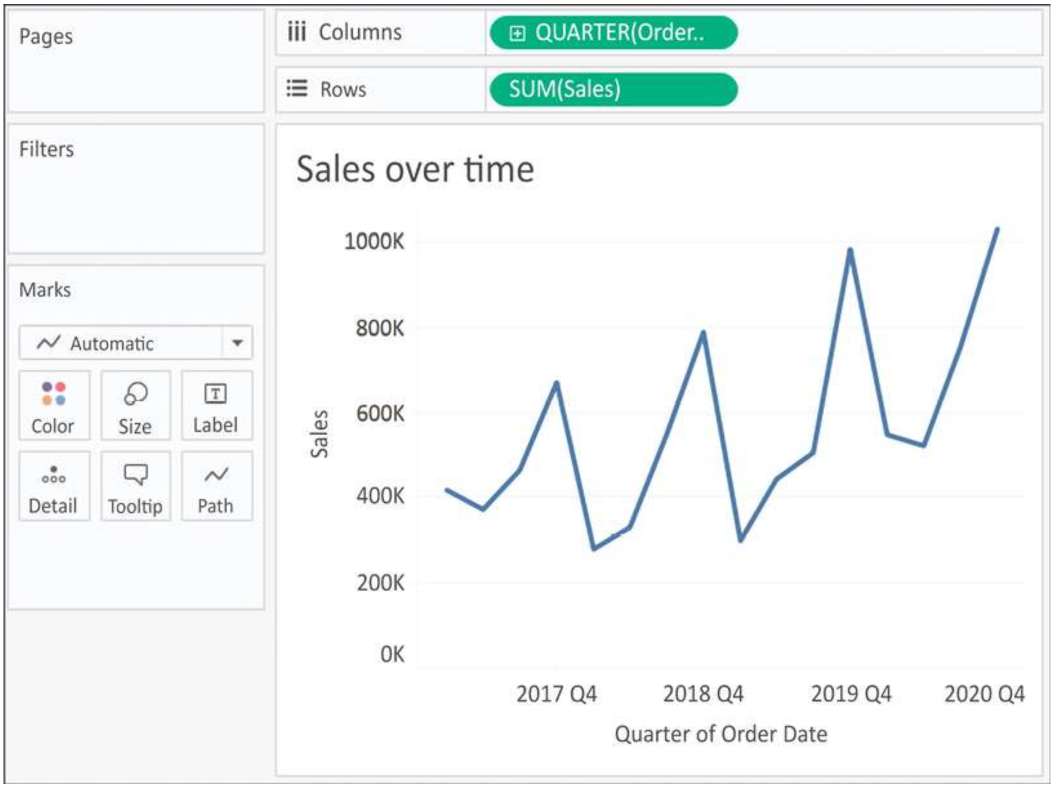


Figure 1.17: Your final view shows sales over each quarter for the last several years

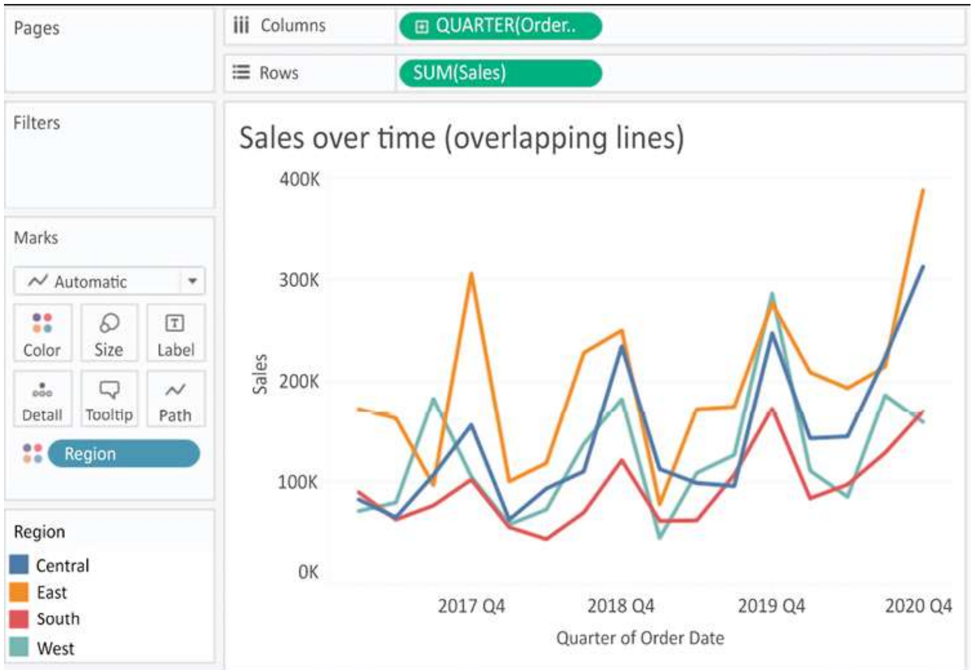


Figure 1.18: This line chart shows the sum of sales by quarter with different colored lines for each region

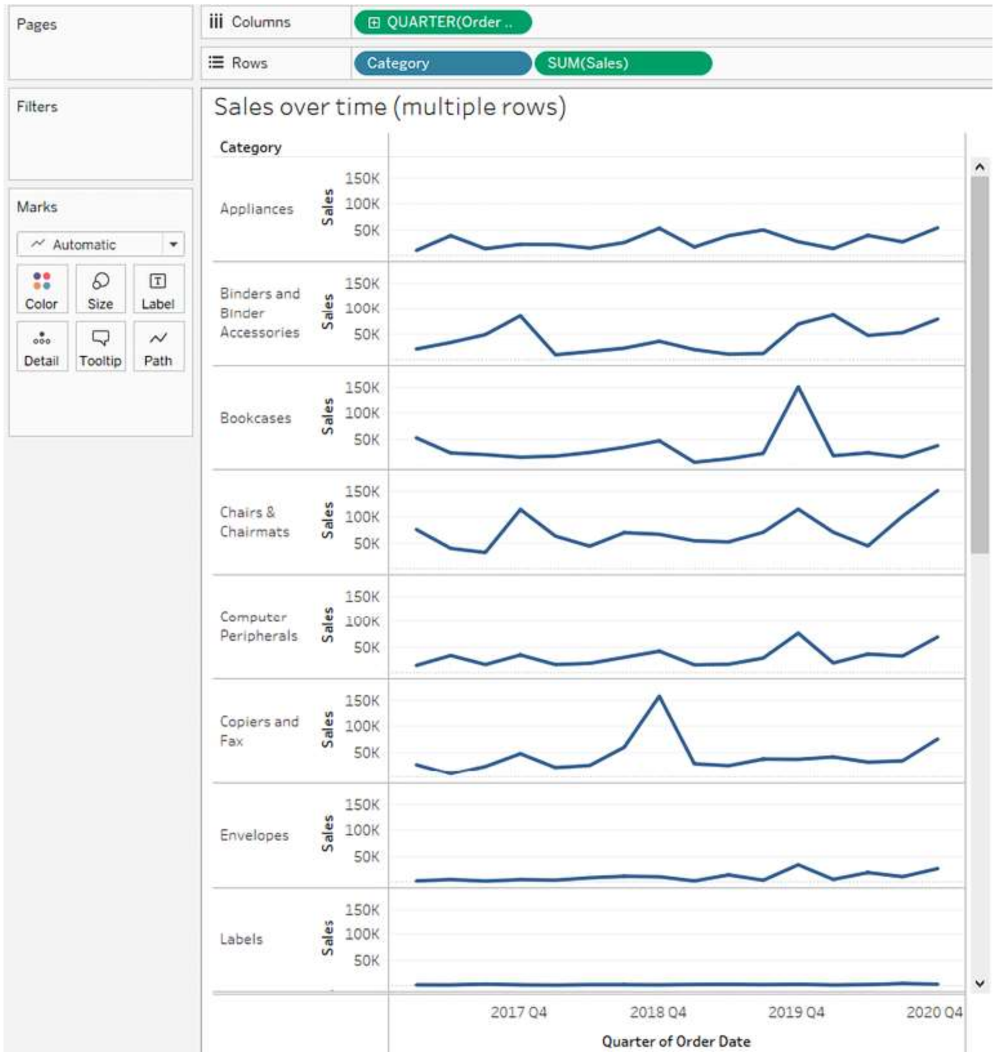


Figure 1.19: Your final view should be a series of line charts for each category

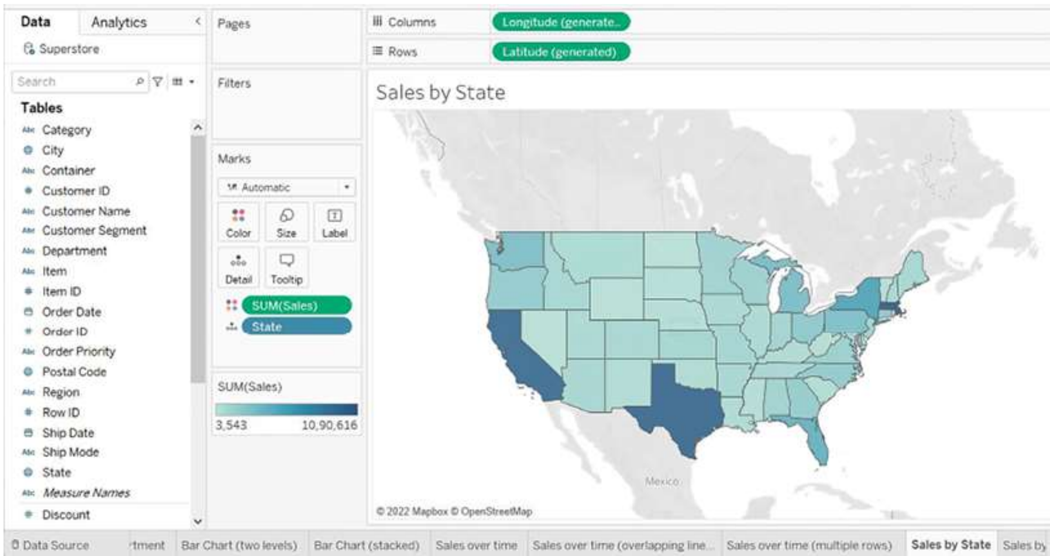


Figure 1.20: A filled map showing the sum of sales per state

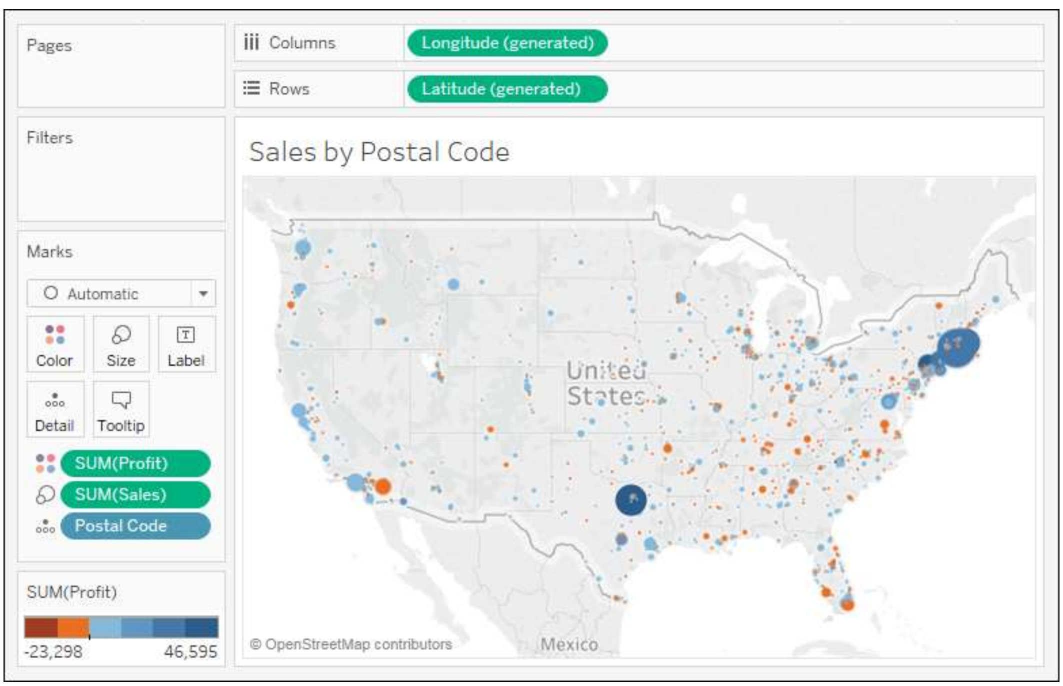


Figure 1.21: A symbol map showing the sum of profit (encoded with color) and the sum of sales (encoded with size) per postal code

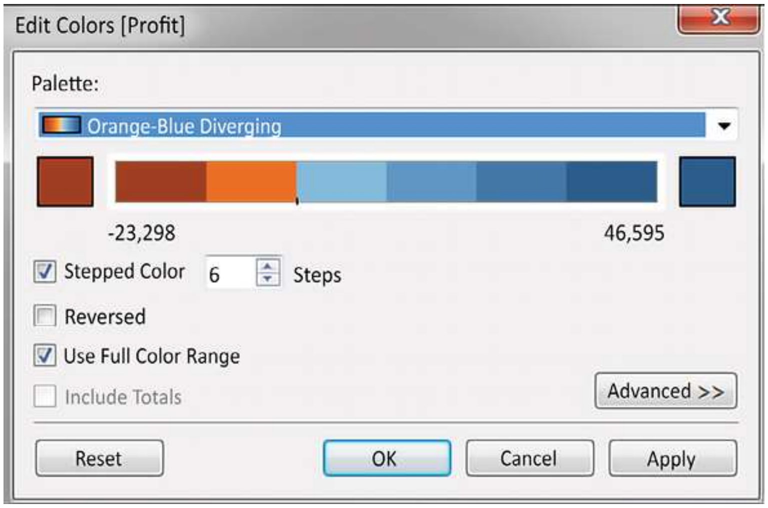


Figure 1.22: The Edit Colors dialog includes options for changing the number of steps, reversing, using the full color range, including totals, and advanced options for adjusting the range and center point

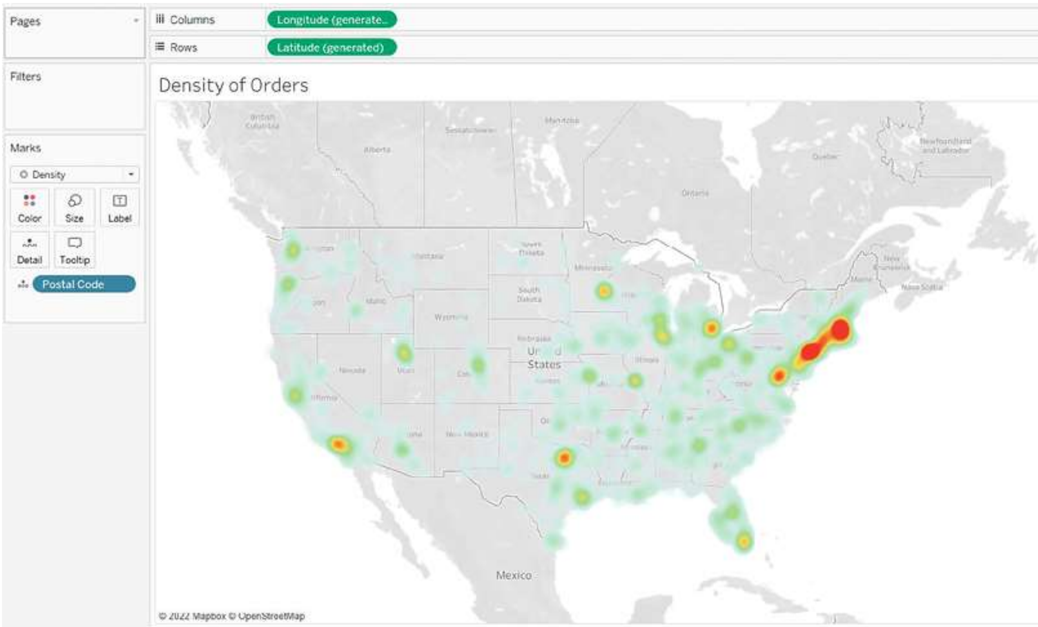


Figure 1.23: A density map showing concentration by postal code

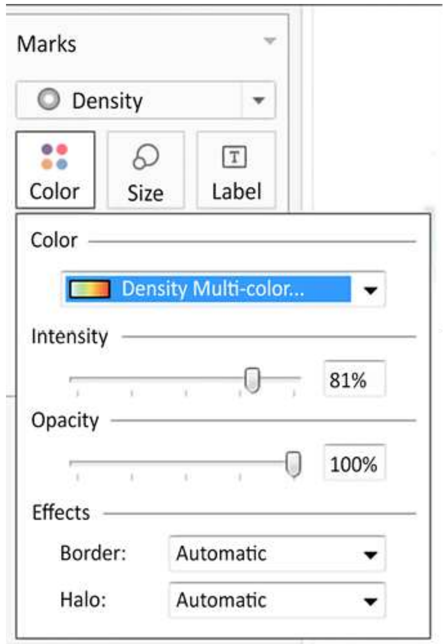


Figure 1.24: Options for adjusting the Color, Intensity, Opacity, and Effects for Density marks

Figure 1.25: The Show Me interface

Figure 1.26: The sidebar for dashboards

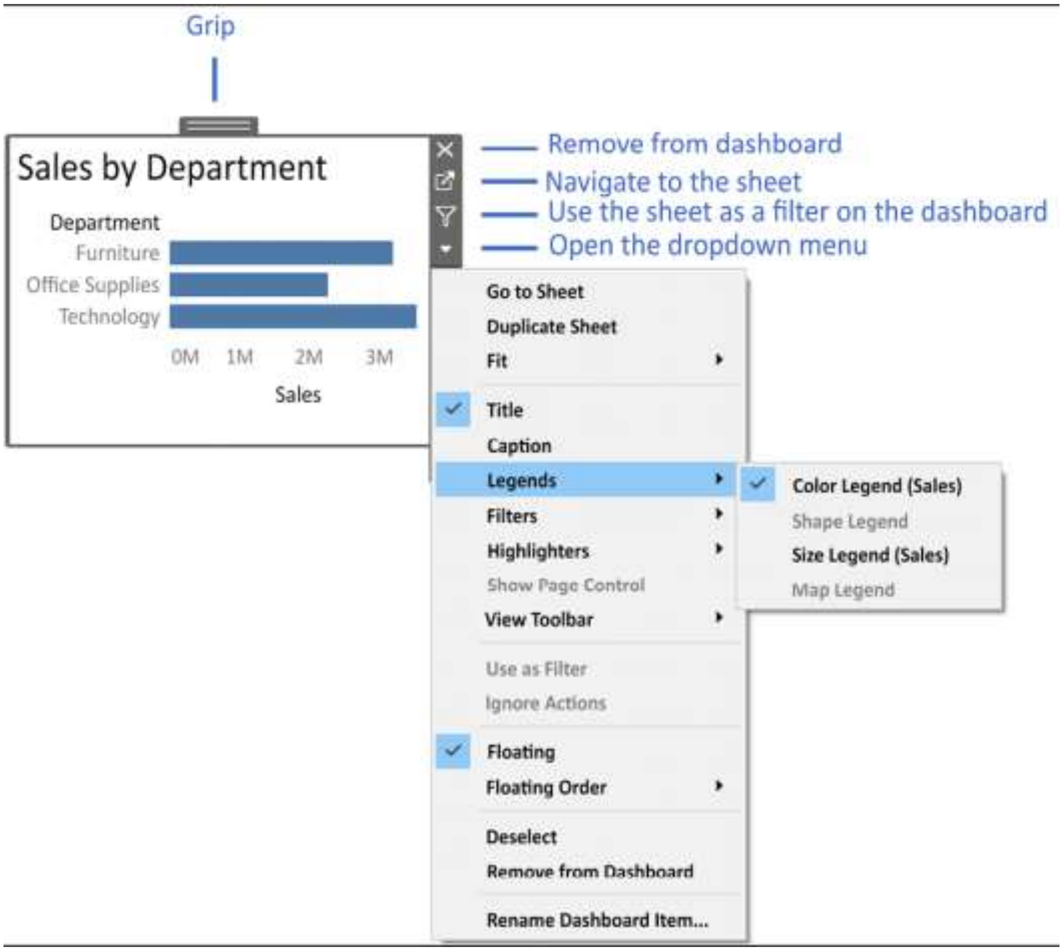


Figure 1.27: Various controls and UI elements become visible when selecting an object on a dashboard

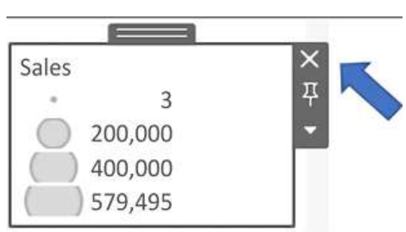


Figure 1.28: Select the legend by clicking on it, then click the X to remove it from the dashboard

Figure 1.29: Click on the Use as Filter button to use a view as a filter in a dashboard

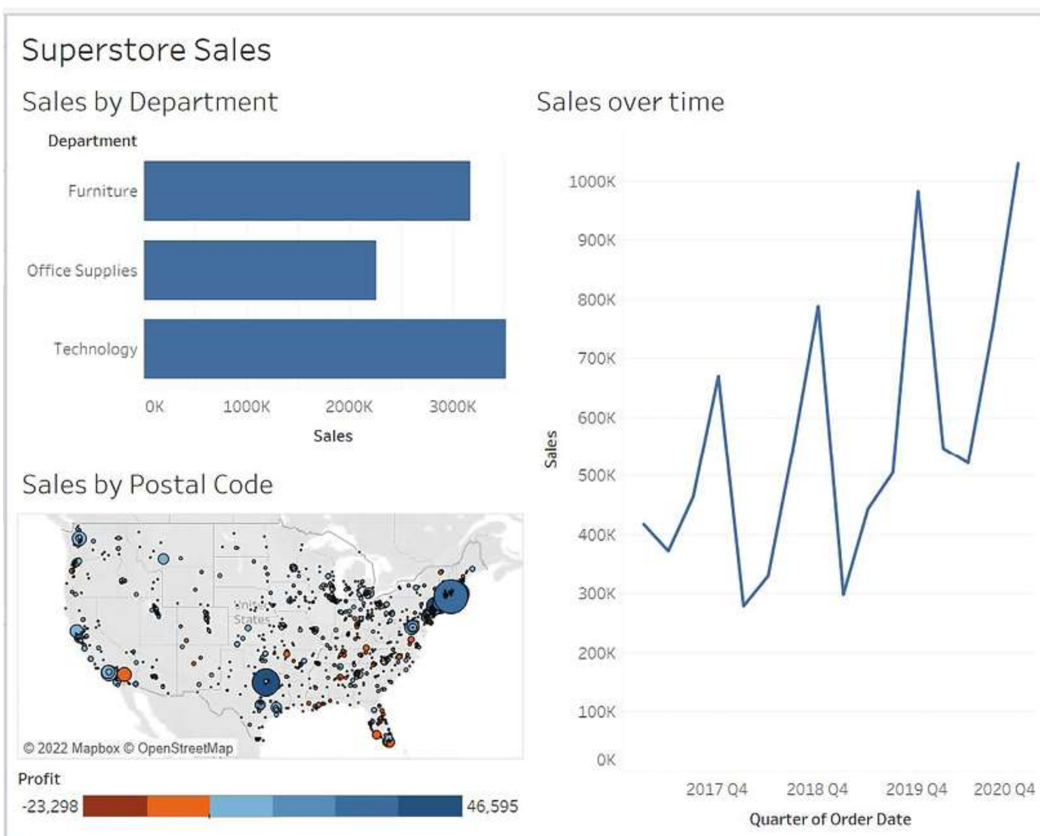


Figure 1.30: The final dashboard consisting of three views

Chapter 2

Figures

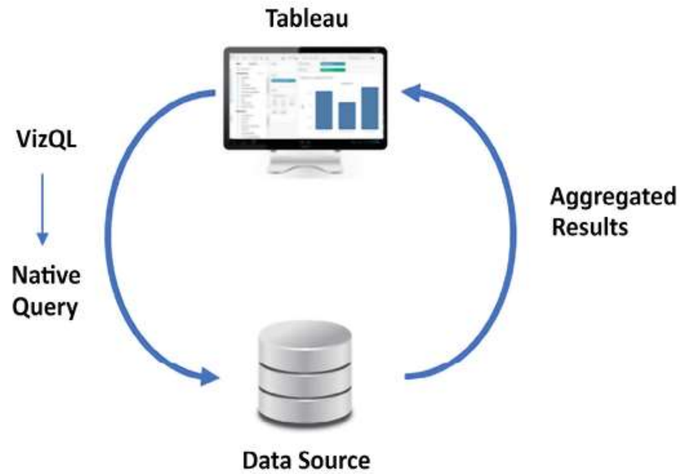


Figure 2.1: The basic Tableau paradigm for working with data

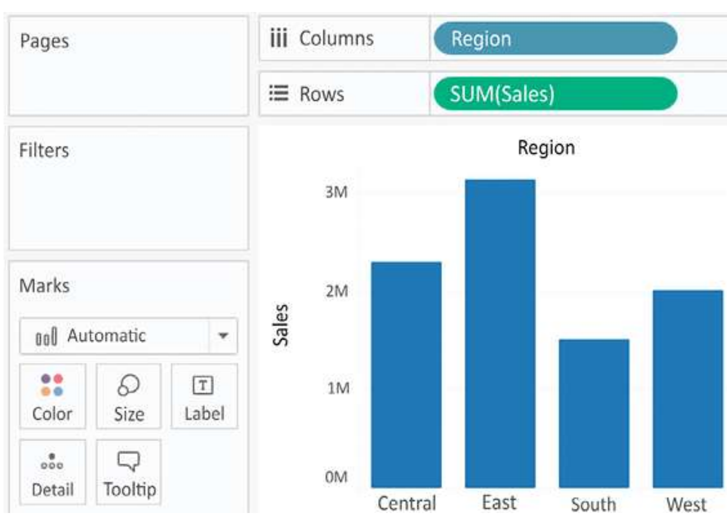
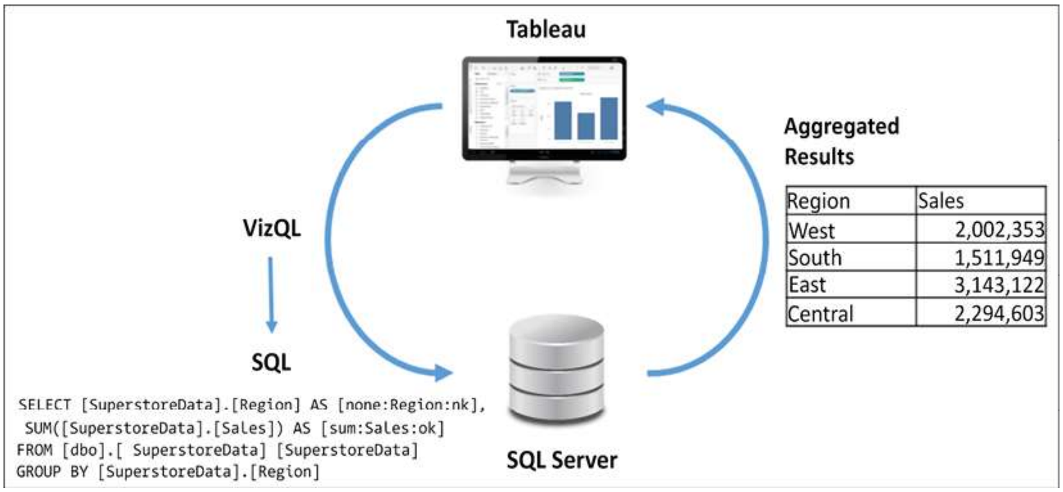


Figure 2.2: This bar chart is the result of a query that returned four aggregate rows of data



Region	Sales
West	2,002,353
South	1,511,949
East	3,143,122
Central	2,294,603

Figure 2.3: Tableau generated the bar chart in the previous image using a paradigm like this

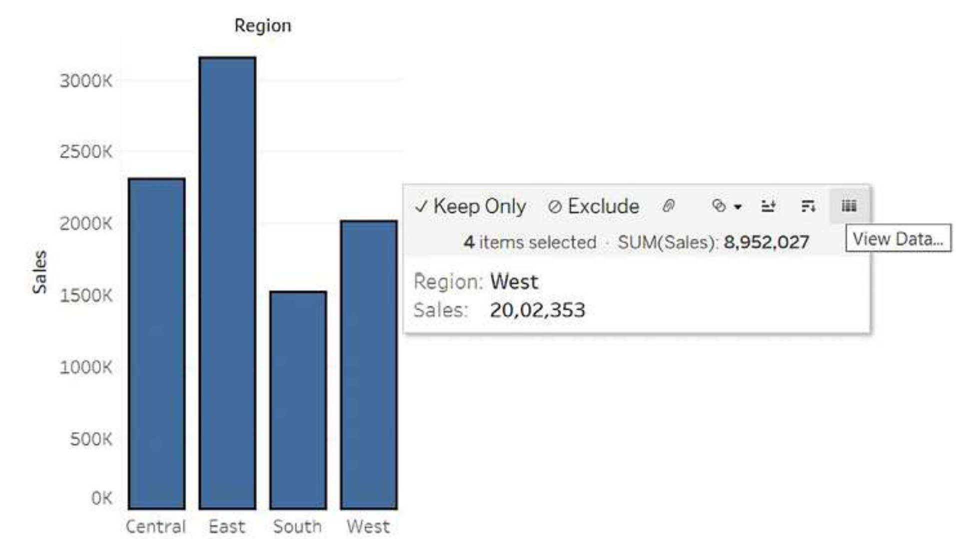


Figure 2.4: Use the View Data… tooltip option to see a summary or underlying data for a mark

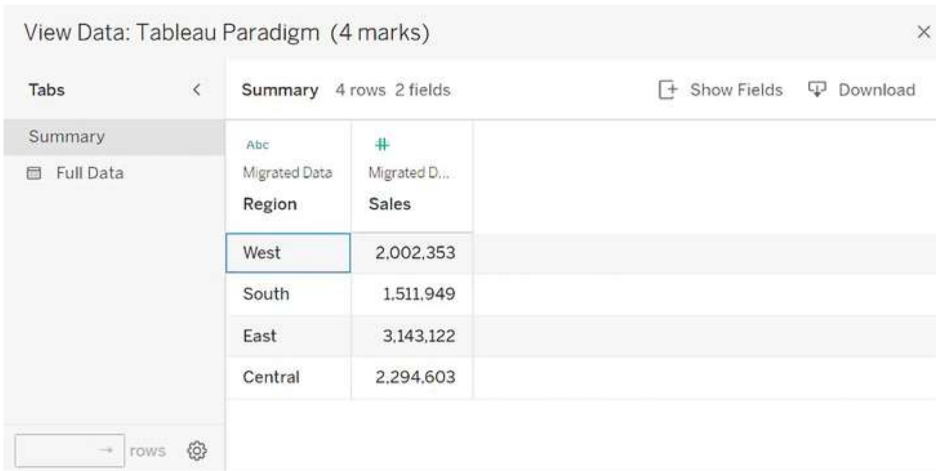


Figure 2.5: The Summary tab displays the aggregate data Tableau used to render each mark in the view

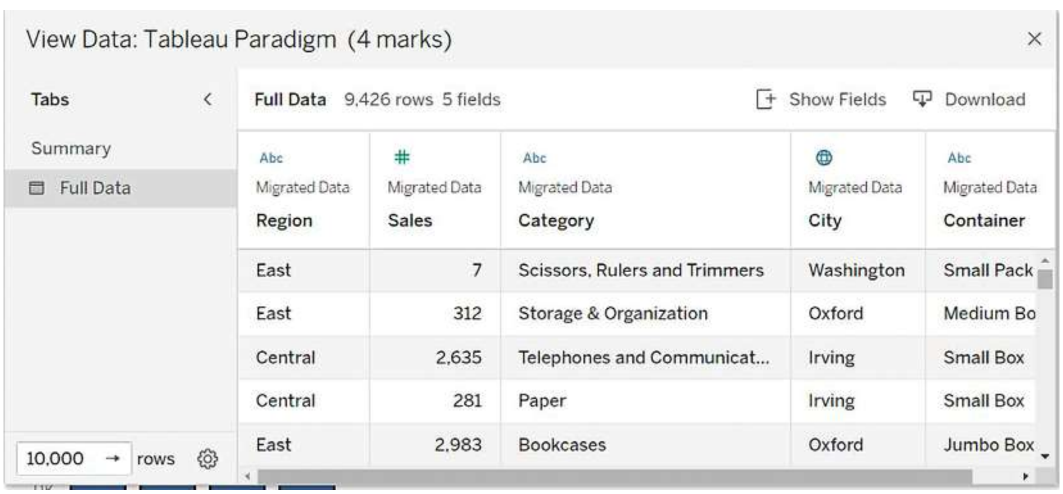


Figure 2.6: The Full Data tab reveals the row-level data in the database

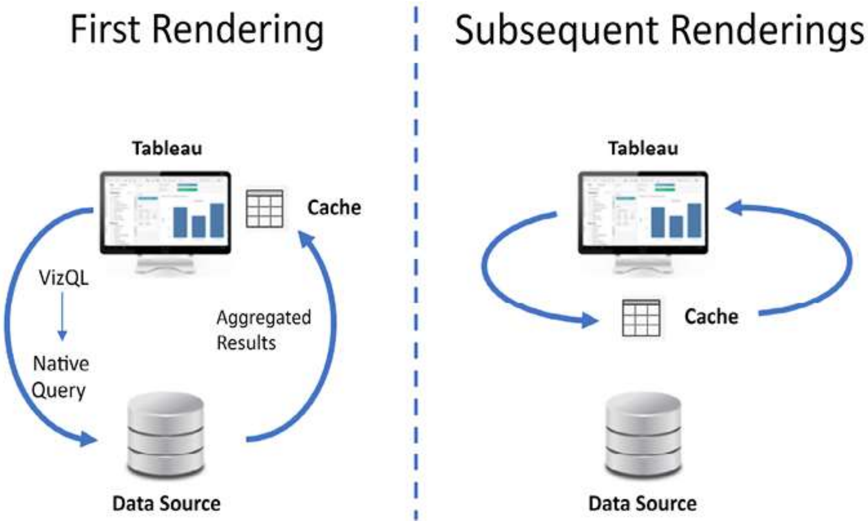


Figure 2.7: The first rendering with a given set of fields queries the data source directly. Subsequent renderings will query the cache, even if the same fields are re-arranged in the view

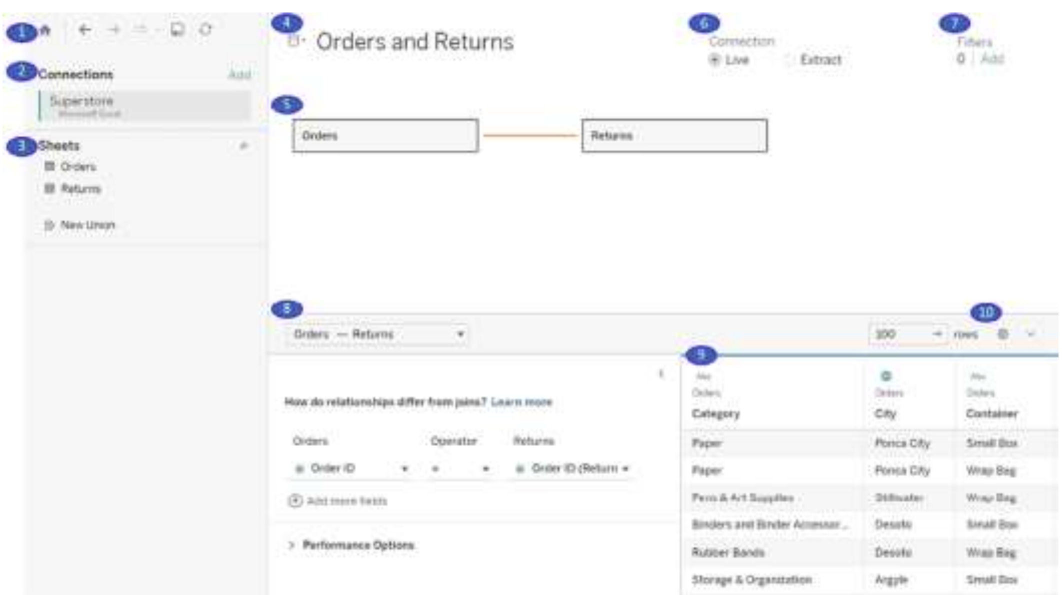


Figure 2.8: The Data Source screen with two objects (Orders and Returns)

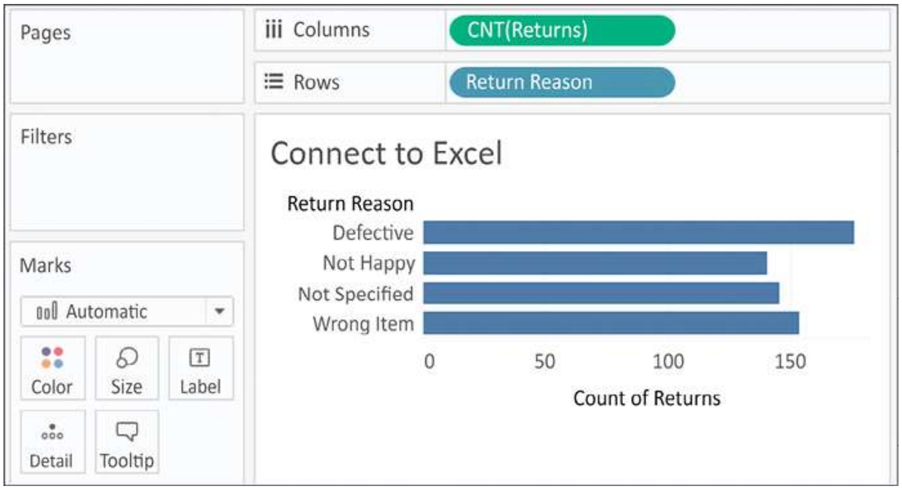


Figure 2.9: The number of returns by return reason

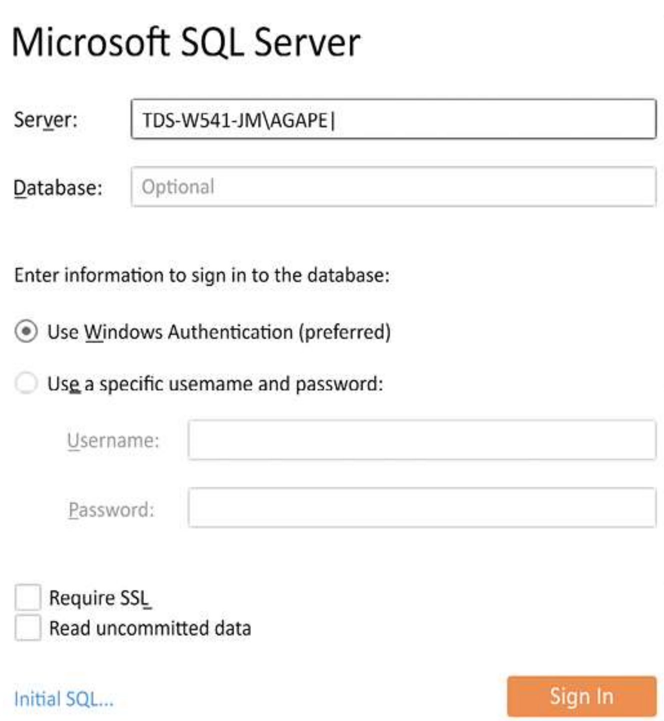


Figure 2.10: The connection editor for Microsoft SQL Server

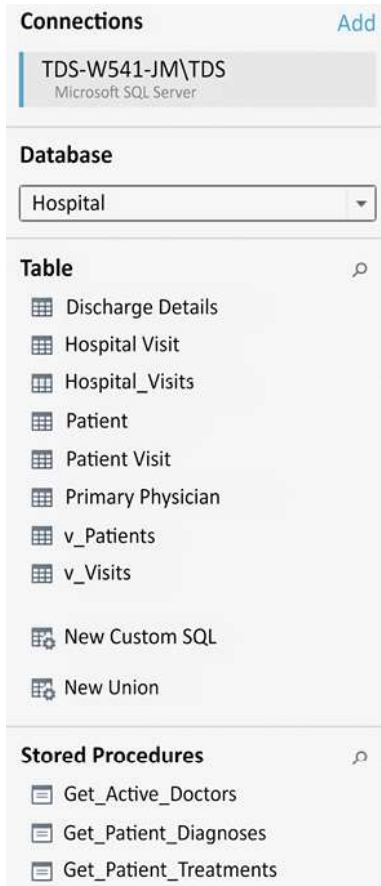


Figure 2.11: Once connected to a database, Tableau will display tables, views, and stored procedures as options to add to the object model

Figure 2.12: The add data button

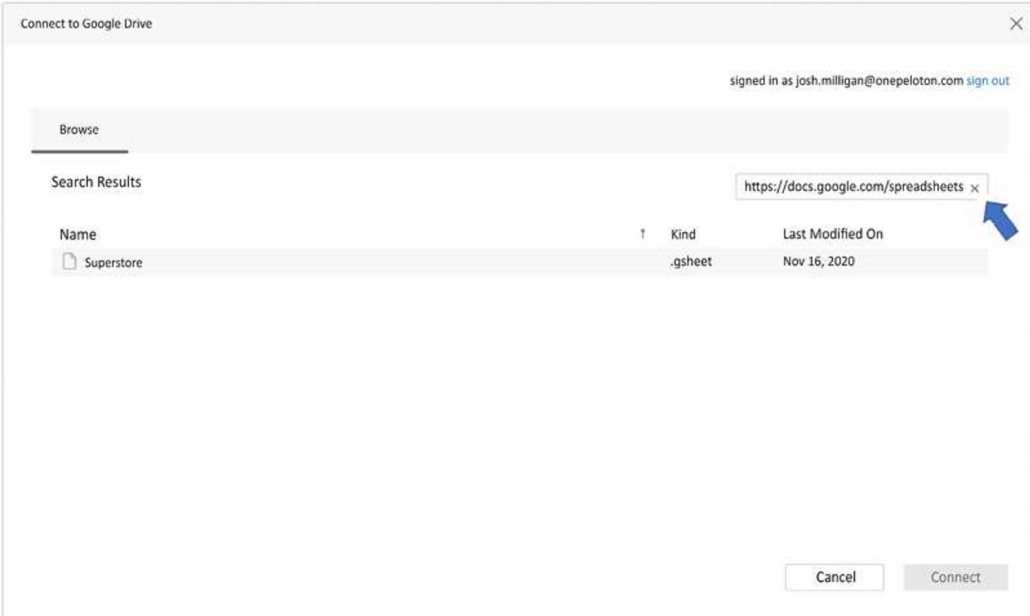


Figure 2.13: When connecting, you may browse the contents of Google Drive. In this example, paste the shared URL into the search box

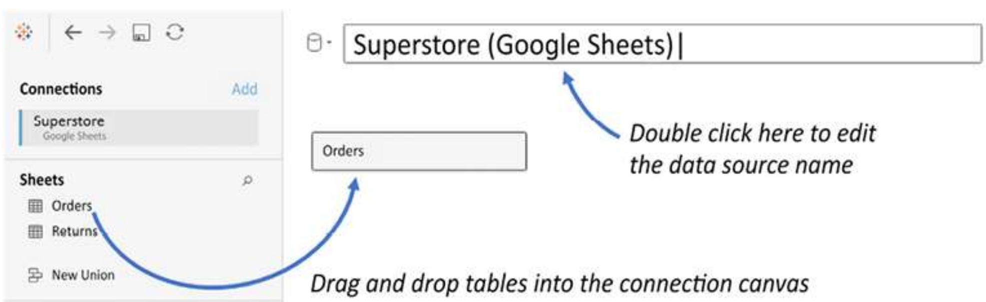


Figure 2.14: Renaming a data source

Figure 2.15: Switch between Live and Extract, edit extract options, and add filters

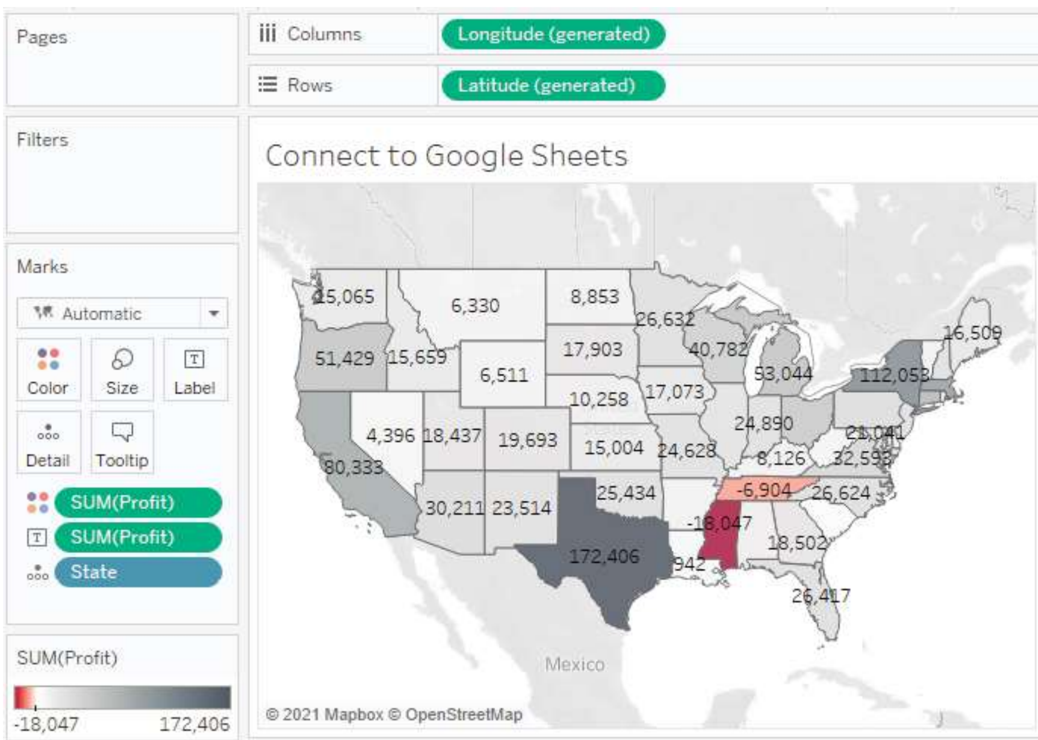


Figure 2.16: The filled map demonstrates the ability to connect to a cloud-based data source

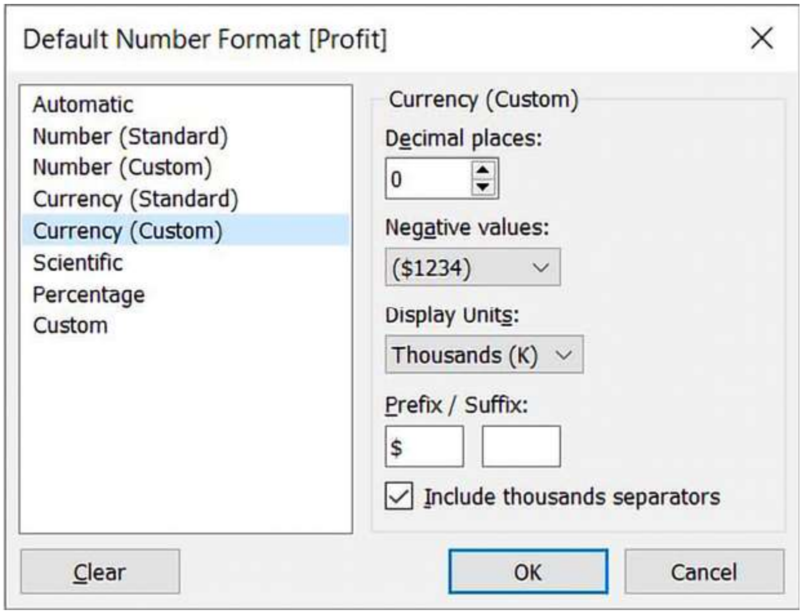


Figure 2.17: Editing the default number format of a field

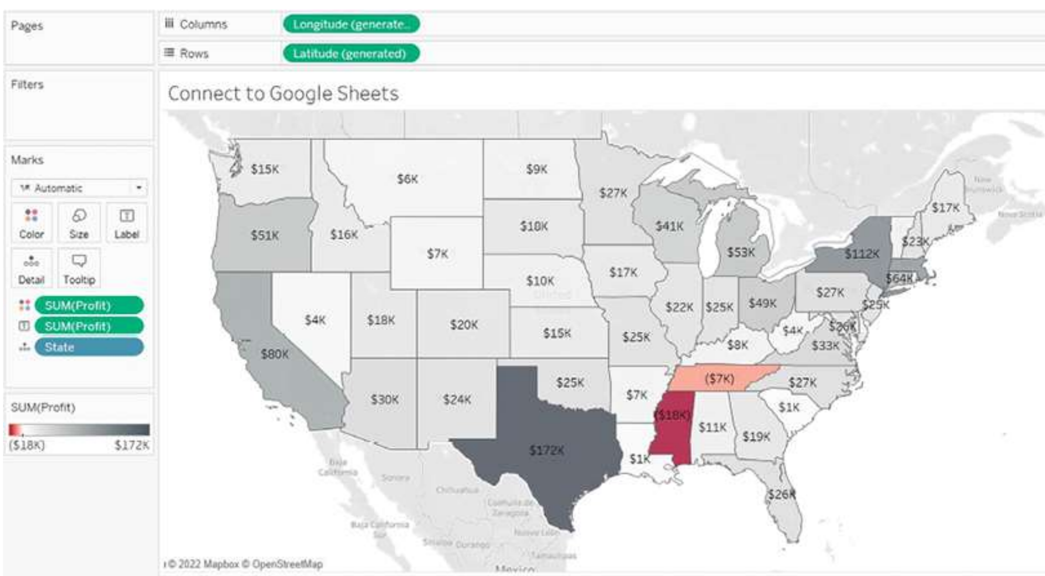


Figure 2.18: Customizing color

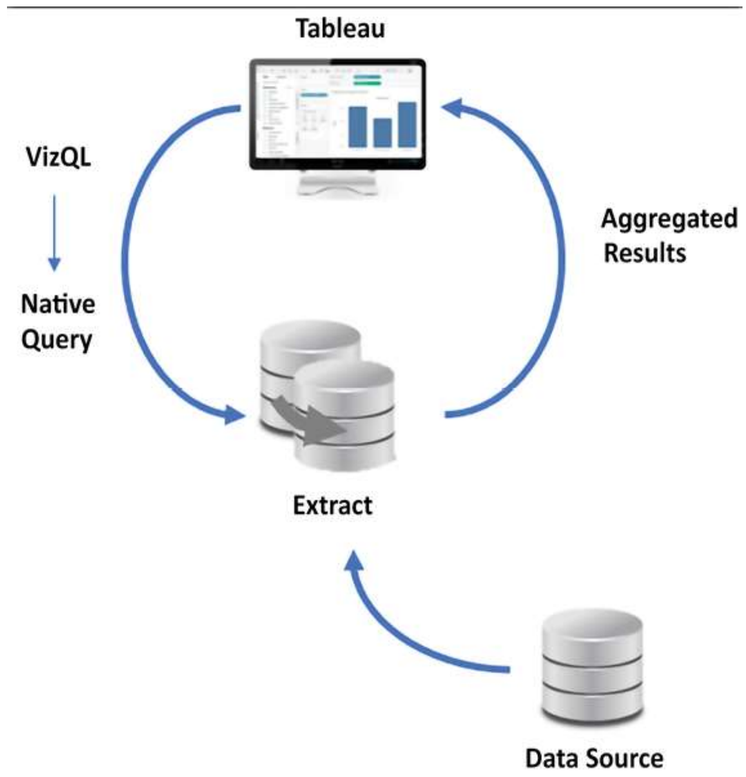


Figure 2.19: Data from the original data source is extracted into a self-contained snapshot of the data

Figure 2.20: Select either Live or Extract for a connection and configure options for the extract by clicking Edit...

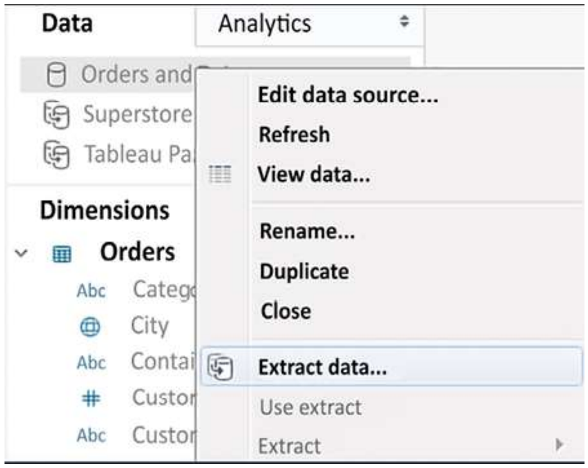


Figure 2.21: The Extract data… option

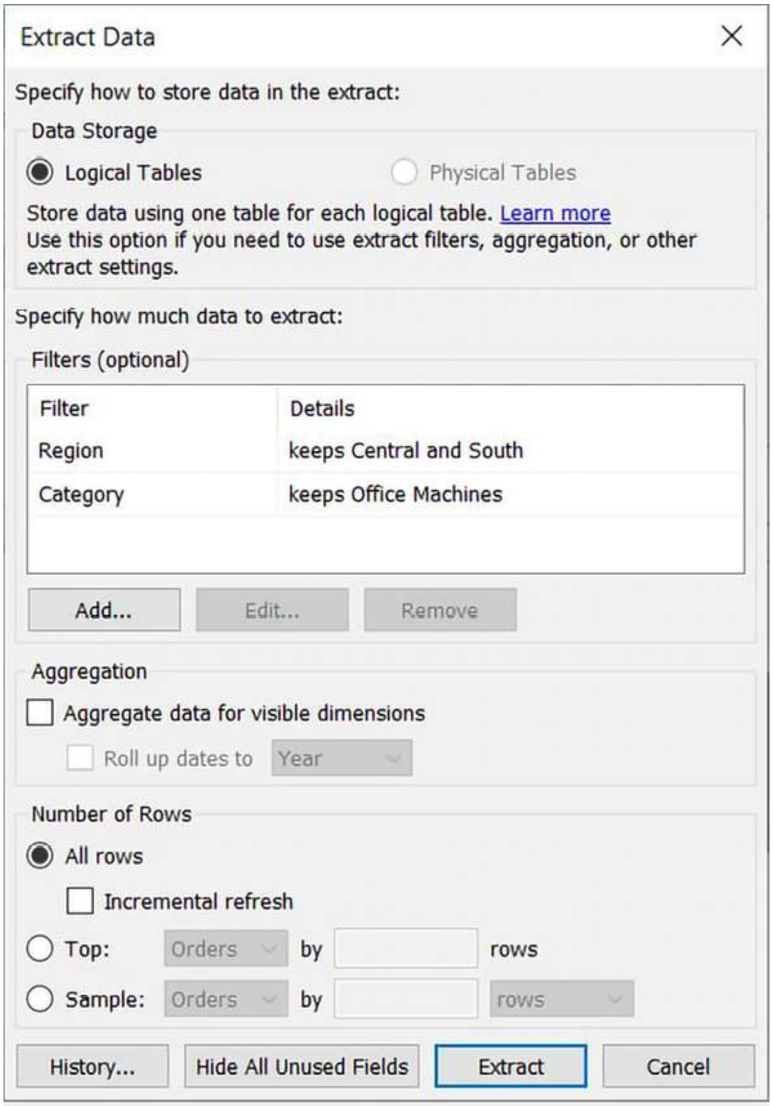


Figure 2.22: The Extract Data dialog gives quite a few options for how to configure the extract

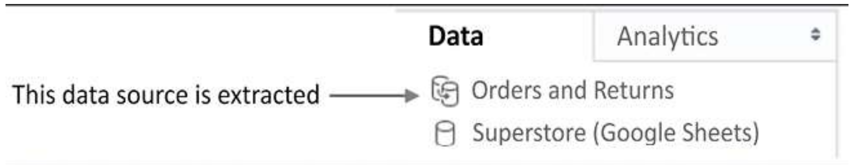


Figure 2.23: The icon next to a data source indicates whether it is extracted or not

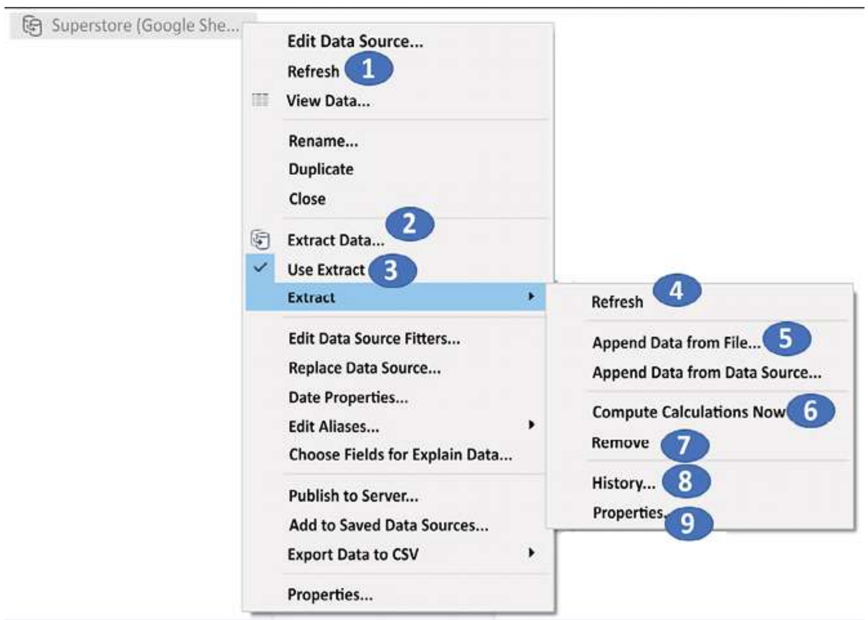


Figure 2.24: The context menu for a data connection in the data pane with Extract options numbered

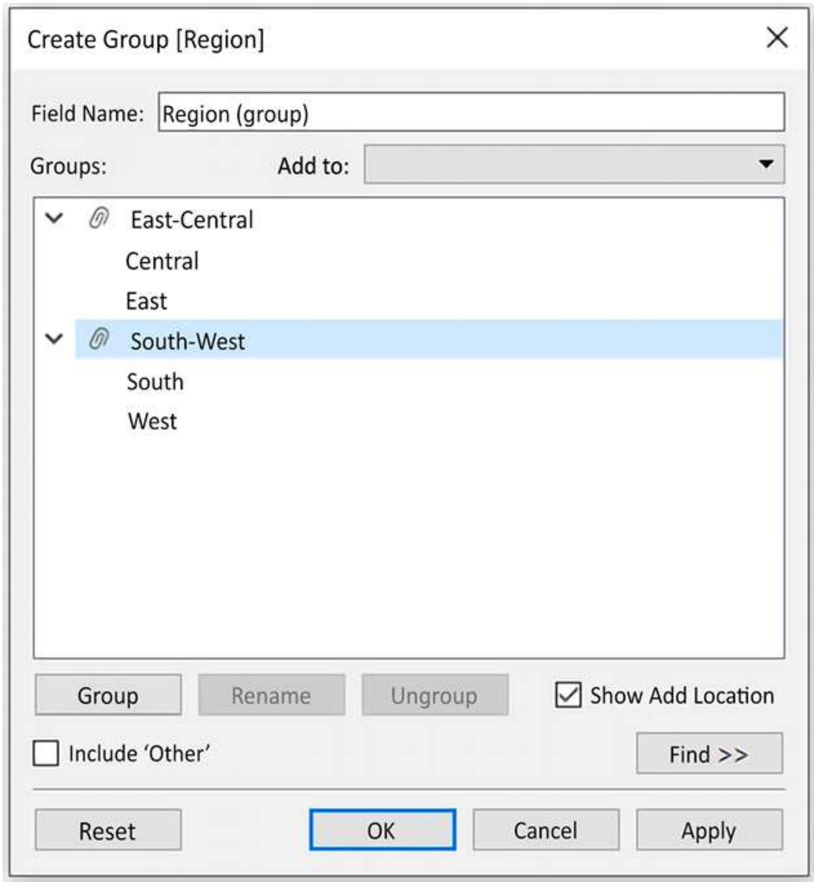


Figure 2.25: The Create Group dialog box allows you to group together values for a given dimension

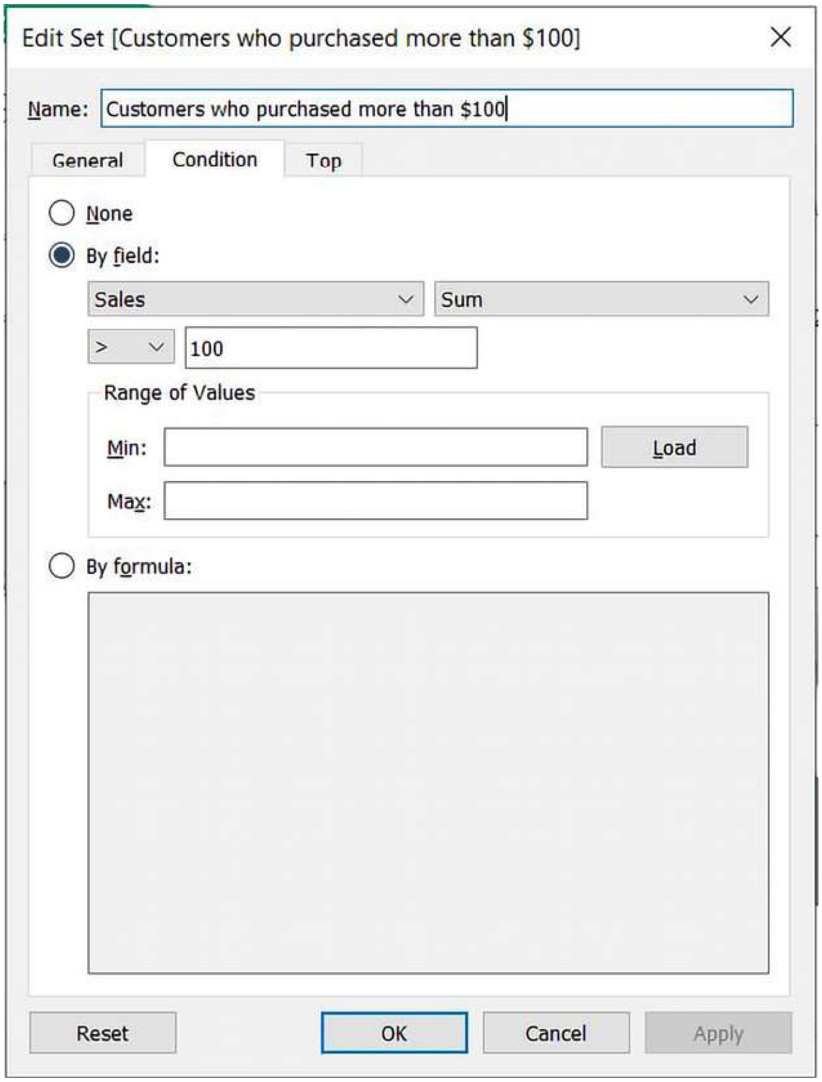


Figure 2.26: A dynamic set based on a condition

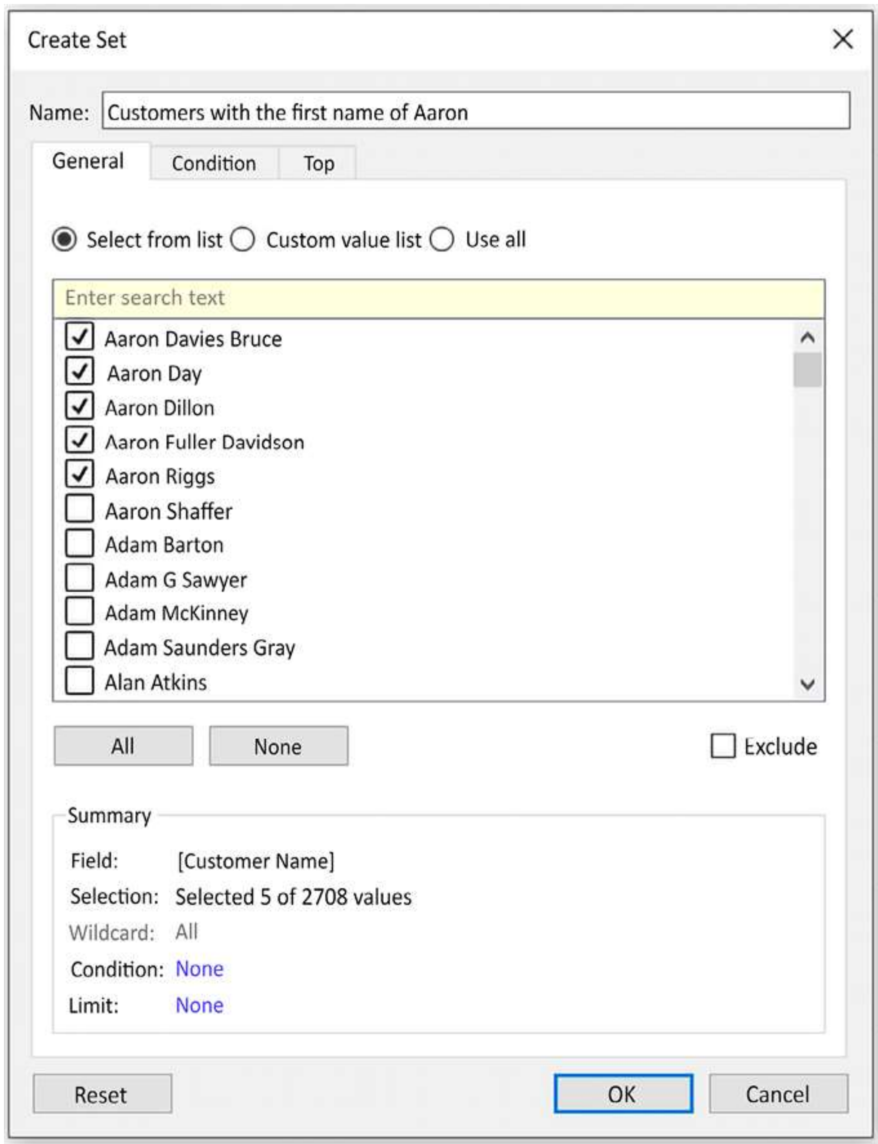


Figure 2.27: A static set based on the selection of members

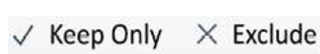


Figure 2.28: Based on the mark selection, you may Keep Only values that match or Exclude such values

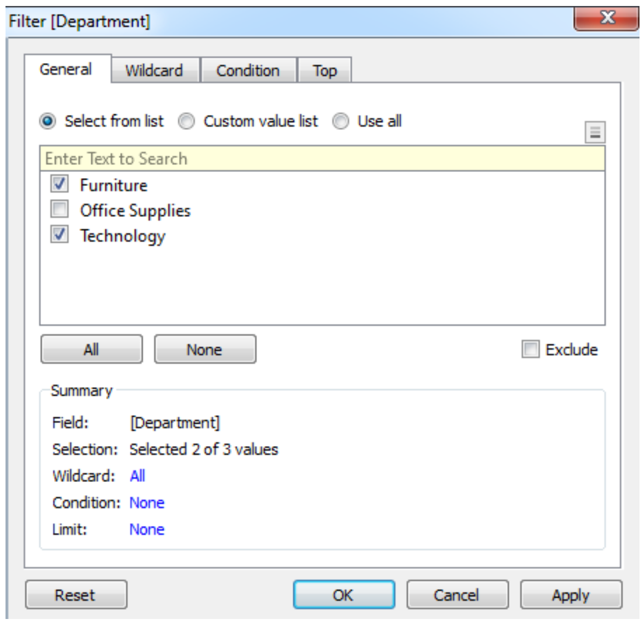


Figure 2.29: A filter for a discrete field will show options for including or excluding individual values

Figure 2.30: For numeric values, you'll often see options for aggregating the value as part of the filter

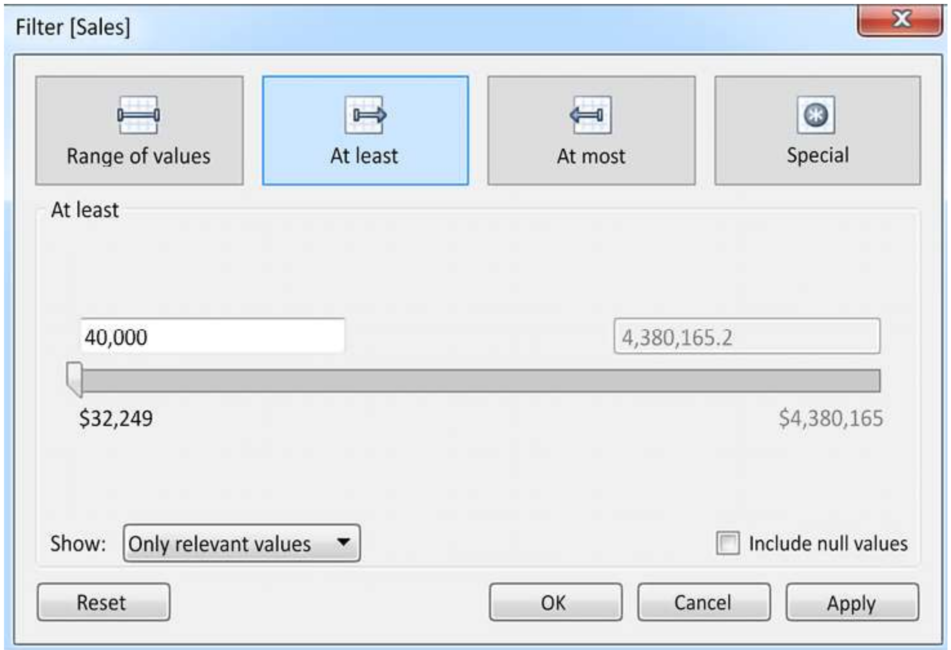


Figure 2.31: Filter options for Sales (as a SUM)

Figure 2.32: Initial filter options for a date field

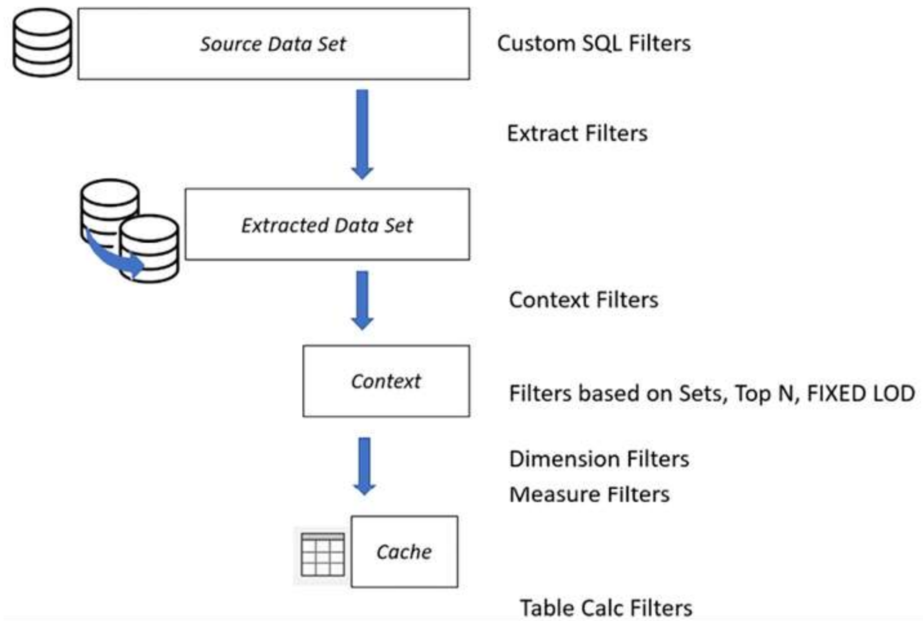


Figure 2.33: Order of Operations

Links

- Connect to Google Drive: https://docs.google.com/spreadsheets/d/1YUZ1UnryuCX7cwbfNptGuzx0G4yth7i-m9Jrkce9_PE/edit?usp=sharing
- Create an extract using the Tableau Hyper API: https://help.tableau.com/current/api/hyper_api/en-us/index.html

Chapter 3

Figures

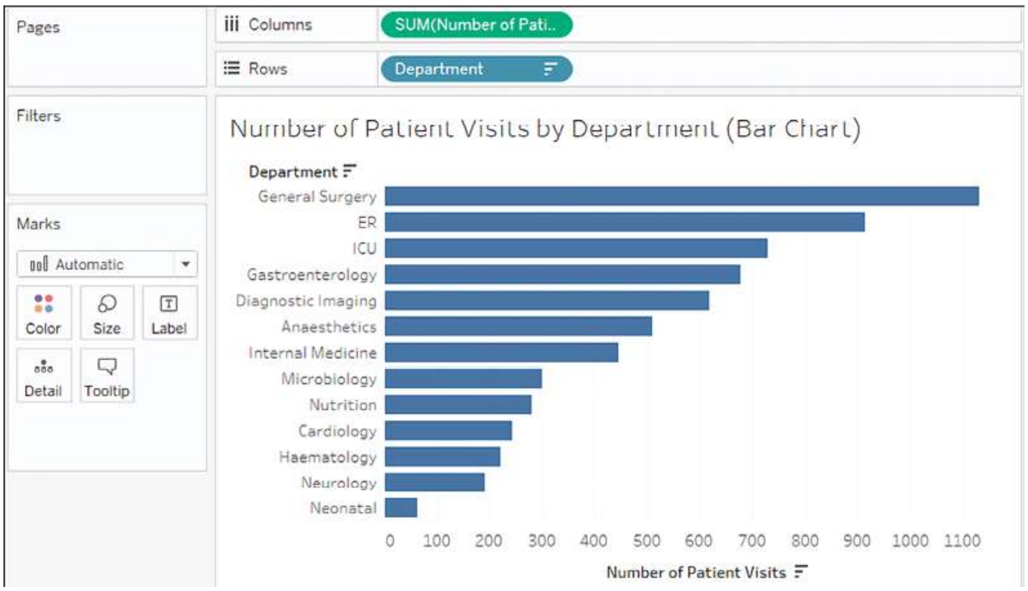


Figure 3.1: A bar chart showing the number of patient visits by department

Figure 3.2: Toolbar sort icons

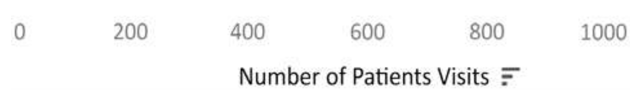


Figure 3.3: Axis sort icon

Figure 3.4: Sorting using the drop-down menu

Figure 3.4: Sorting using the drop-down menu

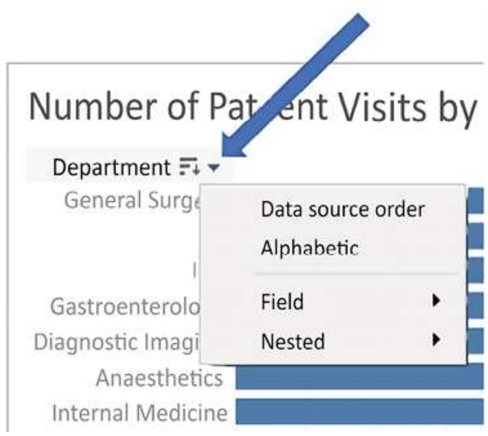


Figure 3.5: Sorting by the field label

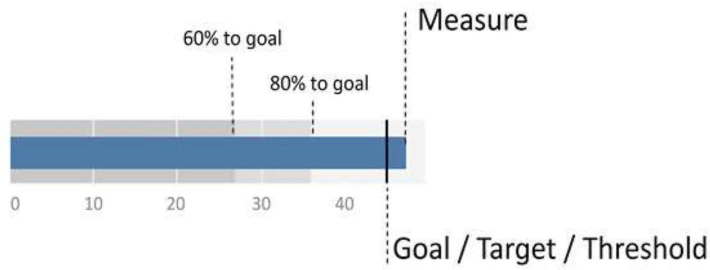


Figure 3.6: Parts of a bullet graph

Department	Goal
Anaesthetics	$300,000
Cardiology	$5,000,000
Diagnostic Imaging	$500,000
ER	$6,000,000
Gastroenterology	$900,000
General Surgery	$8,000,000
Haematology	$800,000
ICU	$3,800,000
Internal Medicine	$200,000
Microbiology	$50,000
Neonatal	$10,000
Neurology	$3,000,000
Nutrition	$10,000

Figure 3.7: Department goals are stored in a spreadsheet as shown here

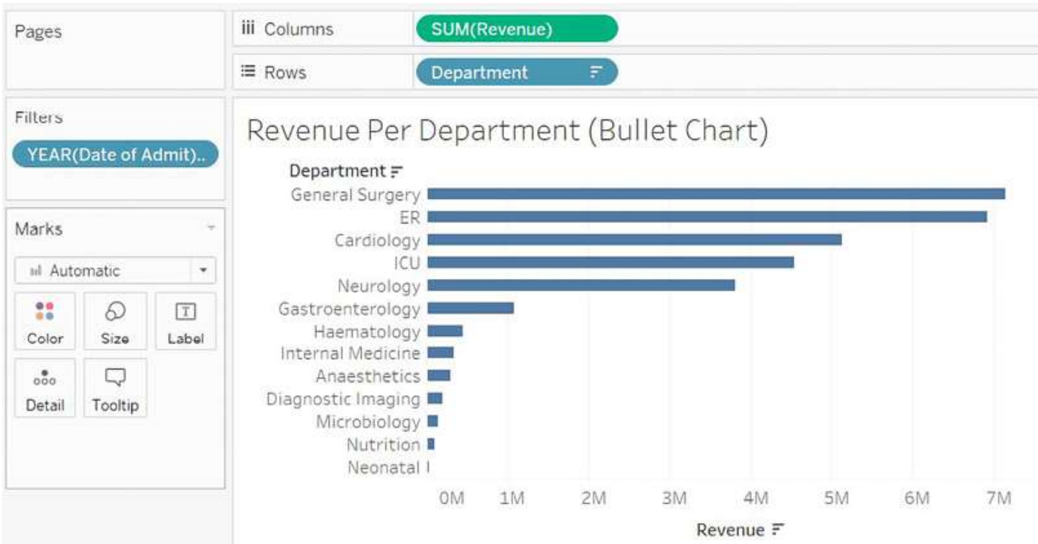


Figure 3.8: Interim steps in creating the bullet graph

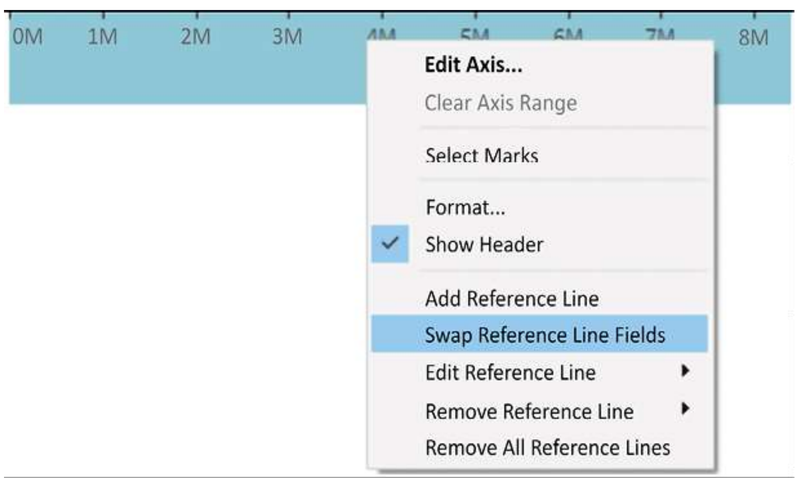


Figure 3.9: The Swap Reference Line Fields option

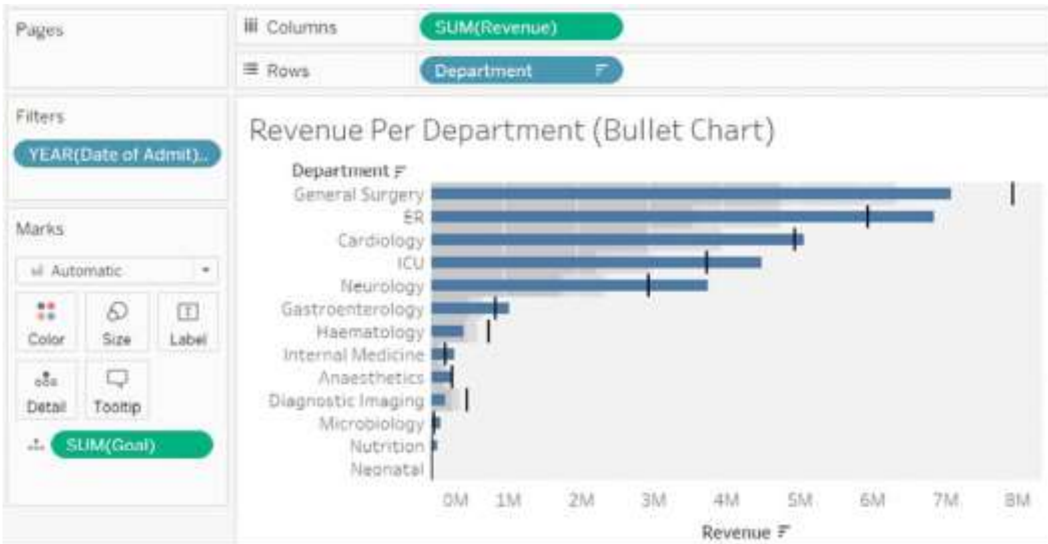


Figure 3.10: The complete bullet graph

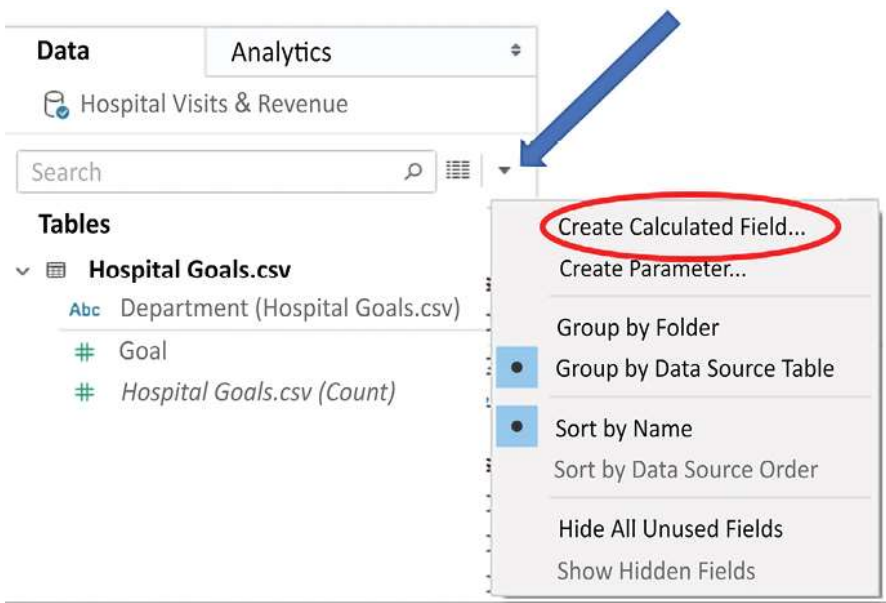


Figure 3.11: Creating a calculated field

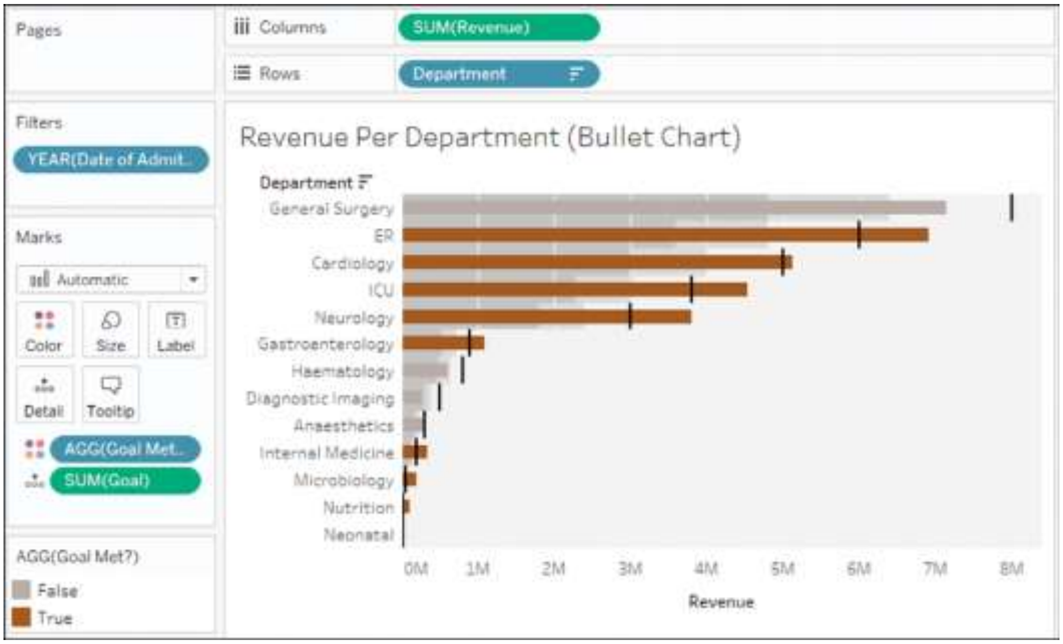


Figure 3.12: Departments that have met their goal are highlighted in this bullet chart

Figure 3.13: Bar-in-bar chart

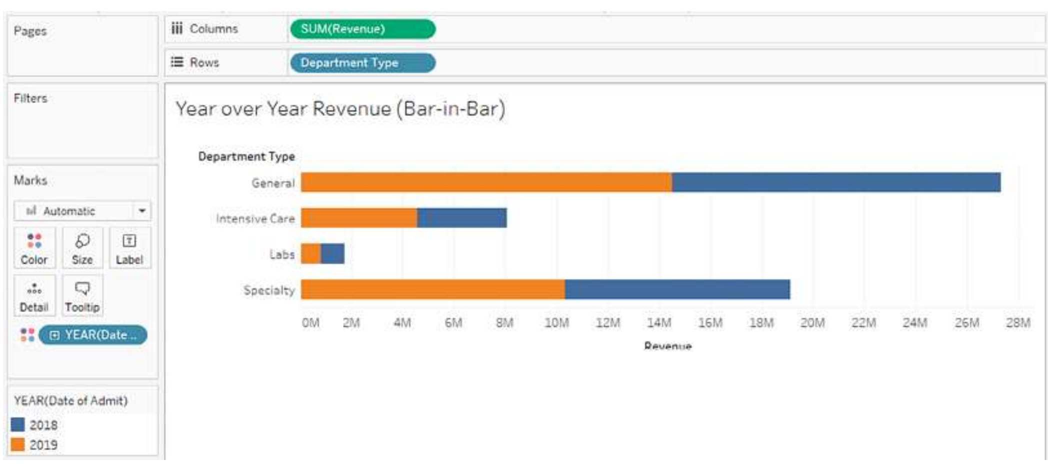


Figure 3.14: Interim steps in creating the bar-in-bar chart

Figure 3.15: You can drag and drop items in legends to reorder them

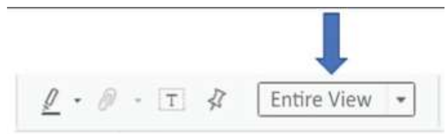


Figure 3.16: This dropdown determines how the current view is sized

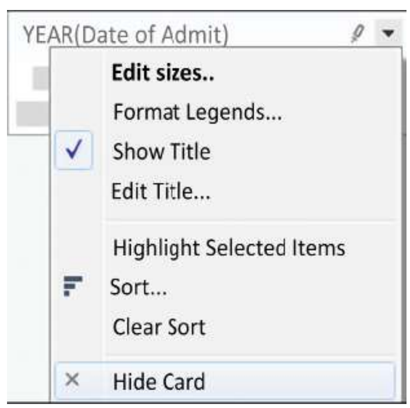


Figure 3.17: The Hide Card option for legends

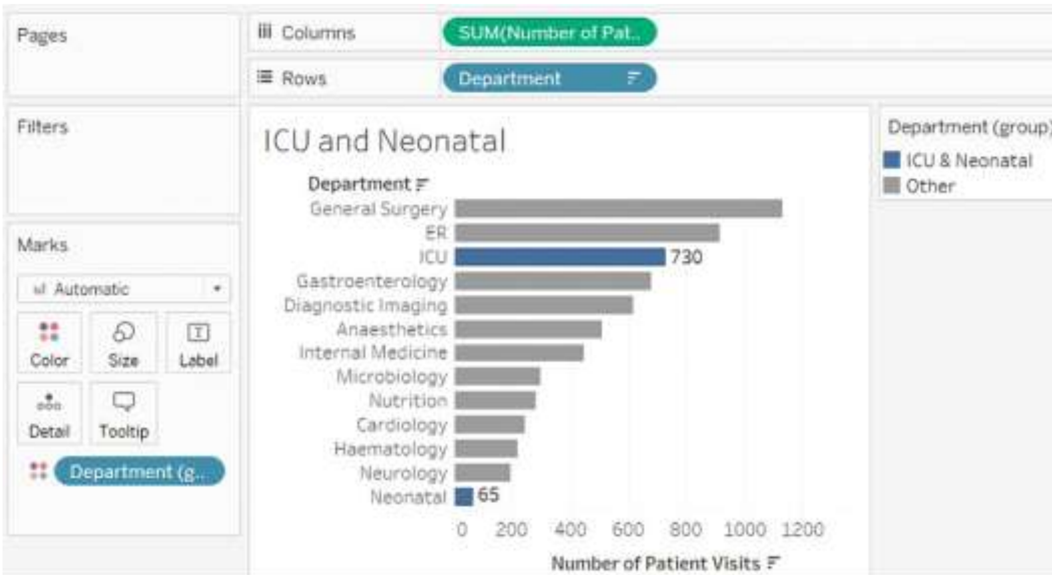


Figure 3.18: A bar chart with two bars highlighted via color

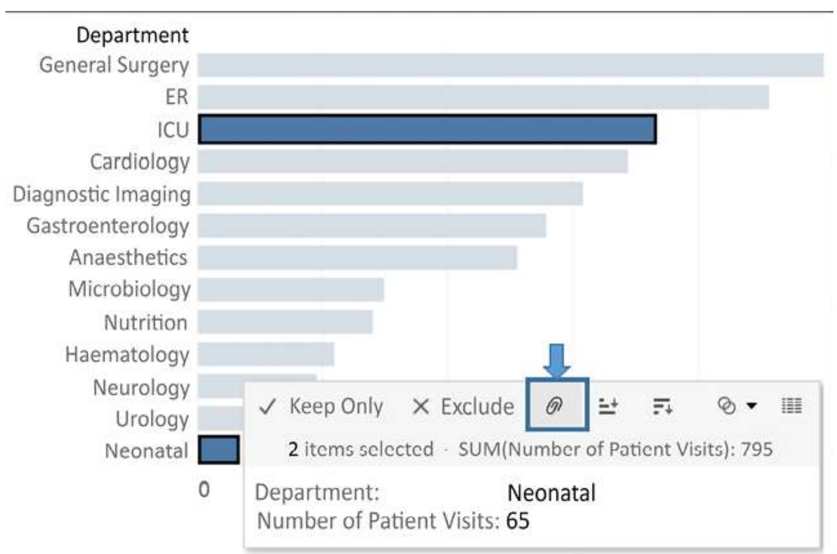


Figure 3.19: After Ctrl + clicking the two bars, use the paperclip icon to group them

Figure 3.20: The field representing the Year part of the date hierarchy

Figure 3.21: A plus icon on the column headers that could be used to expand the hierarchy

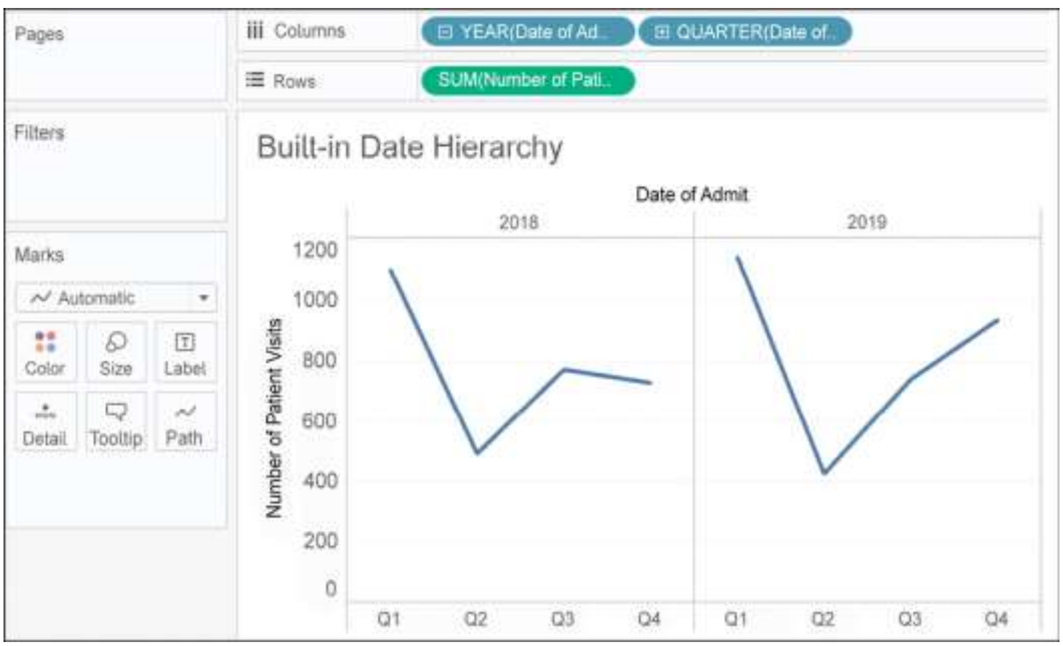


Figure 3.22: The expanded hierarchy with the year and quarter shown

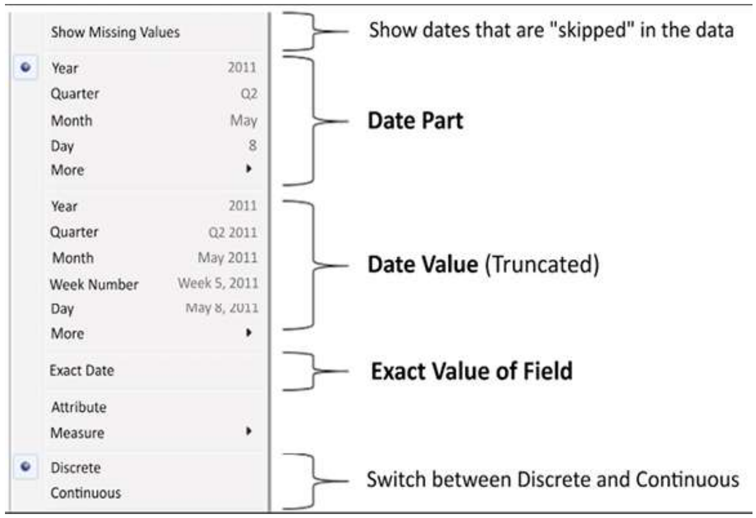


Figure 3.23: The drop-down menu on an active date field demonstrates the various aspects of dates in Tableau

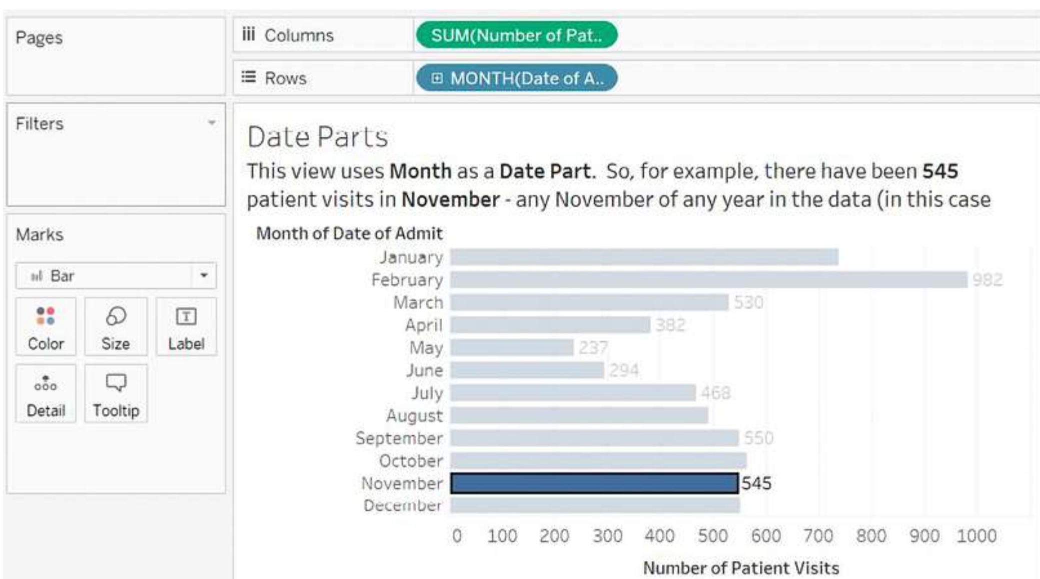


Figure 3.24: This view uses Month as a date part. The number of patient visits is the total for the month, without regard to the year

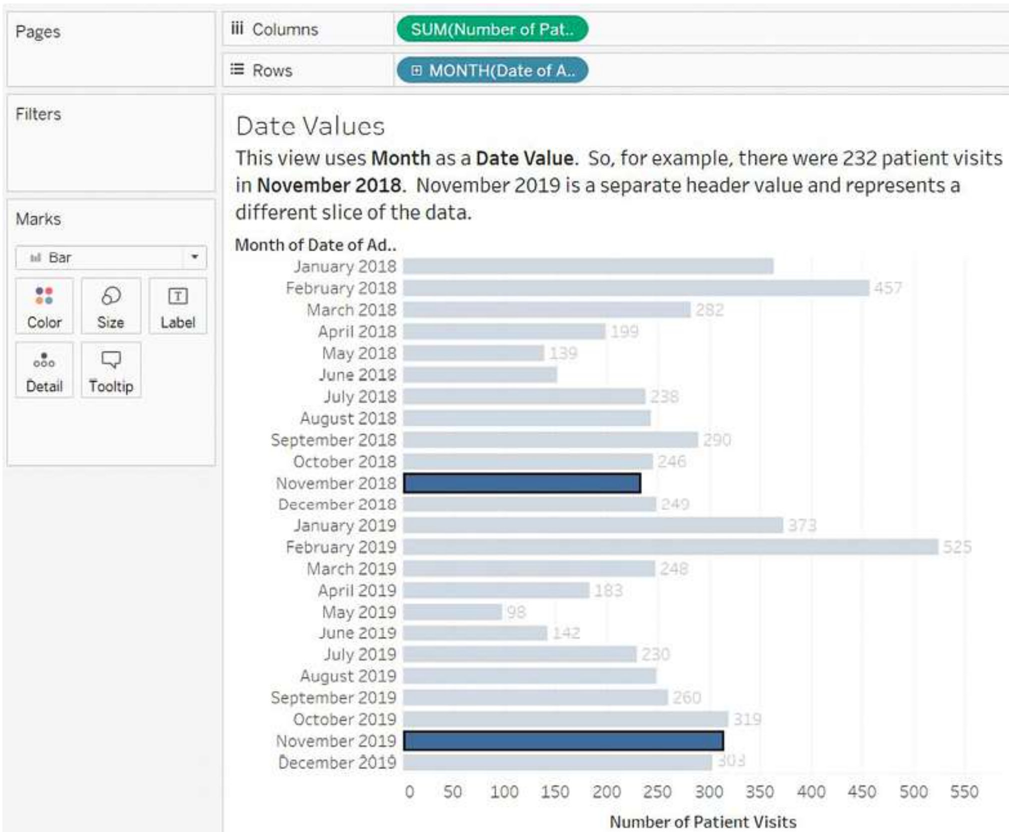


Figure 3.25: This view uses Month as a date value. The number of patient visits is the total for the month with regard to the year

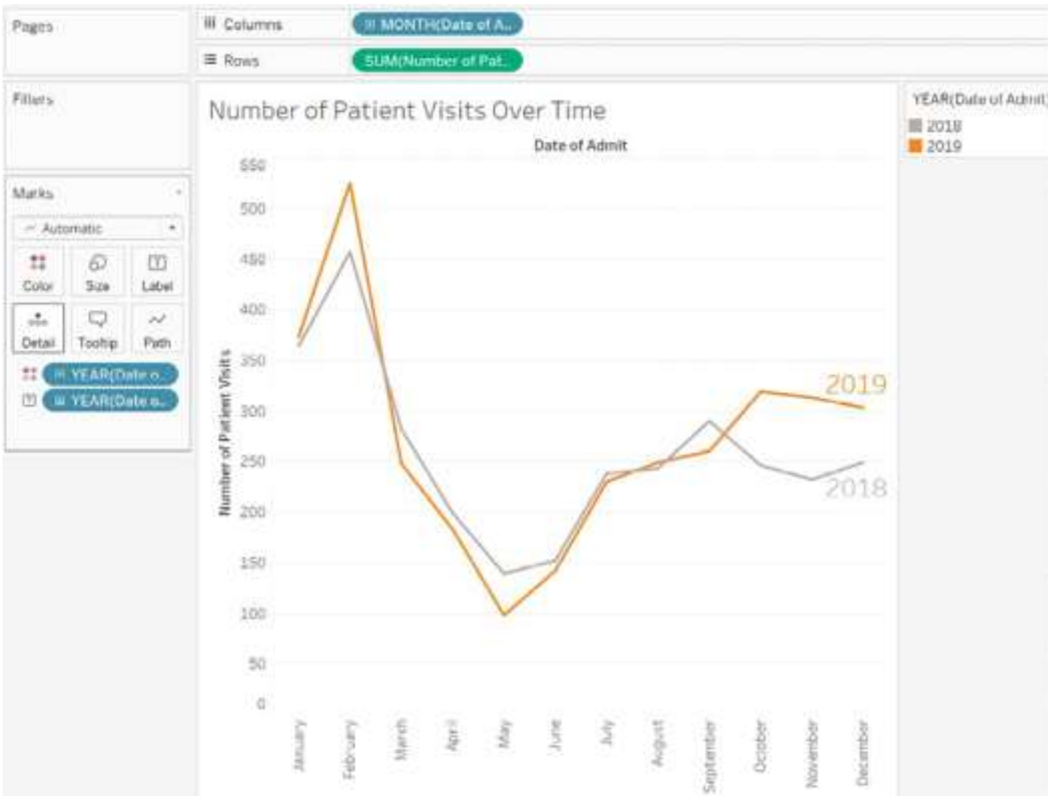


Figure 3.26: The comparison of two years, month by month

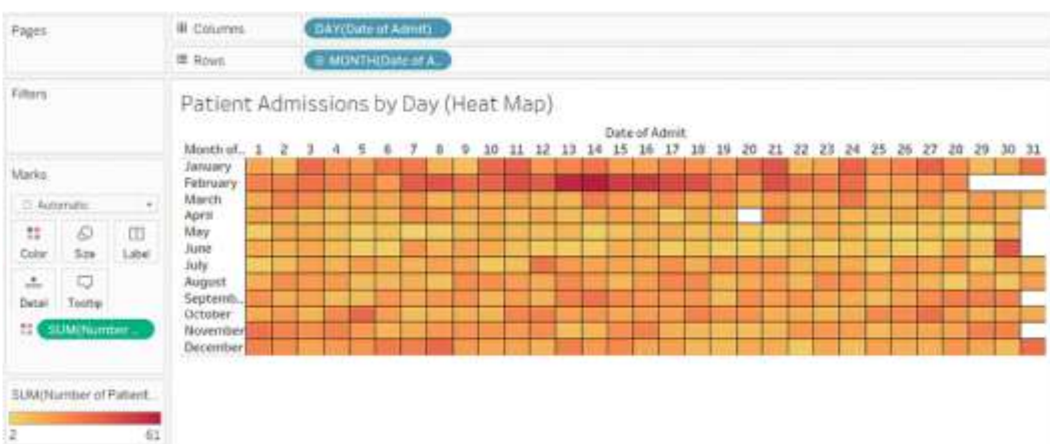


Figure 3.27: A heat map showing the intensity of patient visits by day and month

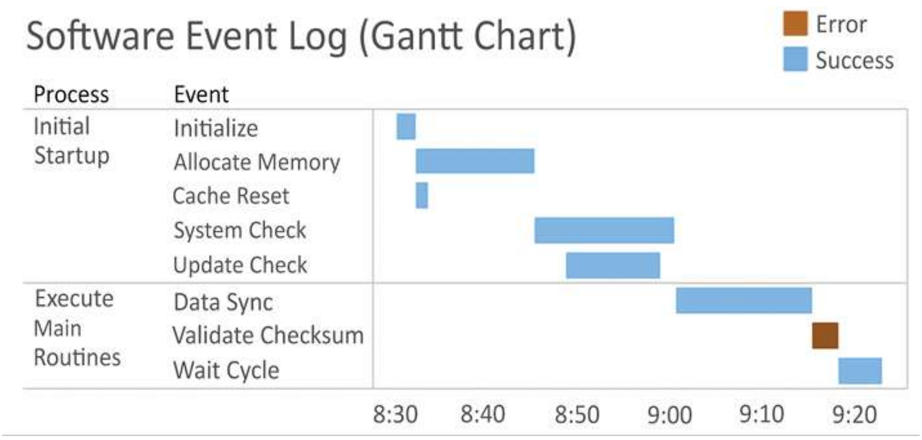


Figure 3.28: A Gantt chart showing the time each process started and how long each took

Figure 3.29: In this case, Gantt bars are the automatic mark type

Figure 3.30: The final Gantt chart, showing each patient, when they were admitted, how long they stayed, and whether they ever returned

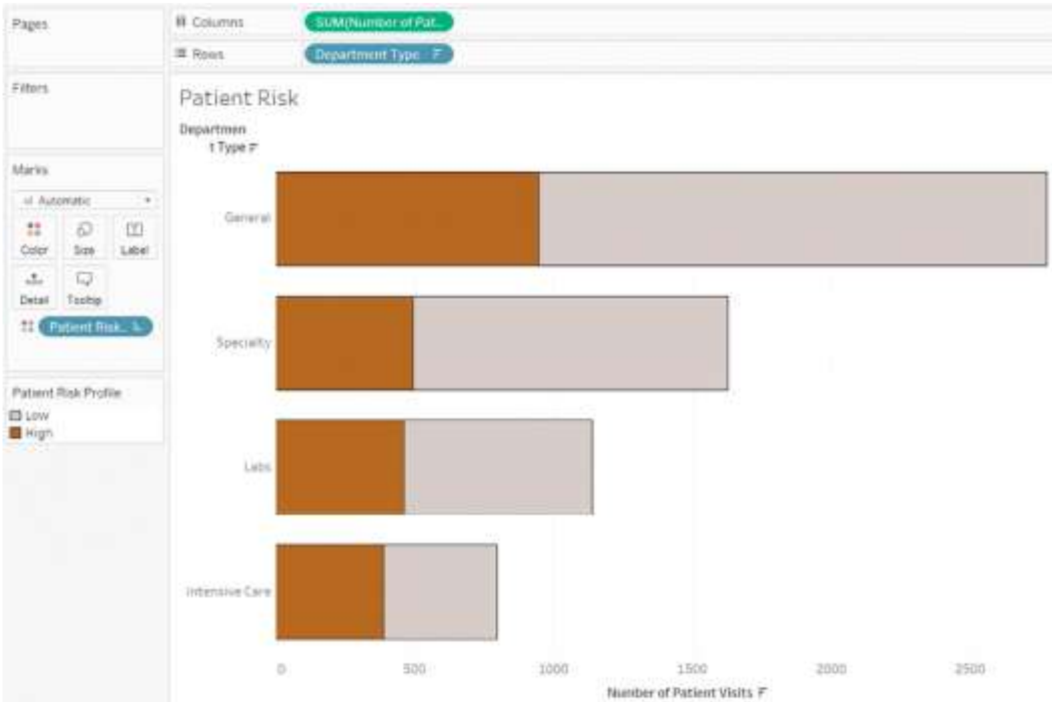


Figure 3.31: A stacked bar chart showing the total number of patients per department and the breakdown of low and high risk

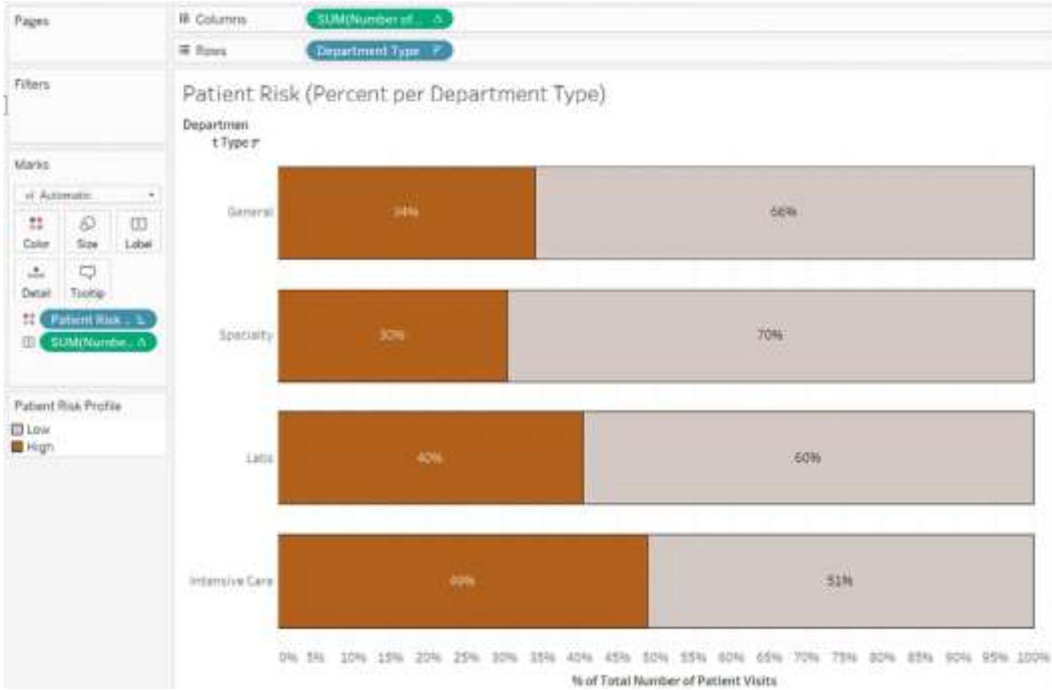


Figure 3.32: A stacked bar chart showing the relative number of high-risk and low-risk patients per department

Figure 3.33: An interim step when creating the stacked bars

Figure 3.34: This toolbar button toggles text labels on/off

Figure 3.35: The final stacked bar view with absolute and relative values

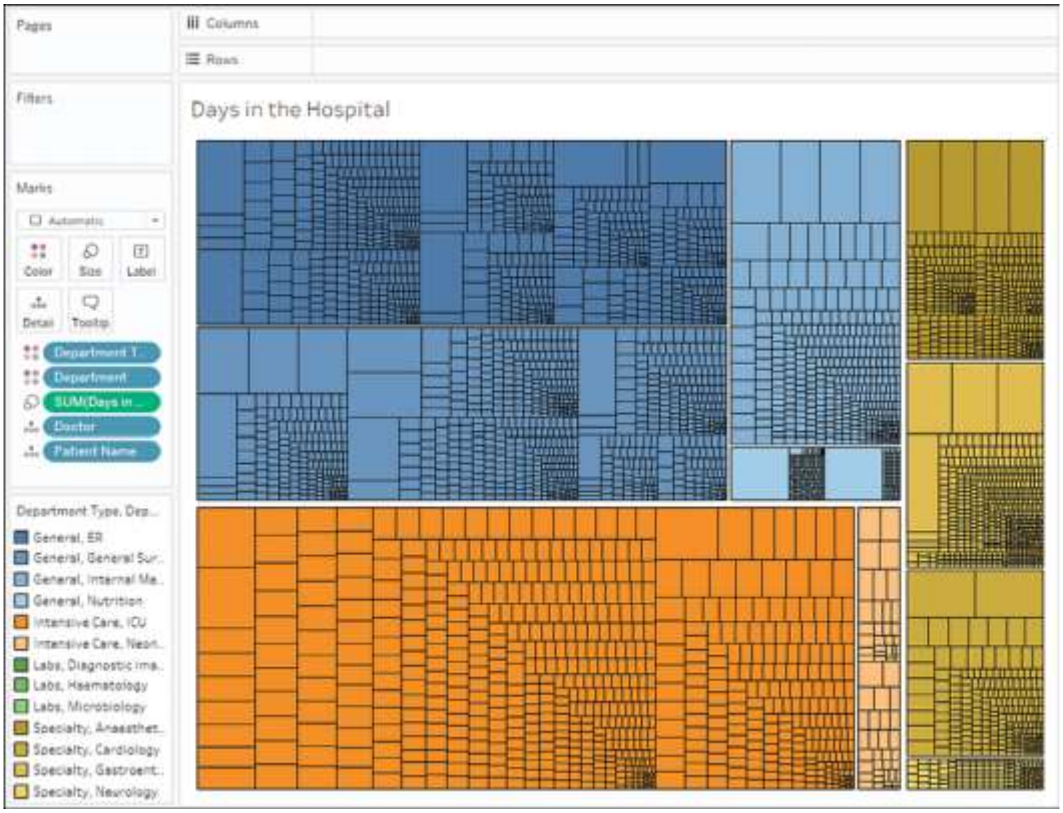


Figure 3.36: A treemap showing the part-to-whole relationship of Department Types/Departments/Doctors/Patients

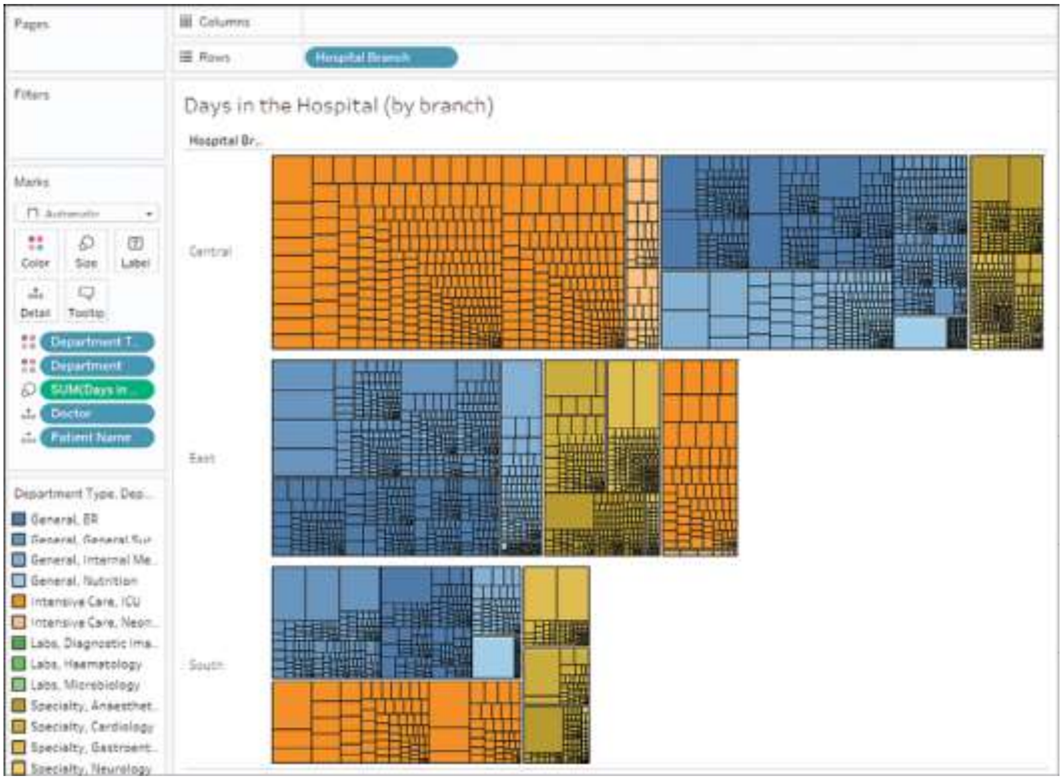


Figure 3.37: Adding a dimension to Rows has effectively made a bar chart of treemaps

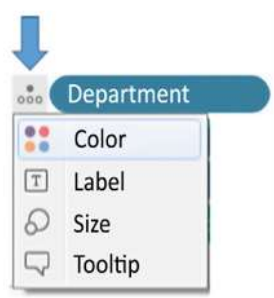


Figure 3.38: Clicking the icon next to a field on the Marks card allows you to change which shelf is used

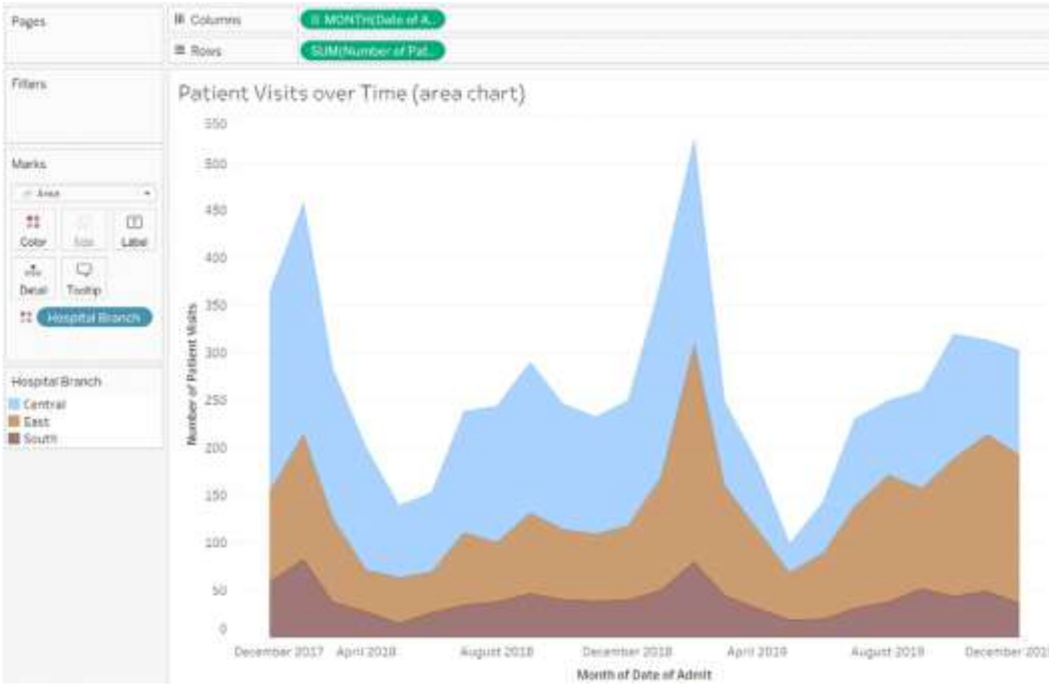


Figure 3.39: An area chart showing patient visits over time by hospital branch

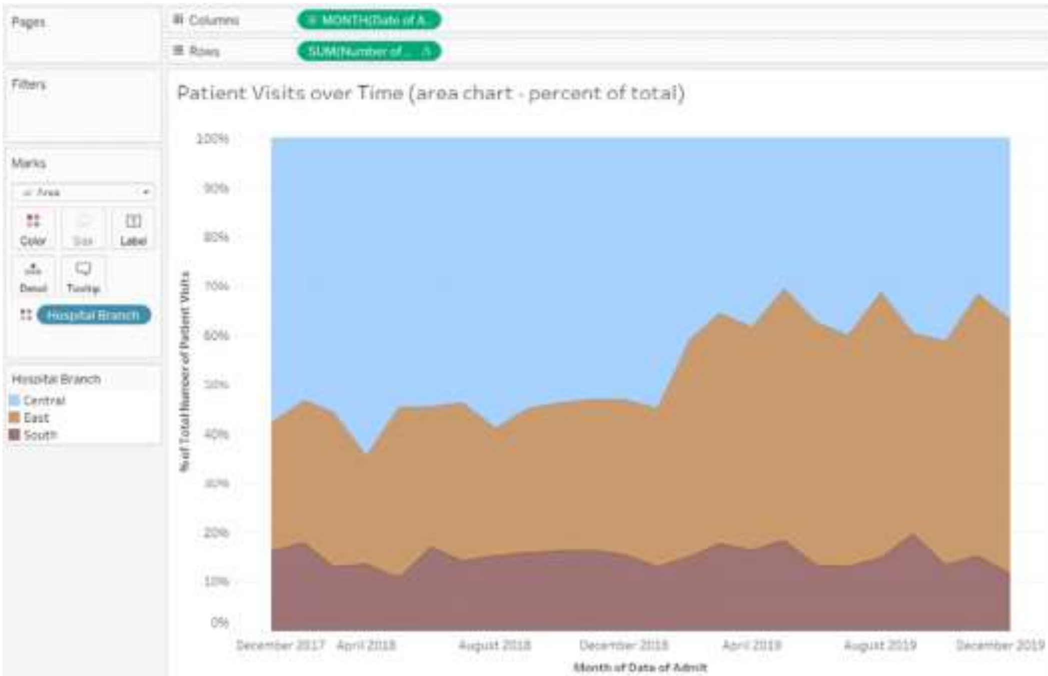


Figure 3.40: An area chart showing percentages instead of absolute values

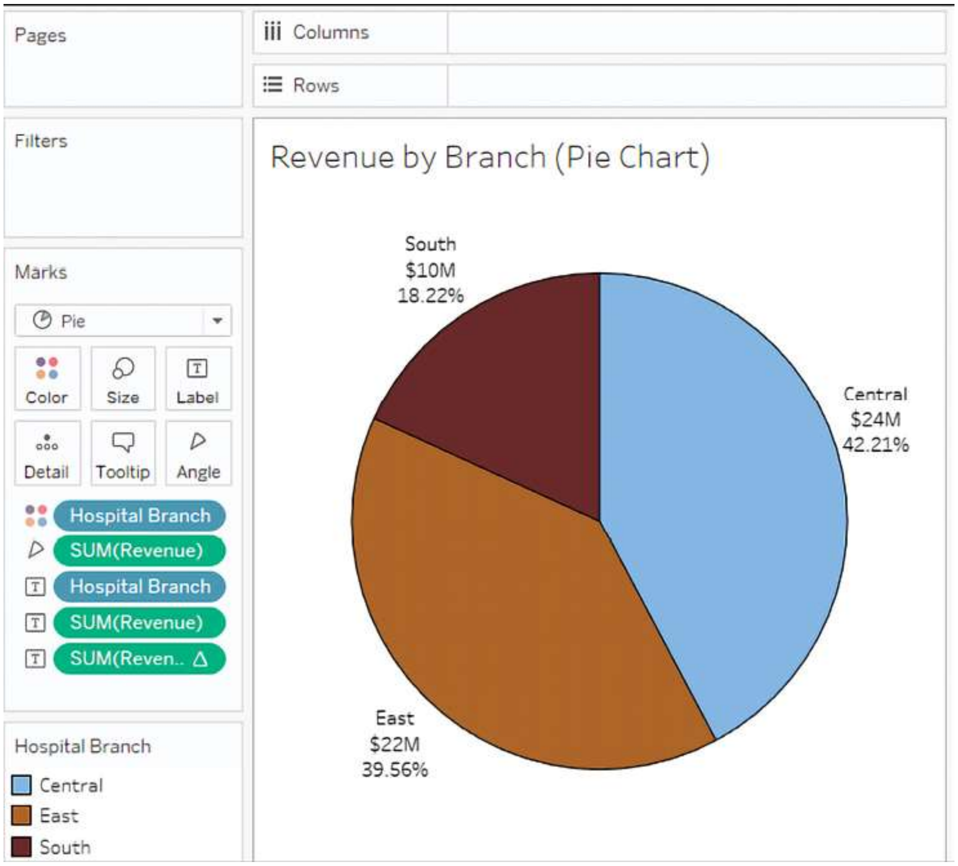


Figure 3.41: A pie chart showing total revenue broken down by branch

Figure 3.42: A circle chart showing the average length of stay for each doctor within each department type

Figure 3.43: You can add reference lines and more by dragging from the Analytics tab to the view

Figure 3.44: Here, INDEX() has been added as a continuous field on Rows (the table calculation is computed along Doctor)

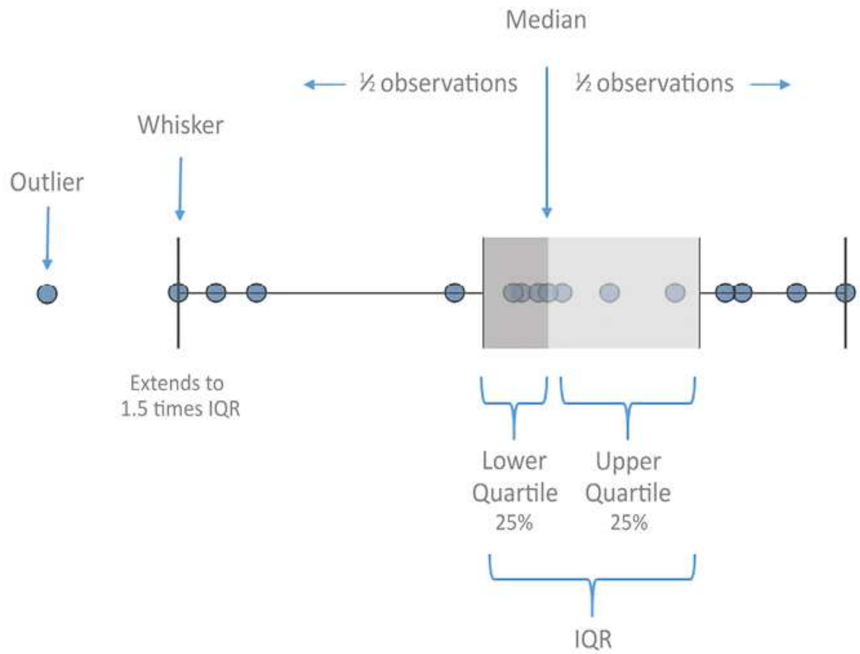


Figure 3.45: Explanation of a box and whisker plot

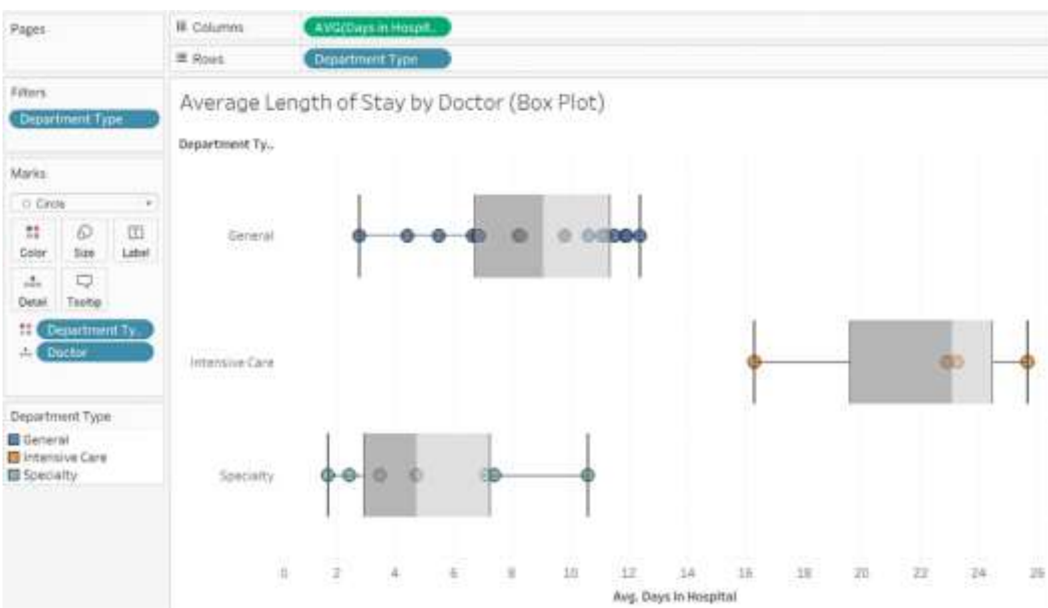


Figure 3.46: A box plot applied to the previous circle chart

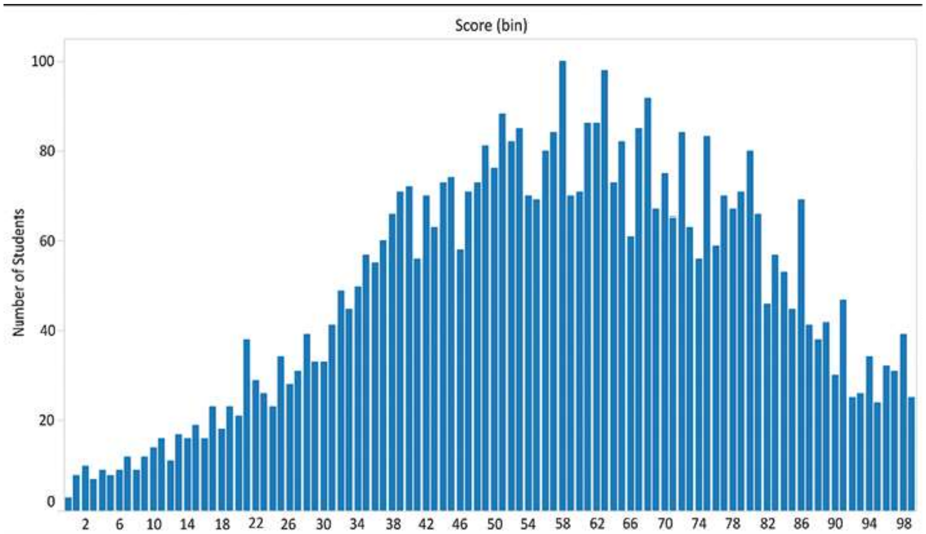


Figure 3.47: A histogram of test scores

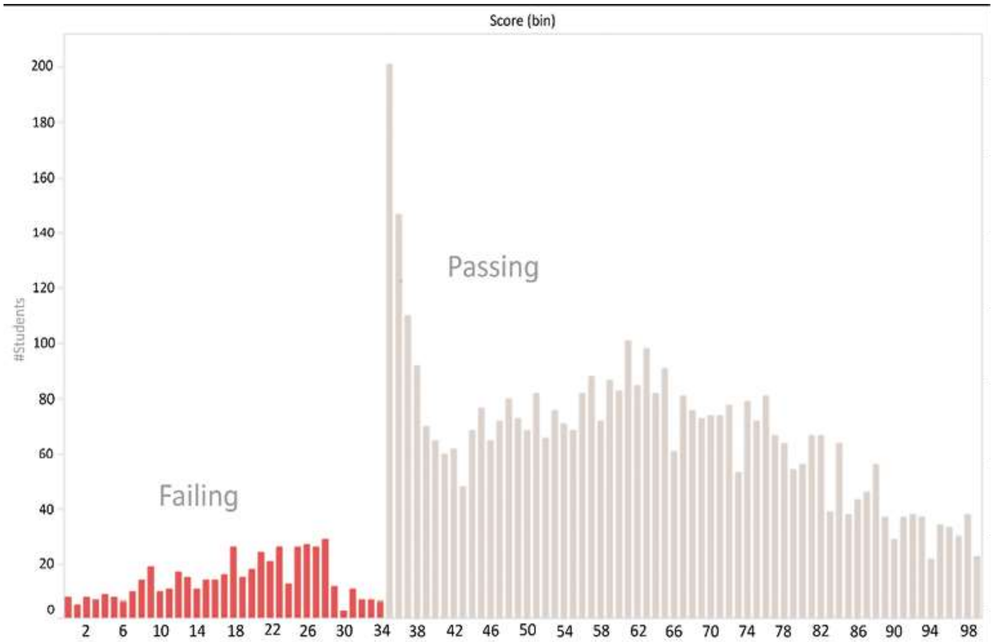


Figure 3.48: A histogram that does not have a typical bell curve, raising some questions

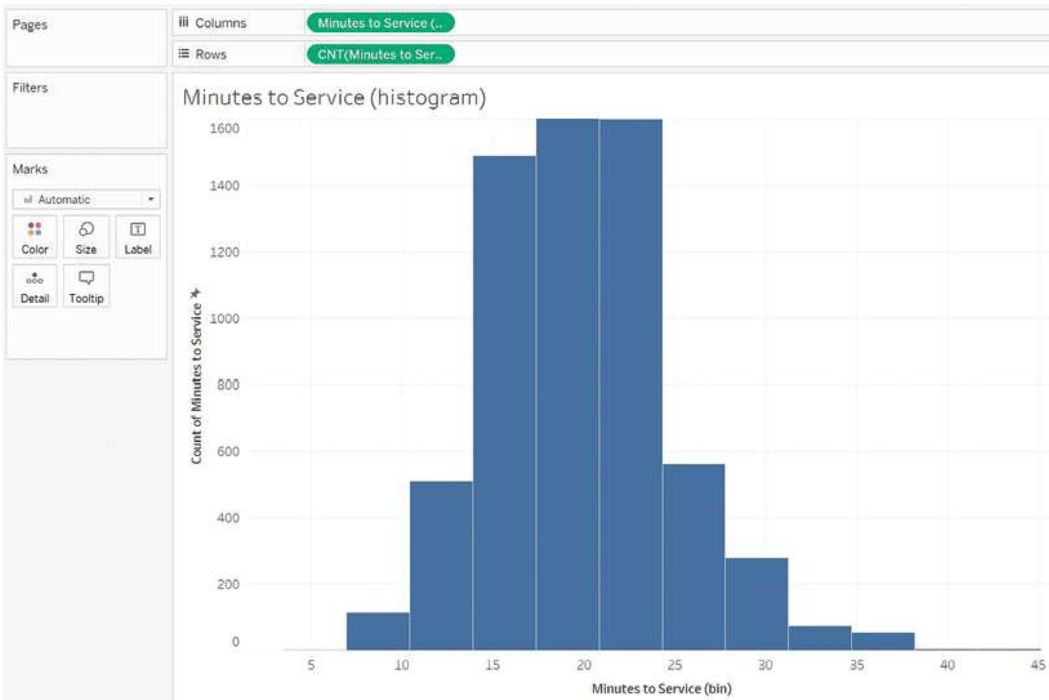


Figure 3.49: A histogram showing the distribution of patients according to minutes to service

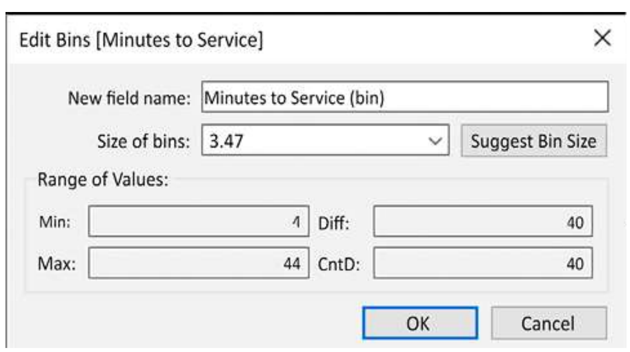


Figure 3.50: Options for editing a bin

Figure 3.51: A histogram with a bin size of 2

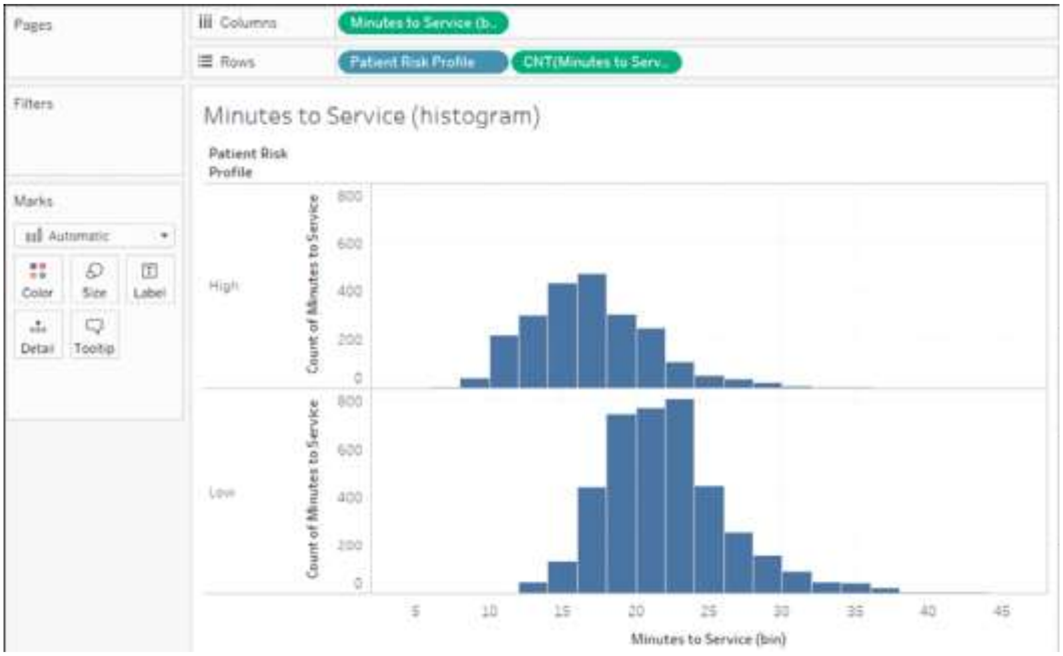


Figure 3.52: Patient risk profile creates two rows of histograms, showing that most high-risk patients receive faster care (as we would hope)

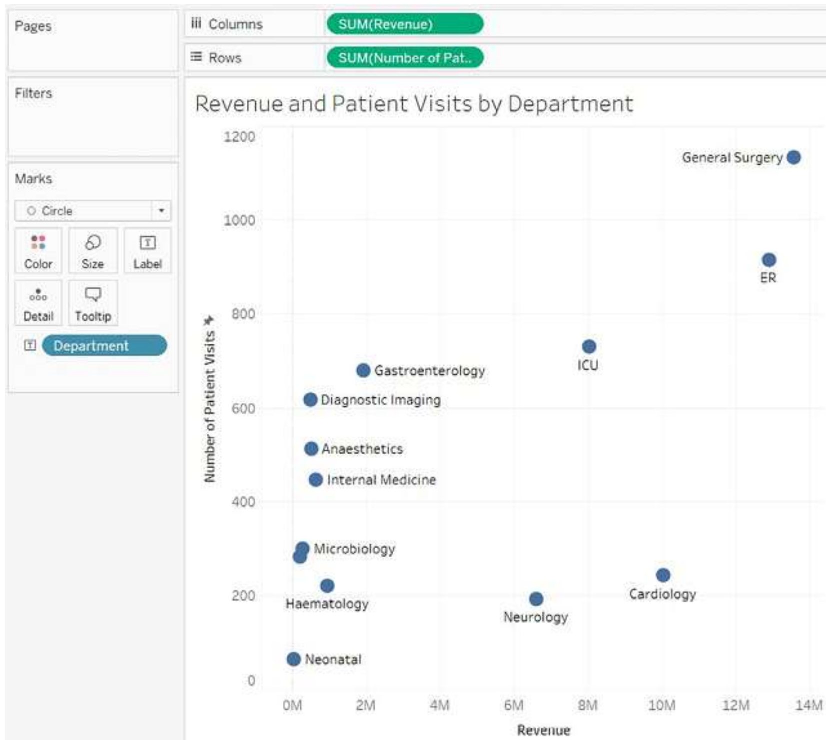


Figure 3.53: A scatterplot showing the correlation between Revenue and Number of Patient Visits

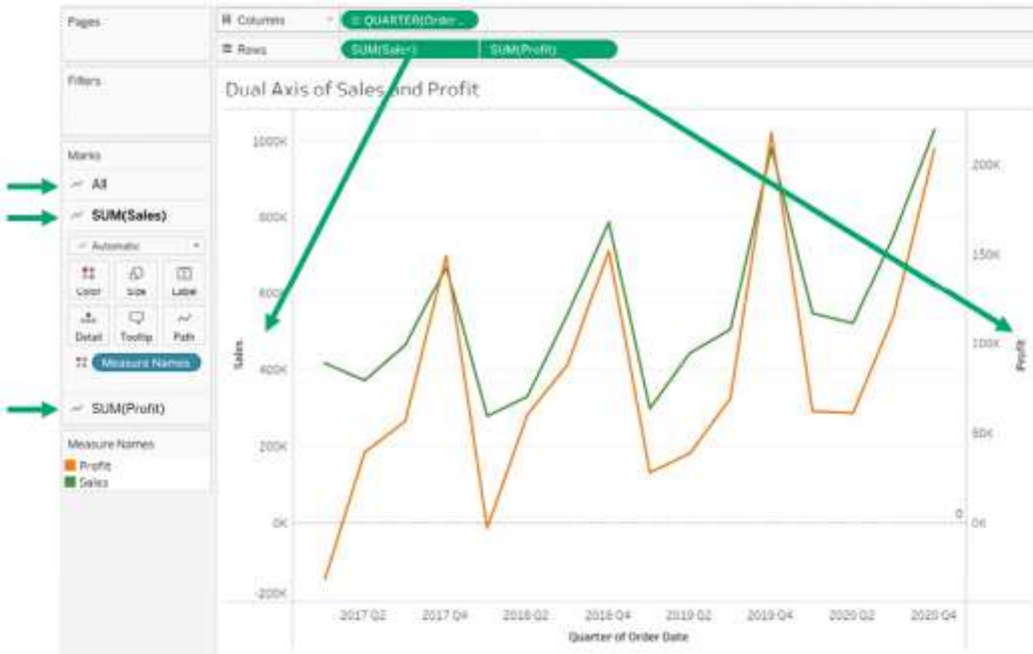


Figure 3.54: A dual-axis chart with an indication of which field defines which axis

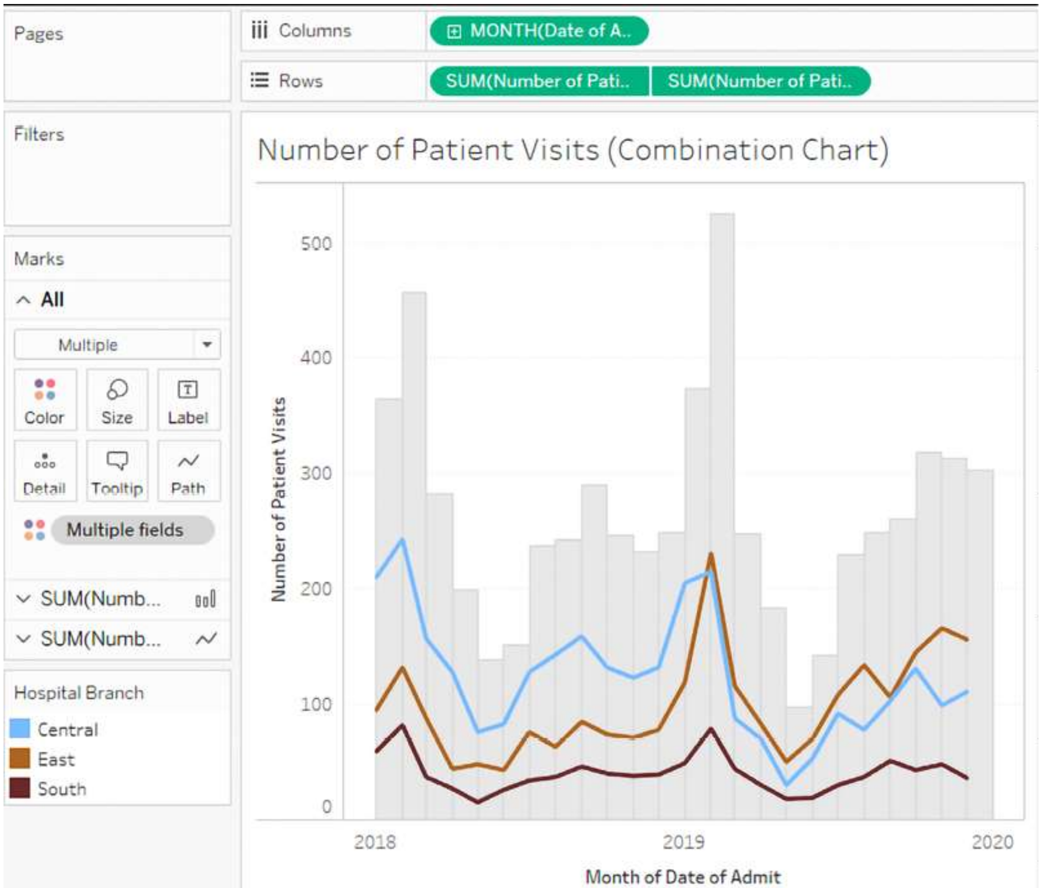


Figure 3.55: A combination chart that shows individual branch visits as lines and total visits as bars

Code

Code 3.1

```
SUM([Revenue]) >= SUM([Goal])
```

Code 3.2

```
DATEDIFF('day', [Date of Admit], [Date of Discharge])
```

Code 3.3

```
DATEDIFF('second', [Start Date], [End Date]) / 86400.
```

Chapter 4

Figures

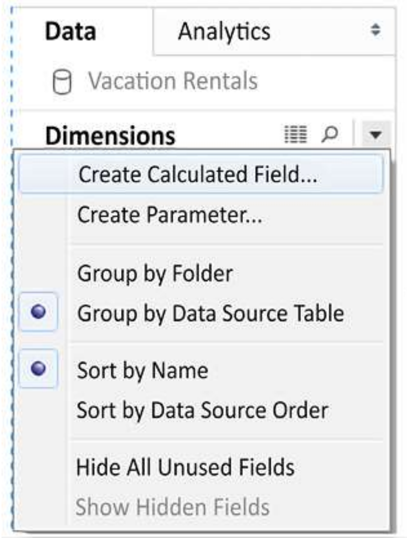


Figure 4.1: The Create Calculated Field... option

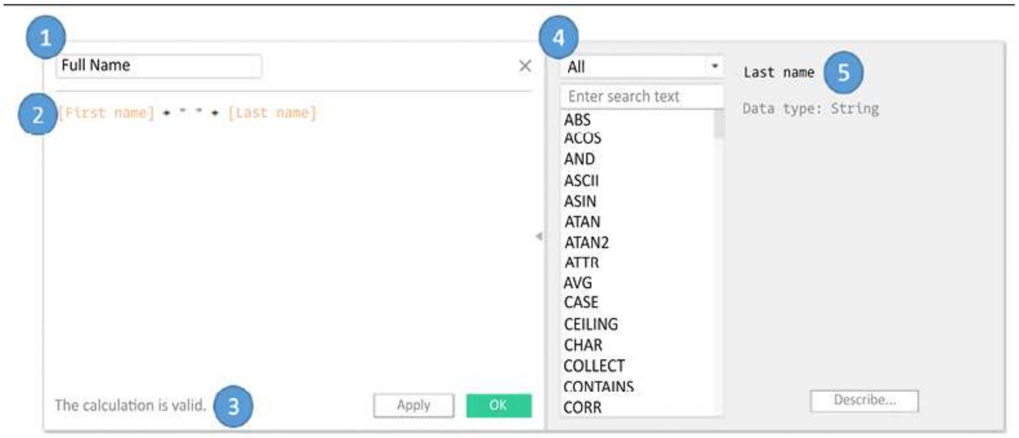


Figure 4.2: The creating and editing calculations interface

Figure 4.3: Creating the Full Name calculation in the editor

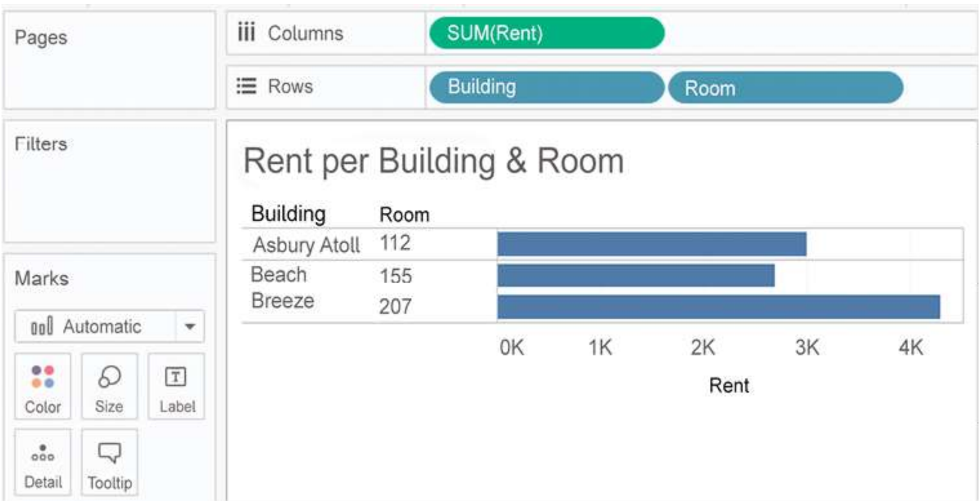


Figure 4.4: Using your calculated fields to build a view

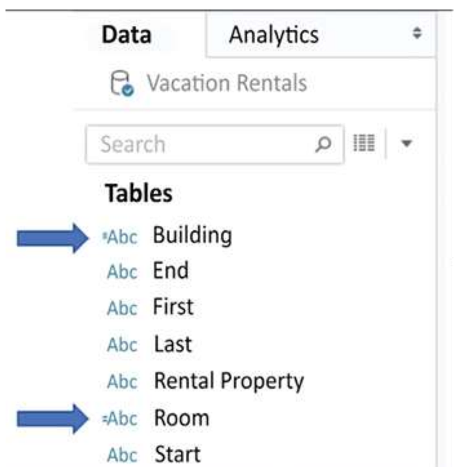


Figure 4.5: The small = sign indicates a field is a calculation

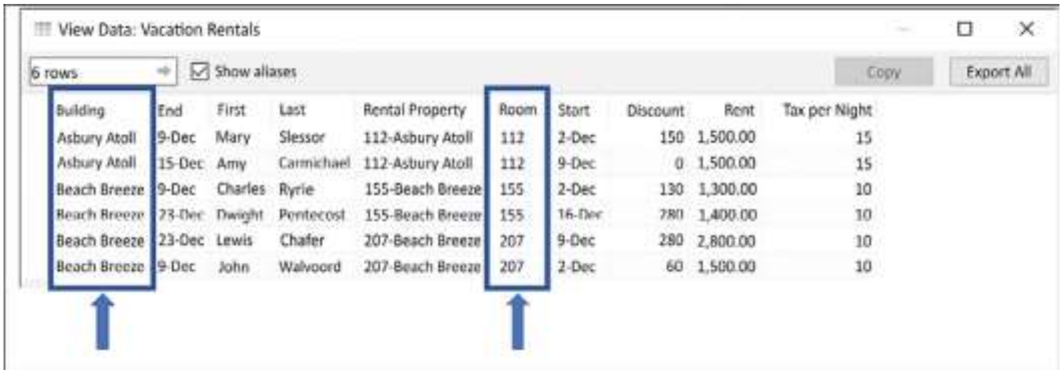


Figure 4.6: Viewing the underlying data shows us the calculation is done per row of data

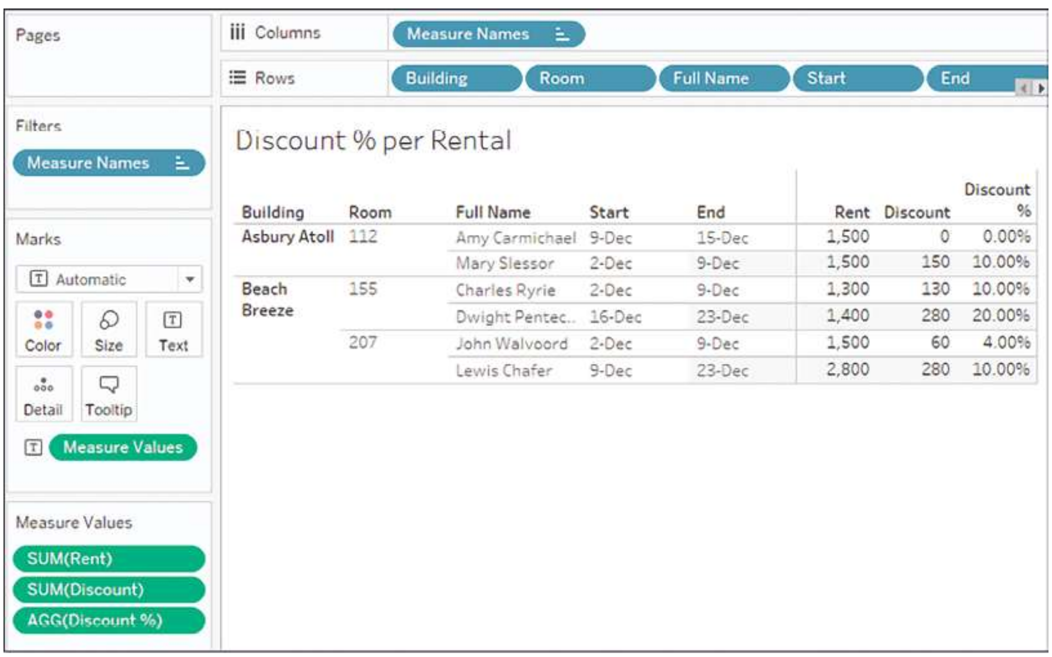


Figure 4.7: Illustrates the Discount % calculated at the level of Building, Room, Full Name, Start, and End

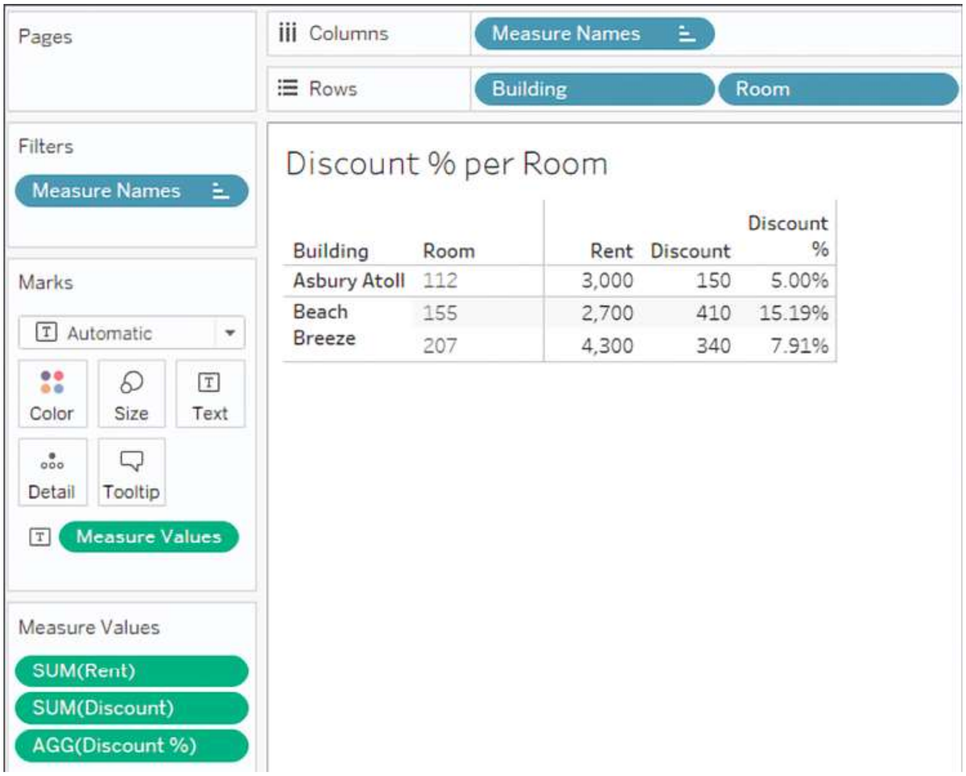


Figure 4.8: Illustrates the Discount % calculated at the level of Building and Room

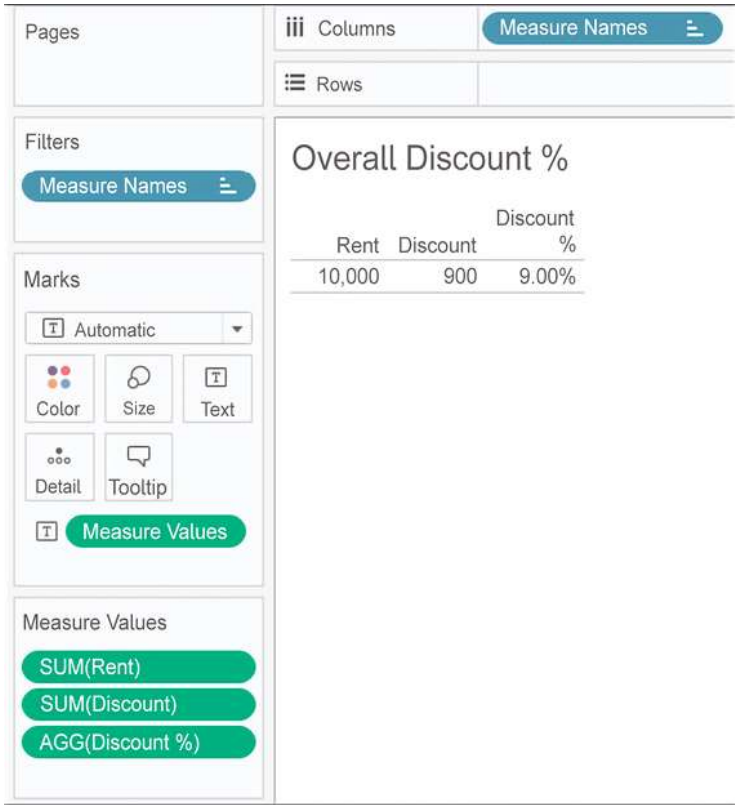


Figure 4.9: Illustrates the Discount % calculated at the highest possible level: the entire dataset

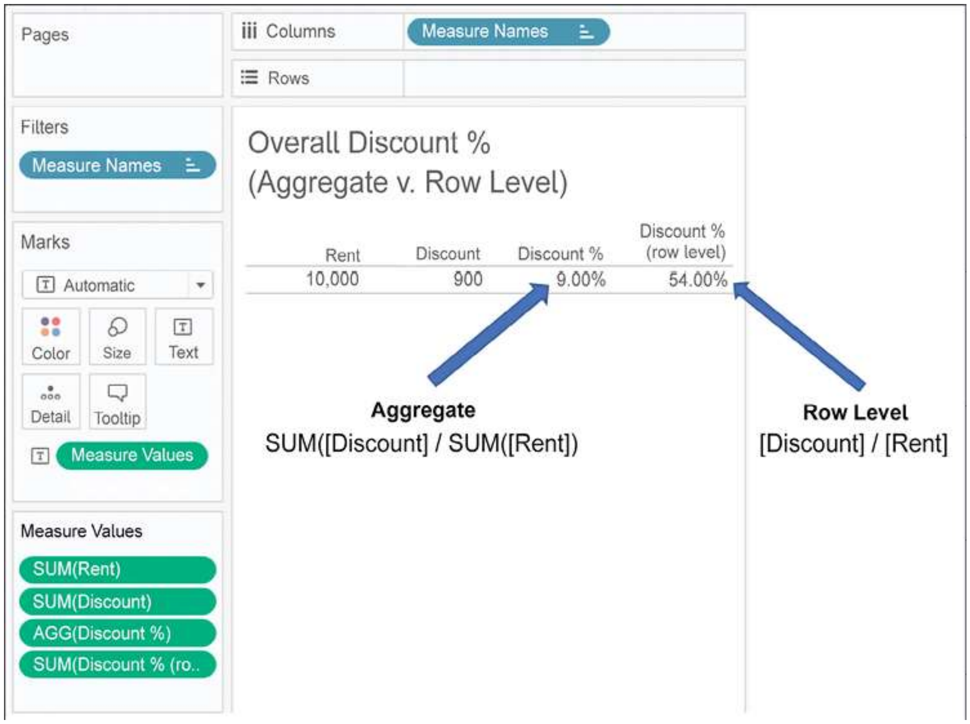


Figure 4.10: Illustrates the Discount % calculated as a row-level value and as an aggregate

Rental Property	First	Discount	Rent	Discount %
112-Asbury Atoll	Mary	150	$1,500	10%
112-Asbury Atoll	Amy	0	$1,500	0%
155-Beach Breeze	Charles	130	$1,300	10%
155-Beach Breeze	Dwight	280	$1,400	20%
207-Beach Breeze	Lewis	280	$2,800	10%
207-Beach Breeze	John	60	$1,500	4%
				54%

Row Level
[Discount] / [Rent]
Results calculated for each row

Final Aggregation
Sum of row level results

Figure 4.11: If each Discount % result is calculated at a row level and then aggregated, the result is wrong

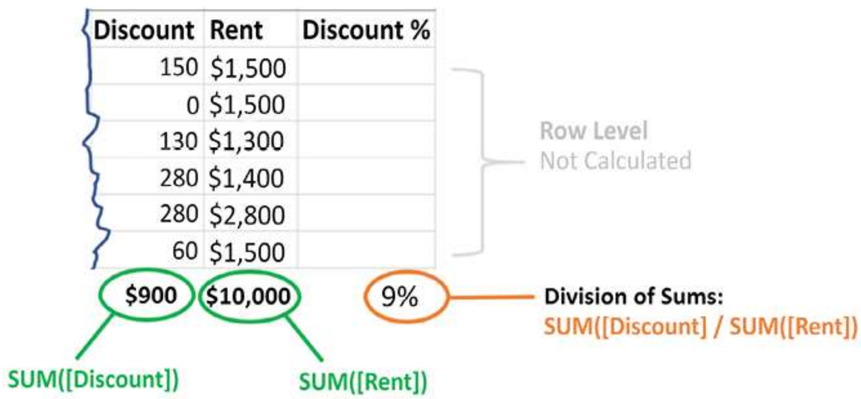


Figure 4.12: If the numerator and denominator are aggregated first, then the Discount % calculation is correct

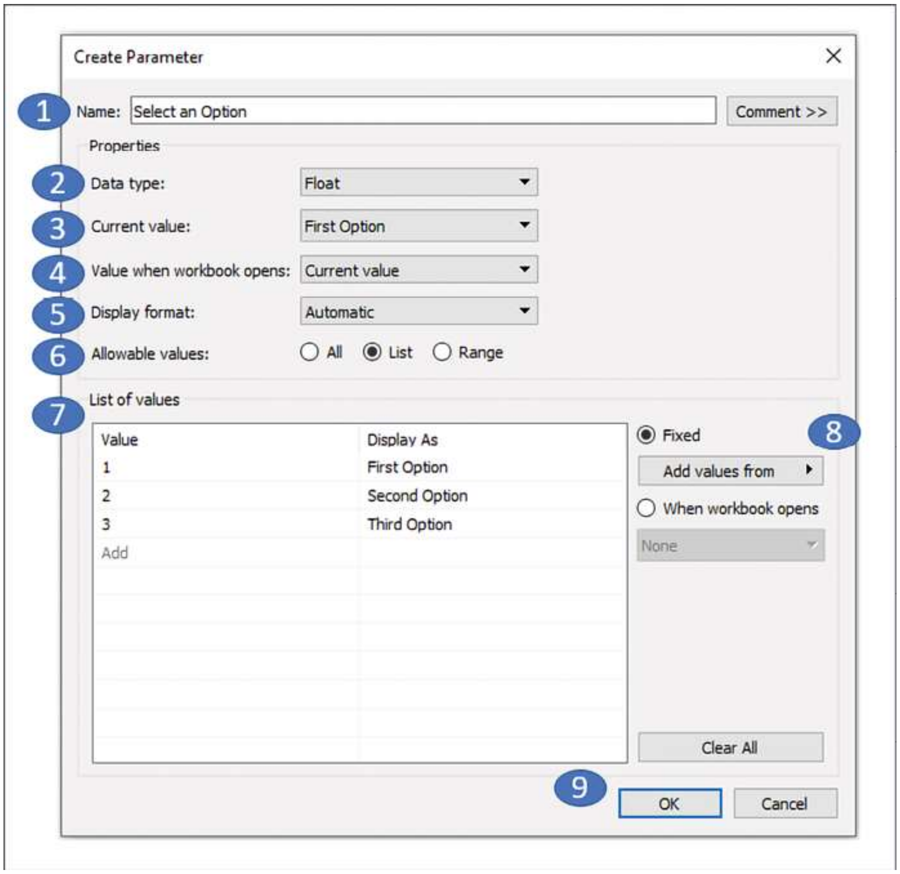


Figure 4.13: The Create Parameter interface numbered with corresponding descriptions below

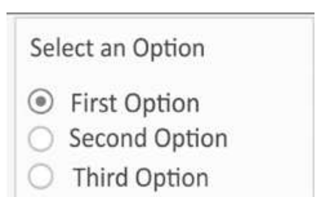


Figure 4.14: The parameter control shown as a single select radio button list

Figure 4.15: The corrected dates appear next to the string versions. All fields are discrete dimensions on Rows (the dates are exact dates)

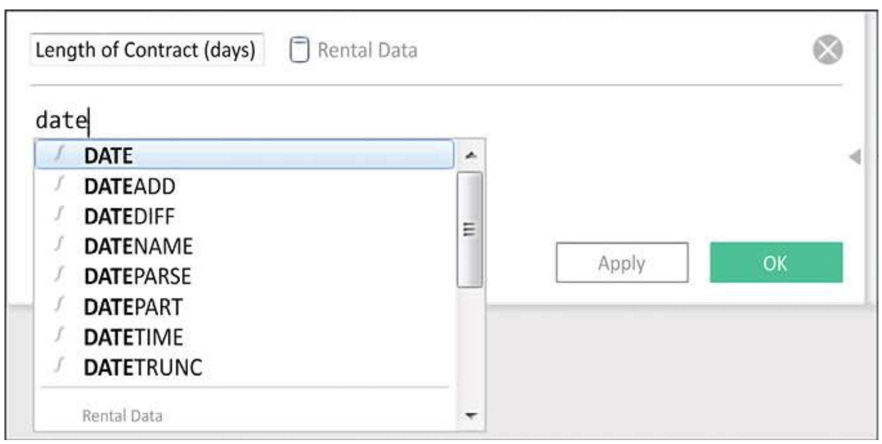


Figure 4.16: The intelligent code completion will suggest possible field names and functions as you type

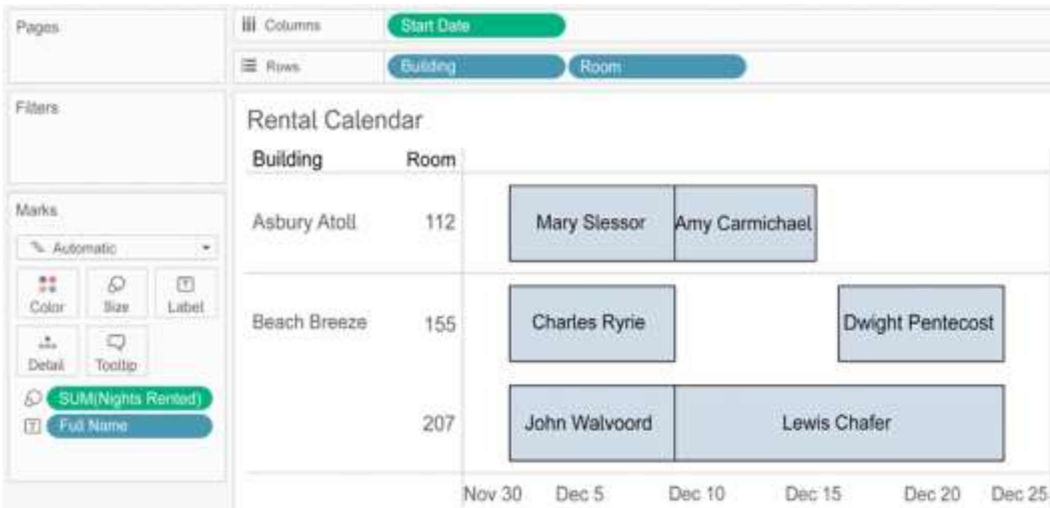


Figure 4.17: The calculated field allows us to create the Gantt chart

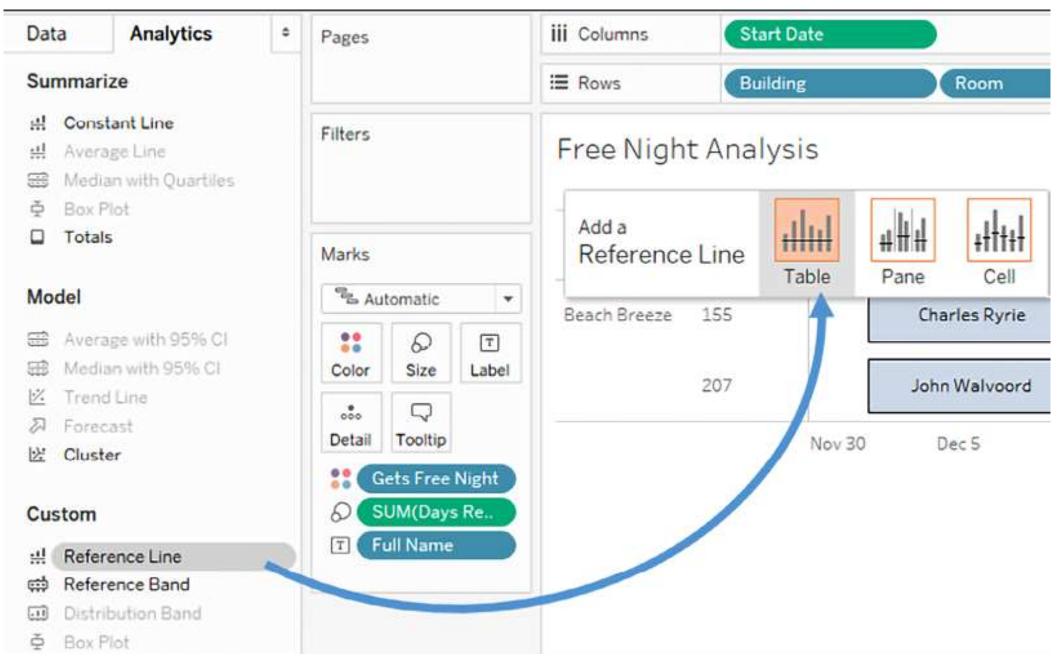


Figure 4.18: Add a reference line by switching to the Analytics pane and dragging the reference line onto the canvas

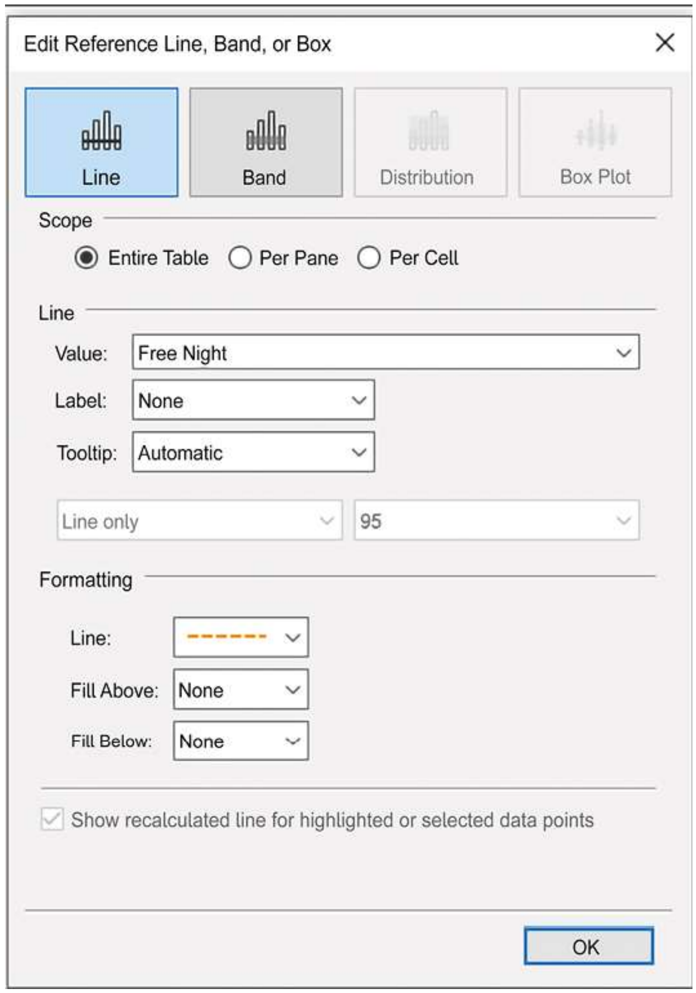


Figure 4.19: Use the Edit Reference Line dialog to adjust the formatting, labels, and tooltips

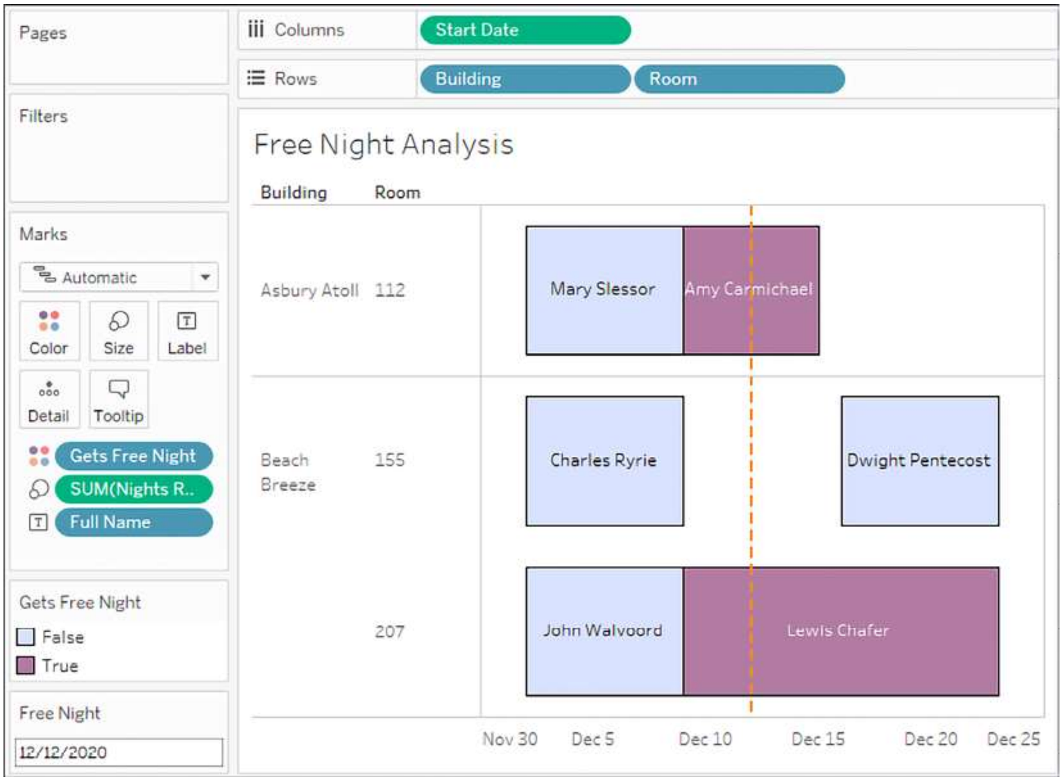


Figure 4.20: The reference line will move, and the affected individuals will be recalculated every time you change the Free Night parameter value

Figure 4.21: The revenue generated from each individual's stay

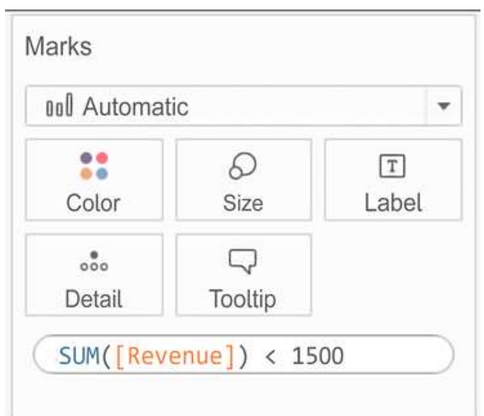


Figure 4.22: Creating an ad hoc calculation on the Marks card

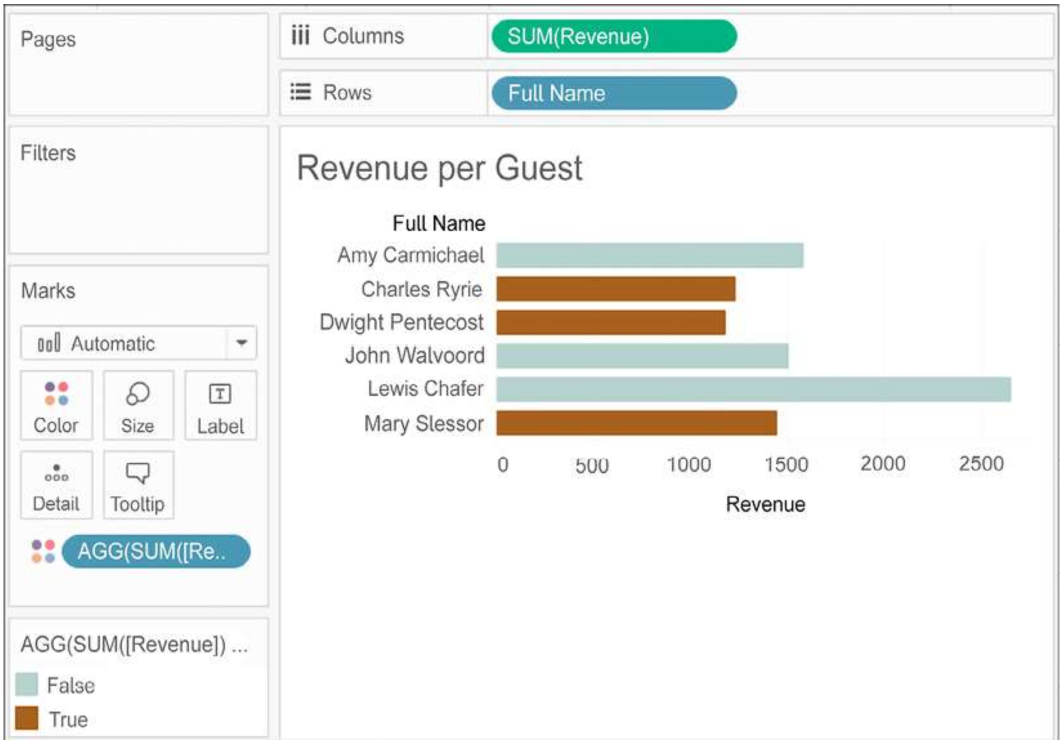


Figure 4.23: Using the ad hoc calculation on Color

Tables

Operator/Keyword	Description
AND	Logical and between two Boolean values or statements
OR	Logical or between two Boolean values or statements
NOT	Logical not to negate a Boolean value or statement

IN	A logical expression that tests for the equality of a value against a list of values and returns true if the value matches one of the items in the list
= or ==	Logical equals to test the equality of two statements or values (single or double equal signs are equivalent in Tableau's syntax)
+	Addition of numeric or date values or the concatenation of strings
-	Subtraction of numeric or date values
*	Multiplication of numeric values
/	Division of numeric values
^	Raise to a power with numeric values
( )	Parentheses to define the order of operations or enclose function arguments
[ ]	Square brackets to enclose field names
{ }	Curly braces to enclose the level of detail calculations
//	Double slash to start a comment

Table 4.1 – Additional functions and operators

Rental Property	First	Last	Start	End	Discount	Rent	Tax per Night
112-Asbury Atoll	Mary	Slessor	Dec 2	Dec 9	150	1,500	15
112-Asbury Atoll	Amy	Carmichael	Dec 9	Dec 15	0	1,500	15
155-Beach Breeze	Charles	Ryrie	Dec 2	Dec 9	260	1,300	10
155-Beach Breeze	Dwight	Pentecost	Dec 16	Dec 23	280	1,400	10
207-Beach Breeze	Lewis	Chafer	Dec 9	Dec 23	280	2,800	10
207-Beach Breeze	John	Walvoord	Dec 2	Dec 9	60	1,500	10

Table 4.2 – Sample dataset for examples

Code

Code 4.1 – Concatenating strings

```
[First] + " " + [Last]
```

Code 4.2 – String manipulation and conditional logic

Name the first Room, and add the following code:

```
SPLIT([Rental Property], "-", 1)
```

Then, create another calculated field named Building with the following code:

```
SPLIT([Rental Property], "-", 2)
```

Code 4.3 – Row-level calculation

```
IF LEFT([Room], 1) = "1"
THEN "First Floor"
ELSEIF LEFT([Room], 1) = "2"
THEN "Second Floor"
END
```

Code 4.4 – Planning data variations

```
LEFT([Room], 1)
```

Code 4.5 – Planning data variations, additional cases

```
MID([Room], 0, LEN([Room]) - 2)
```

Code 4.6 – % discount

```
SUM([Discount]) / SUM([Rent])
```

Code 4.7 –

```
[Discount]/[Rent]
```

Code 4.8 – To get the start date

```
DATE([Start] + ", 2020")
```

Code 4.9 – To get the end date

```
DATE([End] + ", 2020")
```

Code 4.10 – Create a calculated field named Nights Rented with the following code:

```
DATEDIFF('day', [Start Date], [End Date])
```

Code 4.11 – Create a calculated field called Gets free night

```
[Free Night] >= [Start Date]
AND
[Free Night] <= [End Date]
```

Code 4.12 – Analyzing Total Revenue

```
[Rent] - [Discount] + ([Tax per Night] * [Nights Rented])
```

Code 4.13 – Simplifying the code

```
//This is potentially less efficient...
IF [Type] = "Dog" AND [Age] < 1 THEN "Puppy"
ELSEIF [Type] = "Cat" AND [Age] < 1 THEN "Kitten"
END
//...than this code:
IF [Age] < 1 THEN
    IF [Type] = "Dog" THEN "Puppy"
    ELSEIF [Type] = "Cat" THEN "Kitten"
```

```
    END
END
```

Links

A great place to find help and suggestions for calculations is the official Tableau forums at https://community.tableau.com/s/explore-forums.

Chapter 5

Tables

Customer	State	Membership Date	Membership Level	Orders
Neil	Kansas	2009-05-05	Silver	1
Jeane	Kansas	2012-03-17	Gold	5
George	Oklahoma	2016-02-01	Gold	10
Wilma	Texas	2018-09-17	Silver	4

Table 5.1 – Sample dataset

Date	Portfolio	Loan Type	Balance	Open Date	Member Name	Credit Score	Age	State
3/1/2020	Auto	New Auto	15987	9/29/2018	Samuel	678	37	California

7/1/2020	Mortgage	1st Mortgage	96364	8/7/2013	Lloyd	768	62	Ohio
3/1/2020	Mortgage	HELOC	15123	4/2/2013	Inez	751	66	Illinois
3/1/2020	Mortgage	1st Mortgage	418635	9/30/2015	Patrick	766	60	Ohio
5/1/2020	Auto	Used Auto	1151	10/22/2018	Eric	660	44	Pennsylvania
…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…

Table 5.2 – Example data set of fixed level of detail calculations (Chapter 05 Loans data set contained in the Chapter 05 workbook)

Figures

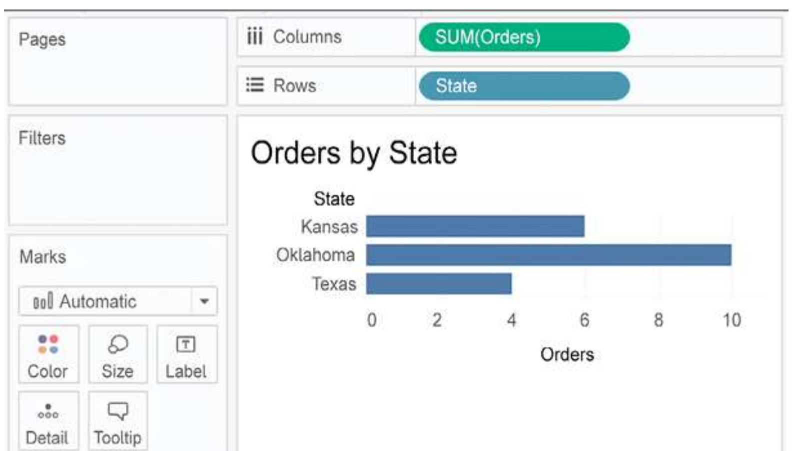


Figure 5.1: The view level of detail of state

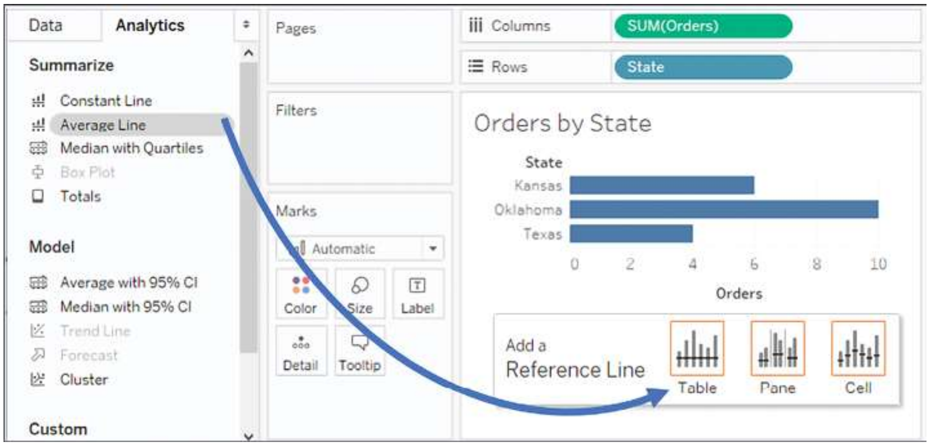


Figure 5.2: Adding an average line to the view

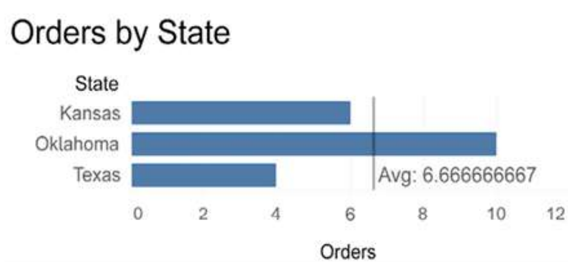


Figure 5.3: The overall average is reported as 6.66667. This is the average per state

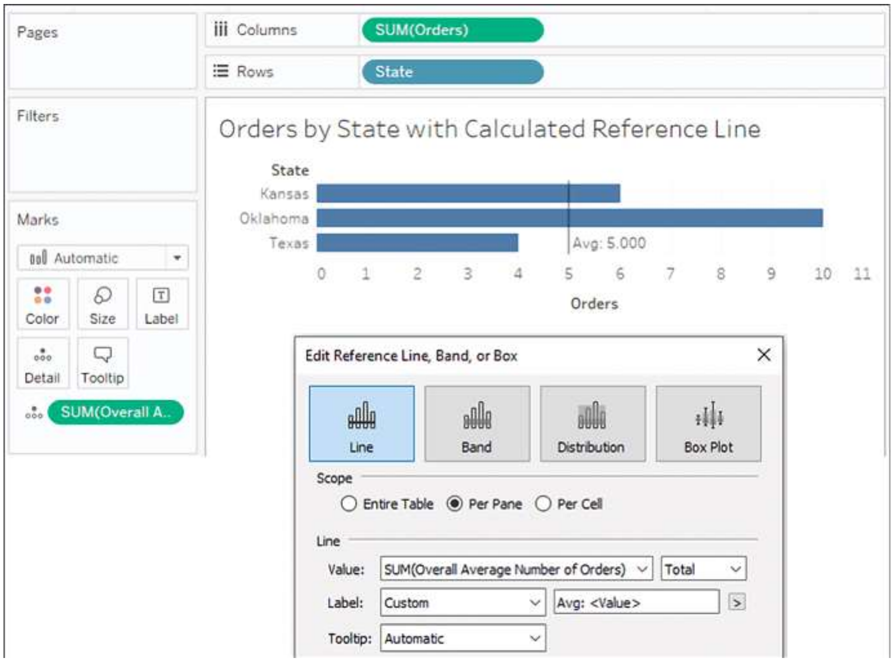


Figure 5.4: The true overall average number of orders per customer is 5

Member ID	Member Name	Loan Type	Date	Balance	Credit Score
158	Vicki Modzelewski	Used Auto	1/1/2020	10,615	712
			2/1/2020	10,441	712
			3/1/2020	10,285	699
			4/1/2020	10,108	699
			5/1/2020	9,891	699
			6/1/2020	9,736	717
			7/1/2020	9,556	717
479	Thomas Villareal	Used Auto	2/1/2020	7,407	526
			3/1/2020	7,191	526
			4/1/2020	6,984	563
			5/1/2020	6,771	563
			6/1/2020	6,551	563
			7/1/2020	6,334	591
			8/1/2020	6,115	591
576	Charles Reeves	Used Auto	1/1/2020	28,145	610
			2/1/2020	27,187	610
			3/1/2020	26,226	535
			4/1/2020	25,267	535
			5/1/2020	24,302	535
			6/1/2020	23,337	530
			7/1/2020	22,366	530

Figure 5.5: Credit scores for three individuals with scores under the 550 threshold indicated via arrows

FIXED Example 1: Member Ever at Risk?

Member ID	Member Name	Loan Type	Date	Member Ever at Risk?	Balance	Credit Score
158	Vicki Modzelewski	Used Auto	1/1/2020	False	10,615	712
			2/1/2020	False	10,441	712
			3/1/2020	False	10,285	699
			4/1/2020	False	10,108	699
			5/1/2020	False	9,891	699
			6/1/2020	False	9,736	717
			7/1/2020	False	9,556	717
479	Thomas Villareal	Used Auto	2/1/2020	True	7,407	526
			3/1/2020	True	7,191	526
			4/1/2020	True	6,984	563
			5/1/2020	True	6,771	563
			6/1/2020	True	6,551	563
			7/1/2020	True	6,334	591
			8/1/2020	True	6,115	591
576	Charles Reeves	Used Auto	1/1/2020	True	28,145	610
			2/1/2020	True	27,187	610
			3/1/2020	True	26,226	535
			4/1/2020	True	25,267	535
			5/1/2020	True	24,302	535
			6/1/2020	True	23,337	530
			7/1/2020	True	22,366	530

Figure 5.6: The Member Ever at Risk? field is True or False for all records of a given member

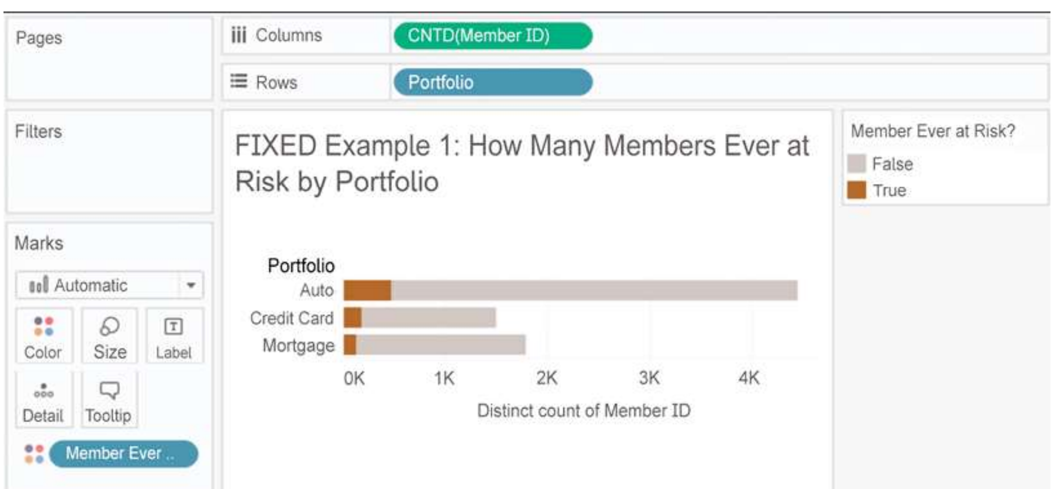


Figure 5.7: We can implement some brushing to show what proportion of members has ever been at risk

Member ID	Member Name	Loan Number	Loan Type	Date	
14827	Kelly Wooldridge	1	New Auto Plus	1/1/2020	21,684
				2/1/2020	21,348
				3/1/2020	21,001
				4/1/2020	21,001
				5/1/2020	20,327
				6/1/2020	19,987
				7/1/2020	19,646
16024	Joseph Clark	1	Used Auto	2/1/2020	19,043
				3/1/2020	18,656
				4/1/2020	18,263
				5/1/2020	17,873
				6/1/2020	17,479
				7/1/2020	17,087
				8/1/2020	16,691
16070	Gerald Quinney	1	1st Mortgage	3/1/2020	144,138
				4/1/2020	140,943
				5/1/2020	137,737
				6/1/2020	134,520
				7/1/2020	131,293
		2	Used Auto	3/1/2020	6,809
				4/1/2020	6,636
				5/1/2020	6,460
				6/1/2020	6,285
				7/1/2020	6,107
				8/1/2020	5,929
				9/1/2020	5,749

Figure 5.8: The data for three selected members in the Loans data set

FIXED Example 2: Most Recent Balance

Member ID	Member Name	Loan Number	Loan Type	Date	Latest Date per Member/ Loan	
14827	Kelly Wooldridge	1	New Auto Plus	1/1/2020	False	21,684
				2/1/2020	False	21,348
				3/1/2020	False	21,001
				4/1/2020	False	21,001
				5/1/2020	False	20,327
				6/1/2020	False	19,987
				7/1/2020	True	19,646
16024	Joseph Clark	1	Used Auto	2/1/2020	False	19,043
				3/1/2020	False	18,656
				4/1/2020	False	18,263
				5/1/2020	False	17,873
				6/1/2020	False	17,479
				7/1/2020	False	17,087
				8/1/2020	True	16,691
16070	Gerald Quinney	1	1st Mortgage	3/1/2020	False	144,138
				4/1/2020	False	140,943
				5/1/2020	False	137,737
				6/1/2020	False	134,520
				7/1/2020	True	131,293
		2	Used Auto	3/1/2020	False	6,809
				4/1/2020	False	6,636
				5/1/2020	False	6,460
				6/1/2020	False	6,285
				7/1/2020	False	6,107
				8/1/2020	False	5,929
				9/1/2020	True	5,749

Figure 5.9: The latest date per loan per person is indicated by a True value for the calculation

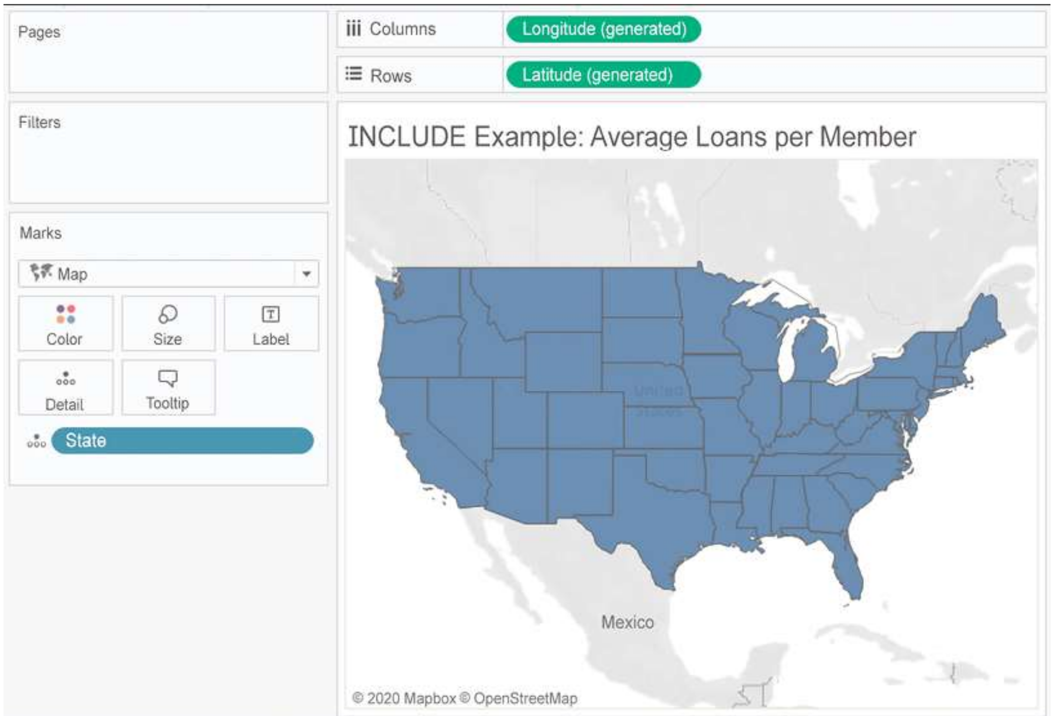


Figure 5.10: The starting place for the example—a filled map by state

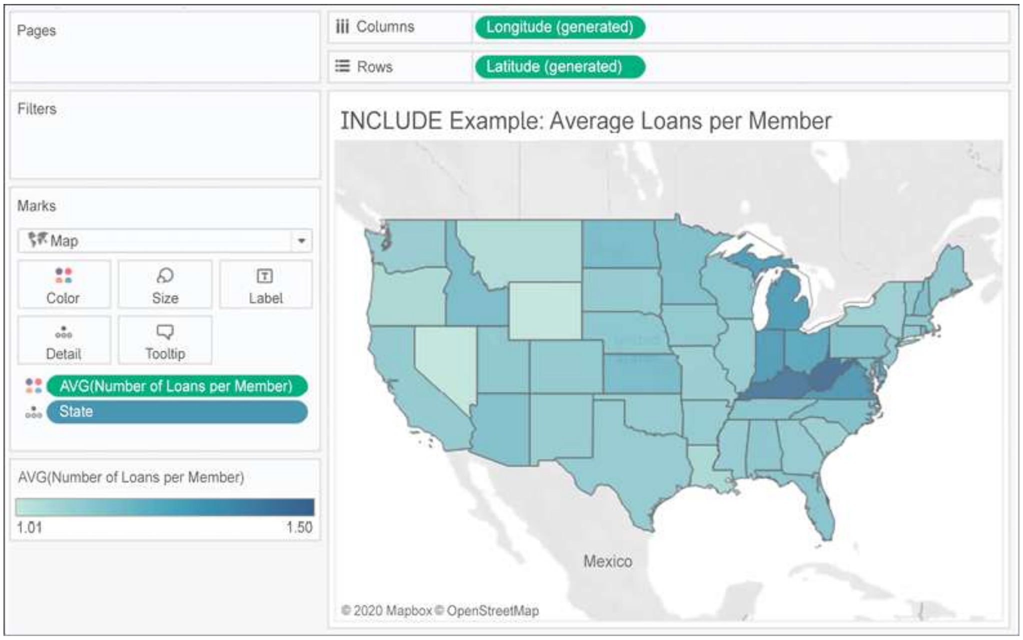


Figure 5.11: Using the include level of detail calculation to create a gradient of color to show average loans per member

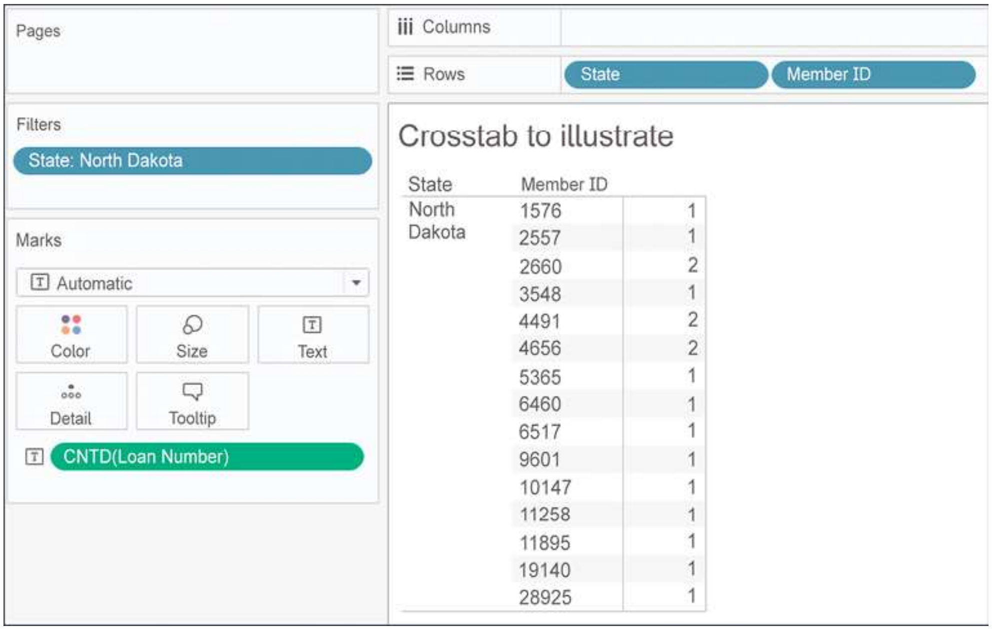


Figure 5.12: A crosstab helps illustrate how the distinct count of loans can be used as a basis for an average

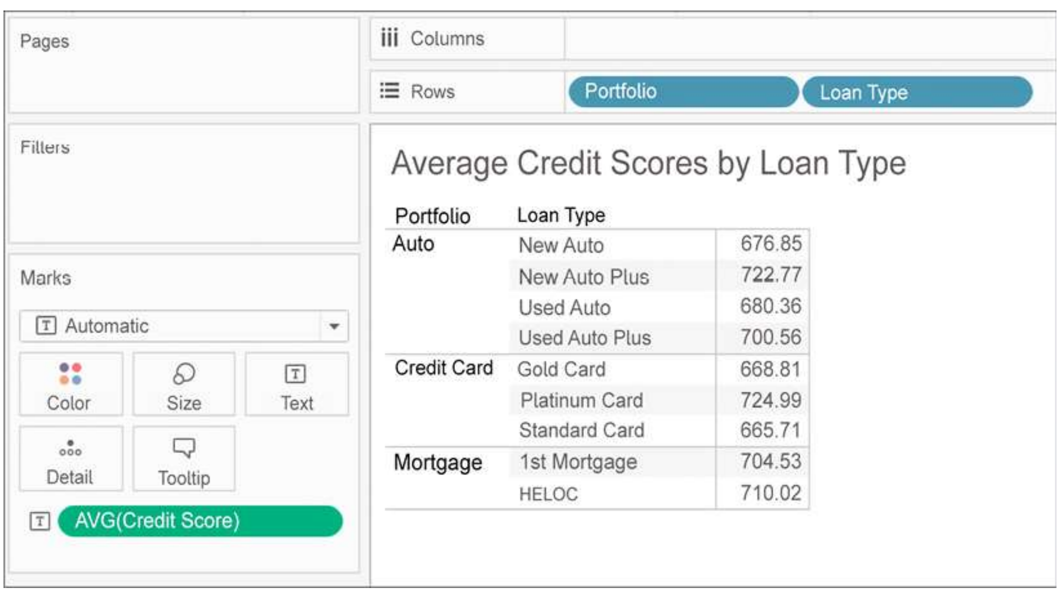


Figure 5.13: This crosstab shows the average credit score per loan type

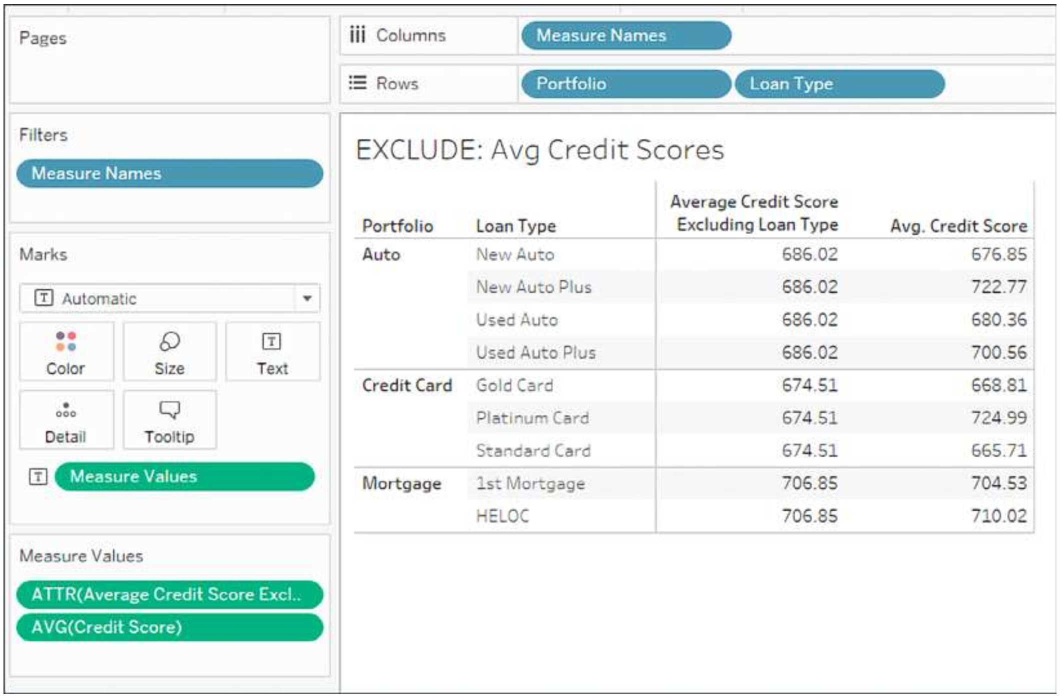


Figure 5.14: The exclude level of detail expression removes Loan Type so the average is only at the portfolio level

Figure 5.15: The final view shows the difference between the loan type average credit score and the overall portfolio average

Code

Code 5.1 – Basic pattern of level of detail syntax

```
{fixed|include|exclude [Dim 1],[Dim 2] : AGG([Field])}
```

Code 5.2 – Minimum number of order per state:

```
{fixed [State] : MIN([Orders])}
```

Code 5.3 – Fixed level of detail expressions

- `{fixed [State] : AVG([Orders])}`
- `{fixed : AVG([Orders])}`
- `{AVG([Orders])}`

Code 5.4 – Calculation named Overall Average Number of Orders and using a fixed level of detail calculation

```
{fixed : AVG([Orders])}
```

Code 5.4 – Fixed level of detail calculation called Latest Date per Member/Loan with code such as this:

```
{fixed [Member ID],[Loan Number] : MAX([Date])} = [Date]
```

Code 5.5 – Level of detail expression named Number of Loans per Member:

```
{include [Member ID] : COUNTD([Loan Number])}
```

Code 5.6

```
COUNTD(STR([Member ID]) + "_" + STR([Loan Number]))
/
COUNTD([Member ID])
```

Code 5.7

```
{fixed [State],[Member ID] : COUNTD([Loan Number])}
```

Code 5.8

```
{exclude [Loan Type] : AVG([Credit Score])}
```

Code 5.9

```
AVG([Credit Score]) - AVG([Average Credit Score Excluding
Loan Type])
```

Chapter 6

Figures

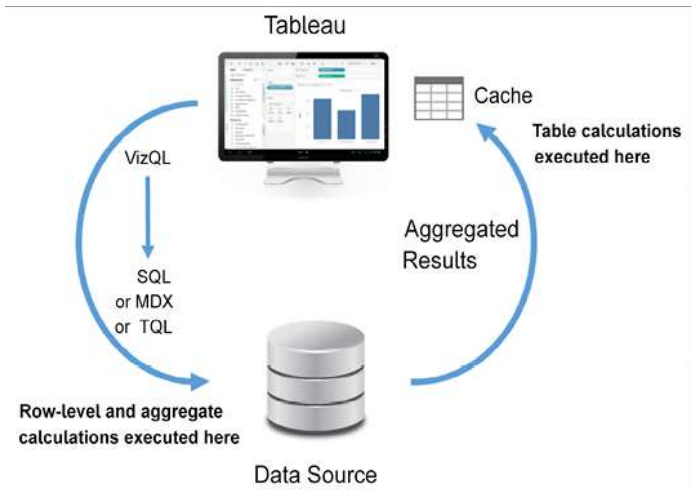


Figure 6.1: Table calculations are computed in Tableau's cache of aggregated data

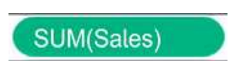


Figure 6.2: An active field without a table calculation applied

Figure 6.3: An active field with a table calculation applied includes the delta symbol

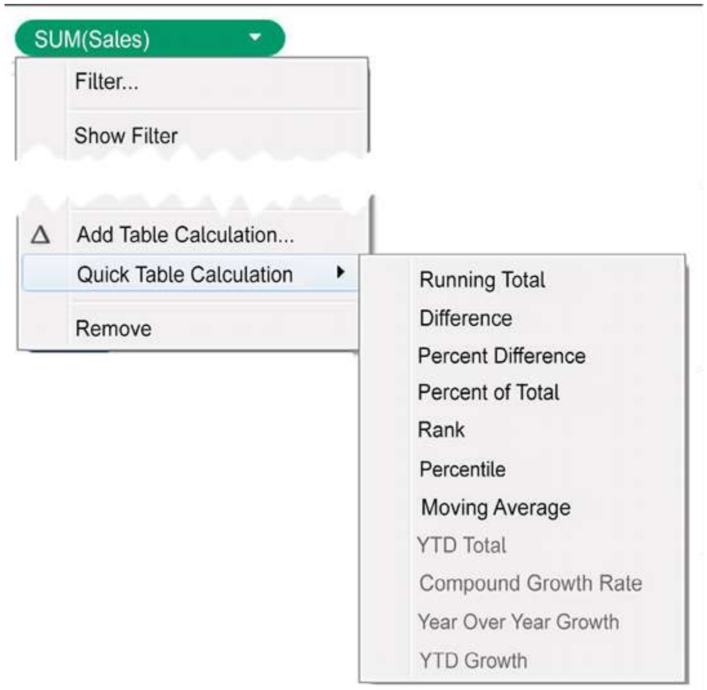


Figure 6.4: Using the dropdown, you can create a quick table calculation from an aggregate field in the view

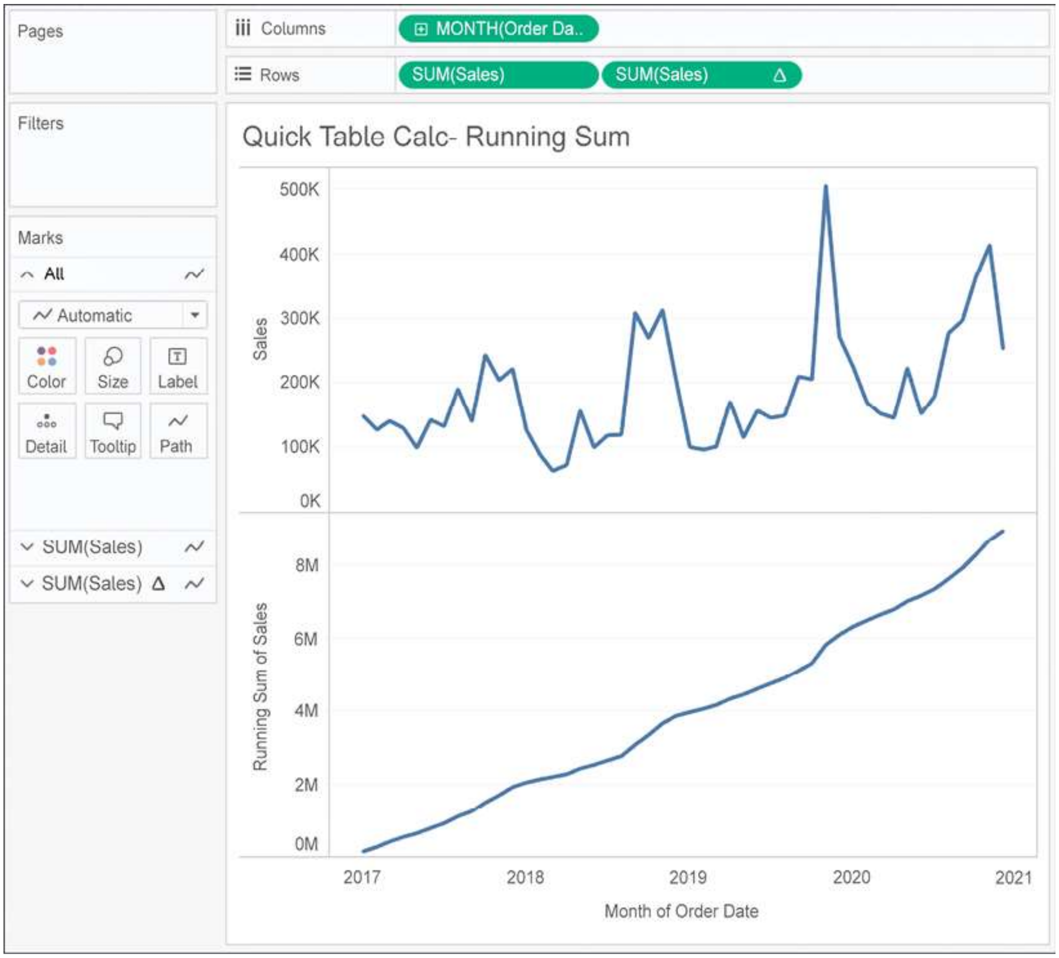


Figure 6.5: The first SUM(Sales) field is a normal aggregate. The second has a quick table calculation of Running Total applied

Some examples of Quick Table Calcs

Year of Order Date	Quarter of Order Date	Sales	Running Sum of Sales along Table (Down)	Difference in Sales from the Previous along Table (Down)	Rank of Sales along Table (Down)
2017	Q1	417,555	417,555		12
	Q2	372,289	789,844	-45,266	13
	Q3	464,319	1,254,163	92,030	10
	Q4	670,182	1,924,345	205,863	5
2018	Q1	279,148	2,203,493	-391,034	16
	Q2	330,269	2,533,762	51,121	14
	Q3	546,875	3,080,637	216,606	7
	Q4	788,255	3,868,892	241,380	3
2019	Q1	298,848	4,167,740	-489,407	15
	Q2	443,764	4,611,504	144,916	11
	Q3	505,453	5,116,957	61,689	9
	Q4	982,675	6,099,632	477,222	2
2020	Q1	547,656	6,647,288	-435,019	6
	Q2	521,650	7,168,938	-26,006	8
	Q3	752,933	7,921,871	231,283	4
	Q4	1,030,156	8,952,027	277,223	1

Figure 6.6: Sales in the first column is simply the SUM(Sales). The three additional columns show various table calculations applied (Running Sum, Difference, Rank)

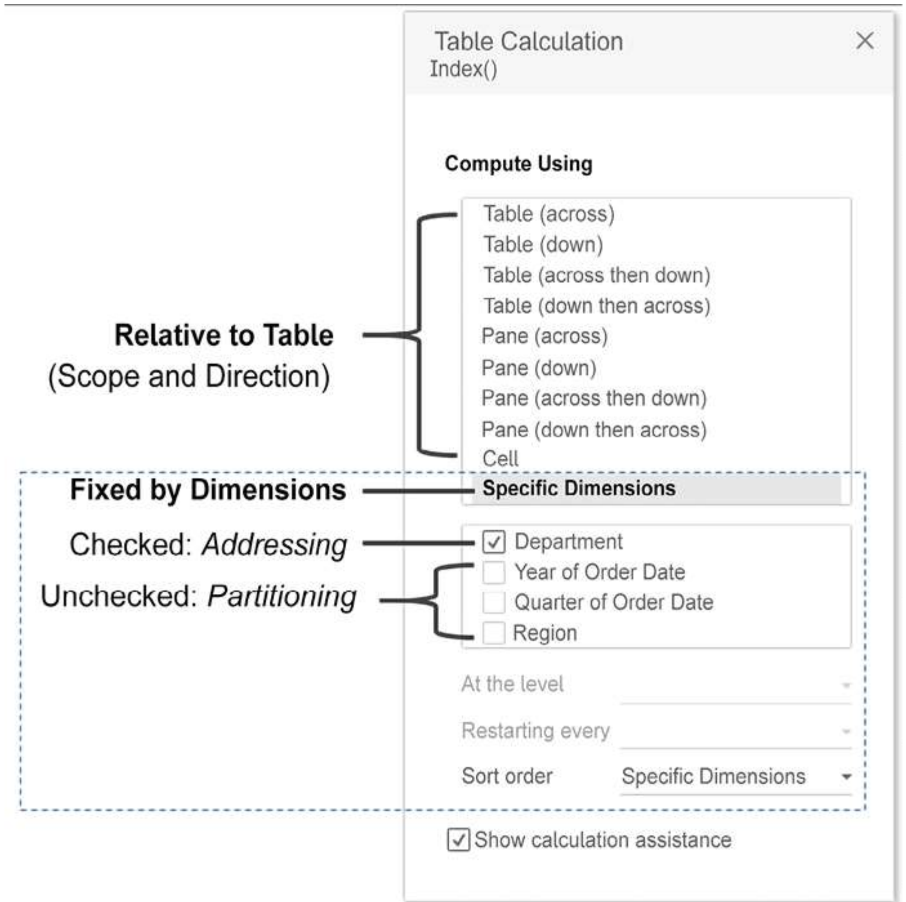
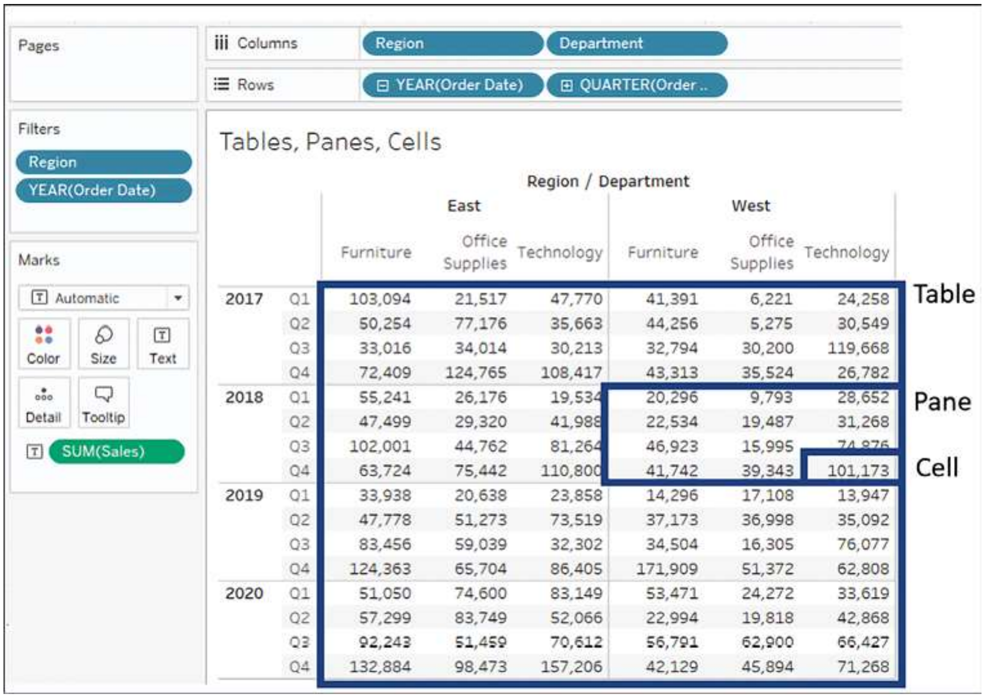
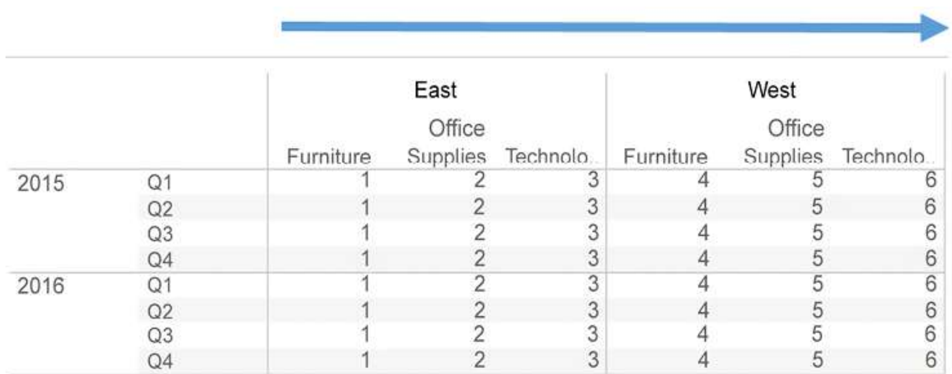


Figure 6.7: The Edit Table Calculation UI demonstrates the difference between Relative and Fixed table calculations



		Region / Department					
		East			West		
		Furniture	Office Supplies	Technology	Furniture	Office Supplies	Technology
2017	Q1	103,094	21,517	47,770	41,391	6,221	24,258
	Q2	50,254	77,176	35,663	44,256	5,275	30,549
	Q3	33,016	34,014	30,213	32,794	30,200	119,668
	Q4	72,409	124,765	108,417	43,313	35,524	26,782
2018	Q1	55,241	26,176	19,534	20,296	9,793	28,652
	Q2	47,499	29,320	41,988	22,534	19,487	31,268
	Q3	102,001	44,762	81,264	46,923	15,995	74,876
	Q4	63,724	75,442	110,800	41,742	39,343	101,173
2019	Q1	33,938	20,638	23,858	14,296	17,108	13,947
	Q2	47,778	51,273	73,519	37,173	36,998	35,092
	Q3	83,456	59,039	32,302	34,504	16,305	76,077
	Q4	124,363	65,704	86,405	171,909	51,372	62,808
2020	Q1	51,050	74,600	83,149	53,471	24,272	33,619
	Q2	57,299	83,749	52,066	22,994	19,818	42,868
	Q3	92,243	51,459	70,612	56,791	62,900	66,427
	Q4	132,884	98,473	157,206	42,129	45,894	71,268

Figure 6.8: The difference between table, pane, and cell in the view



		East			West		
		Furniture	Office Supplies	Technolo..	Furniture	Office Supplies	Technolo..
2015	Q1	1	2	3	4	5	6
	Q2	1	2	3	4	5	6
	Q3	1	2	3	4	5	6
	Q4	1	2	3	4	5	6
2016	Q1	1	2	3	4	5	6
	Q2	1	2	3	4	5	6
	Q3	1	2	3	4	5	6
	Q4	1	2	3	4	5	6

Figure 6.9: Table (across)

		East			West		
		Furniture	Office Supplies	Technolo..	Furniture	Office Supplies	Technolo..
2015	Q1	1	1	1	1	1	1
	Q2	2	2	2	2	2	2
	Q3	3	3	3	3	3	3
	Q4	4	4	4	4	4	4
2016	Q1	5	5	5	5	5	5
	Q2	6	6	6	6	6	6
	Q3	7	7	7	7	7	7
	Q4	8	8	8	8	8	8

Figure 6.10: Table (down)

		East			West		
		Furniture	Office Supplies	Technolo..	Furniture	Office Supplies	Technolo..
2015	Q1	1	2	3	4	5	6
	Q2	7	8	9	10	11	12
	Q3	13	14	15	16	17	18
	Q4	19	20	21	22	23	24
2016	Q1	25	26	27	28	29	30
	Q2	31	32	33	34	35	36
	Q3	37	38	39	40	41	42
	Q4	43	44	45	46	47	48

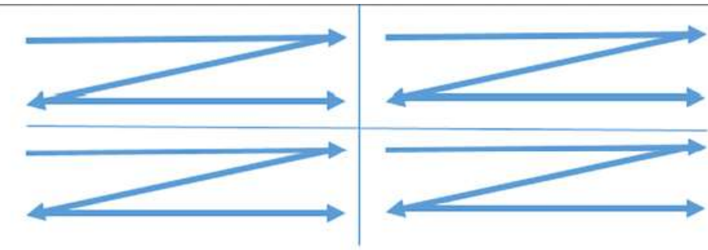
Figure 6.11: Table (across then down)

		East			West		
		Furniture	Office Supplies	Technolo..	Furniture	Office Supplies	Technolo..
2015	Q1	1	2	3	4	5	6
	Q2	1	2	3	4	5	6
	Q3	1	2	3	4	5	6
	Q4	1	2	3	4	5	6
2016	Q1	1	2	3	4	5	6
	Q2	1	2	3	4	5	6
	Q3	1	2	3	4	5	6
	Q4	1	2	3	4	5	6

Figure 6.12: Pane (across)

		East			West		
		Furniture	Office Supplies	Technolo..	Furniture	Office Supplies	Technolo..
2015	Q1	1	1	1	1	1	1
	Q2	2	2	2	2	2	2
	Q3	3	3	3	3	3	3
	Q4	4	4	4	4	4	4
2016	Q1	1	1	1	1	1	1
	Q2	2	2	2	2	2	2
	Q3	3	3	3	3	3	3
	Q4	4	4	4	4	4	4

Figure 6.13: Pane (down)



		East			West		
		Furniture	Office Supplies	Technolo..	Furniture	Office Supplies	Technolo..
2015	Q1	1	2	3	1	2	3
	Q2	4	5	6	4	5	6
	Q3	7	8	9	7	8	9
	Q4	10	11	12	10	11	12
2016	Q1	1	2	3	1	2	3
	Q2	4	5	6	4	5	6
	Q3	7	8	9	7	8	9
	Q4	10	11	12	10	11	12

Figure 6.14: Pane (across then down)

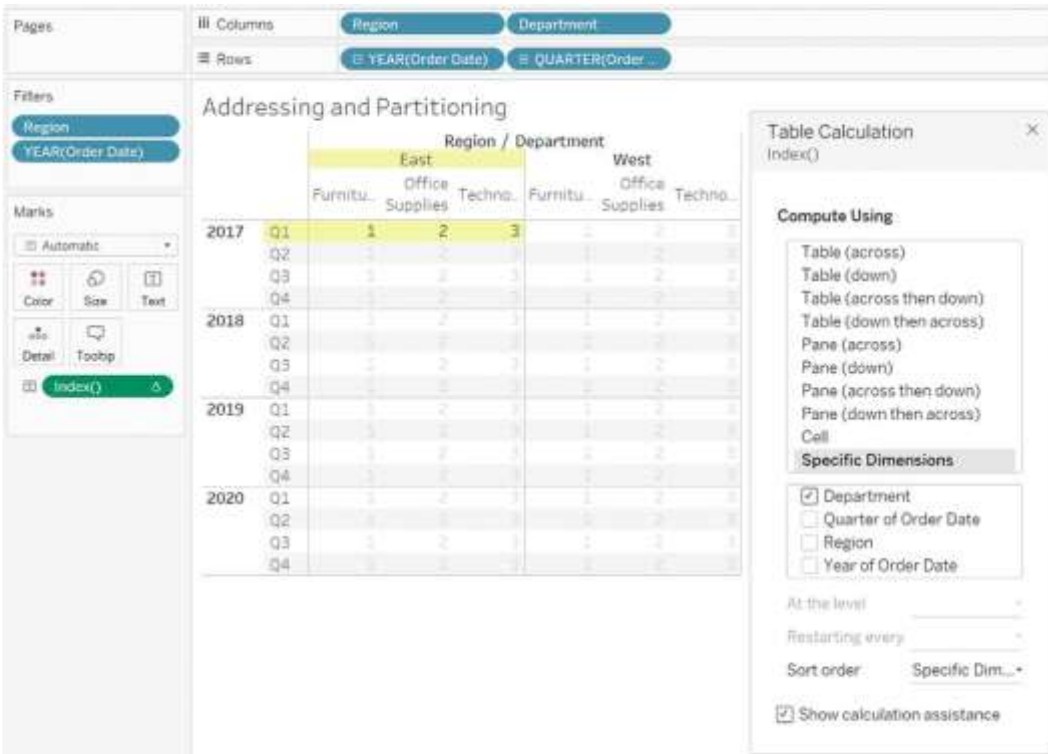


Figure 6.15: Setting the table calculation to Compute Using Specific Dimensions uses addressing and partitioning

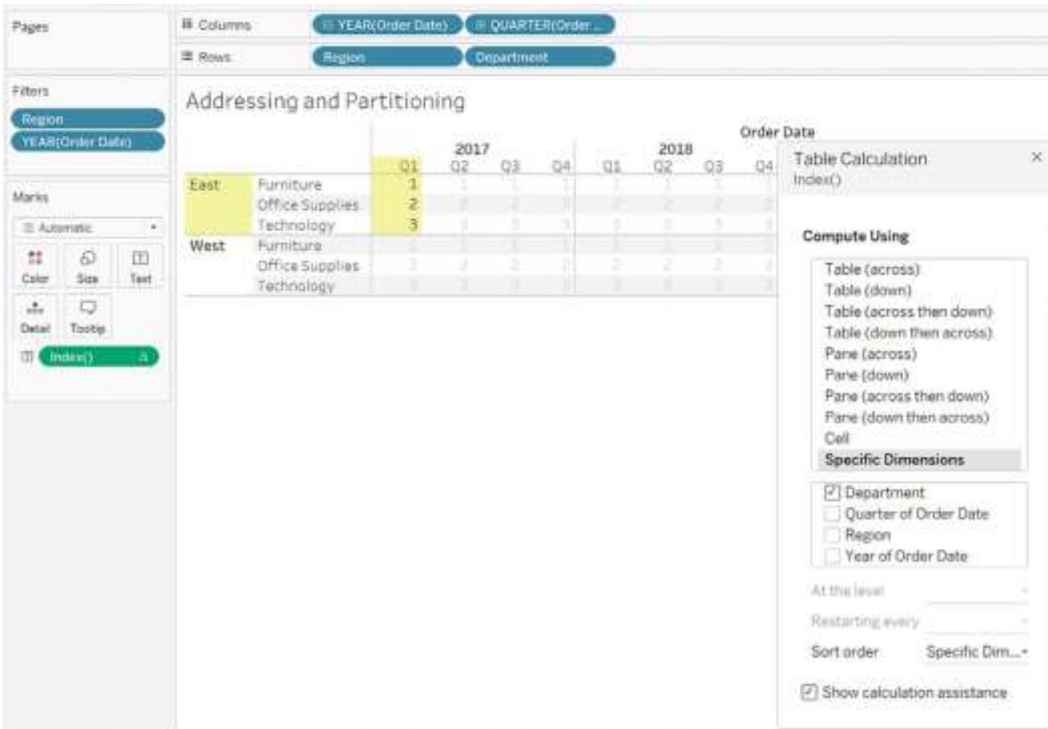


Figure 6.16: Swapping Rows and Columns does not change how this table calculation was computed as it is fixed to the dimensions rather than the table layout

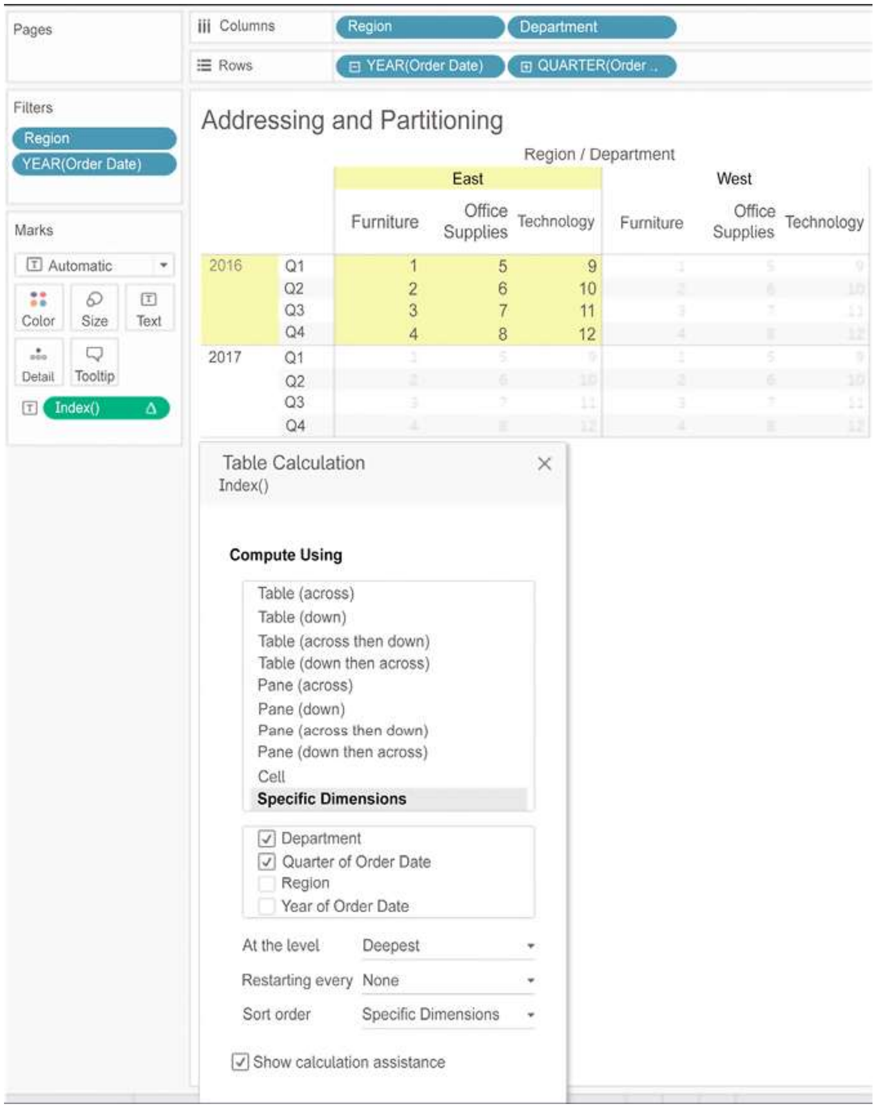


Figure 6.17: Adding dimensions alters the table calculation's behavior. One of the resulting partitions is highlighted

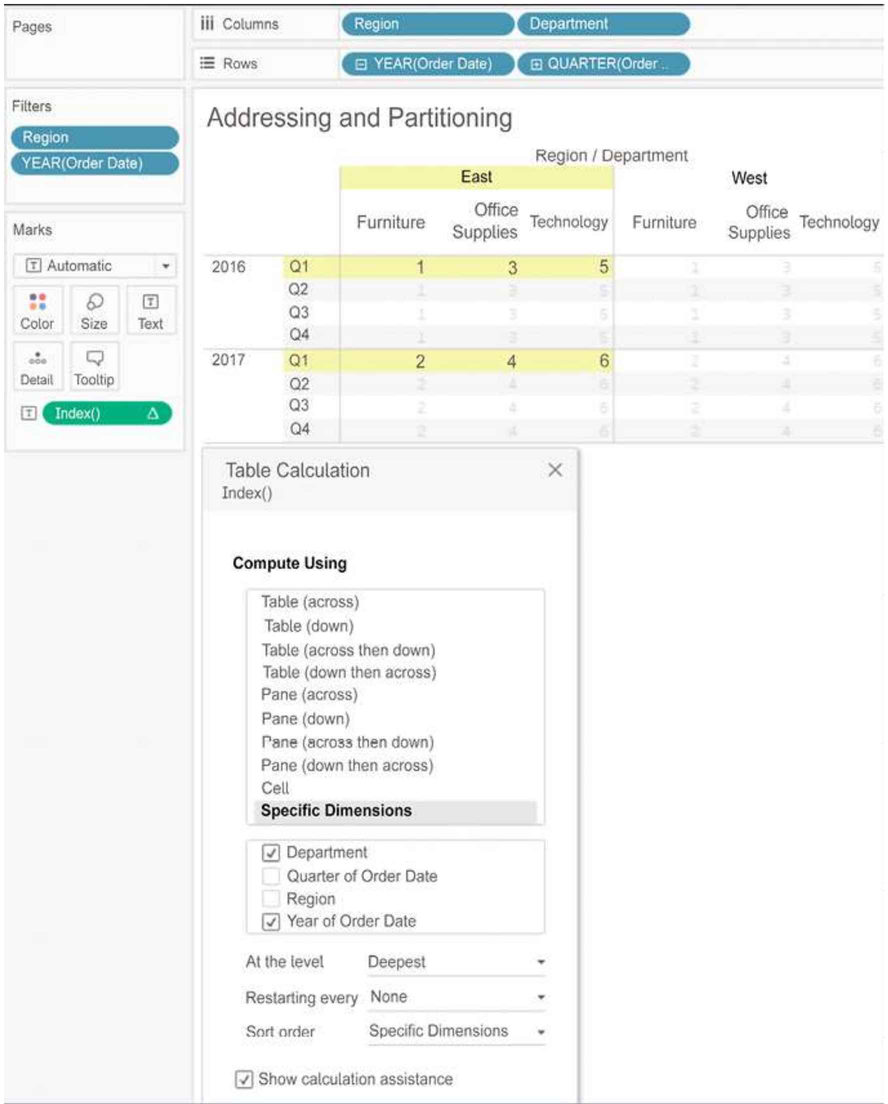


Figure 6.18: Changing the checked dimensions alters the table calculation's behavior. One of the resulting partitions is highlighted

Meta Table Calculations

Department	Category	Index along Category	First along Category	Last along Category	Size along Category
Furniture	Bookcases	1	0	3	4
	Chairs & Chairmats	2	-1	2	4
	Office Furnishings	3	-2	1	4
	Tables	4	-3	0	4
Office Supplies	Appliances	1	0	8	9
	Binders and Binder Accessories	2	-1	7	9
	Envelopes	3	-2	6	9
	Labels	4	-3	5	9
	Paper	5	-4	4	9
	Pens & Art Supplies	6	-5	3	9
	Rubber Bands	7	-6	2	9
	Scissors, Rulers and Trimmers	8	-7	1	9
	Storage & Organization	9	-8	0	9
Technology	Computer Peripherals	1	0	3	4
	Copiers and Fax	2	-1	2	4
	Office Machines	3	-2	1	4
	Telephones and Communication	4	-3	0	4

Figure 6.19: Meta table calculations

Lookup and Previous Value

Departme..	Category	Lookup	Previous Value
Furniture	Bookcases	Null	,Bookcases
	Chairs & Chairmats	Bookcases	,Bookcases,Chairs & Chairmats
	Office Furnishings	Chairs & Chairmats	,Bookcases,Chairs & Chairmats,Office Furnishings
	Tables	Office Furnishings	,Bookcases,Chairs & Chairmats,Office Furnishings,Tables
Office Supplies	Appliances	Null	,Appliances
	Binders and Binder Accessor..	Appliances	,Appliances,Binders and Binder Accessories
	Envelopes	Binders and Binder Acc..	,Appliances,Binders and Binder Accessories,Envelopes
	Labels	Envelopes	,Appliances,Binders and Binder Accessories,Envelopes,Labels
	Paper	Labels	,Appliances,Binders and Binder Accessories,Envelopes,Labels,Paper
	Pens & Art Supplies	Paper	,Appliances,Binders and Binder Accessories,Envelopes,Labels,Paper,Pens & Art Supplies
	Rubber Bands	Pens & Art Supplies	,Appliances,Binders and Binder Accessories,Envelopes,Labels,Paper,Pens & Art Supplies,Rubber Bands
	Scissors, Rulers and Trimmers	Rubber Bands	,Appliances,Binders and Binder Accessories,Envelopes,Labels,Paper,Pens & Art Supplies,Rubber Bands,Scissors,
	Storage & Organization	Scissors, Rulers and..	,Appliances,Binders and Binder Accessories,Envelopes,Labels,Paper,Pens & Art Supplies,Rubber Bands,Scissors,
Technology	Computer Peripherals	Null	,Computer Peripherals
	Copiers and Fax	Computer Peripherals	,Computer Peripherals,Copiers and Fax
	Office Machines	Copiers and Fax	,Computer Peripherals,Copiers and Fax,Office Machines
	Telephones and Communication	Office Machines	,Computer Peripherals,Copiers and Fax,Office Machines,Telephones and Communication

Figure 6.20: Lookup and Previous_Value functions (though Previous_Value includes some additional logic described)

Running Functions

Department	Category	Sales	Running Sum of Sales along Category	Running Min of Sales along Category
Furniture	Bookcases	507,496	507,496	507,496
	Chairs & Chairmats	1,164,586	1,672,082	507,496
	Office Furnishings	444,634	2,116,716	444,634
	Tables	1,061,922	3,178,638	444,634
Office Supplies	Appliances	456,736	456,736	456,736
	Binders and Binder ..	638,583	1,095,319	456,736
	Envelopes	147,915	1,243,234	147,915
	Labels	23,446	1,266,680	23,446
	Paper	253,620	1,520,300	23,446
	Pens & Art Supplies	103,265	1,623,565	23,446
	Rubber Bands	8,670	1,632,235	8,670
	Scissors, Rulers and ..	40,432	1,672,667	8,670
	Storage & Organizat..	585,717	2,258,384	8,670
Technology	Computer Periphera..	490,851	490,851	490,851
	Copiers and Fax	661,215	1,152,066	490,851
	Office Machines	1,218,655	2,370,721	490,851
	Telephones and Com..	1,144,284	3,515,005	490,851

Figure 6.21: Running Functions

Window Functions

Departm..	Category	Sales	Window Sum along Catego..	Window Max along Catego..
Furniture	Bookcases	507,496	3,178,638	1,164,586
	Chairs & Chairmats	1,164,586	3,178,638	1,164,586
	Office Furnishings	444,634	3,178,638	1,164,586
	Tables	1,061,922	3,178,638	1,164,586
Office Supplies	Appliances	456,736	2,258,384	638,583
	Binders and Binder ..	638,583	2,258,384	638,583
	Envelopes	147,915	2,258,384	638,583
	Labels	23,446	2,258,384	638,583
	Paper	253,620	2,258,384	638,583
	Pens & Art Supplies	103,265	2,258,384	638,583
	Rubber Bands	8,670	2,258,384	638,583
	Scissors, Rulers and ..	40,432	2,258,384	638,583
	Storage & Organizat..	585,717	2,258,384	638,583
Technolo..	Computer Peripherals	490,851	3,515,005	1,218,655
	Copiers and Fax	661,215	3,515,005	1,218,655
	Office Machines	1,218,655	3,515,005	1,218,655
	Telephones and Com..	1,144,284	3,515,005	1,218,655

Figure 6.22: Examples of Window functions

Rank Functions

Department	Category	Sales	Rank along Category
Furniture	Bookcases	507,496	3
	Chairs & Chairmats	1,164,586	1
	Office Furnishings	444,634	4
	Tables	1,061,922	2
Office Supplies	Appliances	456,736	3
	Binders and Binder ..	638,583	1
	Envelopes	147,915	5
	Labels	23,446	8
	Paper	253,620	4
	Pens & Art Supplies	103,265	6
	Rubber Bands	8,670	9
	Scissors, Rulers and ..	40,432	7
	Storage & Organizat..	585,717	2
Technology	Computer Periphera..	490,851	4
	Copiers and Fax	661,215	3
	Office Machines	1,218,655	1
	Telephones and Com..	1,144,284	2

Figure 6.23: Examples of rank functions

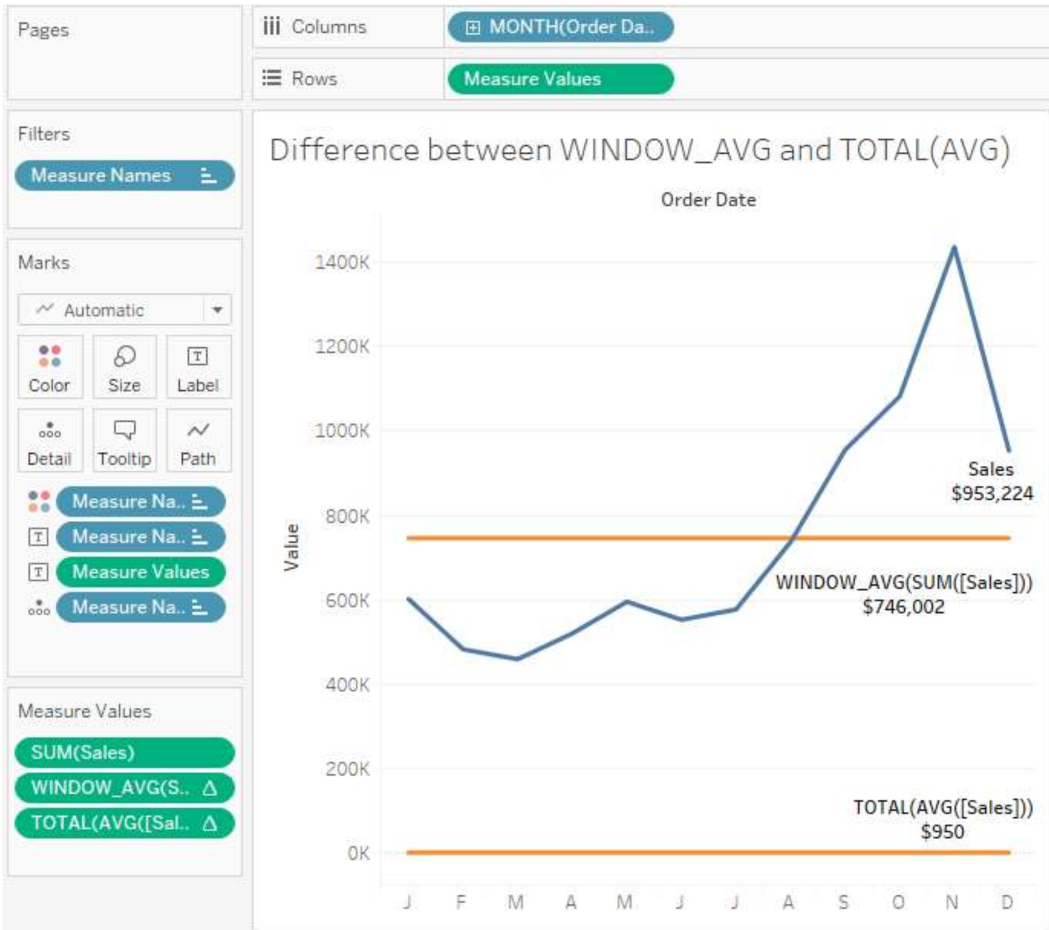


Figure 6.24: The difference between WINDOW_AVG() and TOTAL(AVG)

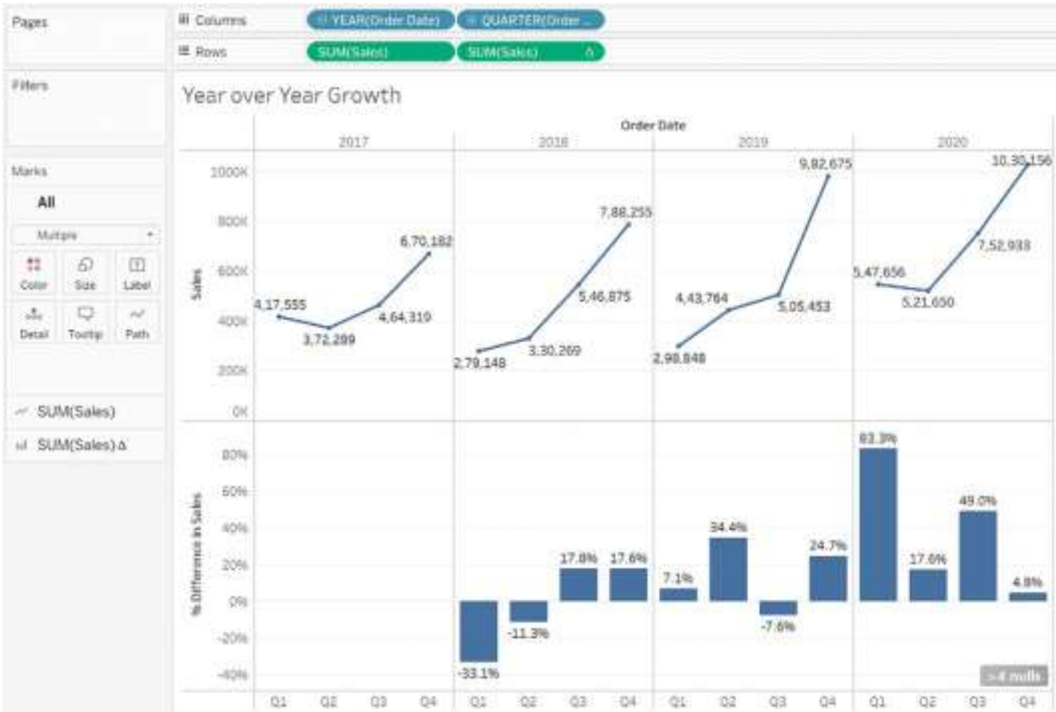


Figure 6.25: Year over year growth of Sales

Figure 6.26: Year over year growth of Sales—but it doesn't work with Q1 missing in the first year

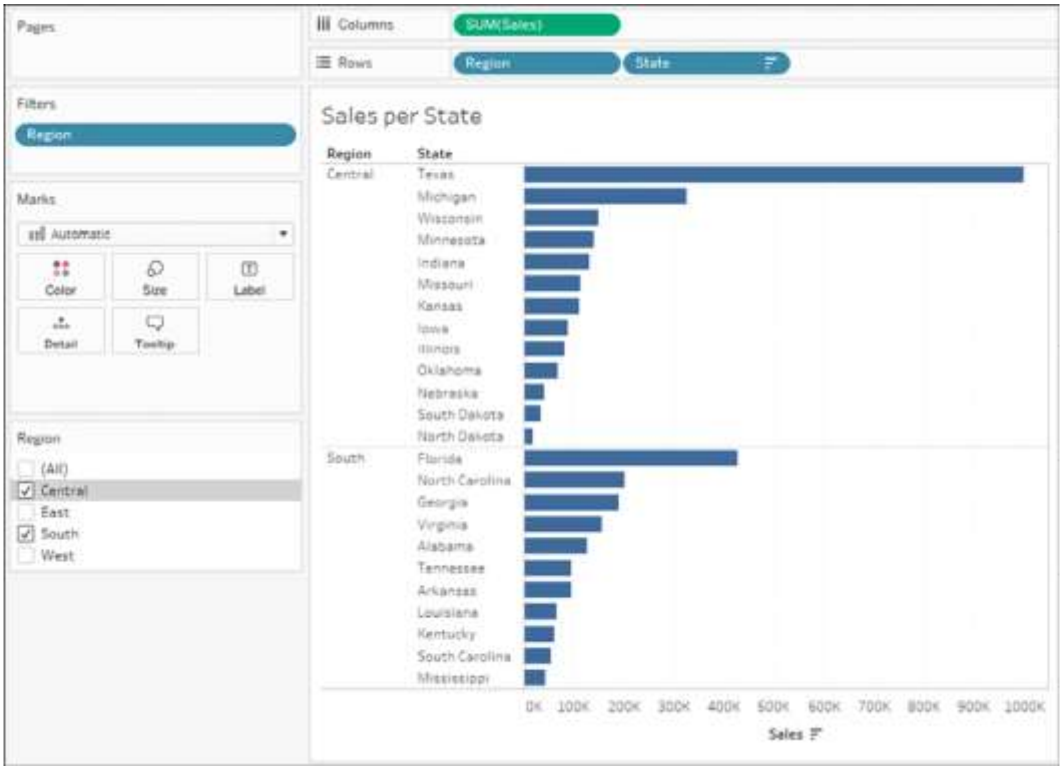


Figure 6.27: Sales per State for two regions

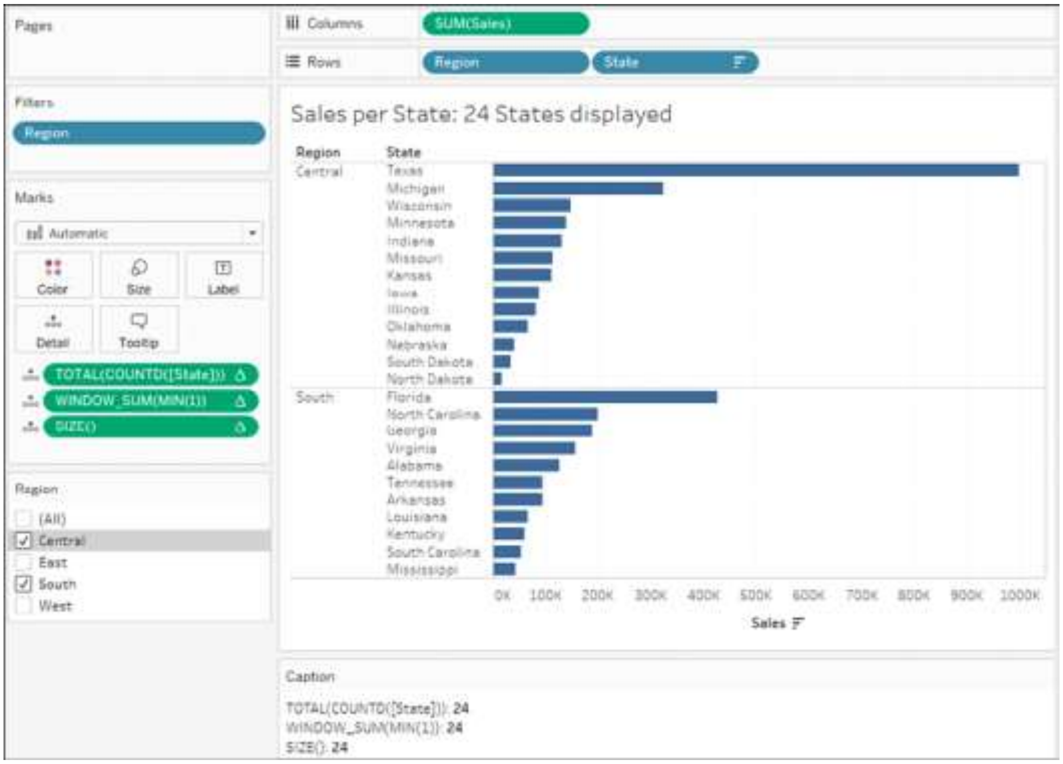


Figure 6.28: Various table calculations could be employed to achieve the total in the title

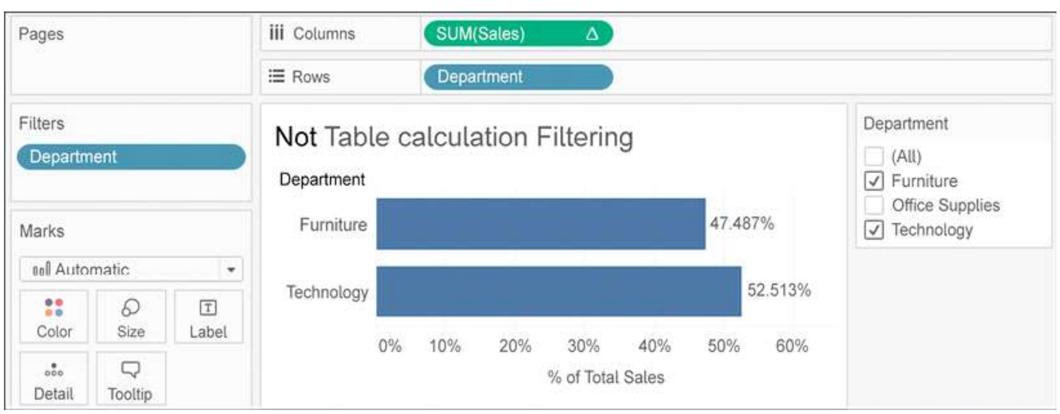


Figure 6.29: When Office Supplies is filtered out, the percentage table calculation adds up to 100% for the departments remaining in the view

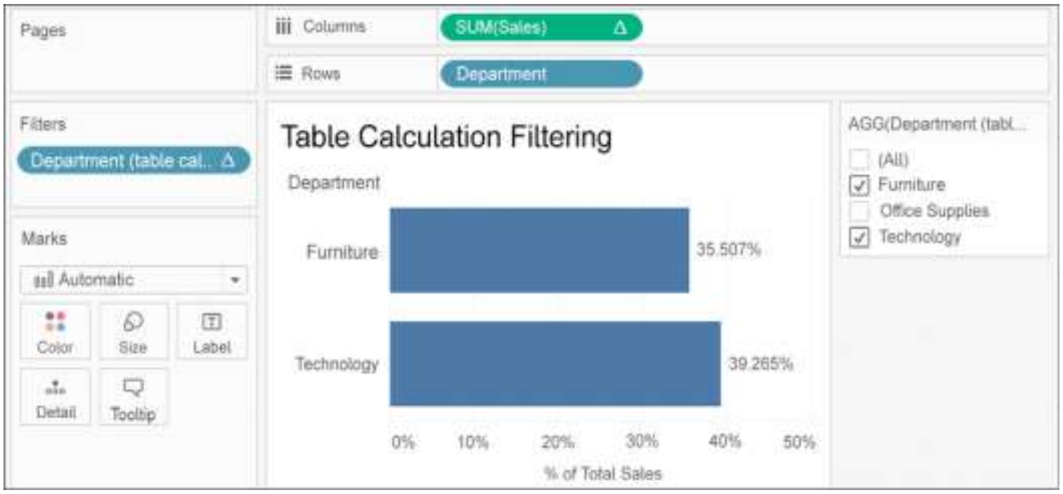


Figure 6.30: When a table calculation filter is used, all the aggregate data is available in the cache for the % of Total Sales to be calculated for all departments

Code

Code 6.1

```
Lookup(ATTR([Category]), -1)
```

Code 6.2

```
Previous_Value("") + "," + ATTR([Category])
```

Code 6.3

```
IF LEN(Previous_Value("")) == 0
THEN ATTR([Category])
ELSE Previous_Value("") + "," + ATTR([Category])
END
```

Code 6.4

```
Running_Sum(SUM[Sales]))
```

Code 6.5

```
Running_Min(SUM[Sales]))
```

Code 6.6

```
Window_Sum(SUM([Sales])
```

Code 6.7

```
Window_Max(SUM([Sales])
```

Code 6.8 – Example of Total function

For example, `Total(SUM([Sales]))` gives the same result as `Window_Sum(SUM([Sales]))`, but `Total(AVG([Sales]))` will almost certainly give a different result from `Window_AVG(SUM([Sales]))` because Total is giving you the actual average of underlying rows, while the Window function is averaging the sums

Code 6.9

```
LOOKUP(ATTR([Department]), 0)
```

Chapter 7

Figures

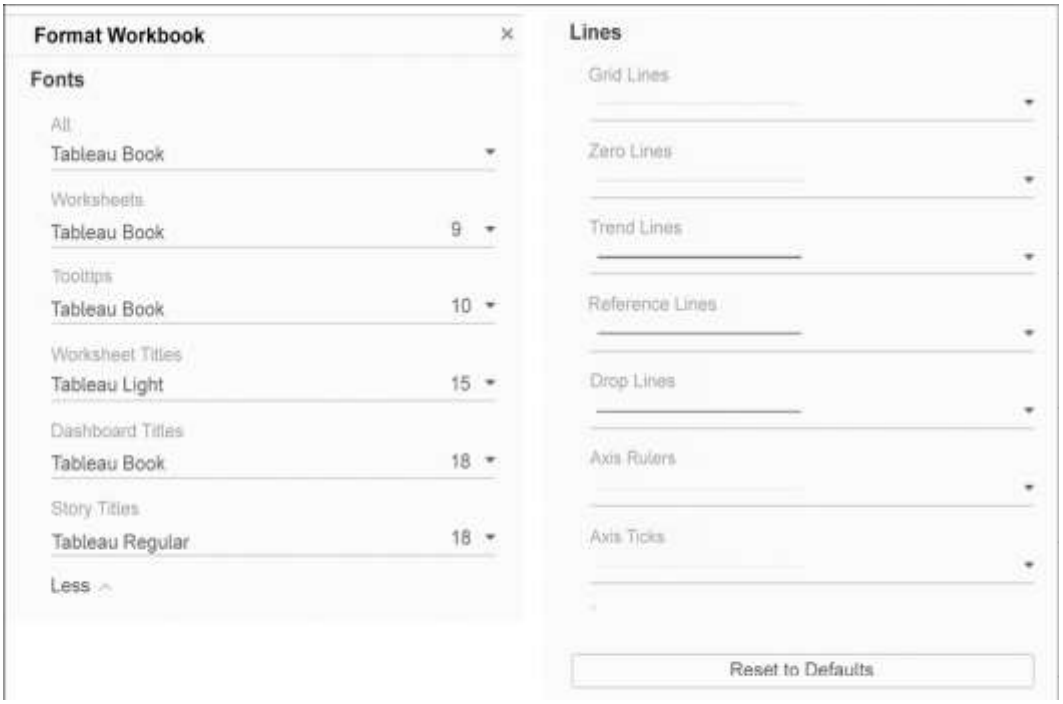
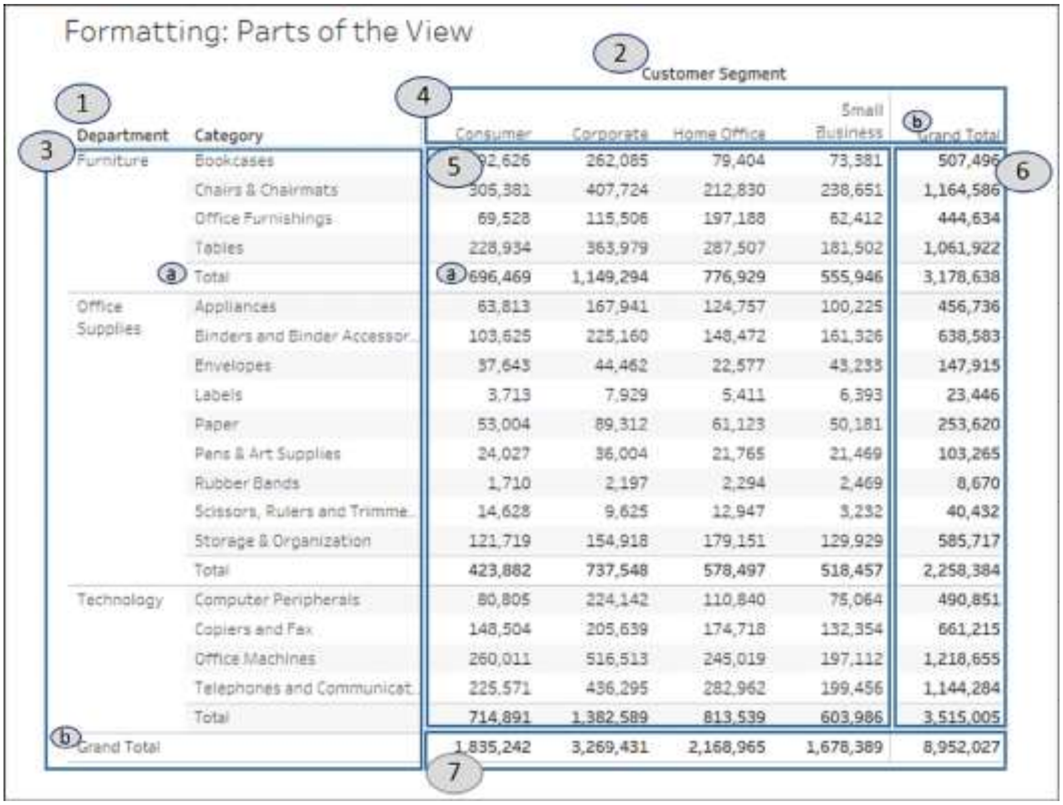


Figure 7.1: Workbook formatting options



Formatting: Parts of the View

Department	Category	Consumer	Corporate	Home Office	Small Business	Grand Total
		Customer Segment				
Furniture	Bookcases	92,626	262,085	79,404	73,381	507,496
	Chairs & Chairmats	305,381	407,724	212,830	238,651	1,164,586
	Office Furnishings	69,528	115,506	197,188	62,412	444,634
	Tables	228,934	363,979	287,507	181,502	1,061,922
	Total	696,469	1,149,294	776,929	555,946	3,178,638
Office Supplies	Appliances	63,813	167,941	124,757	100,225	456,736
	Binders and Binder Accessor..	103,625	225,160	148,472	161,326	638,583
	Envelopes	37,643	44,462	22,577	43,233	147,915
	Labels	3,713	7,929	5,411	6,393	23,446
	Paper	53,004	89,312	61,123	50,181	253,620
	Pens & Art Supplies	24,027	36,004	21,765	21,469	103,265
	Rubber Bands	1,710	2,197	2,294	2,469	8,670
	Scissors, Rulers and Trimme..	14,628	9,625	12,947	3,232	40,432
	Storage & Organization	121,719	154,918	179,151	129,929	585,717
	Total	423,882	737,548	578,497	518,457	2,258,384
Technology	Computer Peripherals	80,805	224,142	110,840	75,064	490,851
	Copiers and Fax	148,504	205,639	174,718	132,354	661,215
	Office Machines	260,011	516,513	245,019	197,112	1,218,655
	Telephones and Communicat..	225,571	436,295	282,962	199,456	1,144,284
	Total	714,891	1,382,589	813,539	603,986	3,515,005
Grand Total		1,835,242	3,269,431	2,168,965	1,678,389	8,952,027

Figure 7.2: Parts of a view that can be formatted using worksheet-level formatting

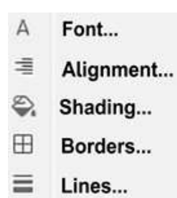


Figure 7.3: Formatting options for a worksheet

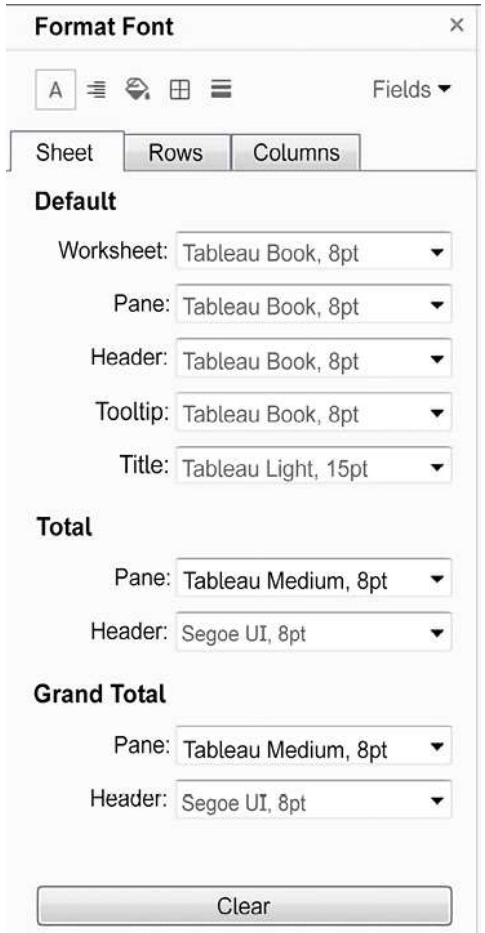


Figure 7.4: The Format Font pane

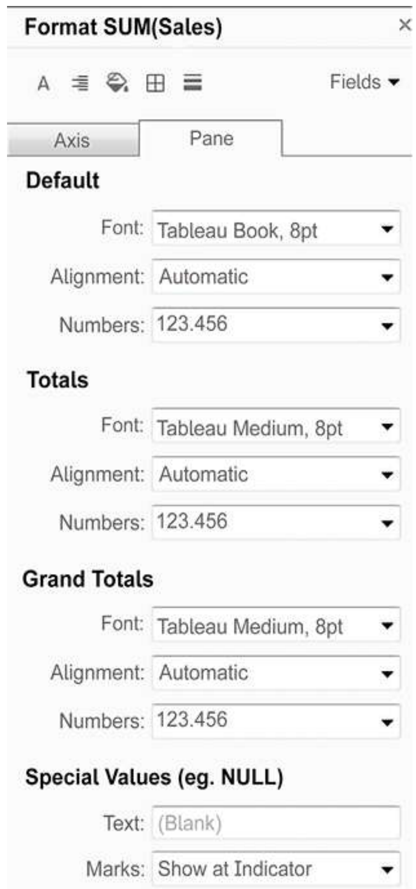


Figure 7.5: The Format pane for field-level formatting

Format String	Resulting Value
#;-#	34331 and -8157
#,###.##; (#,###.##)	34,331.34 and (8,156.78)
#,###.00000;-#,###.00000	34,331.33600 and -8,156.77770
"up "#,###;"down "#,###; "same"	up 34,331 and down 8,157
#,###"▲";#,###" ▼"	34,331▲ and 8,157▼

Figure 7.6: Examples of format strings and resulting values

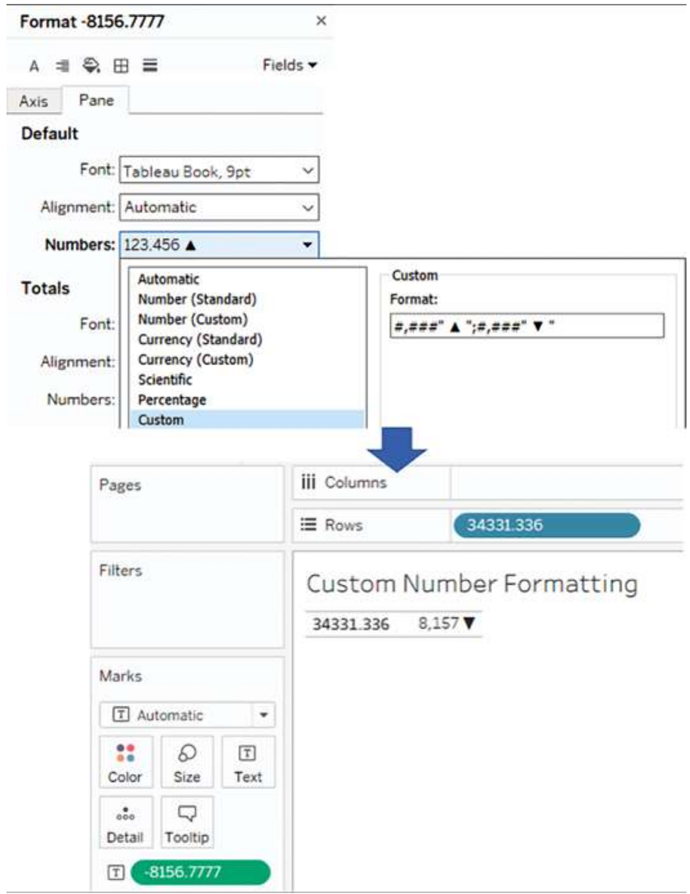


Figure 7.7: Experiment with format strings using the Custom Number Formatting view in the Chapter 7 workbook

Format String	Resulting Value
m/d/yyyy	11/8/2018
dd/mm/yyyy	08/11/2018
"The date is" m/d/yyyy	The date is 11/8/2018
mmm d, yyyy	Nov 8, 2018
mmmm dd, yyyy	November 08 2018
mm/dd/yyyy h:mm AM/PM	11/08/2018 1:30PM
ttttt	1:30:28 PM
dddd, mmmm d, HH:MM:SS	Thursday, November 8, 13:30:28
ddd	Thu

Figure 7.8: Some possible date formatting examples

Figure 7.9: The custom format string used here is mmmmm, which results in a single letter for the month

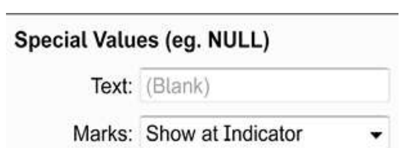


Figure 7.10: The Special Values options appear on the Format pane

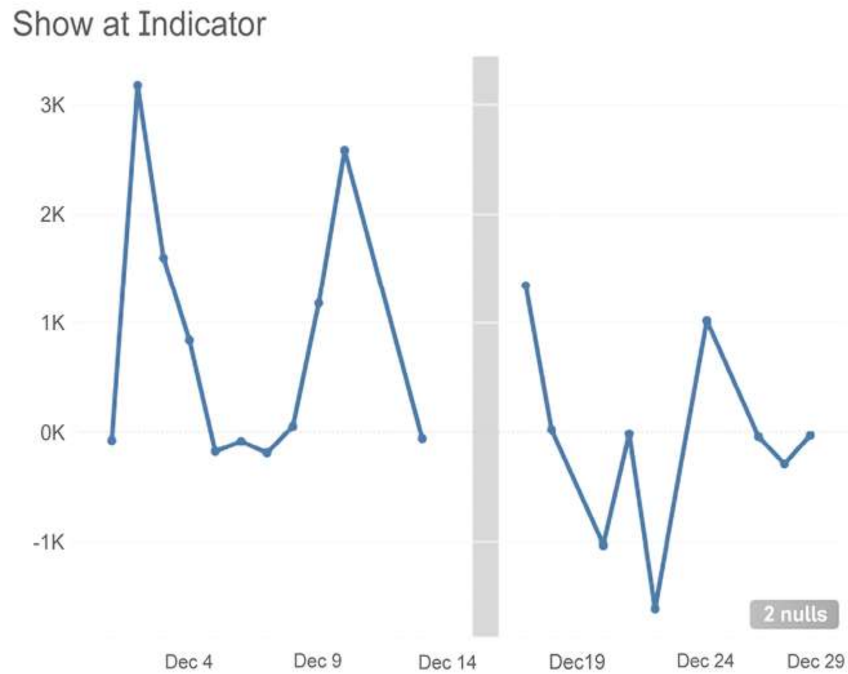


Figure 7.11: Show at Indicator

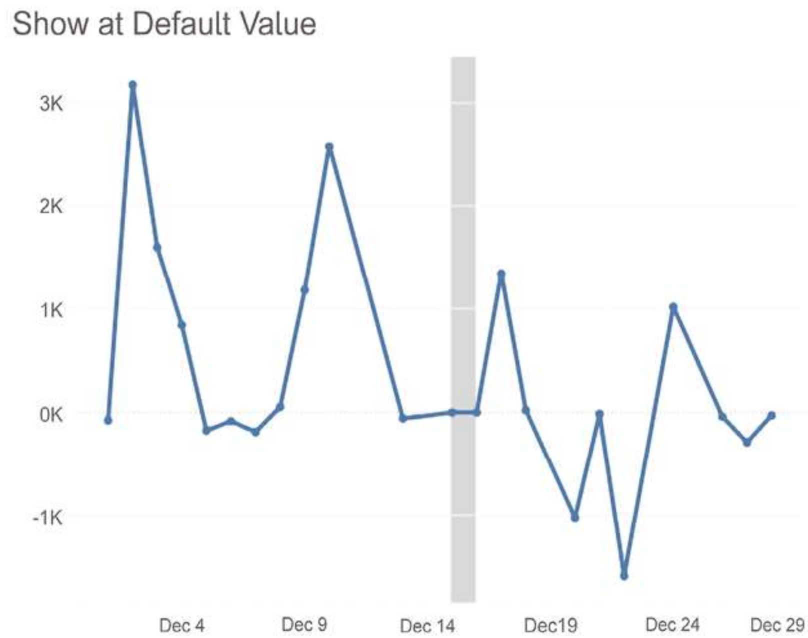


Figure 7.12: Show at Default Value

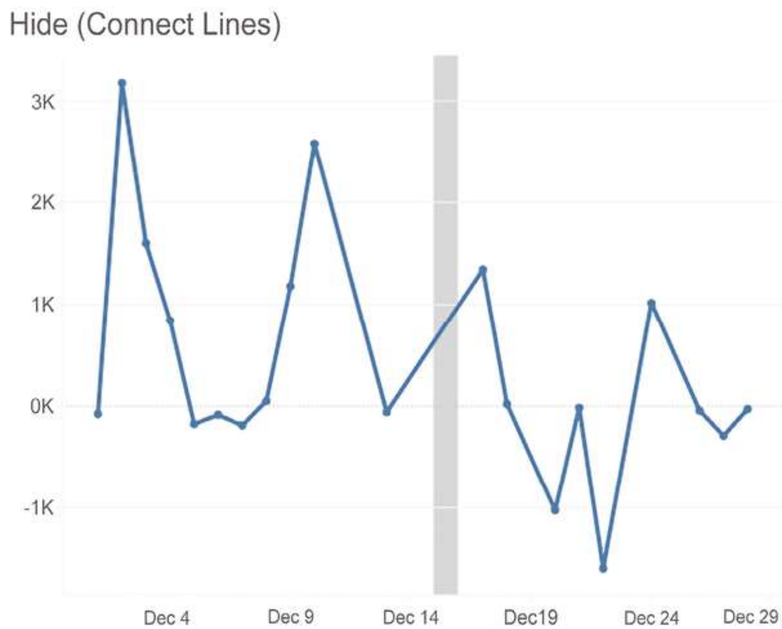


Figure 7.13: Hide (Connect Lines)

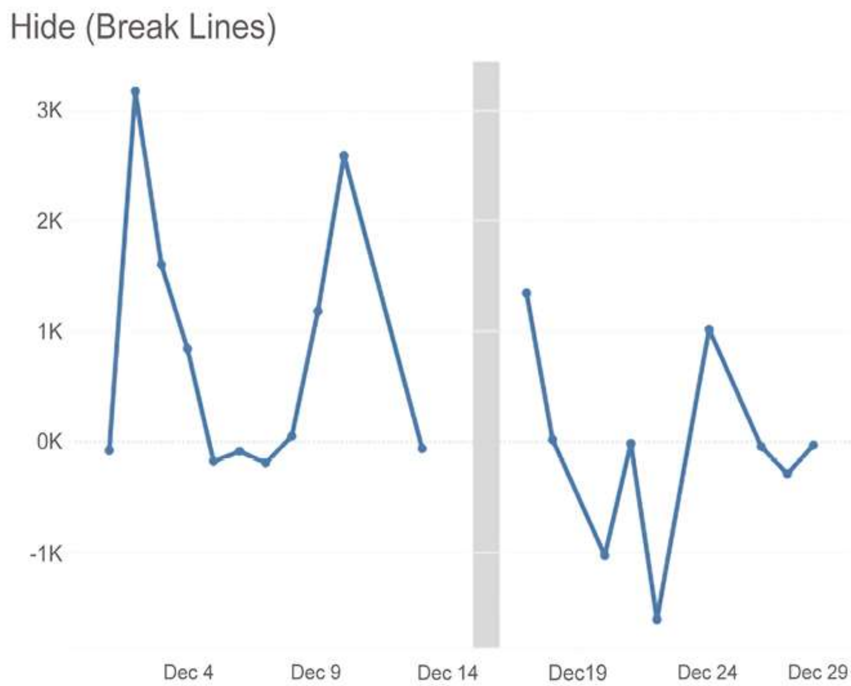


Figure 7.14: Hide (Break Lines)

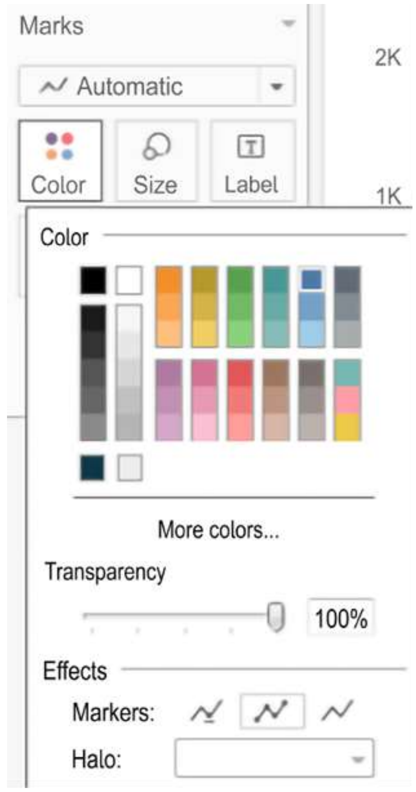


Figure 7.15: Adding Markers to lines

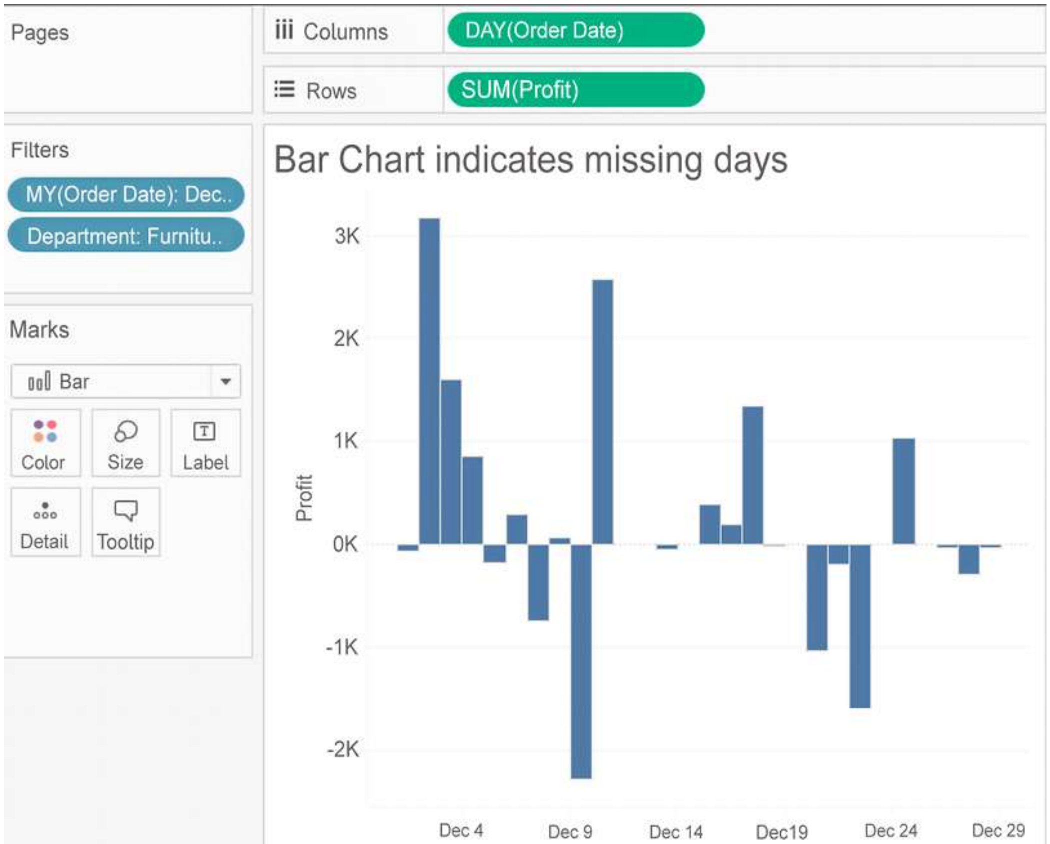


Figure 7.16: Bar charts are sometimes better than line charts for showing missing values

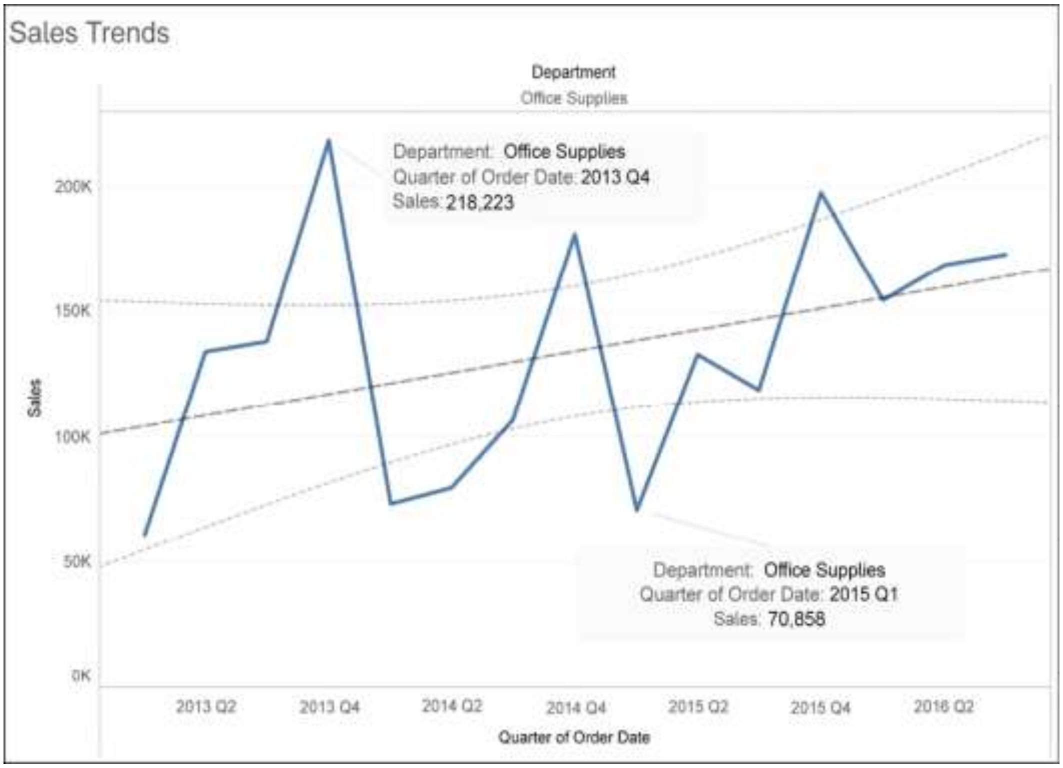


Figure 7.17: The default formatting is often great for data discovery and quick analysis but may be more cluttered than desired for clearly communicating and emphasizing the data story to others

Where do you want to be … I want to be okay with my alone-ness by my next birthday …

Because my first thought was I don't want to be alone.

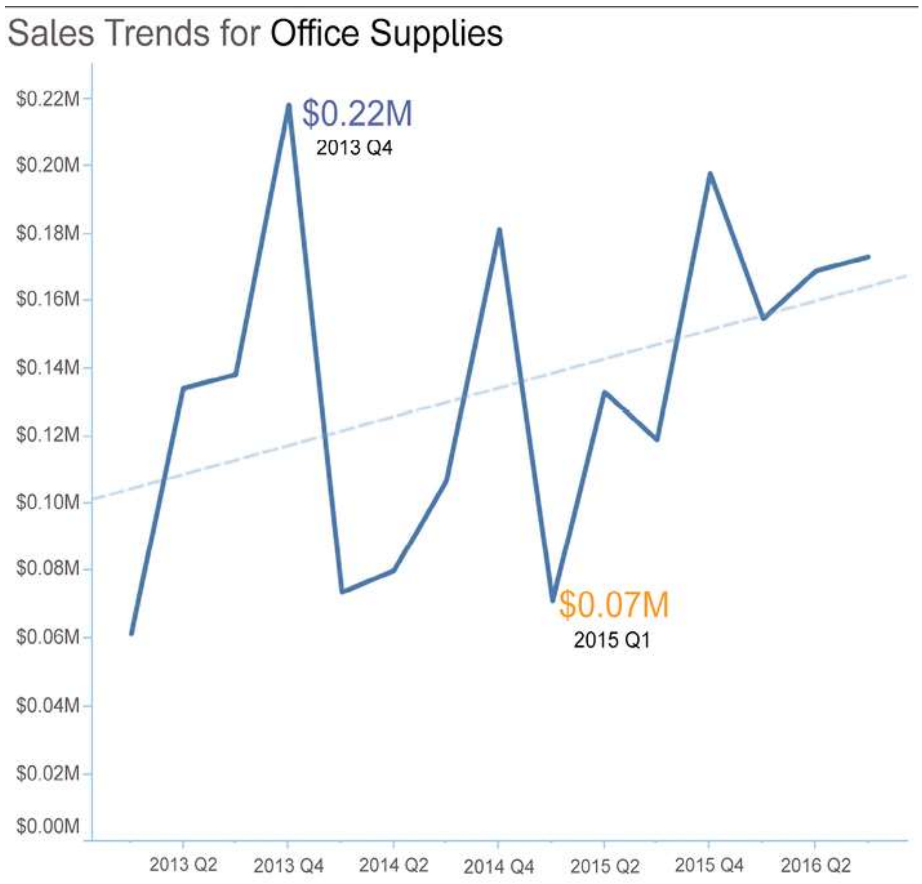


Figure 7.18: Formatting can make a visualization less cluttered and communicate data more effectively

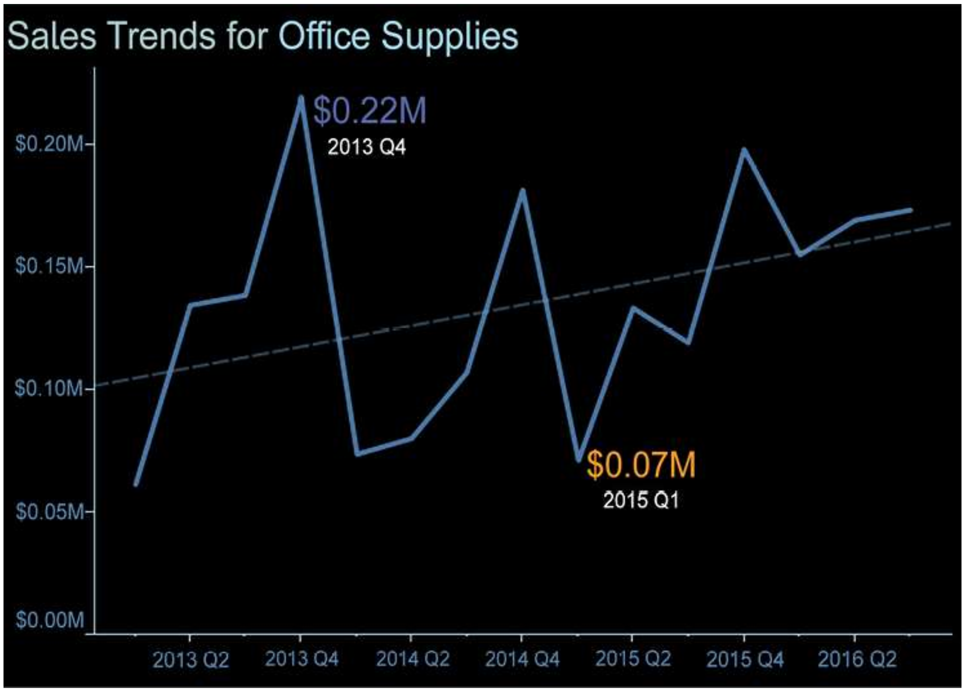


Figure 7.19: A dark background can be set by formatting the shading of a view

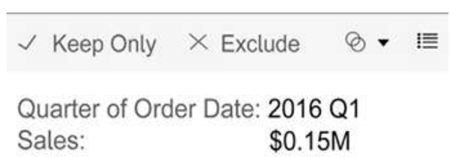


Figure 7.20: Default tooltip

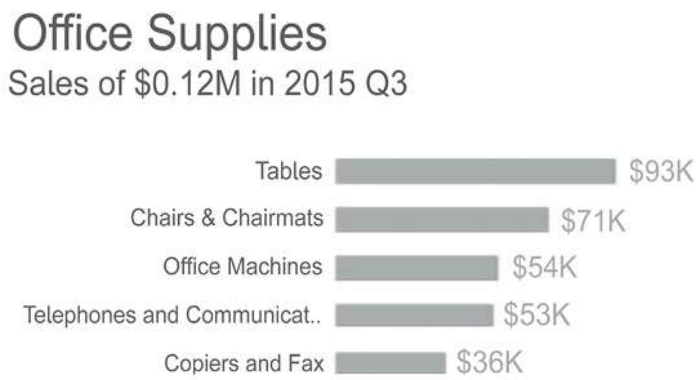


Figure 7.21: Customized tooltip

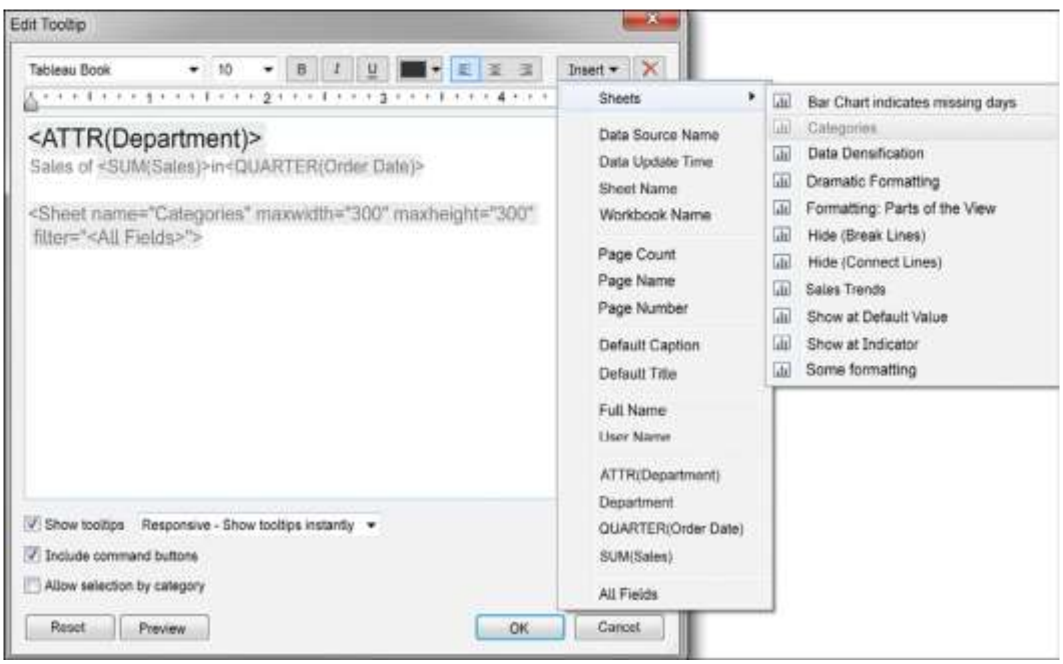


Figure 7.22: Tooltip editor

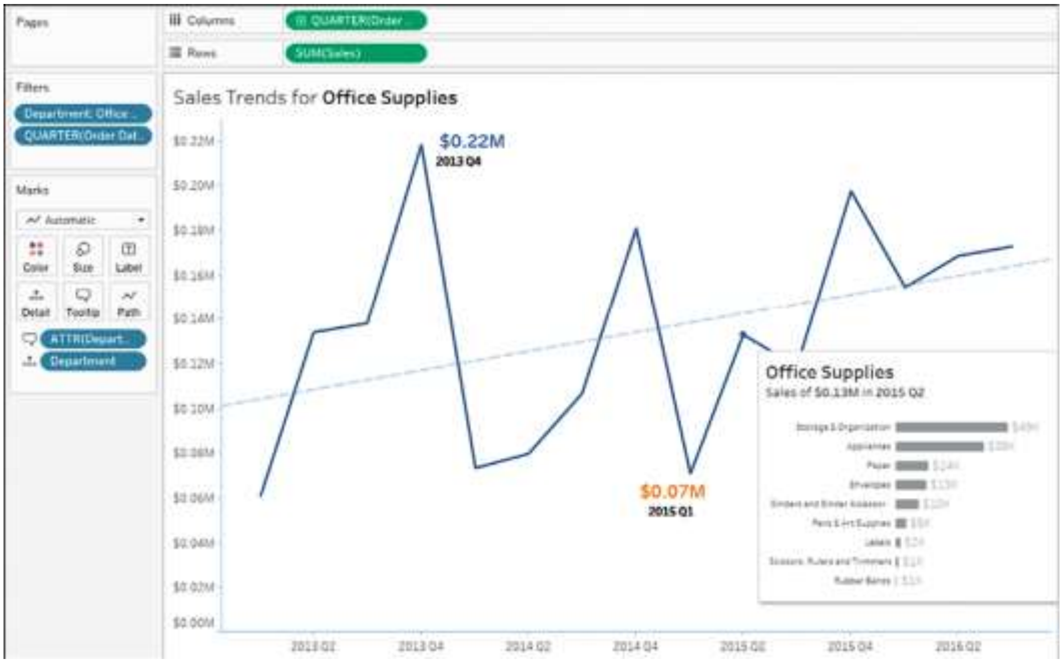


Figure 7.23: Viz in Tooltip

Tag for Viz in Tooltip

```
<Sheet name="Categories" maxwidth="300" maxheight="300"
filter="<All Fields>">
```

Links

For a complete list of custom date format string options, check out https://onlinehelp.tableau.com/current/pro/desktop/en-us/dates_custom_date_formats.htm.

Chapter 8

Figures

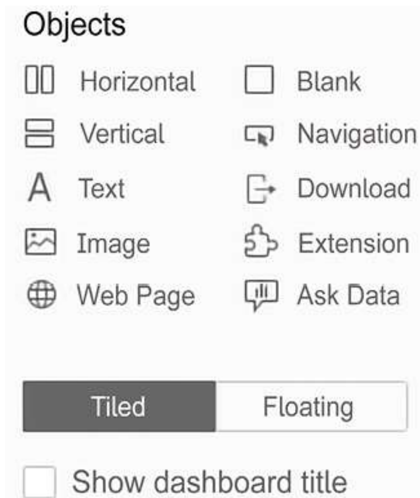


Figure 8.1: Objects available to add to a dashboard

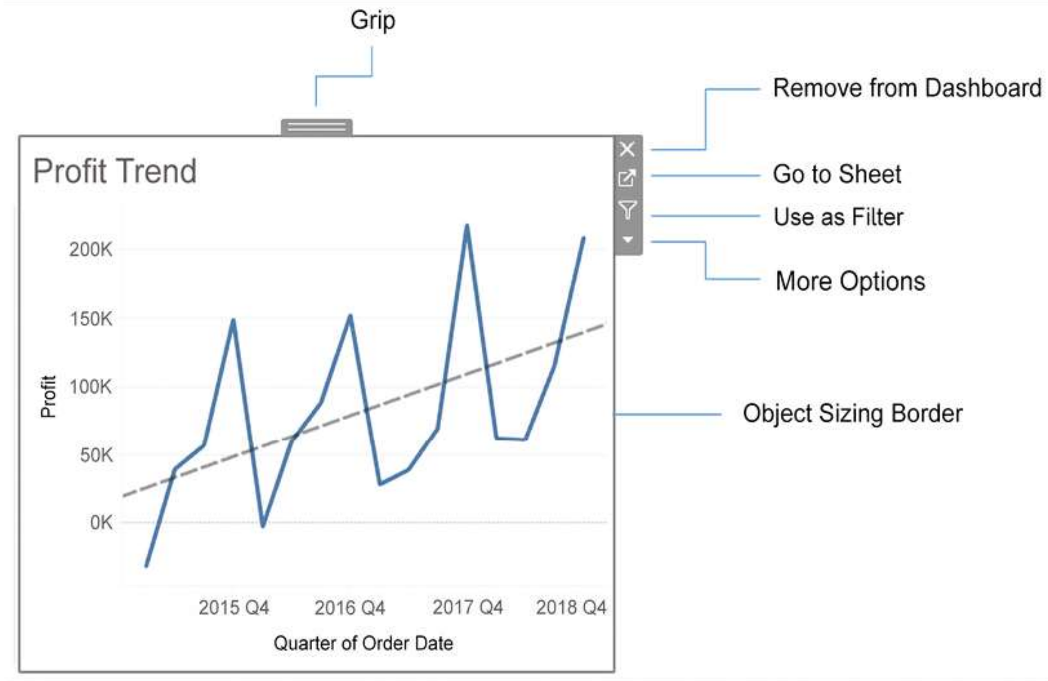


Figure 8.2: Various controls become accessible once you select a dashboard object

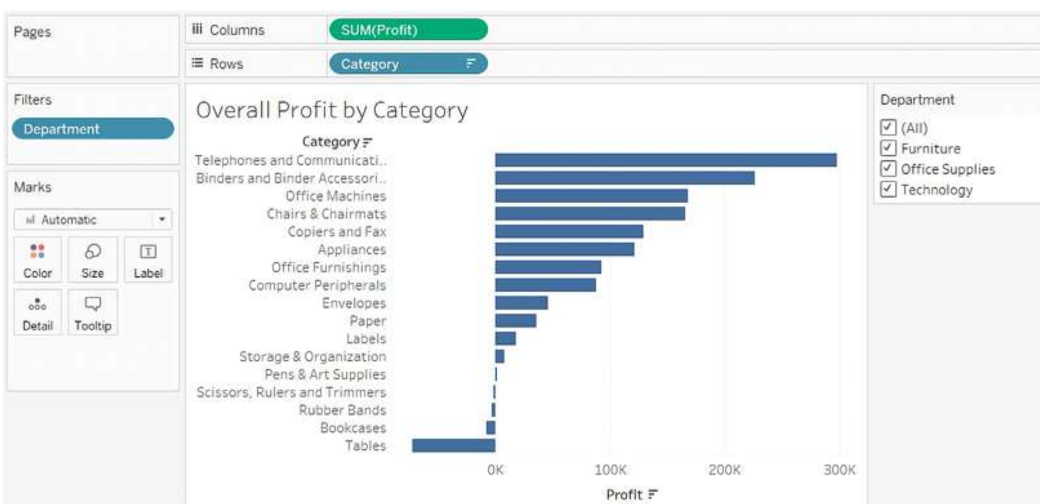


Figure 8.3: A bar chart showing the sum of profit by category with Department as a filter

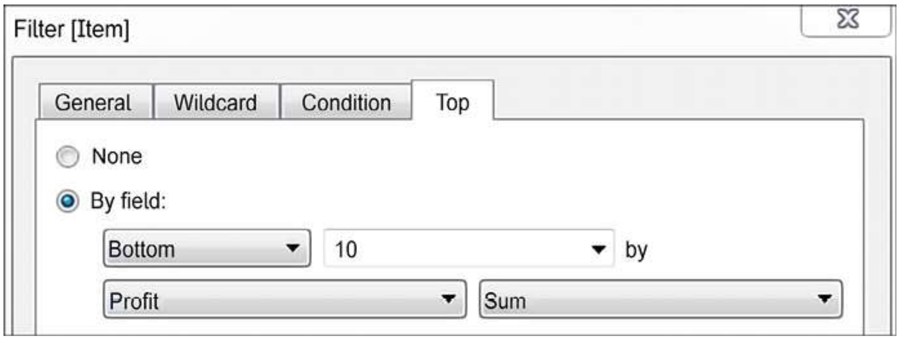


Figure 8.4: Use the Top tab to set the number of items to display

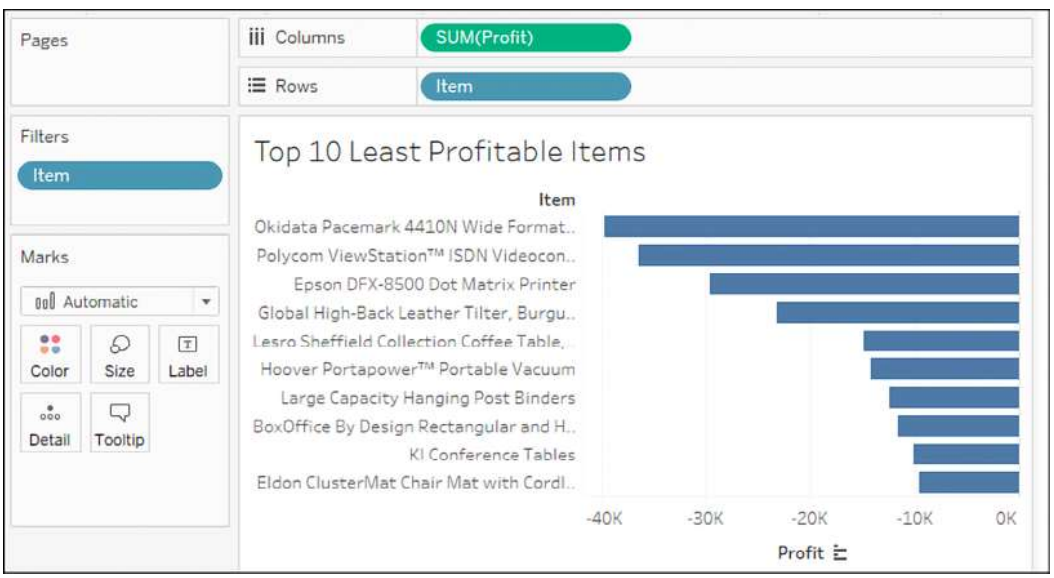


Figure 8.5: The resulting bar chart shows the top 10 least profitable items

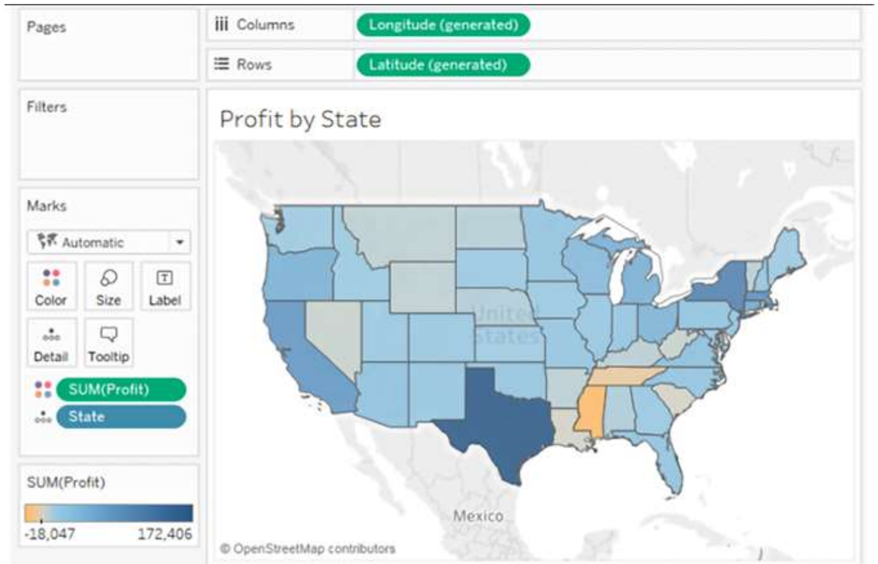


Figure 8.6: A filled map showing profit by state

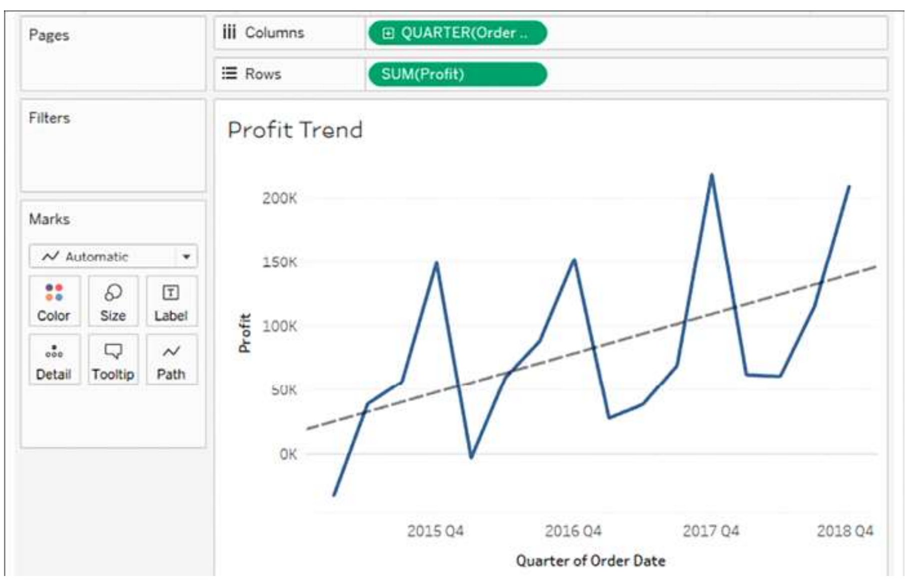


Figure 8.7: A line chart showing the trend of profit by quarter

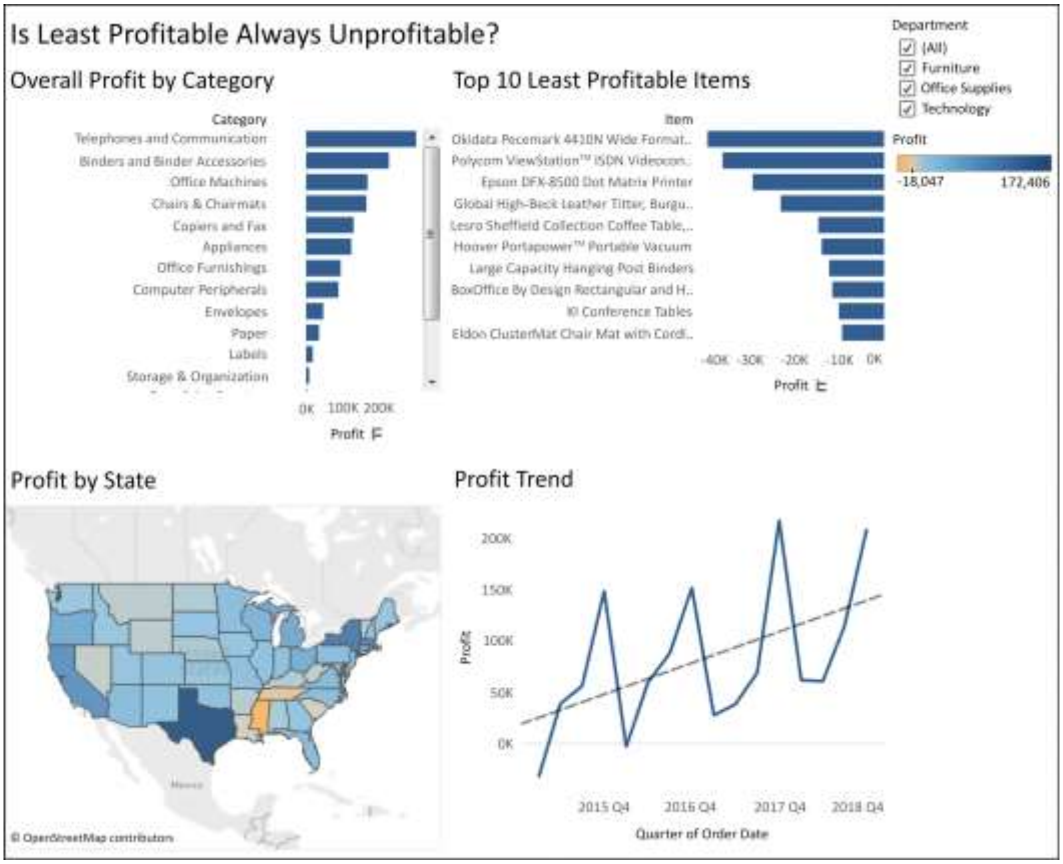


Figure 8.8: All views are placed on the dashboard

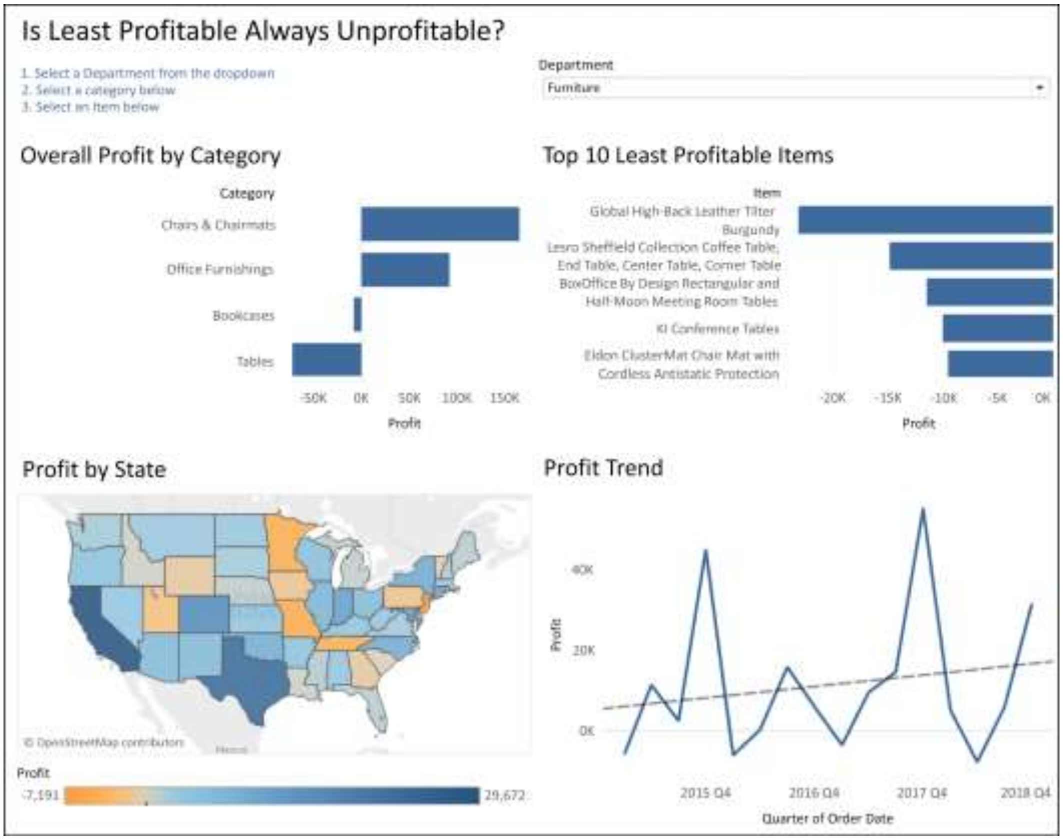


Figure 8.9: The polished dashboard with rearranged and resized objects

Figure 8.10: The Use As Filter control on the Profit by Category view

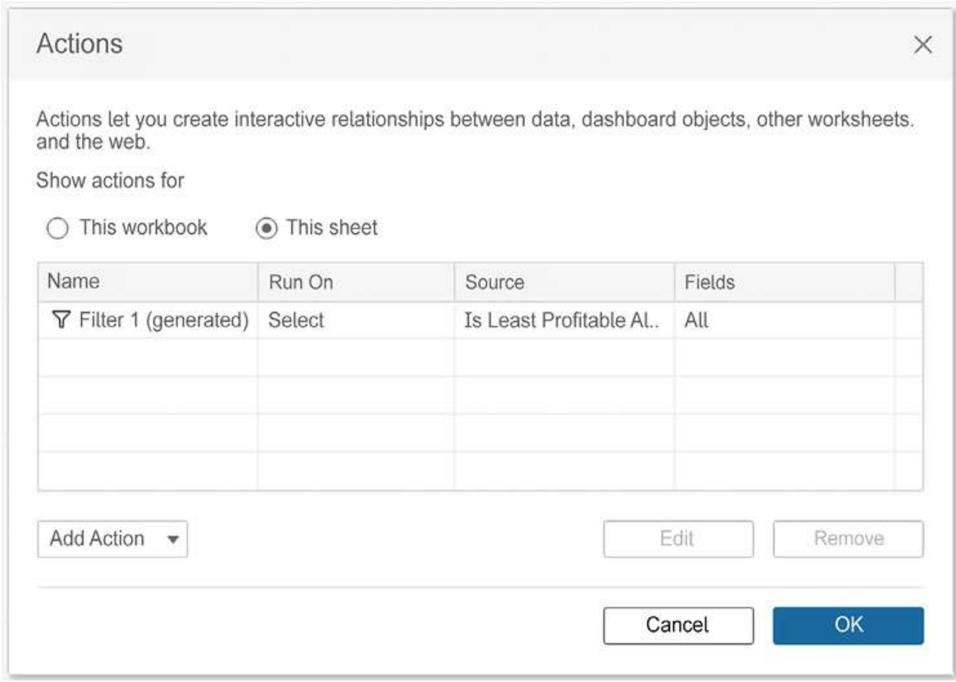


Figure 8.11: Filter 1 (generated) was created when the filter control was clicked

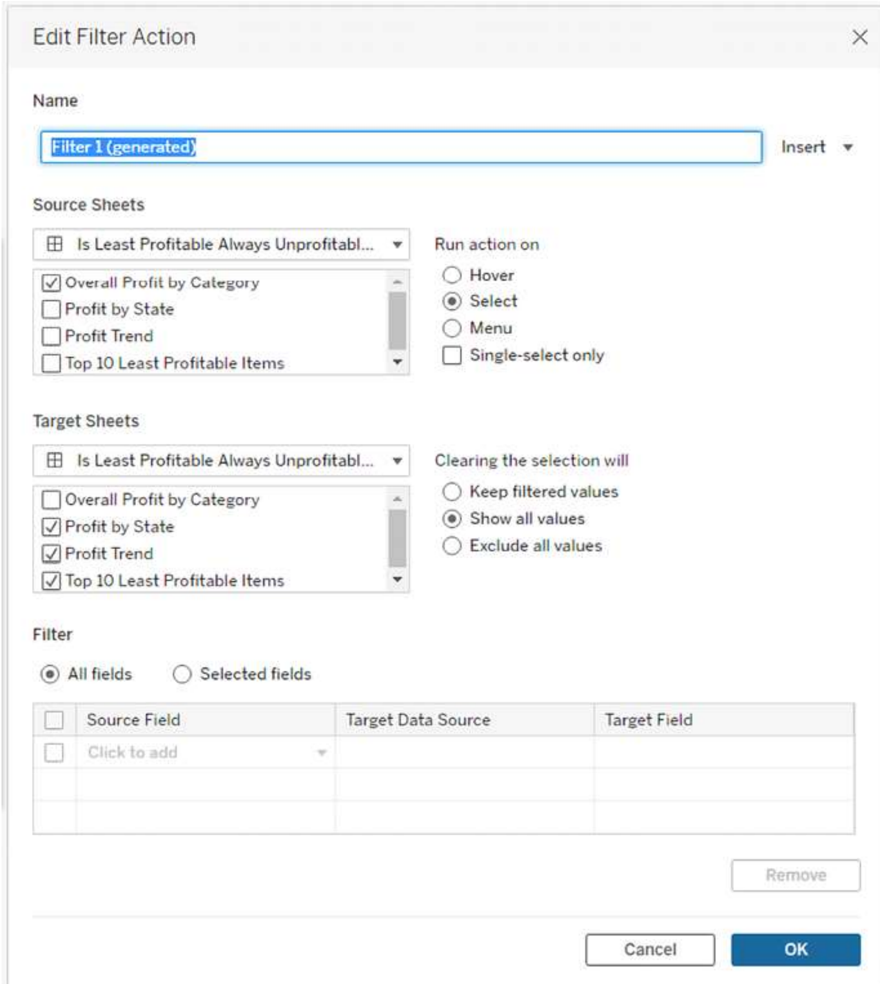


Figure 8.12: Setting options for the Filter by Item action

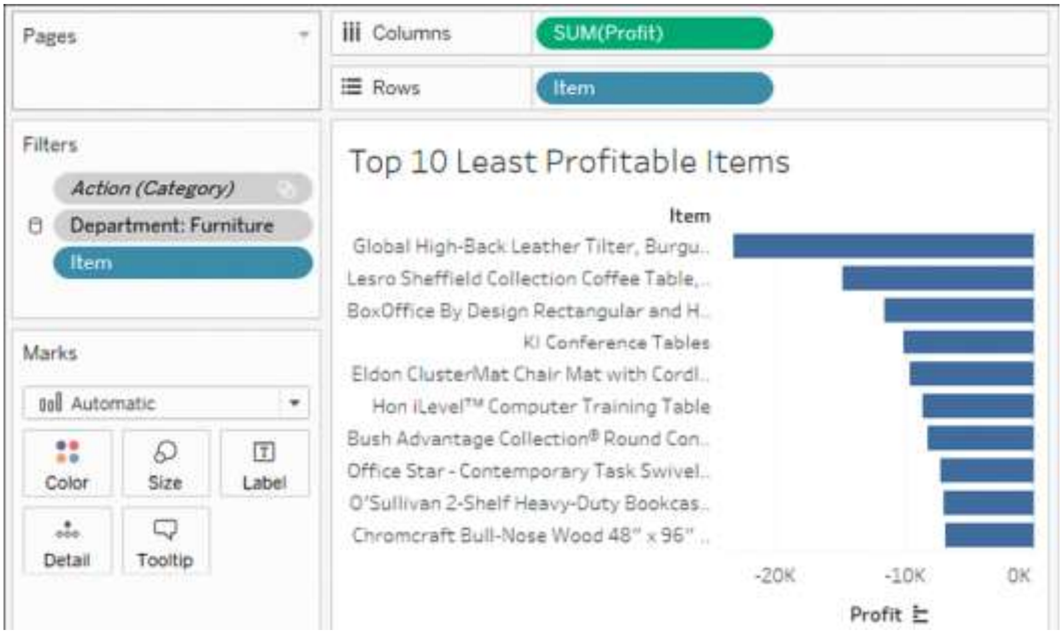


Figure 8.13: The 10 least profitable items will be within the context of the Action (Category) and Department filters

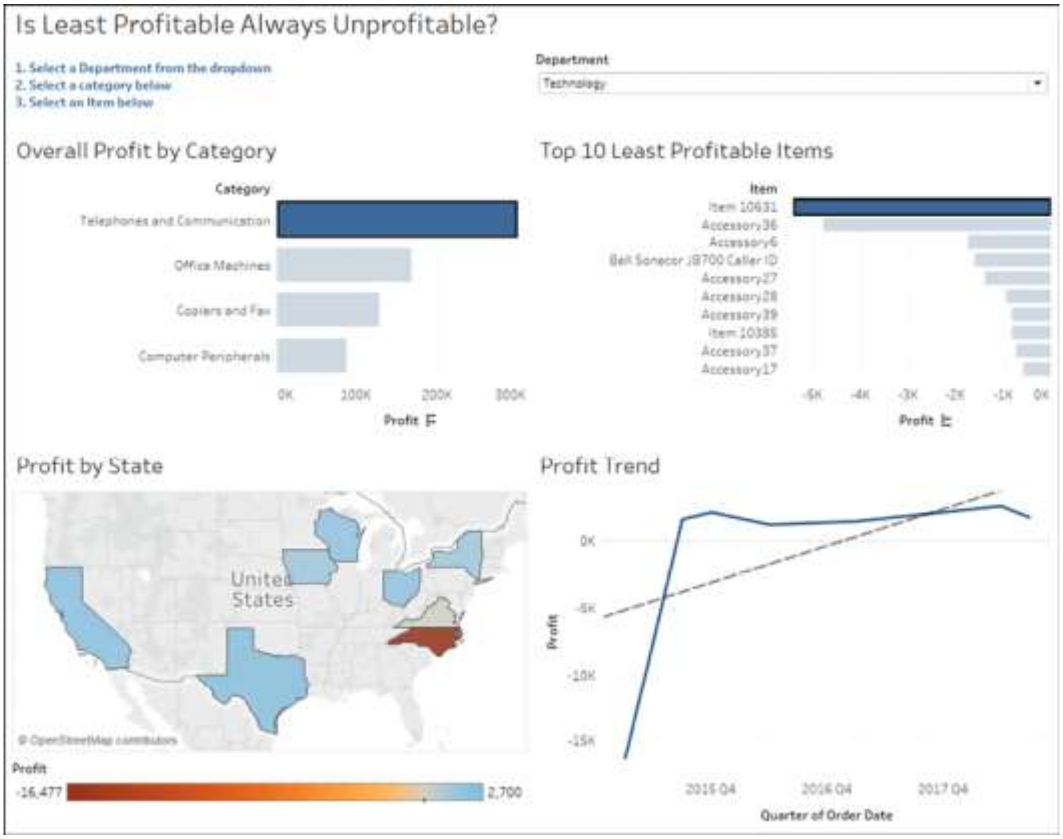


Figure 8.14: The final dashboard with filters triggered by selection

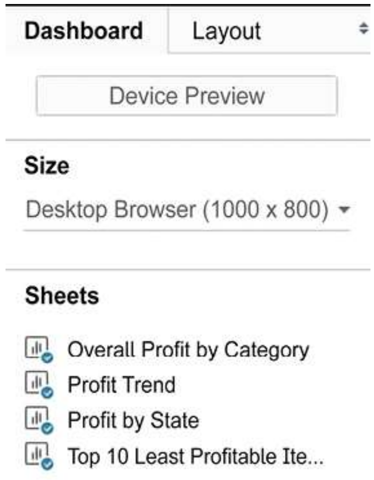


Figure 8.15: The Device Preview option allows you to design and preview your dashboard for other devices

Figure 8.16: Customizable options for devices

Figure 8.17: Each layout can be configured with various options

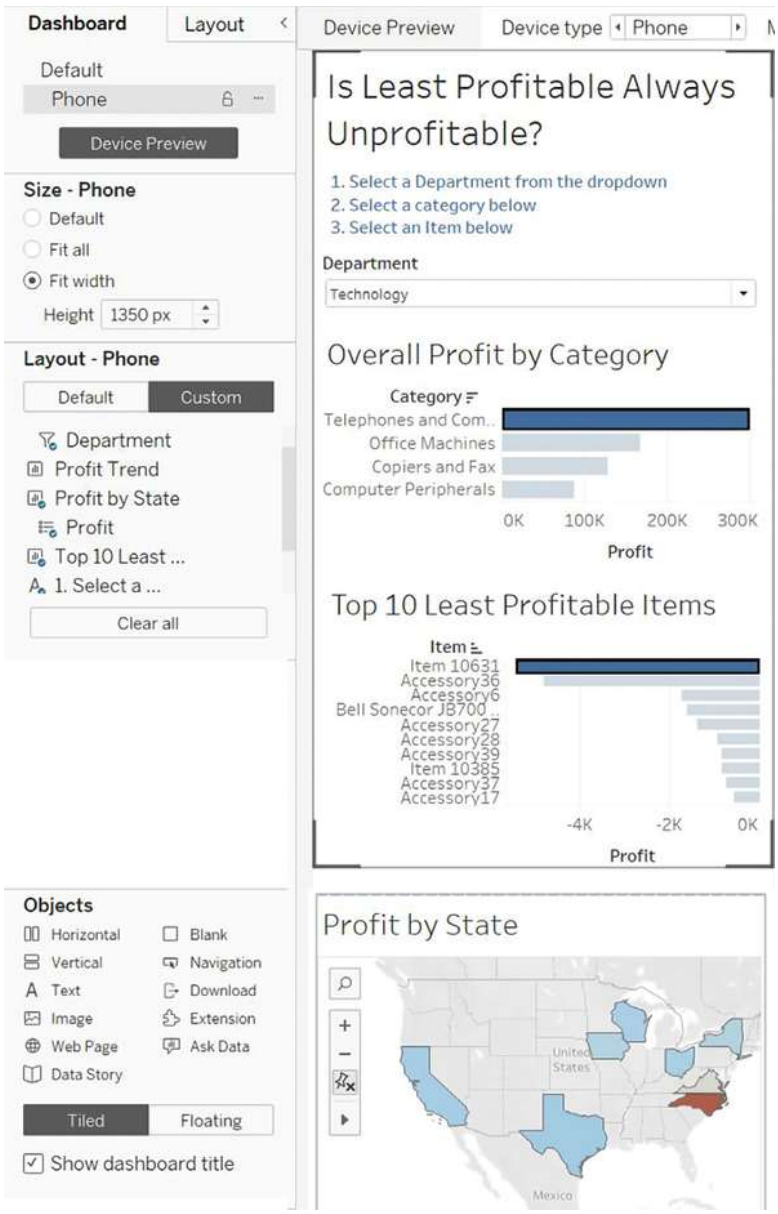


Figure 8.18: The phone layout of the dashboard

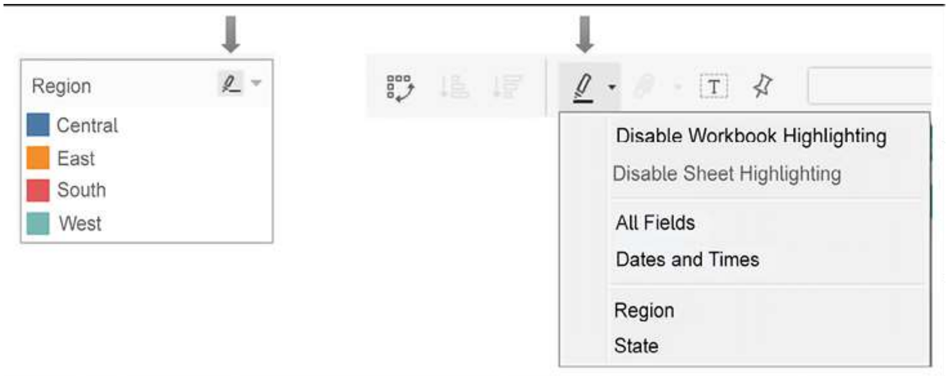


Figure 8.19: Options for enabling highlighting

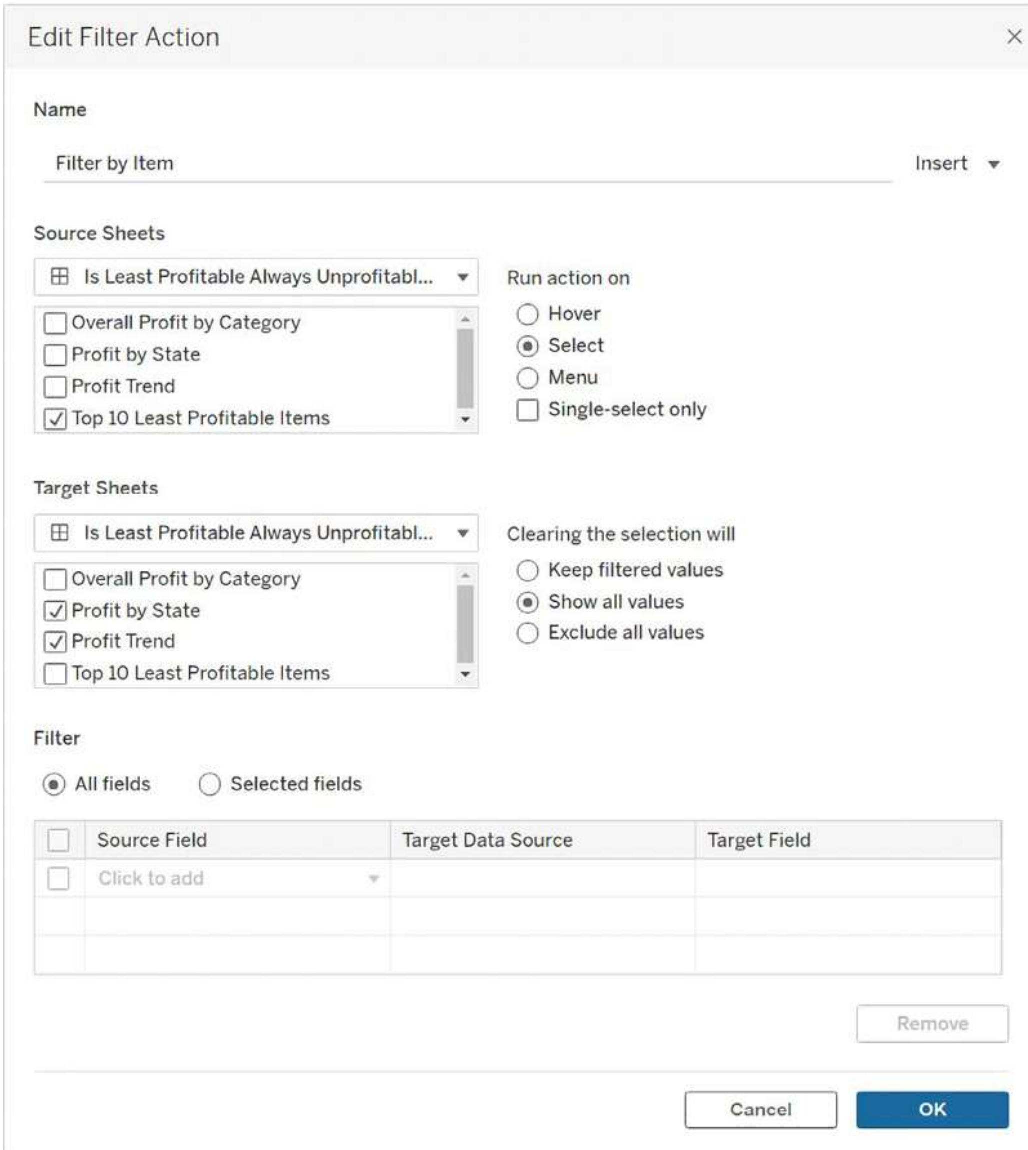


Figure 8.20: Options for filter actions

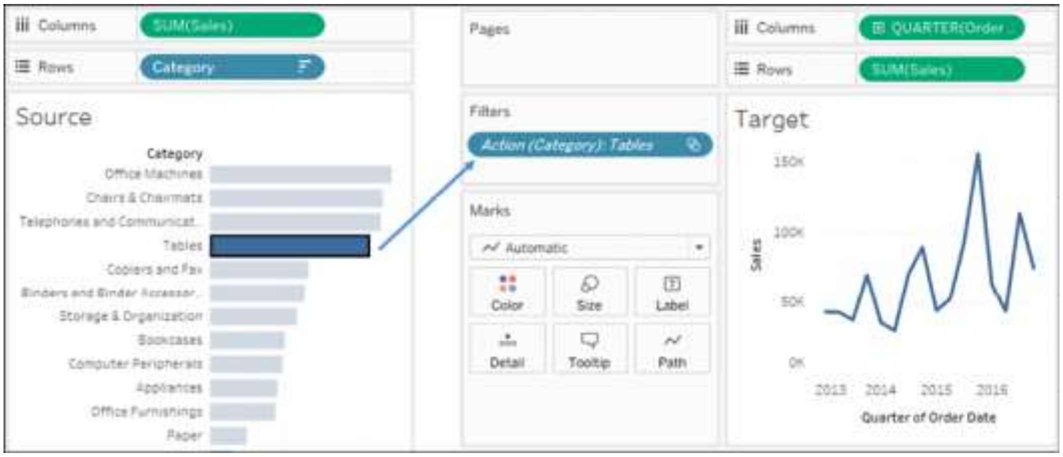


Figure 8.21: Clicking the bar for Tables passes Category as a filter to the Target sheet

Figure 8.22: Clicking the square for the intersection of Office Machines and West passes both dimensional values as a single action filter to the target

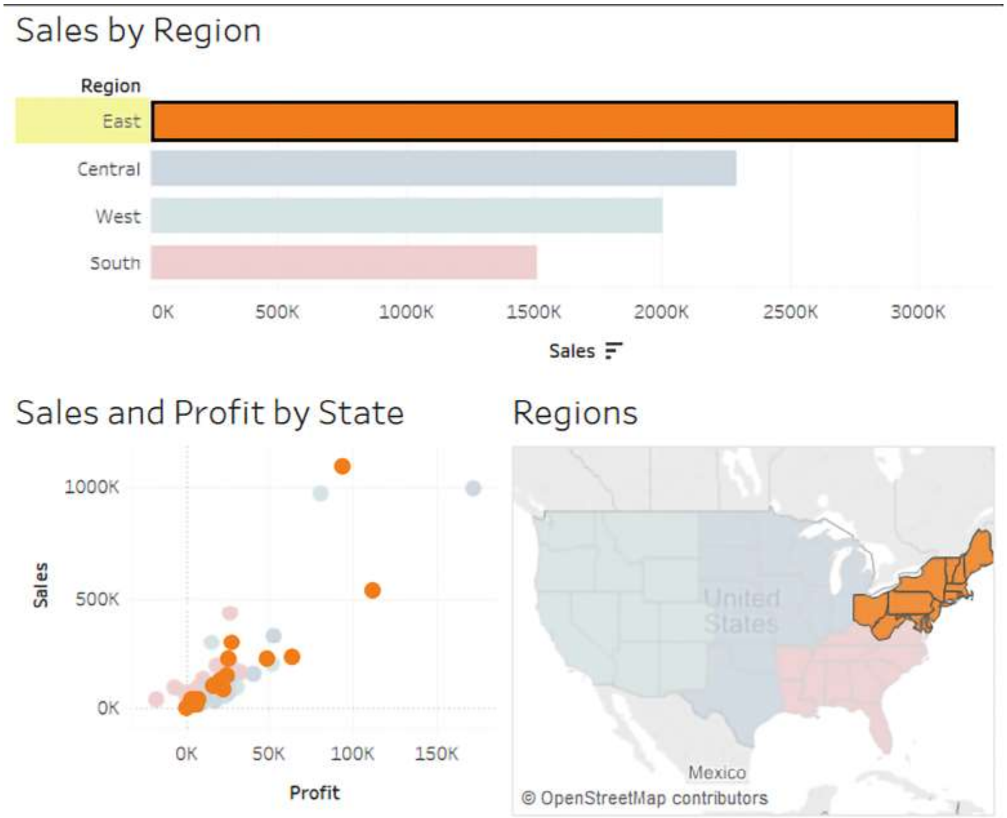


Figure 8.23: Clicking the bar for East has highlighted all other marks associated with that dimensional value

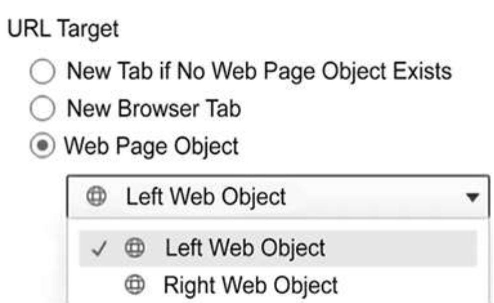


Figure 8.24: Options for a URL action

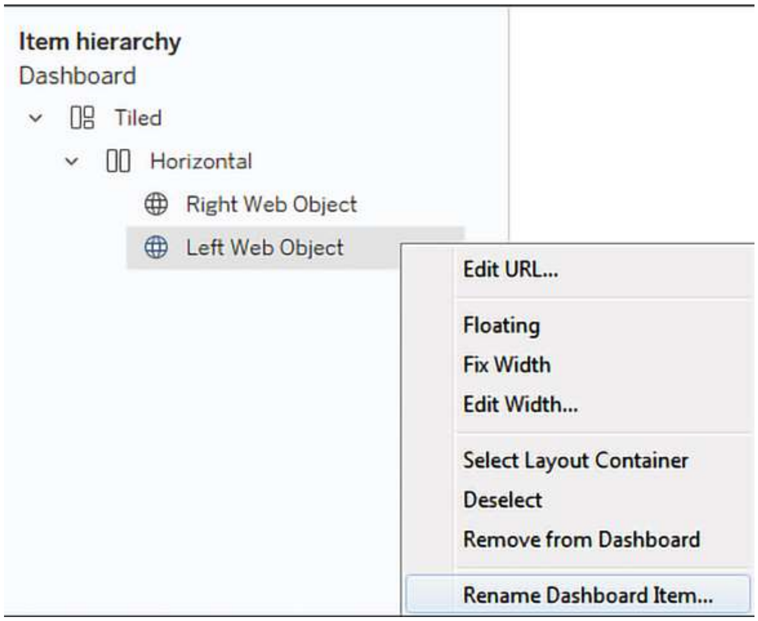


Figure 8.25: Dashboard objects can be renamed using the item hierarchy and the right-click context menu

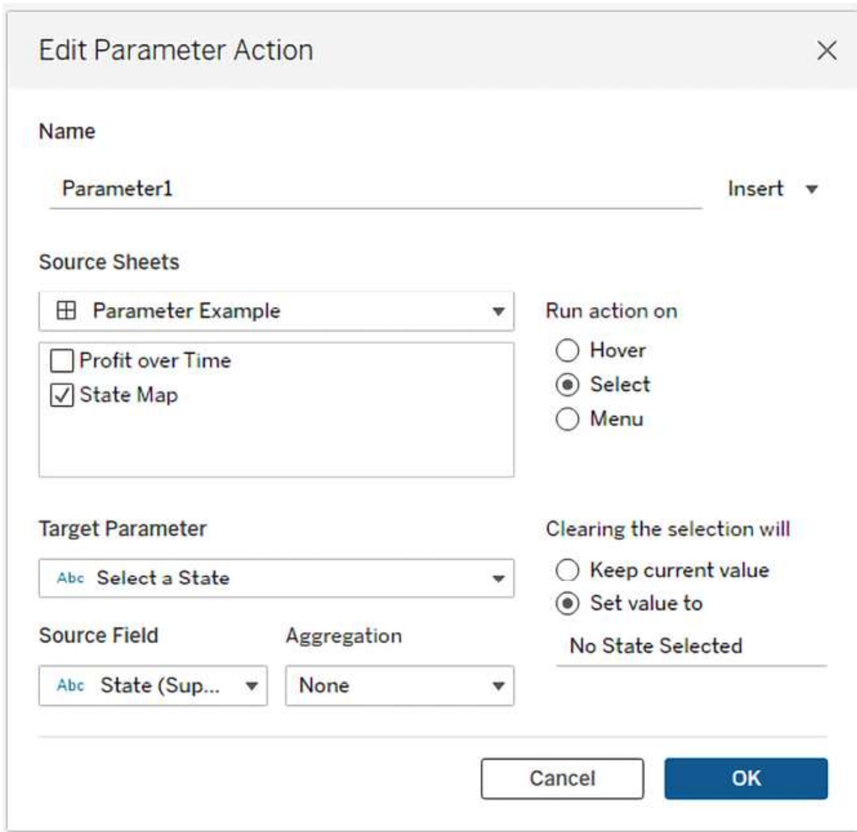


Figure 8.26: Options for a parameter action

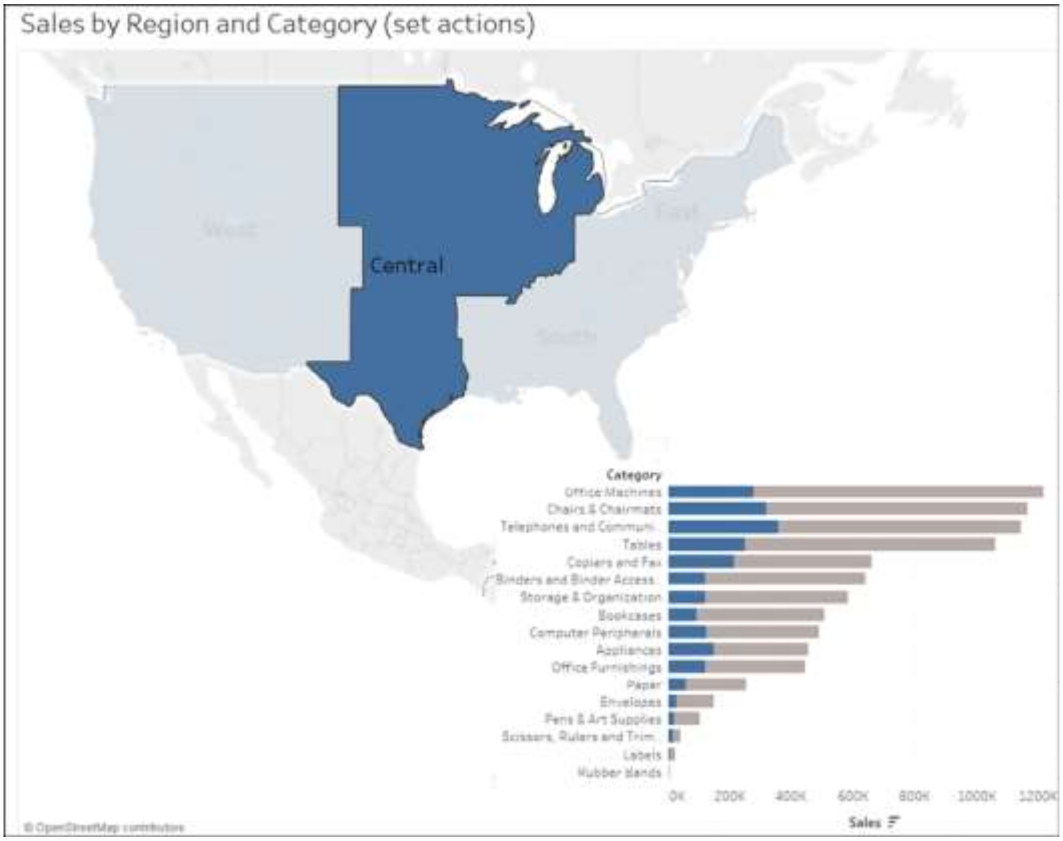


Figure 8.27: The set action allows brushing: highlighting the portion of the bars that belong to the selection

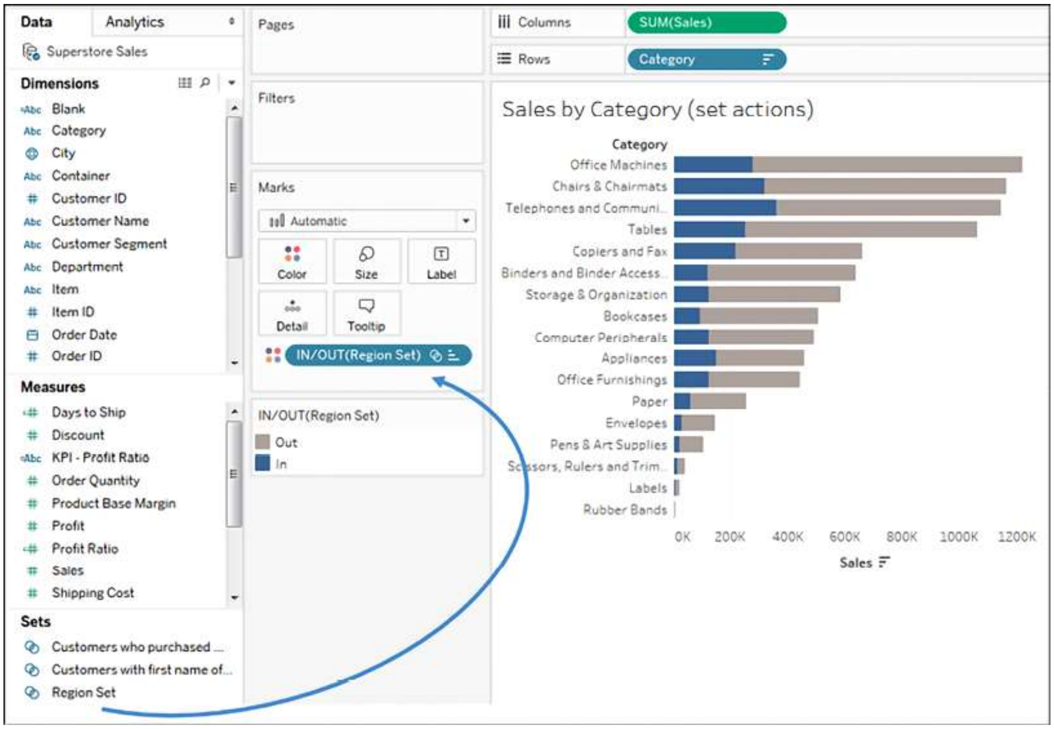


Figure 8.28: Drag and drop the set on Color to show the difference between In and Out of the set

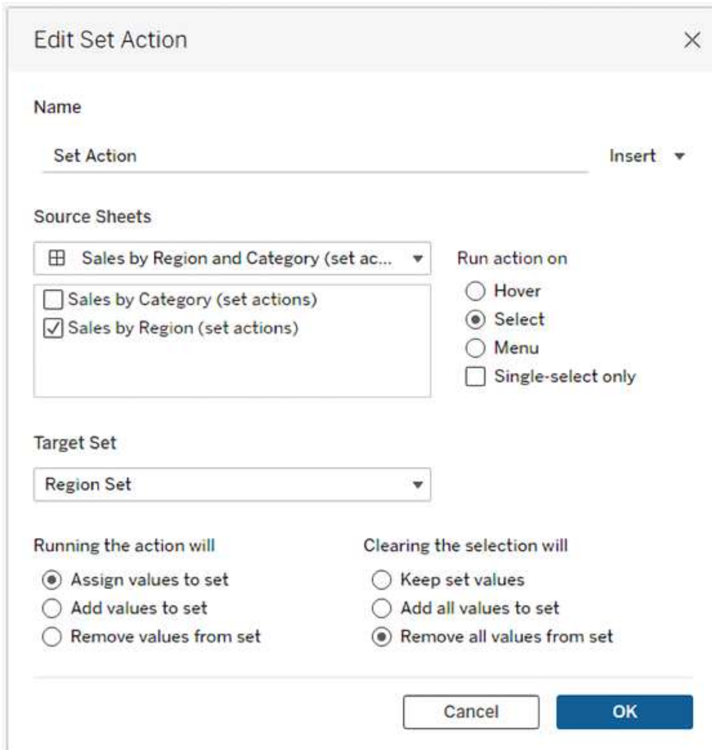


Figure 8.29: Options for the set action

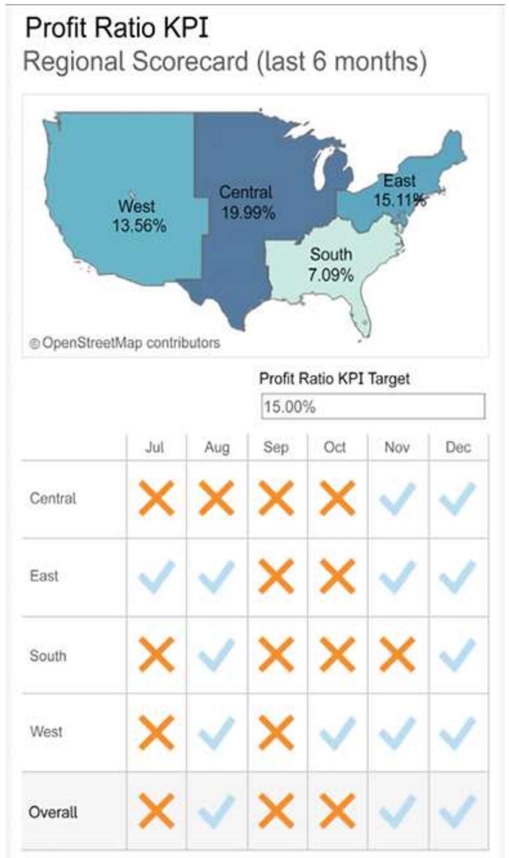


Figure 8.30: A simple profit KPI dashboard

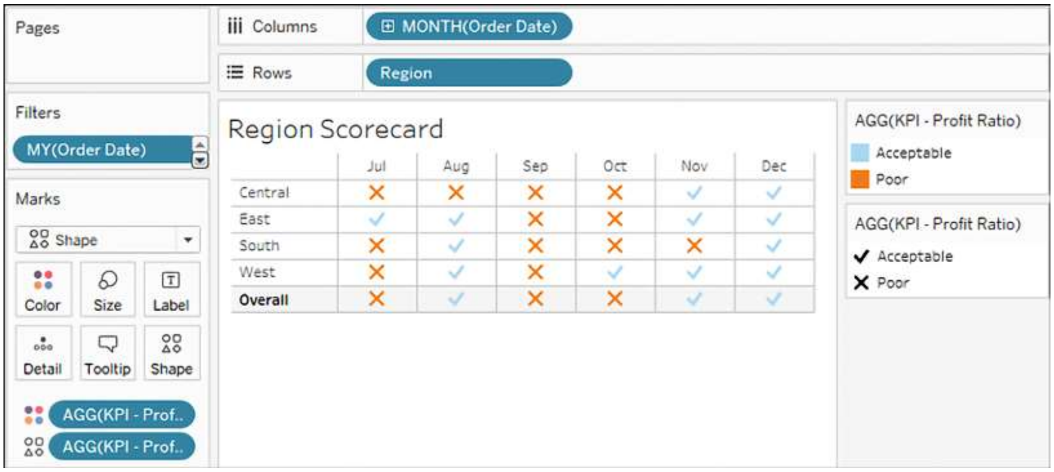


Figure 8.31: This view defines the Region scorecard showing Acceptable versus Poor results per region per month

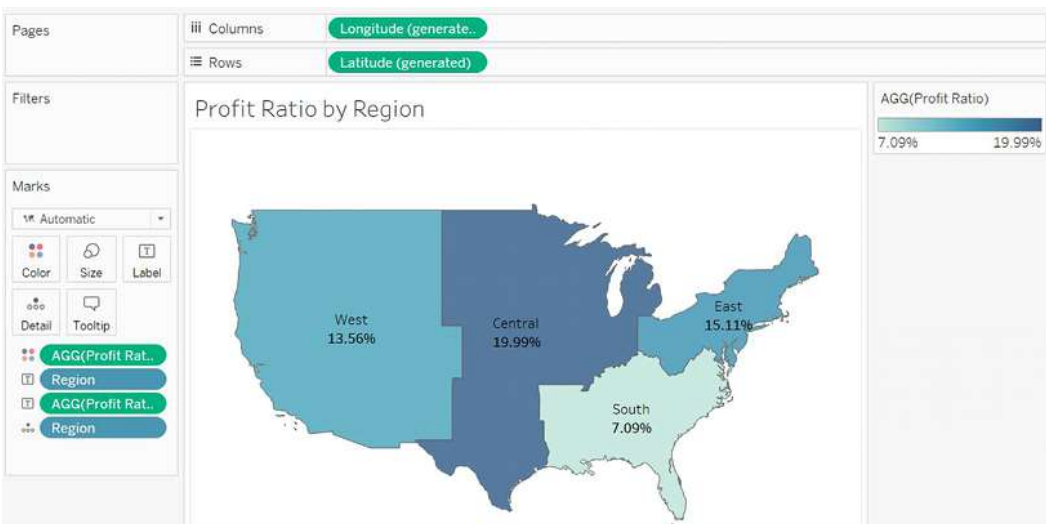


Figure 8.32: The filled map shows profit by region

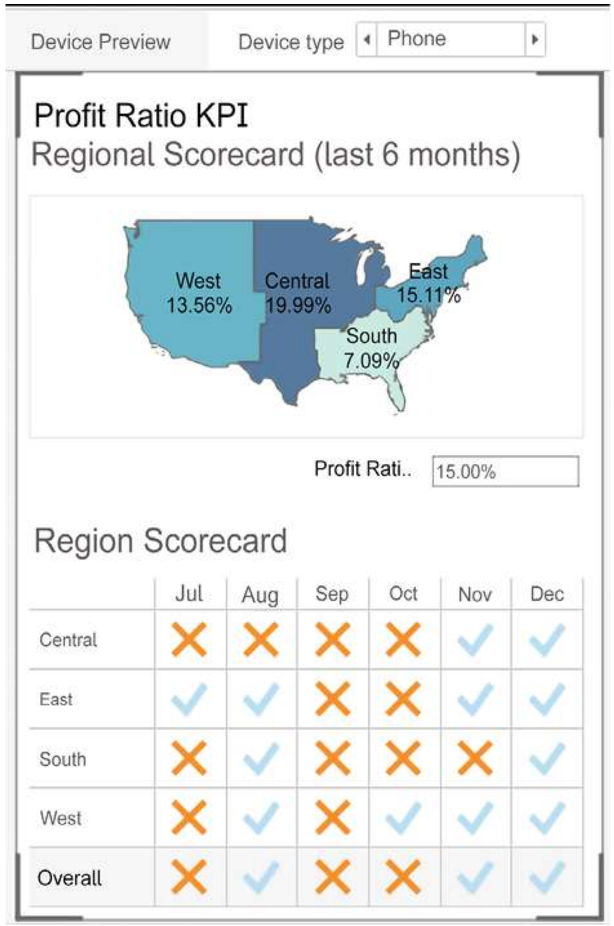


Figure 8.33: A phone layout for the KPI dashboard

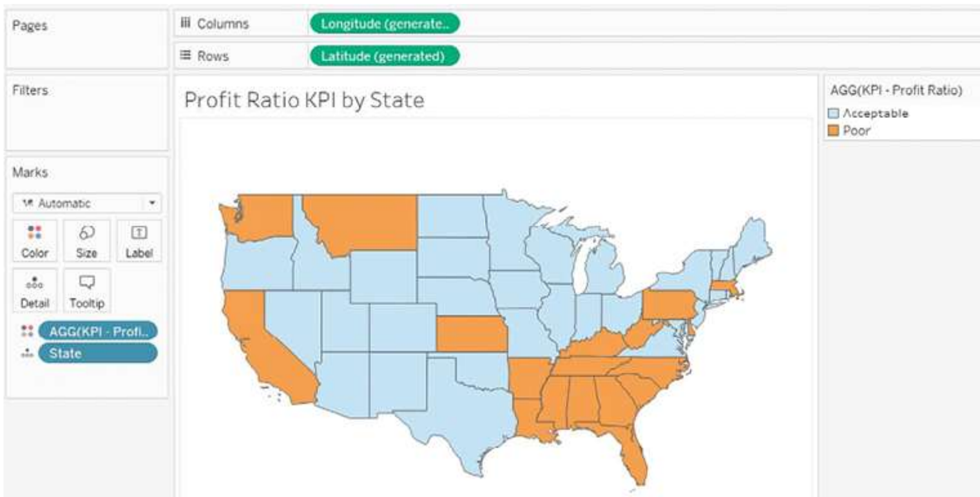


Figure 8.34: Profit Ratio KPI by State uses color encoding to show Acceptable versus Poor

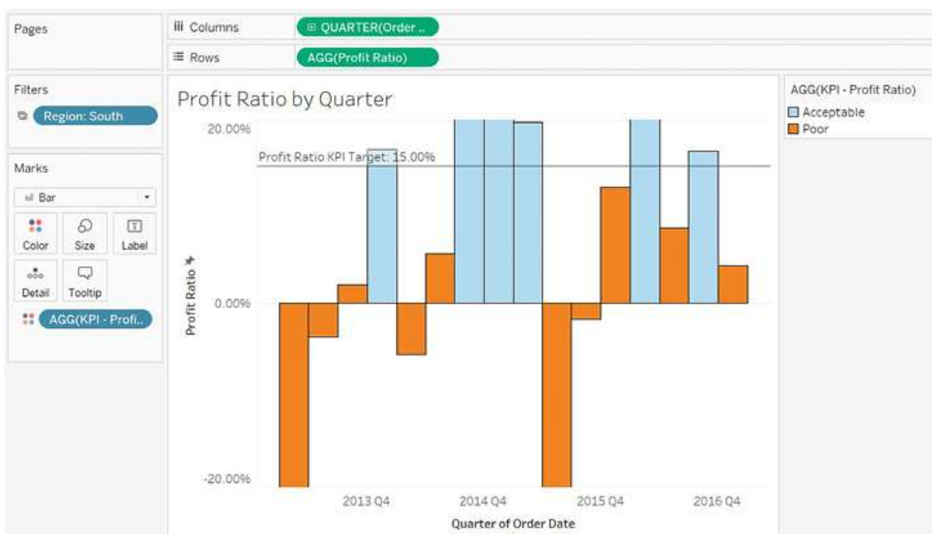


Figure 8.35: Profit Ratio by Quarter shows whether a given quarter was Acceptable or Poor based on the target

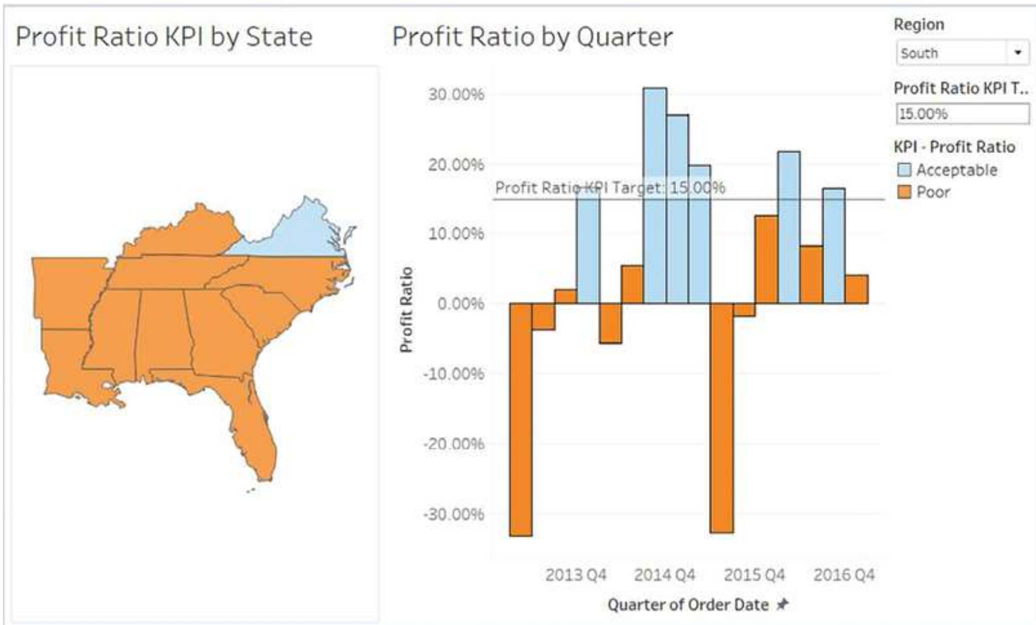


Figure 8.36: The Profit Ratio KPI by State and by Quarter views on the same dashboard

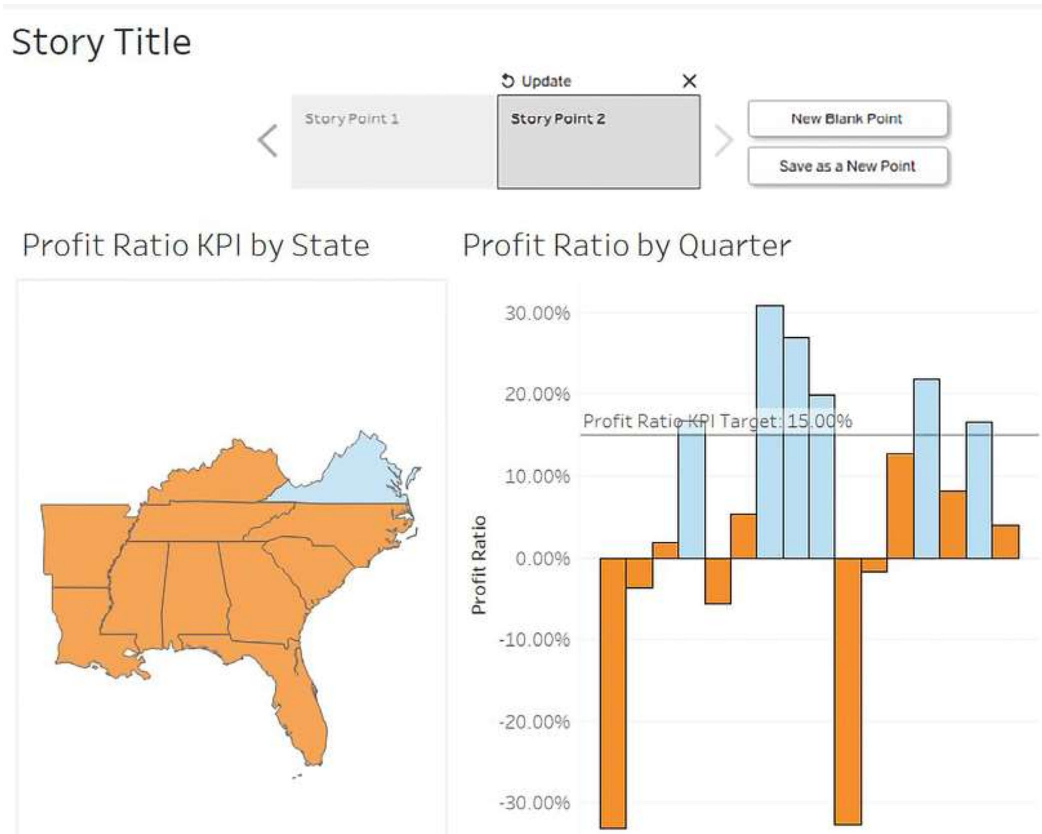


Figure 8.37: Adding a story point

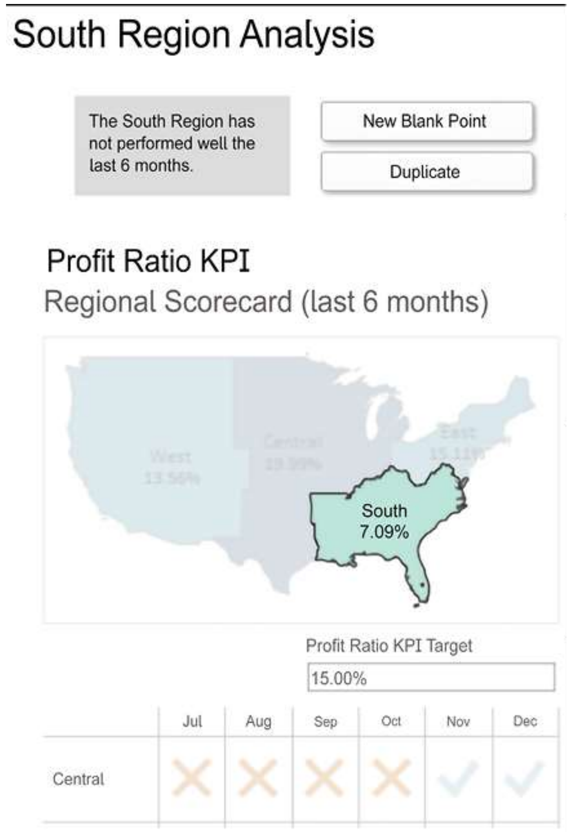


Figure 8.38: The first story point highlights performance in the South region

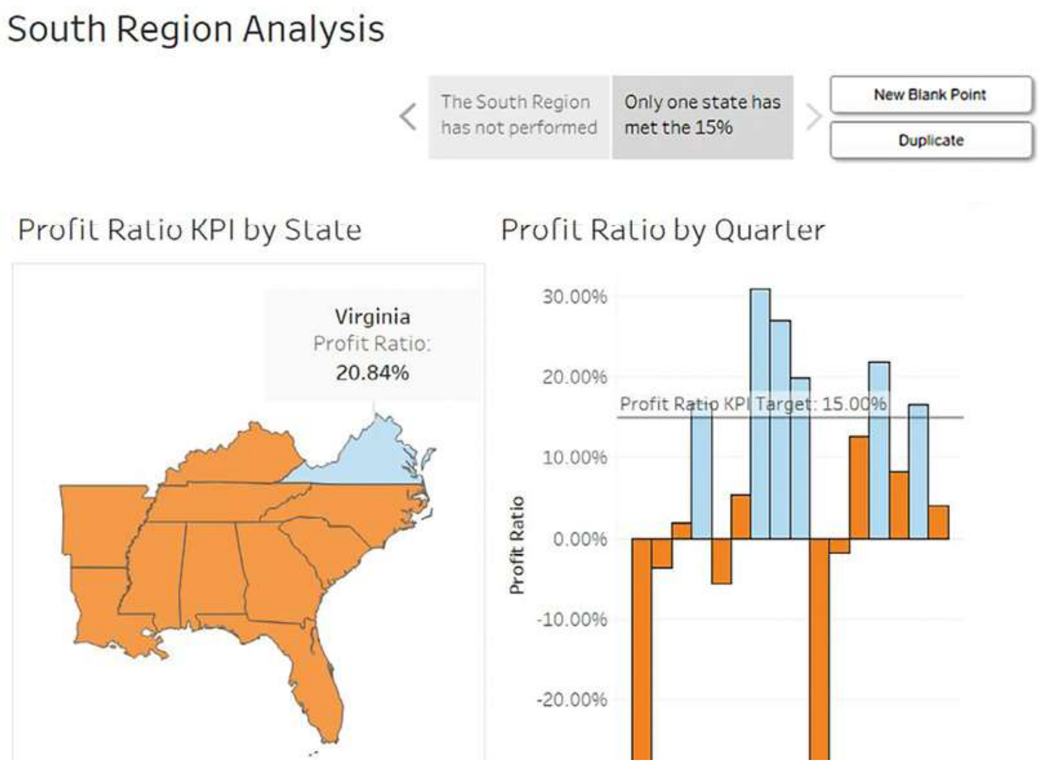


Figure 8.39: The second story point dives into the details

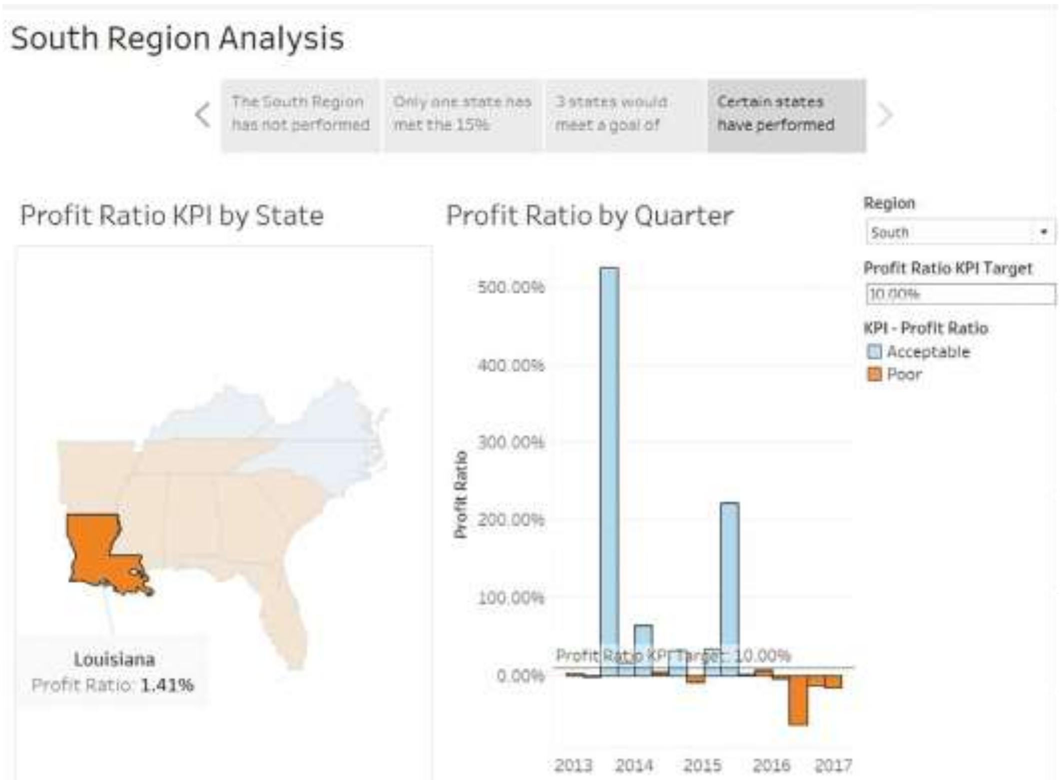


Figure 8.40: This story point highlights quarterly results for a single state

Code

Code 8.1

```
IF [Profit Ratio] >= [Profit Ratio KPI Target]
THEN "Acceptable"
ELSE "Poor"
END
```

Code 8.2 – Profit ratio

```
SUM([Profit]) / SUM([Sales])
```

Links

- This definition of dashboard is drawn from Few's paper *Dashboard Confusion*, which can be read here: https://www.perceptualedge.com/articles/ie/dashboard_confusion.pdf.
- Viz of the day: http://www.tableau.com/public/community/viz-of-the-day

Chapter 9

Figures

Figure 9.1: Population values for Afghanistan and Australia over time

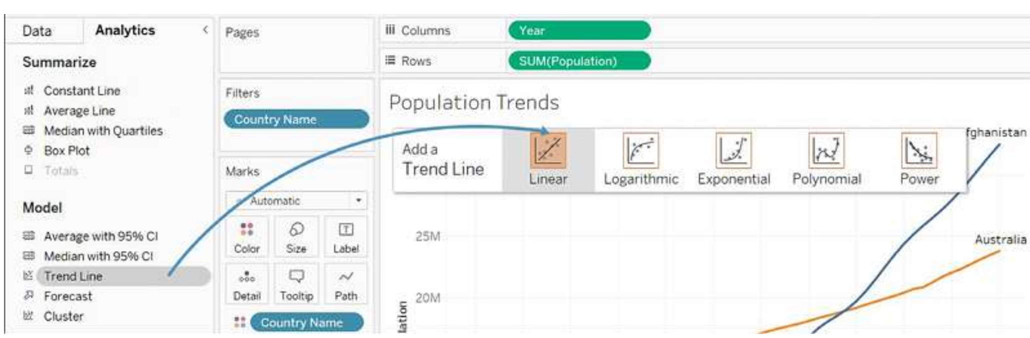


Figure 9.2: Adding a trend line to a view by dragging and dropping from the Analytics pane

Figure 9.3: Each trend line shows the overall trend for the respective country

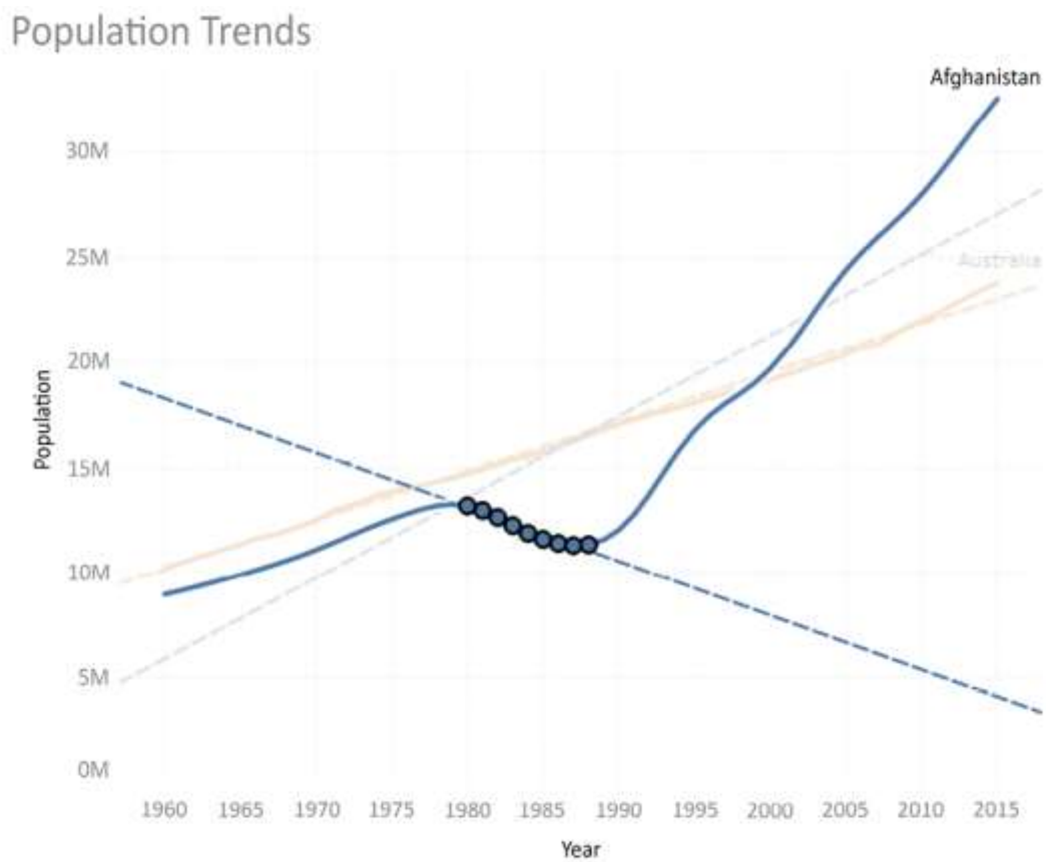


Figure 9.4: The default settings specify that trend lines will be drawn for selections

Figure 9.5: Here, the discrete dimension Period creates three separate time periods and a trend for each one

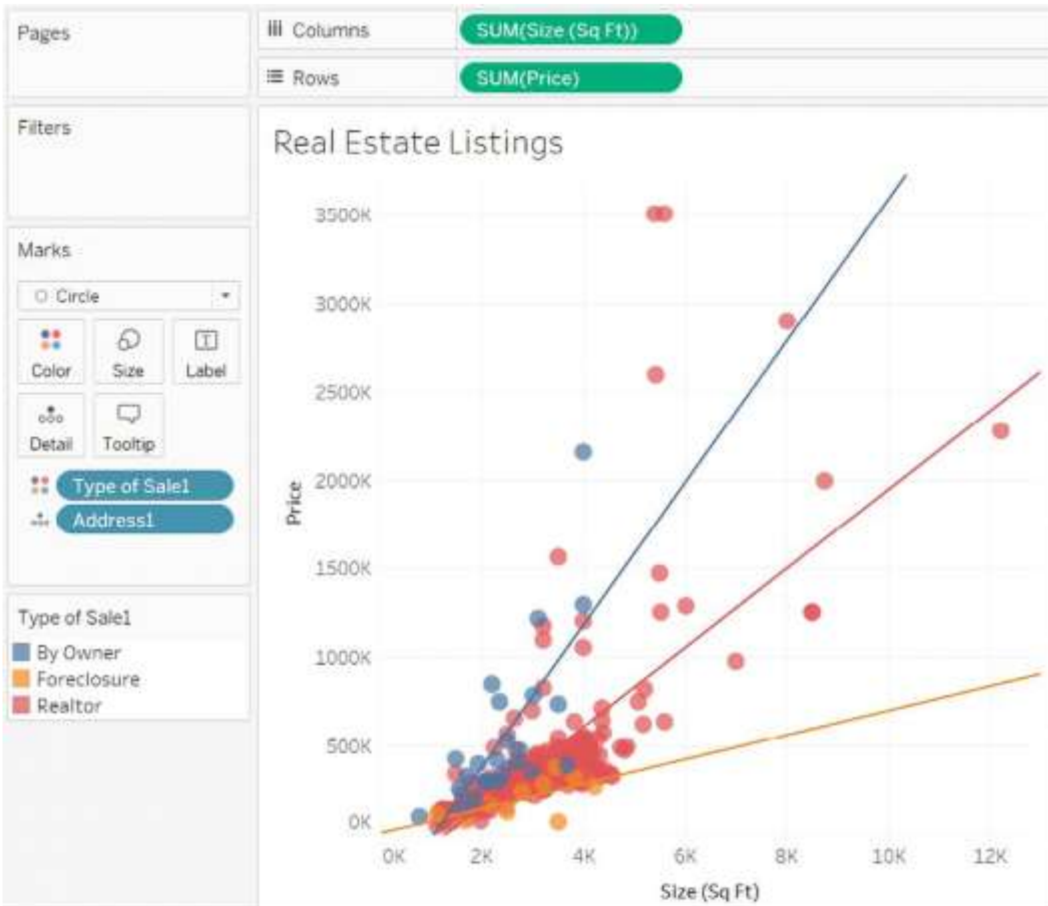


Figure 9.6: Trend lines on a scatterplot are often useful for better understanding correlation and outliers

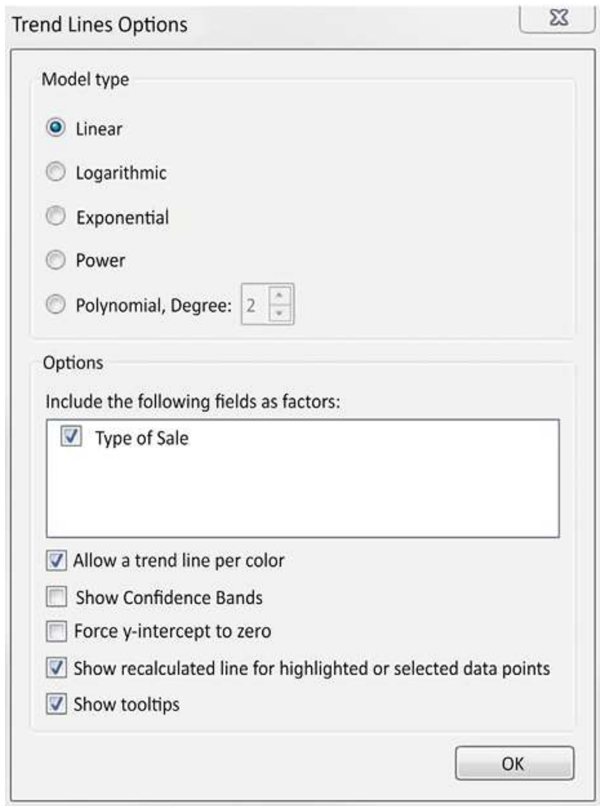


Figure 9.7: Tableau offers many options for configuring trend lines

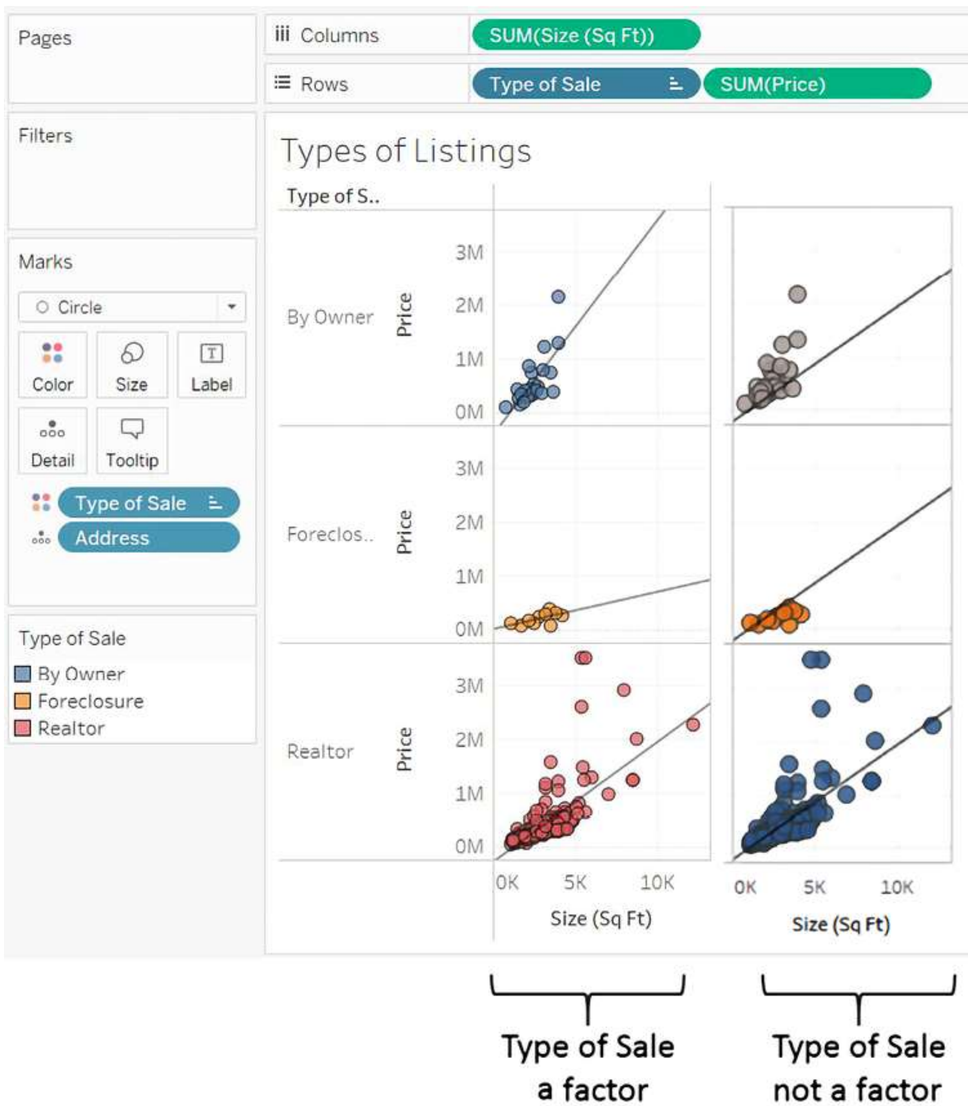


Figure 9.8: Including a field as a factor tells Tableau whether it contributes to the trend model

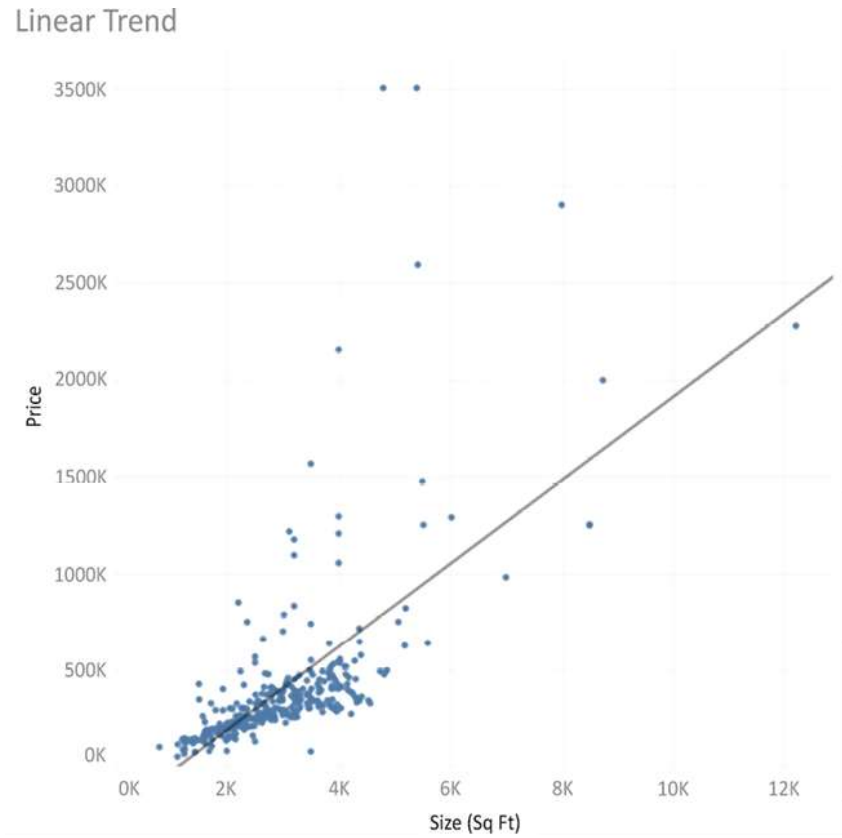


Figure 9.9: Linear trend

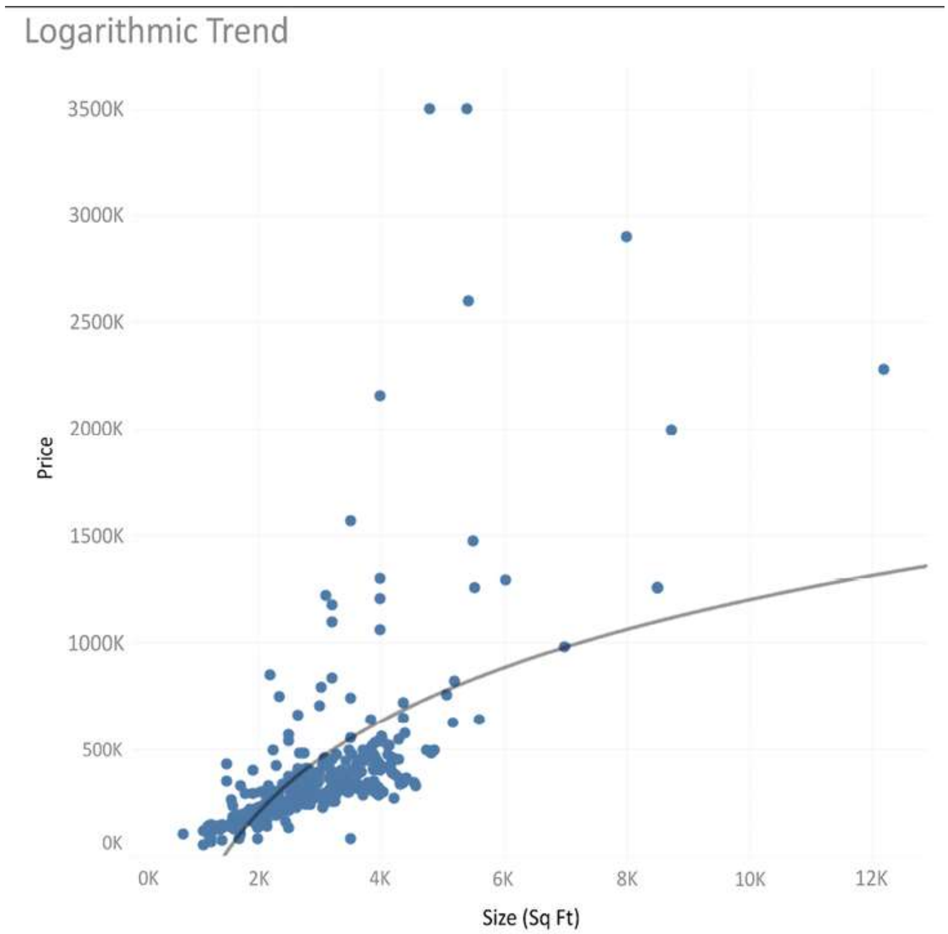


Figure 9.10: Logarithmic trend

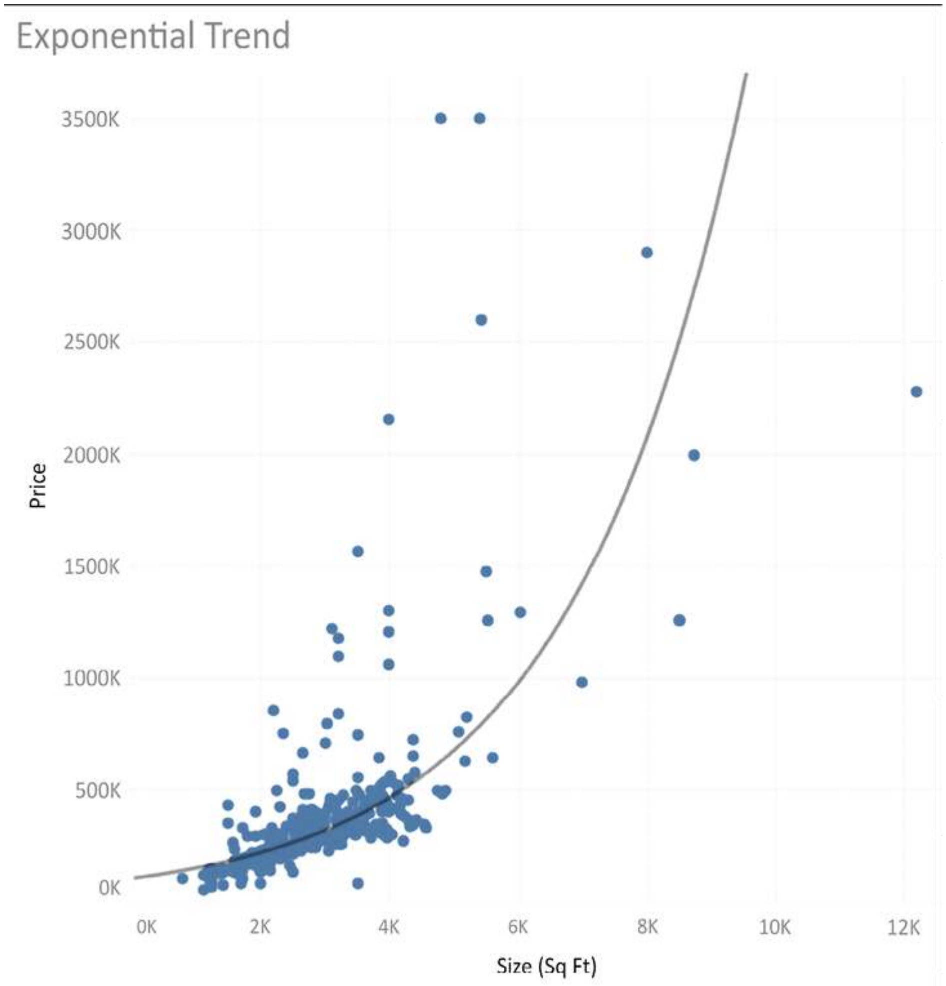


Figure 9.11: Exponential trend

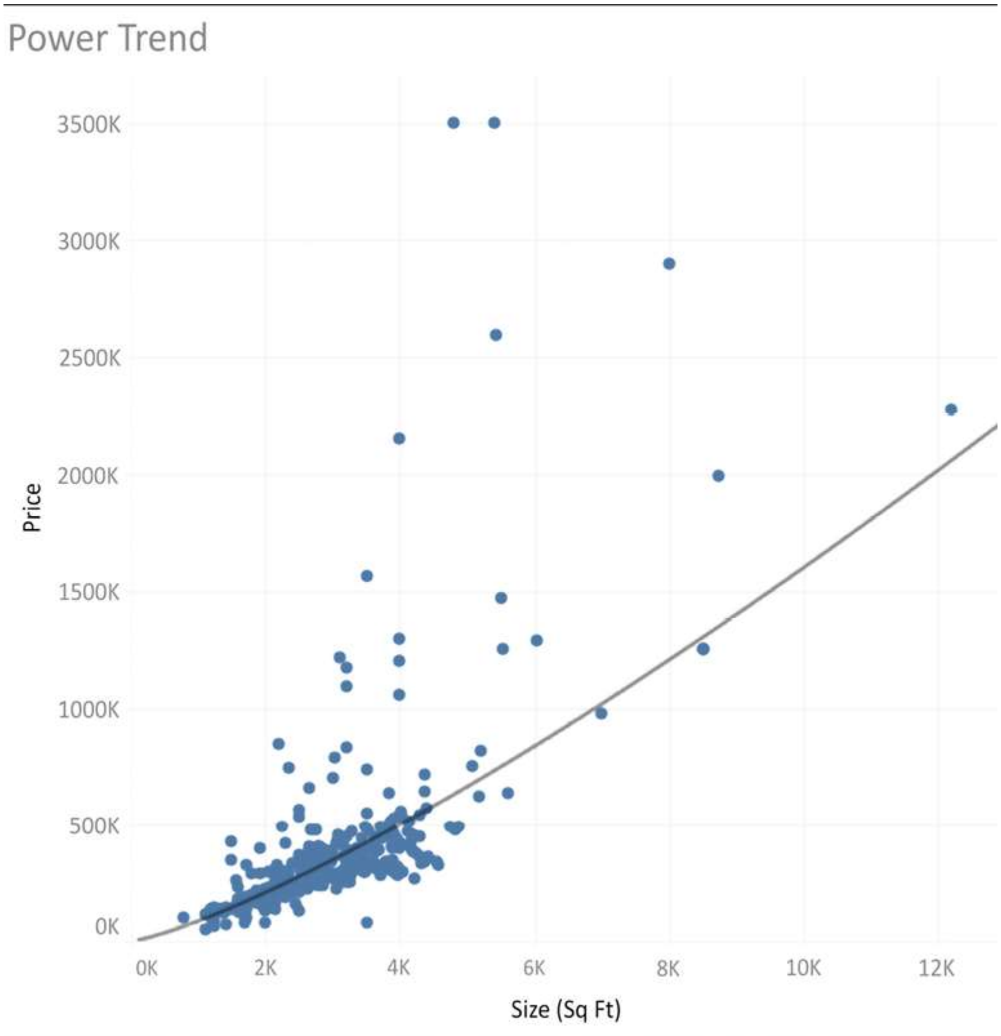


Figure 9.12: Power trend

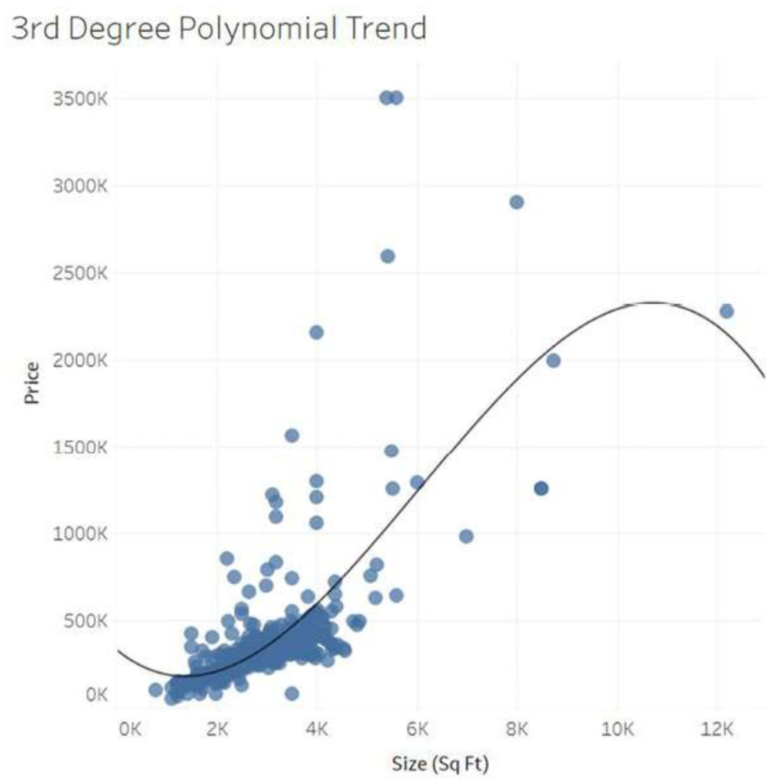


Figure 9.13: 3rd-degree polynomial trend

Figure 9.14: Tooltip displayed by hovering over the trend line

Describe Trend Line

P-value: < 0.0001

Equation: Price = -5.25244e-06*Size (Sq Ft)^3 + 0.0953132*Size (Sq Ft)^2 + -232.162*Size (Sq Ft) + 329855

Coefficients

Term	Value	StdErr	t-value	p-value
Size (Sq Ft)^3	-5.252e-06	1.226e-06	-4.2833	< 0.0001
Size (Sq Ft)^2	0.0953132	0.0208252	4.57683	< 0.0001
Size (Sq Ft)	-232.162	96.273	-2.4115	0.0164296
intercept	329855	129714	2.54295	0.0114456

Copy

Figure 9.15: The Describe Trend Line window

Summary

Count:	336
SUM(Price)	
Sum:	134,239,384
Average:	399,521.98
Minimum:	50,000
Maximum:	3,500,000
Median:	300,000.00
Standard deviation:	414,528
First quartile:	211,500.00
Third quartile:	410,000.00
Skewness:	4.59
Excess Kurtosis:	25.74
SUM(Size (Sq Ft))	
Sum:	1,001,318
Average:	2,980.11
Minimum:	784
Maximum:	12,200
Median:	2,800.50
Standard deviation:	1,264
First quartile:	2,108.50
Third quartile:	3,627.00
Skewness:	2.27
Excess Kurtosis:	10.95

Figure 9.16: Summary information

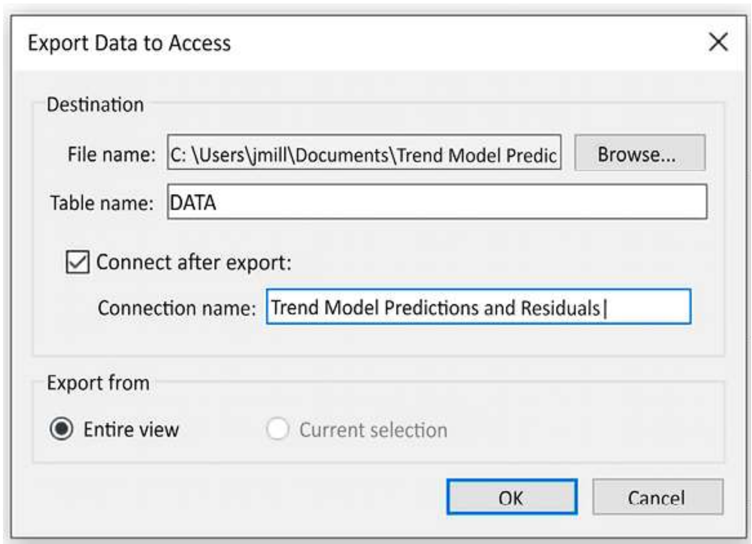


Figure 9.17: The Export Data to Access dialog box

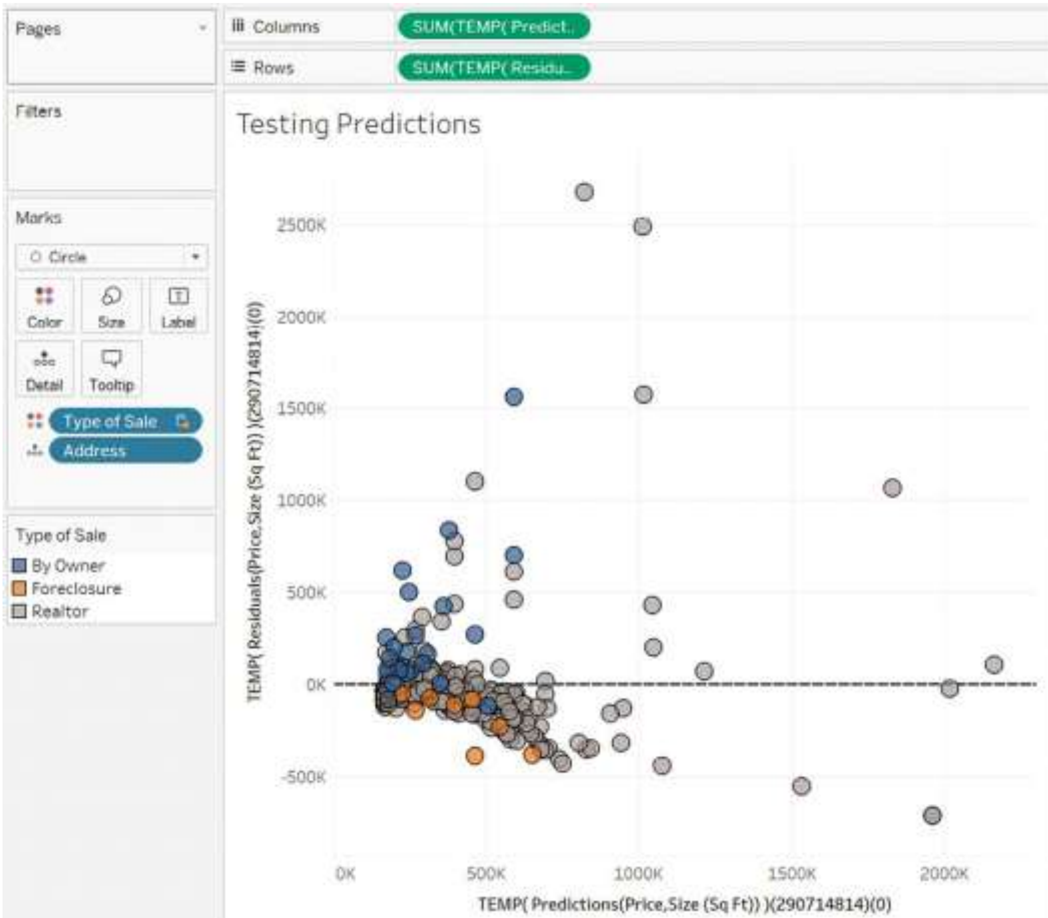


Figure 9.18: A view using residuals and predictions to test the model

Figure 9.19: The highlight button

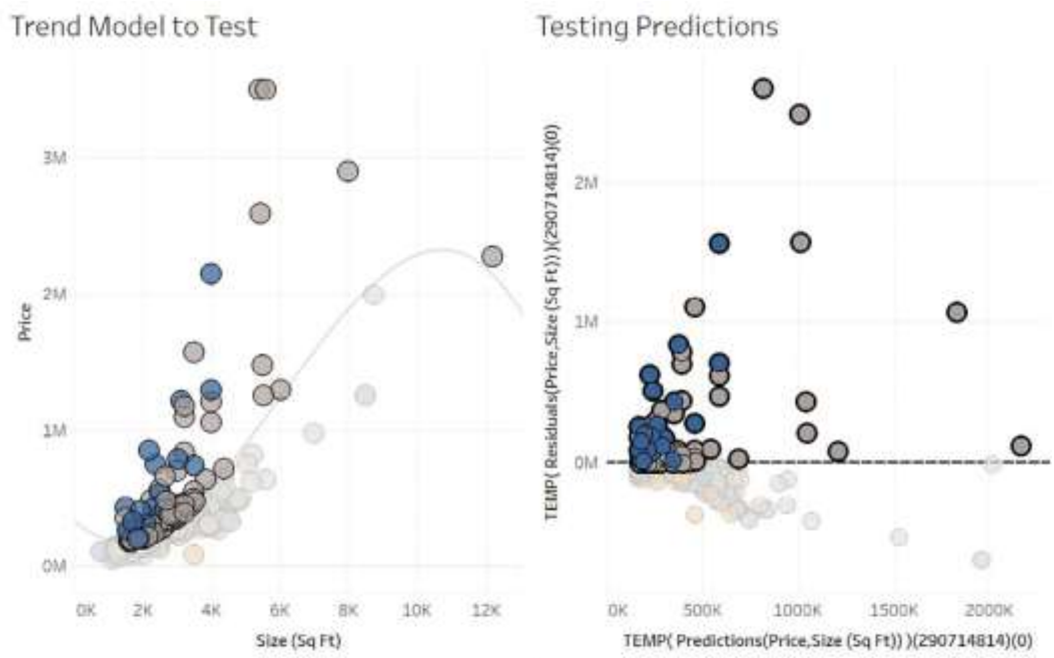


Figure 9.20: Placing the original view alongside the testing view allows you to see the relationship

Figure 9.21: Selecting a mark for Explain Data

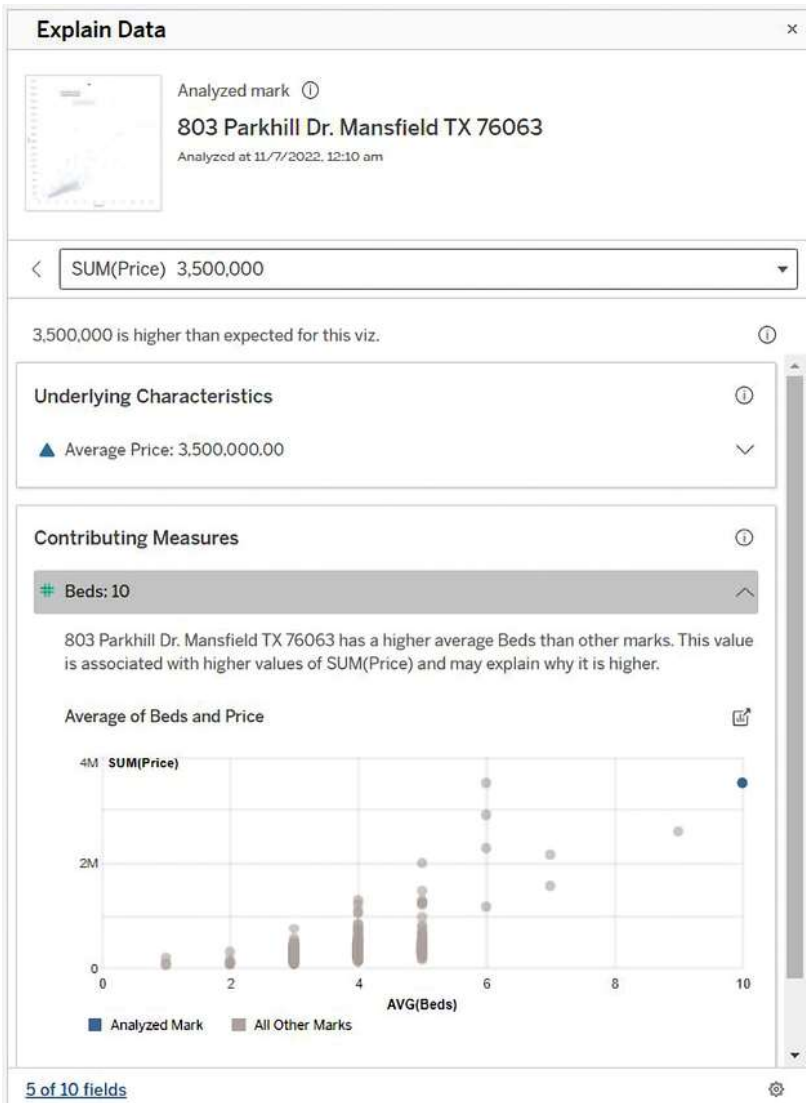


Figure 9.22: The Explain Data pane

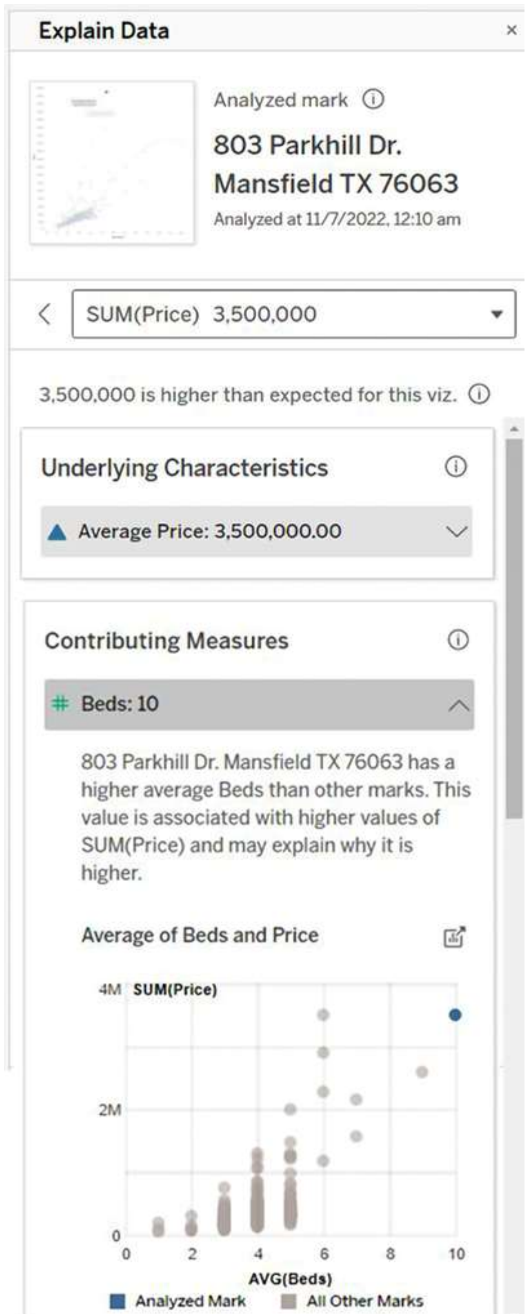


Figure 9.23: Exploring the contributing measures for values

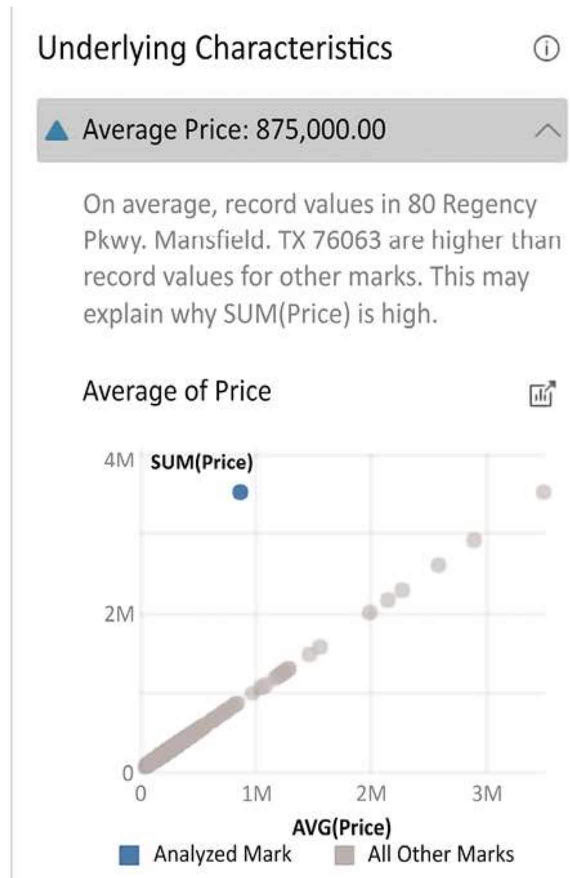


Figure 9.24: Exploring the underlying characteristics of a mark

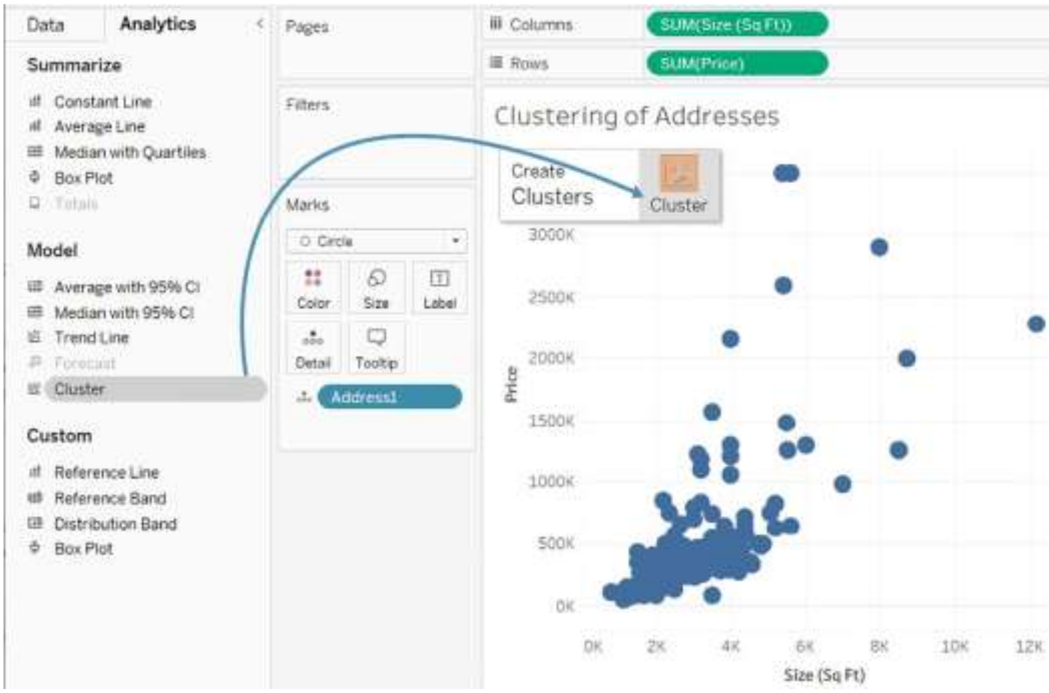


Figure 9.25: Adding clusters by dragging and dropping from the Analytics pane

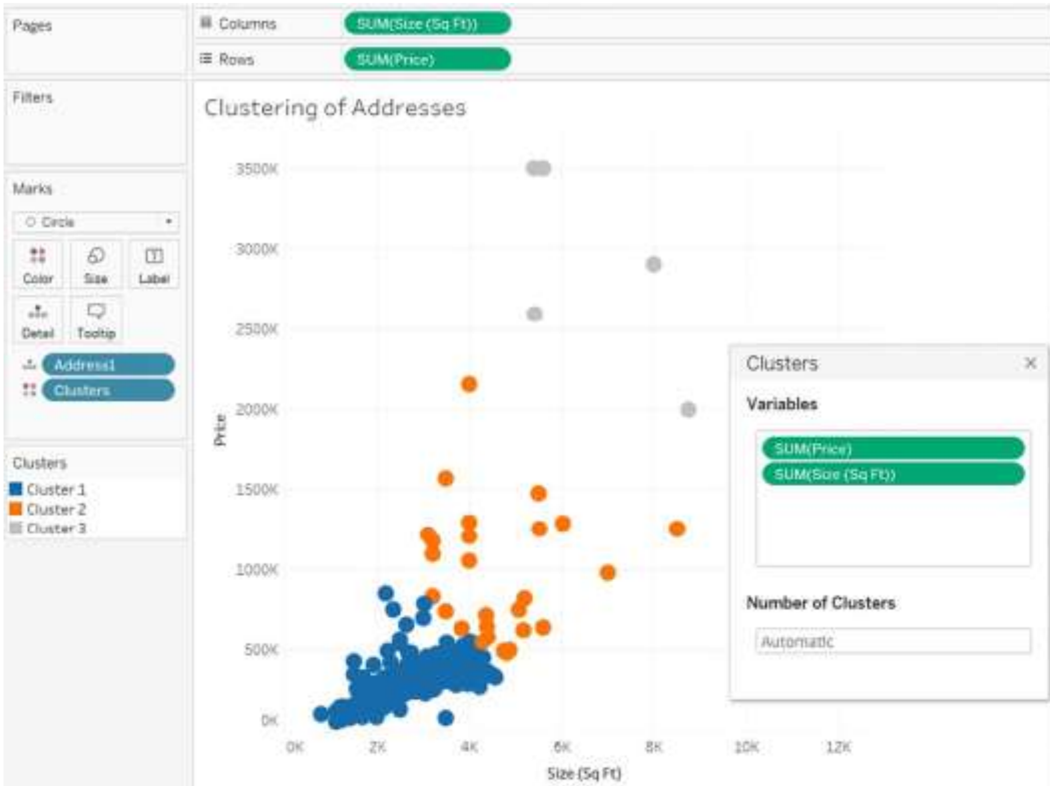


Figure 9.26: Clusters of individual addresses based on Price and Size

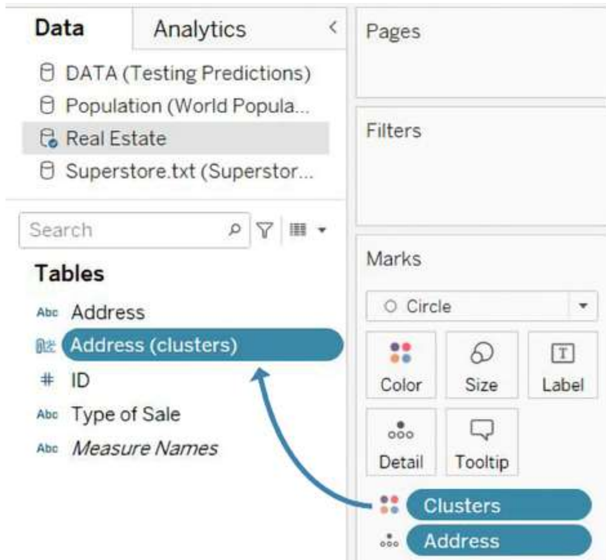


Figure 9.27: Materializing a cluster by dragging the Clusters field from the view to the Data pane

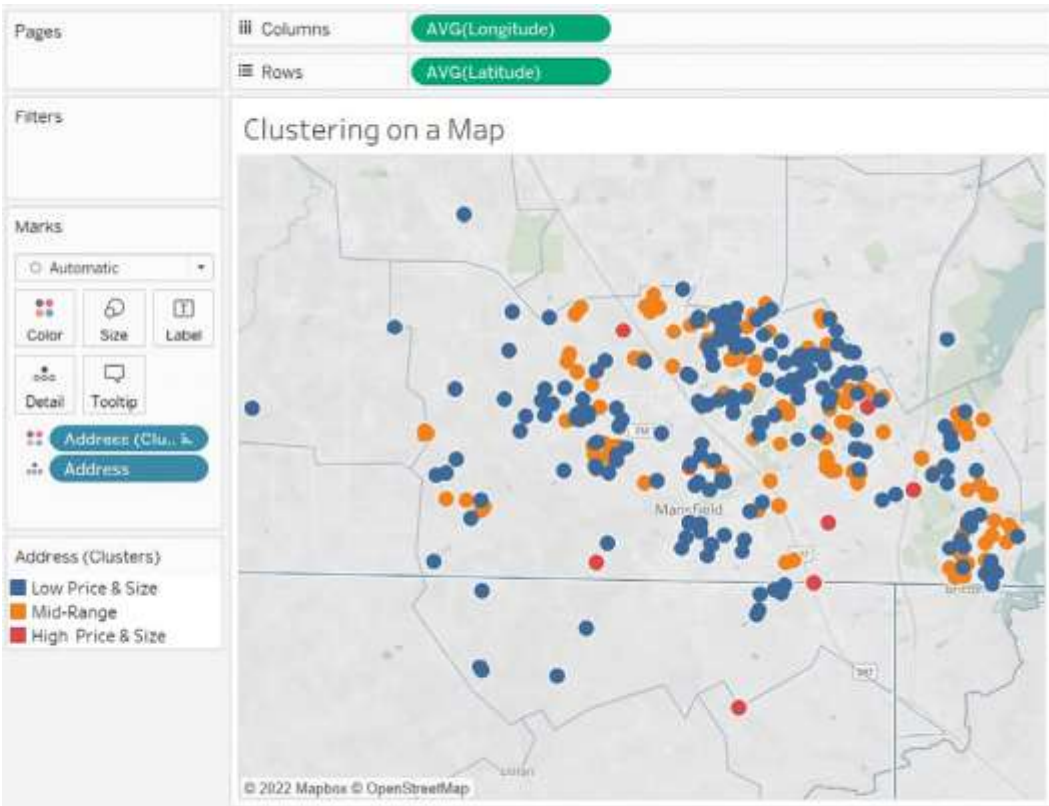


Figure 9.28: This view uses the clusters we identified to additionally understand any geospatial relationships

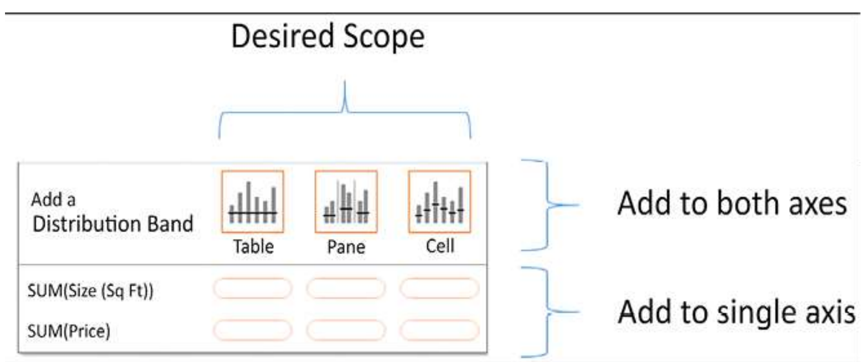


Figure 9.29: Defining the scope and axis as you add reference lines and distributions from the Analytics pane

Figure 9.30: A scatterplot divided into three columns

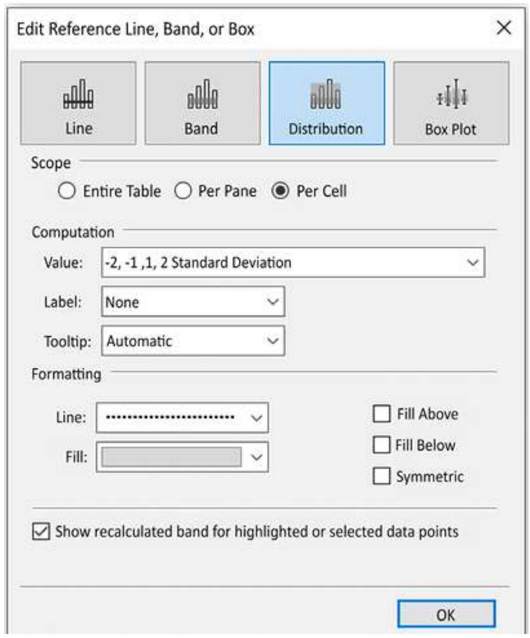


Figure 9.31: The dialog box for adding or editing lines, bands, distributions, or box plots

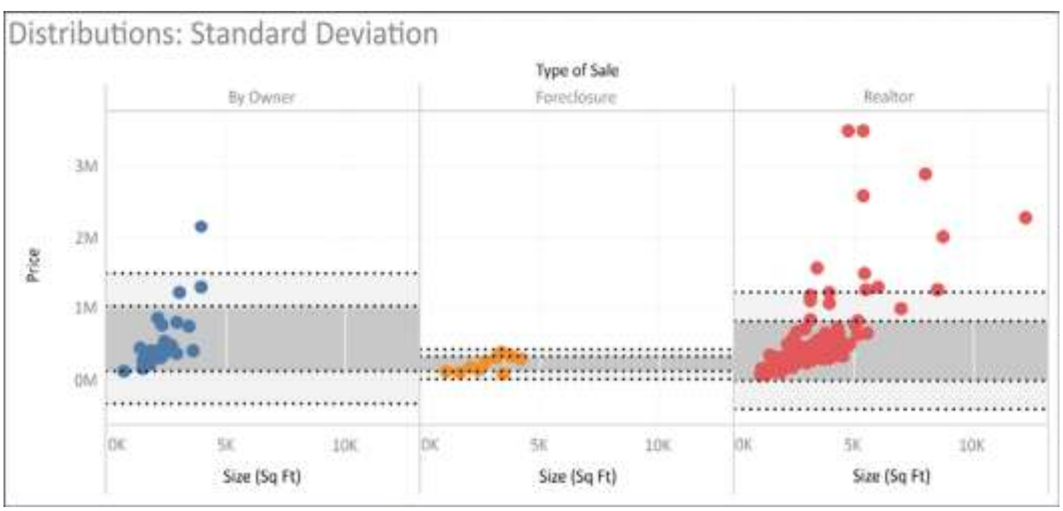


Figure 9.32: Two standard deviations of Price for each Type of Sale

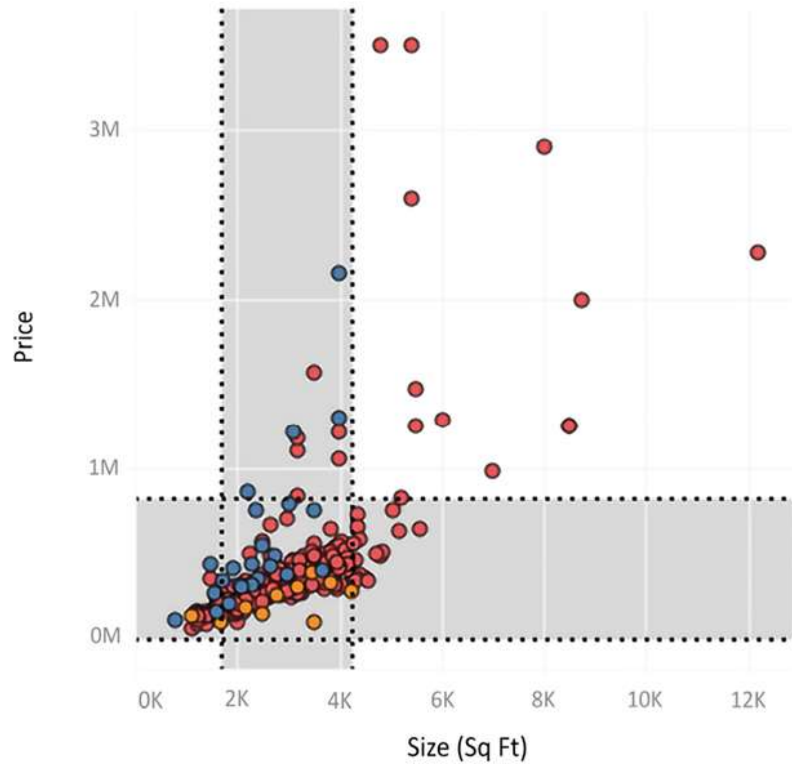


Figure 9.33: One standard deviation for both Price and Size of all houses

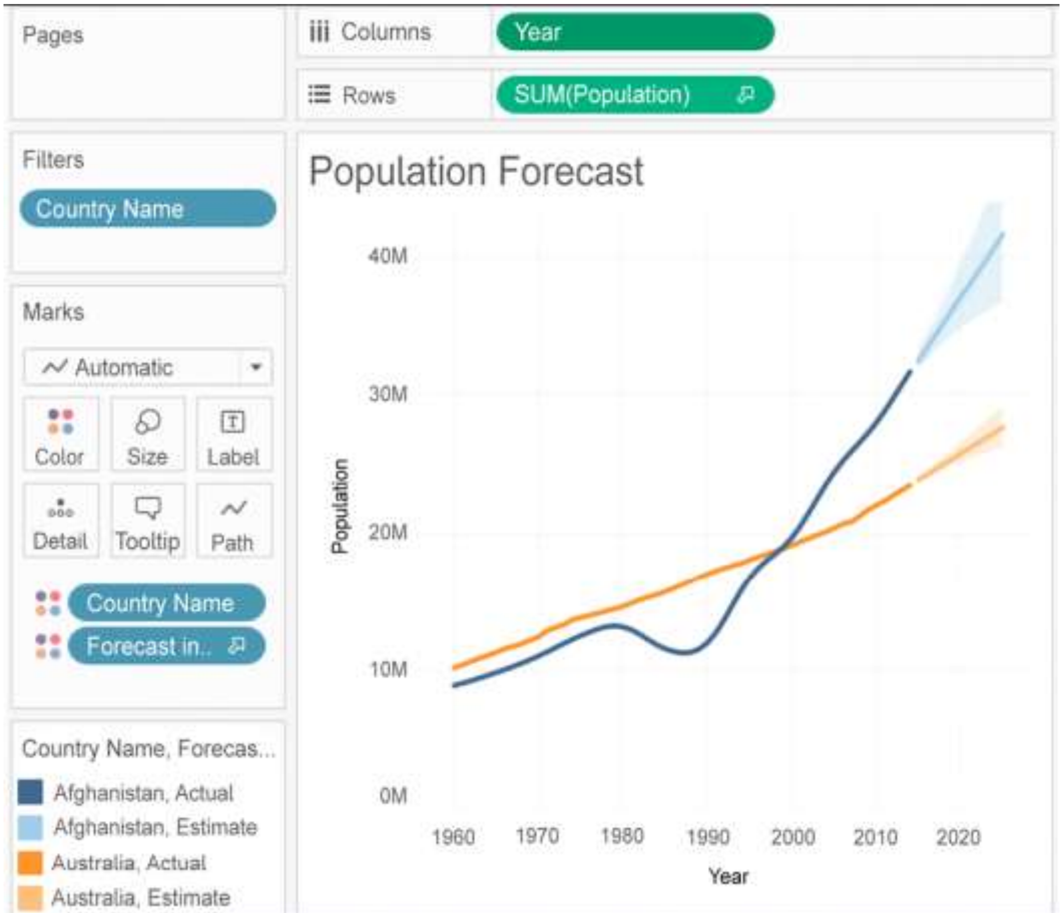


Figure 9.34: A forecast of the population for both Afghanistan and Australia

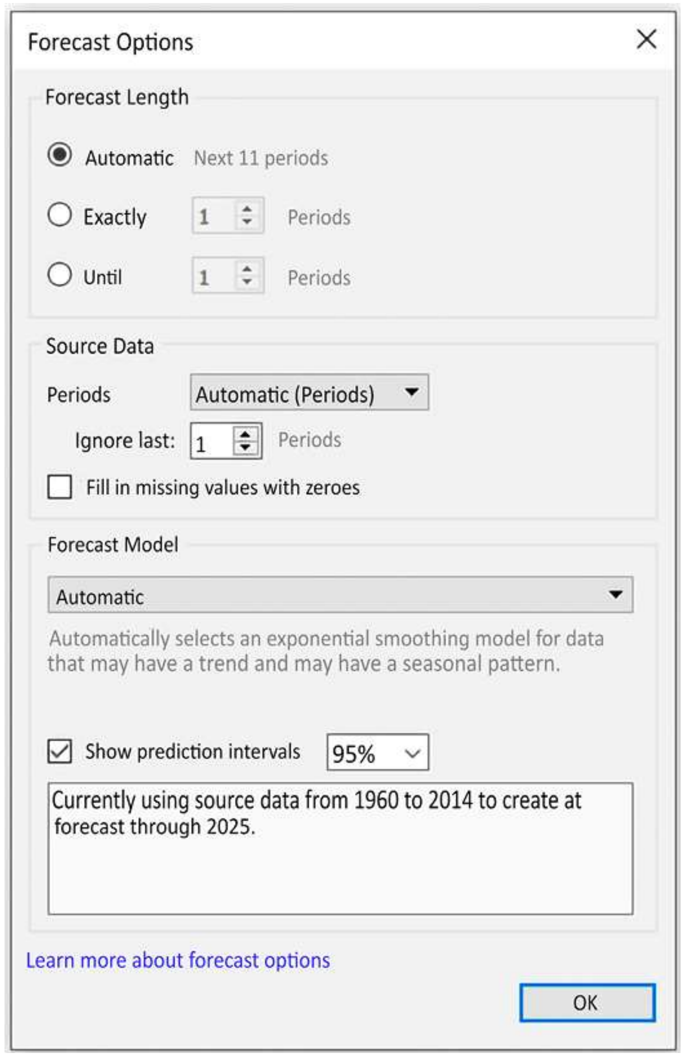


Figure 9.35: The Forecast Options dialog box

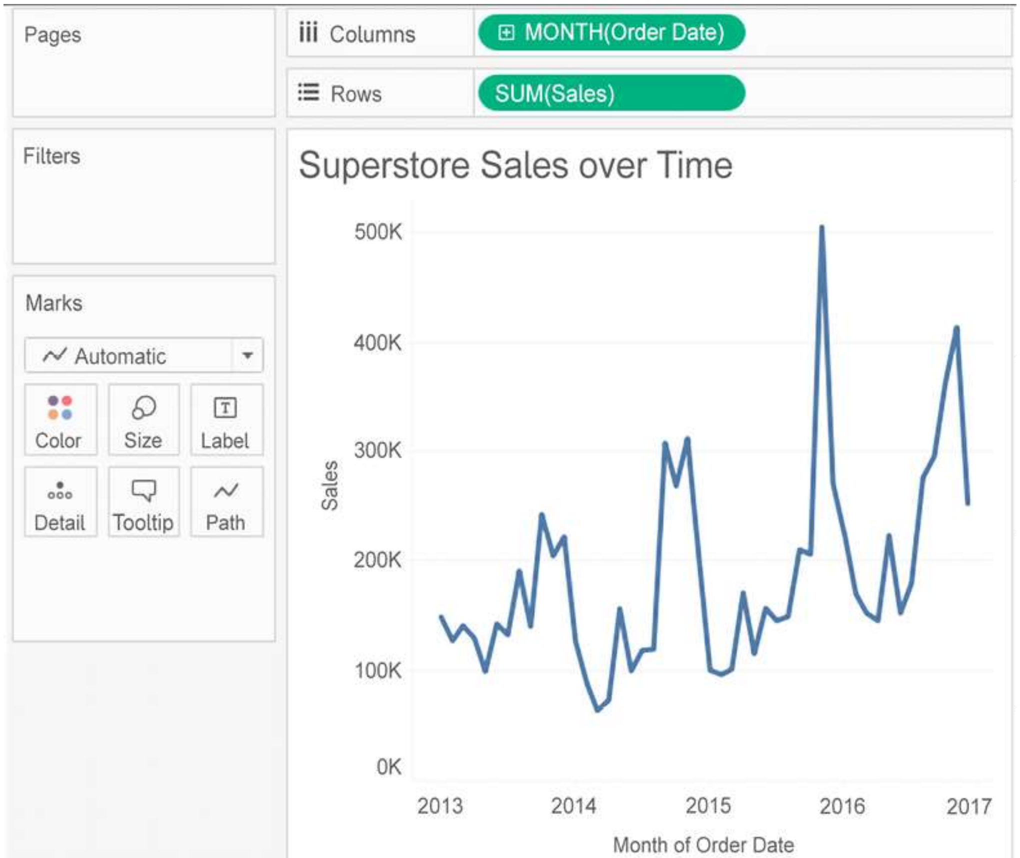


Figure 9.36: This time series shows a cyclical or seasonal pattern

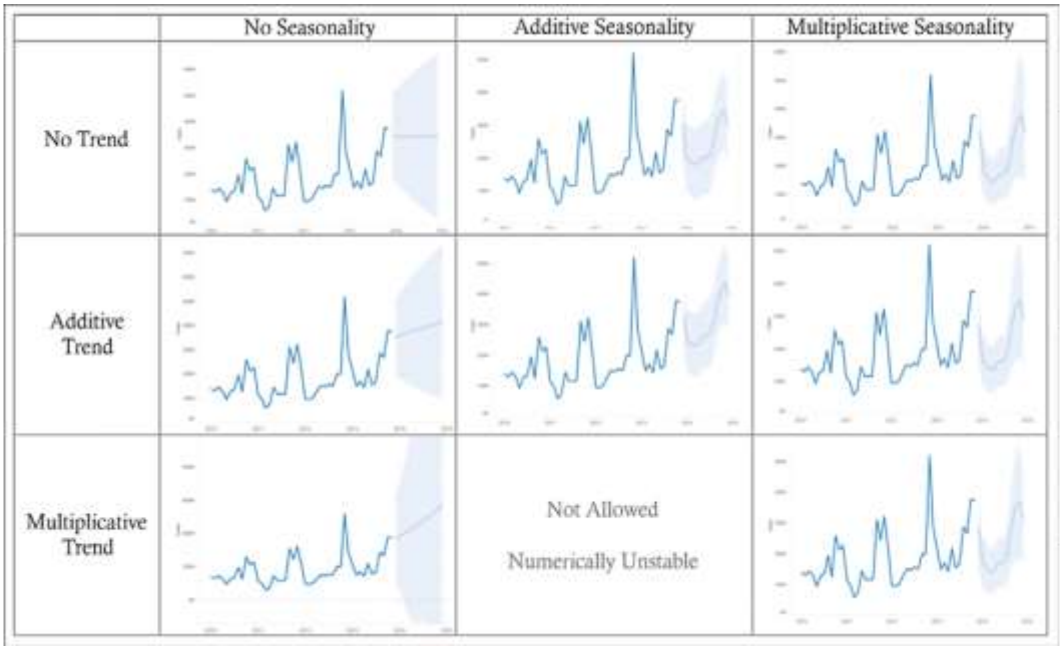


Figure 9.37: Selecting various forecast models will yield different results

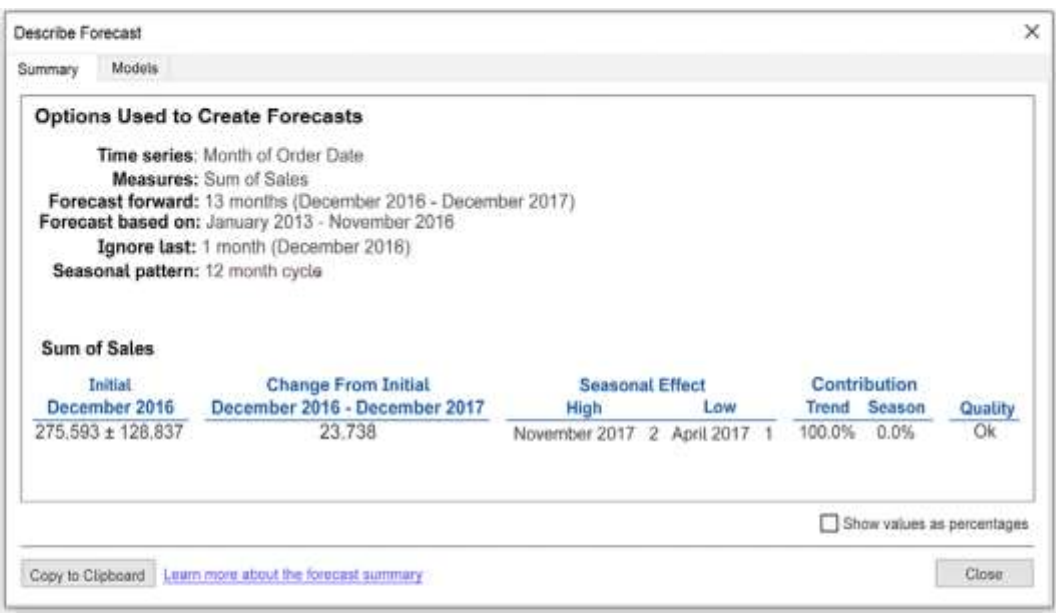


Figure 9.38: Tableau describes the forecast model

Code

Code 9.1

```
IF [Year] <= 1979
  THEN "1960 to 1979"
ELSEIF [Year] <= 1988
  THEN "1980 to 1988"
ELSE "1988 to 2015"
END
```

Chapter 10

Figures

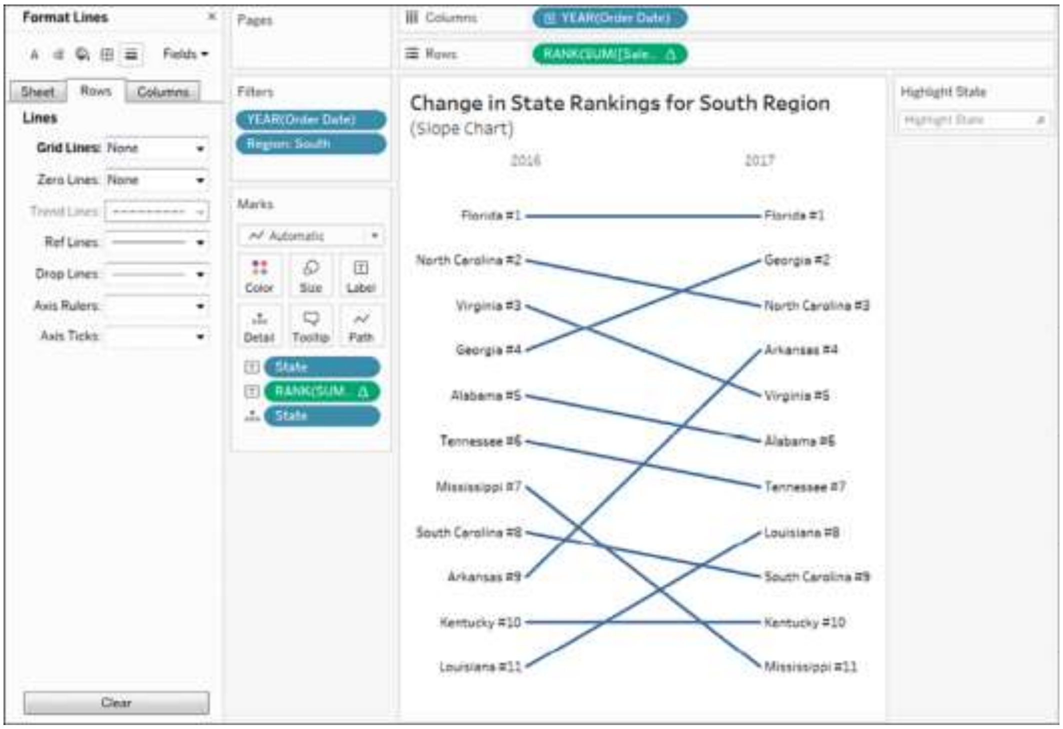


Figure 10.1: A slope chart is useful to compare the change of rank or absolute values from one period or status to another

Figure 10.2: This bump chart shows the change in rank for each state over time and makes use of a highlighter

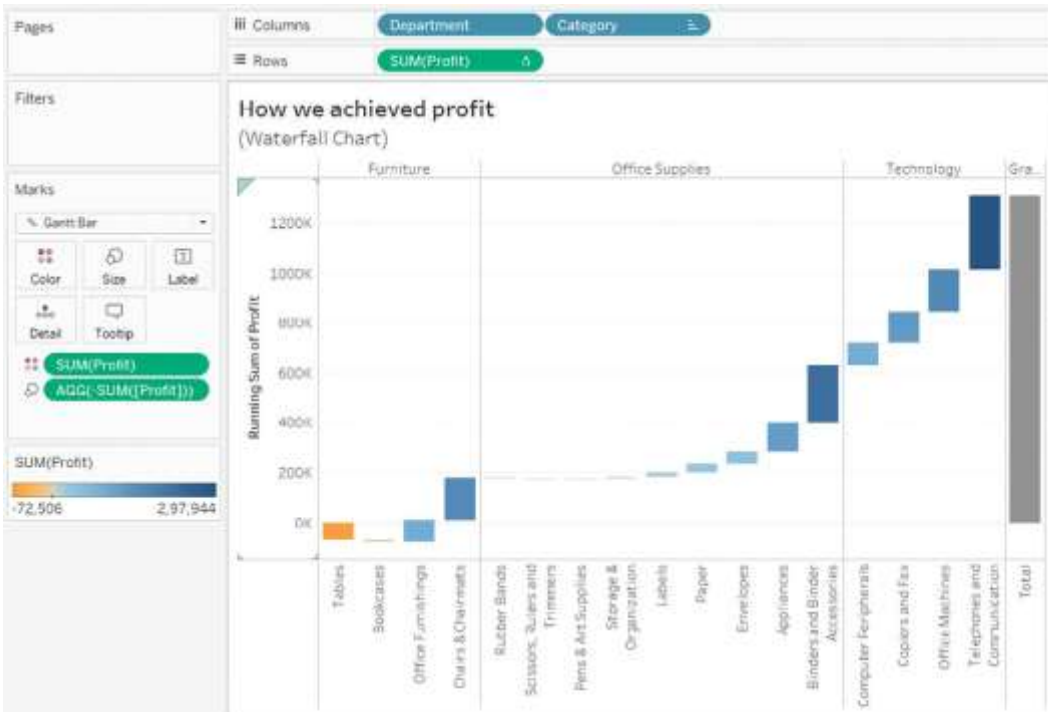


Figure 10.3: This waterfall chart shows how each Category adds (or subtracts) profit to build toward the total

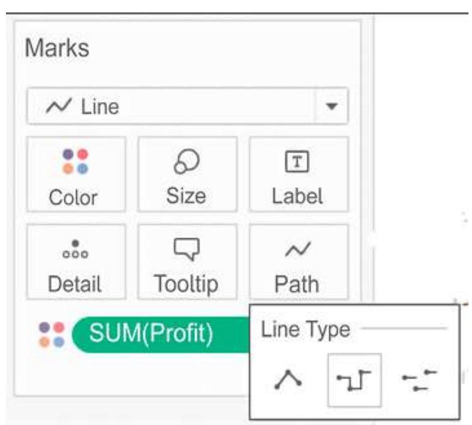


Figure 10.4: Change the type of Line by clicking Path on the Marks card

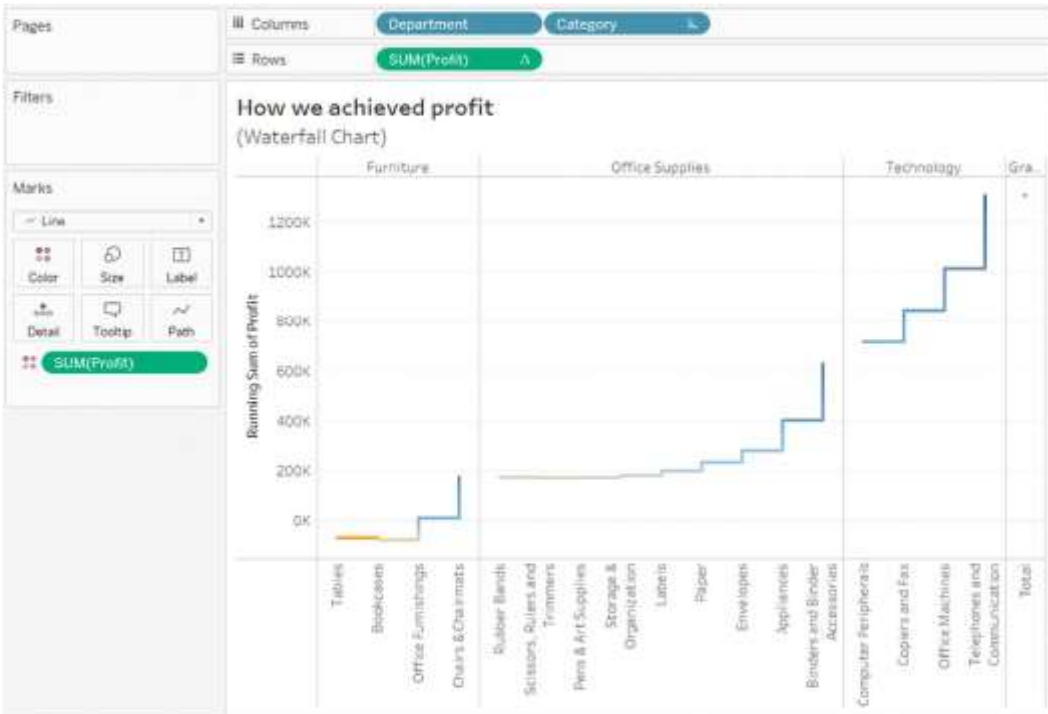


Figure 10.5: A step line chart emphasizes an abrupt change or discrete difference

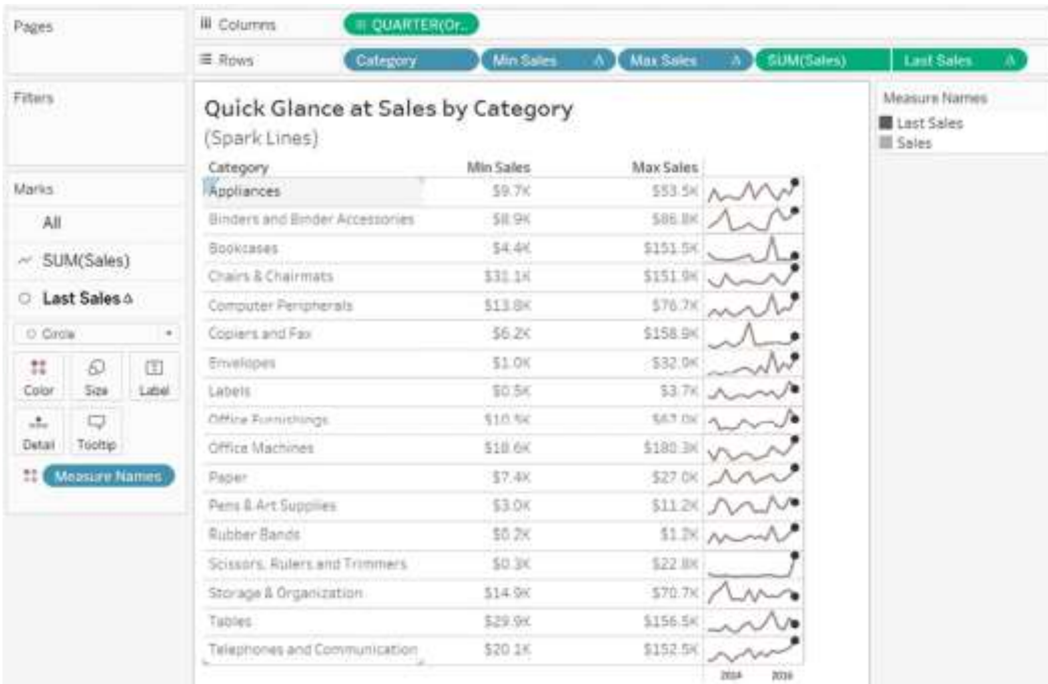


Figure 10.6: Spark Lines give you a quick glance at the "shape" of change over time across multiple categories

Figure 10.7: A dumbbell chart emphasizes the distance/difference between two values

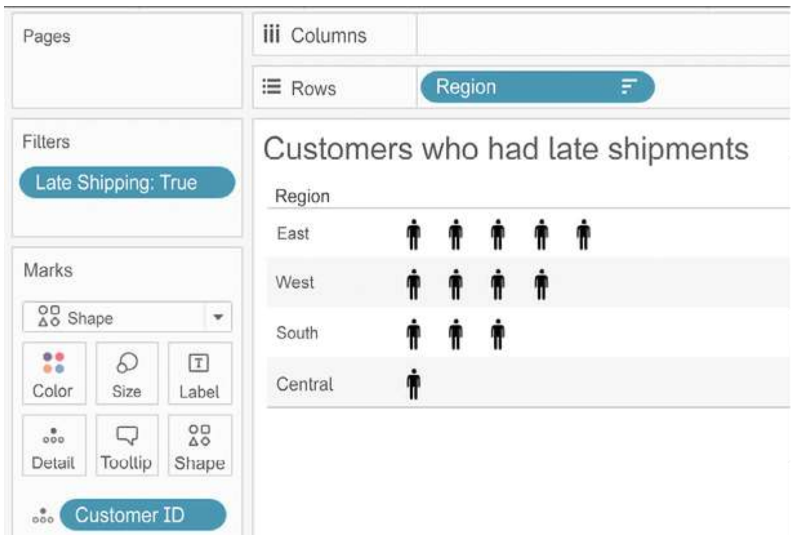


Figure 10.8: Each image represents a real person and is less abstract than circles or squares

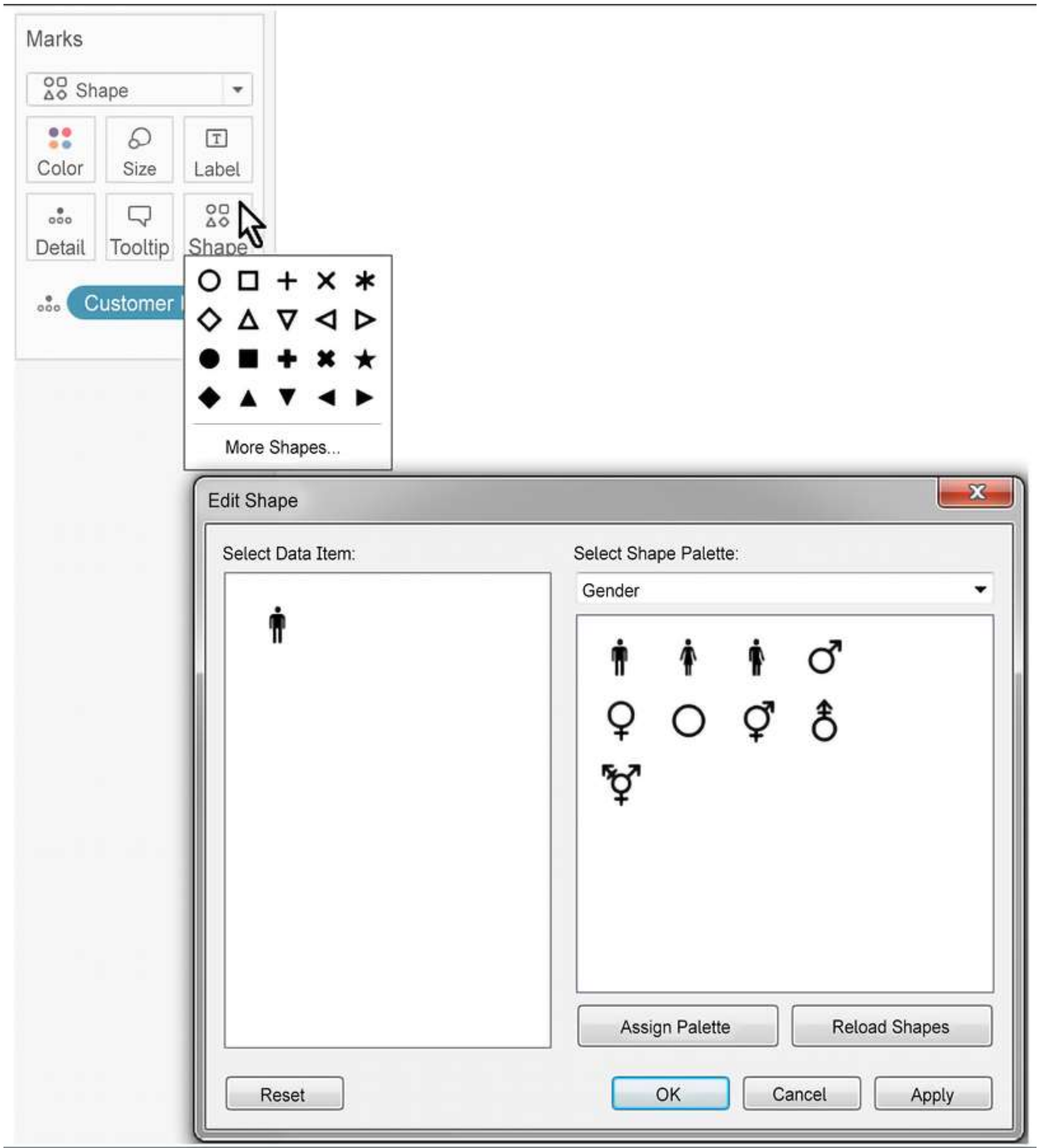


Figure 10.9: You can assign shapes to dimensional values using the Shape shelf

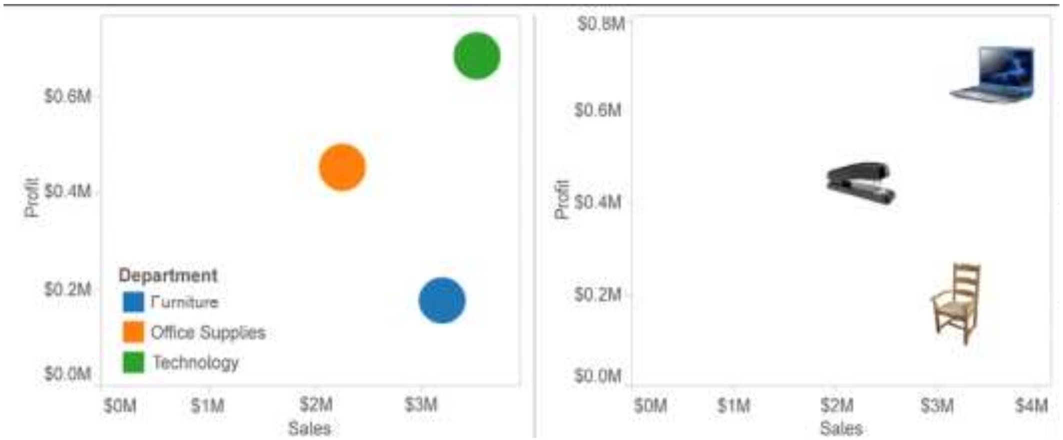


Figure 10.10: Notice the difference in "cognitive load" between the left chart and the right

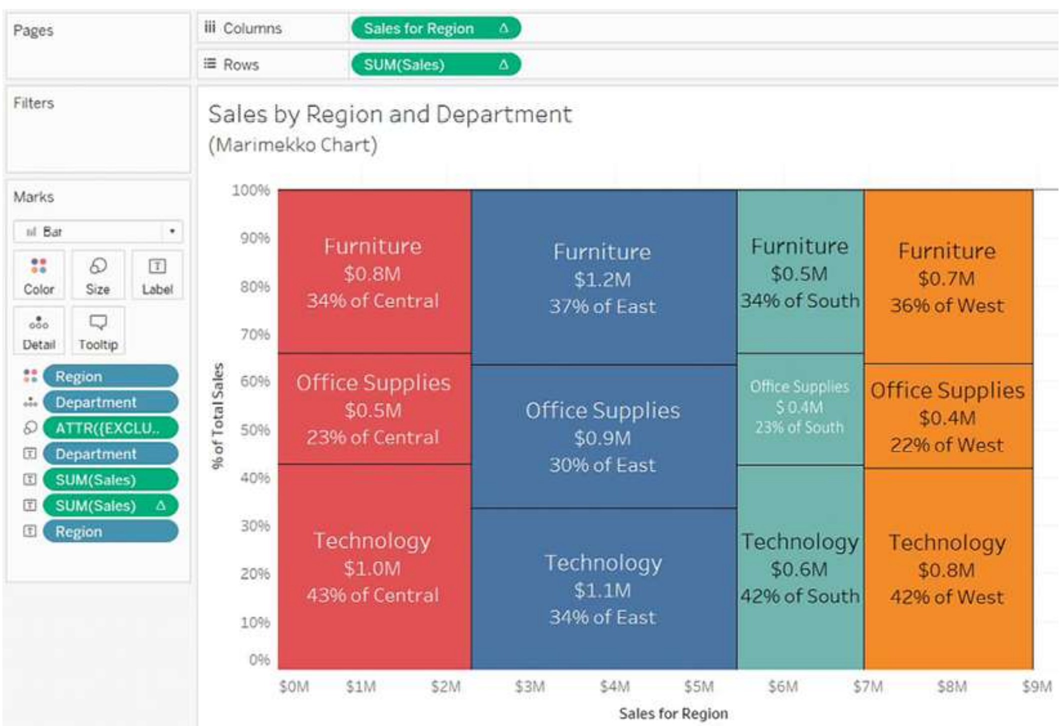


Figure 10.11: The amount of sales per Department is indicated by the height of each segment, while the width of each bar indicates the overall sales per Region

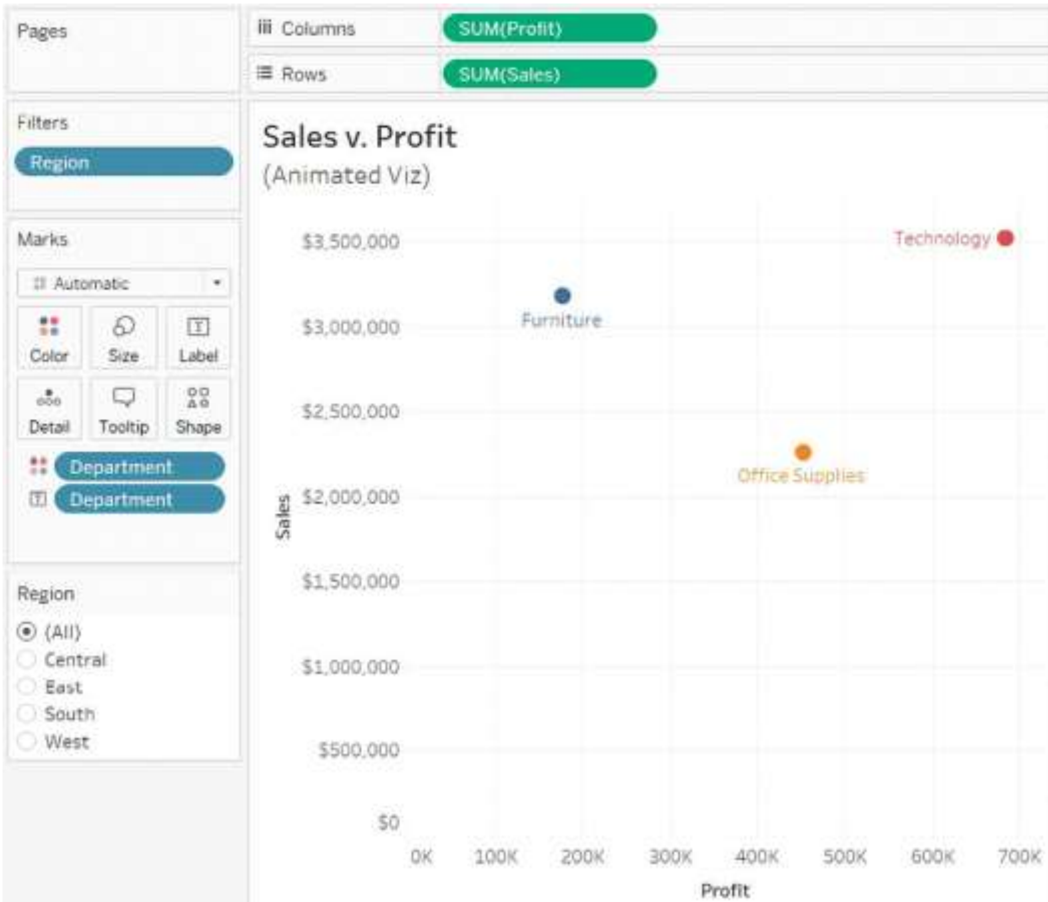


Figure 10.12: Sales and profit per Department

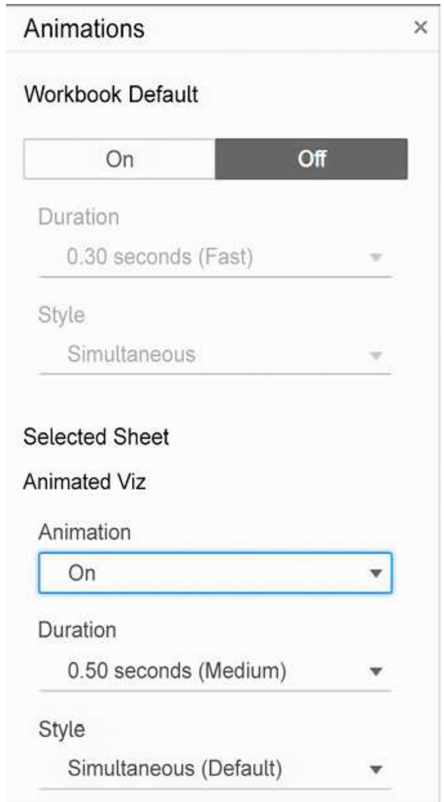


Figure 10.13: The Animations format pane gives various options for workbook and individual sheet animation settings

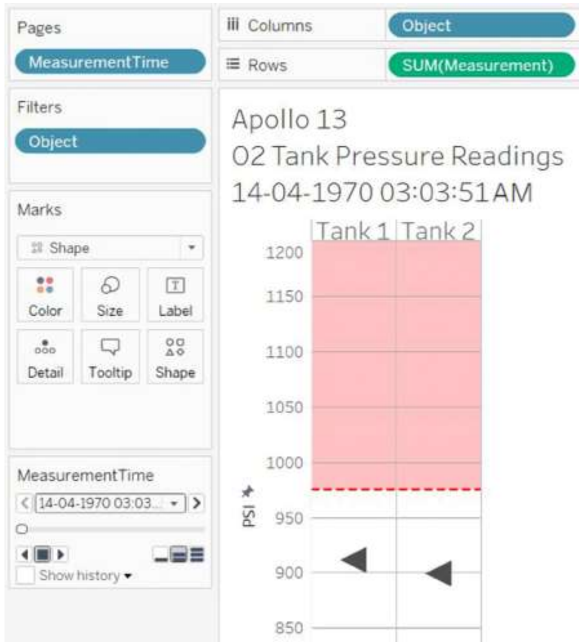


Figure 10.14: O2 Tank 1 and 2 pressure readings over time during the Apollo 13 mission

Links

For a more comprehensive discussion of Marimekko charts, along with approaches that work with sparse data, see Jonathan Drummey's blog post at https://www.tableau.com/about/blog/2016/8/how-build-marimekko-chart-tableau-58153.

Code

Code 10.1 – Building Sparklines

Creating two calculated fields to show the minimum and maximum quarterly sales values for each category.

Min Sales:

```
WINDOW_MIN(SUM(Sales))
```

Max Sales:

```
WINDOW_MAX(SUM(Sales)). Add both to Rows as discrete (blue)
fields.
```

Place the Last Sales calculation with the following code:

```
IF LAST() == 0 THEN SUM([Sales]) END
```

Code 10.2 – Creating symbol charts

The view is filtered where **Late Shipping** is **True**. **Late Shipping** is a calculated field that determines if it took more than 14 days to ship an order. The code is as follows:

```
DATEDIFF('day', [Order Date], [Ship Date]) > 14
```

Code 10.3 – Marimekko Charts

The calculated field **Sales for Region** calculates the *x* axis location for the right-side position of each bar. See code 10.3.

```
IF FIRST() = 0
    THEN MIN({EXCLUDE [Department] : SUM(Sales)})
ELSEIF LOOKUP(MIN([Region]), -1) <> MIN([Region])
    THEN PREVIOUS_VALUE(0) + MIN({EXCLUDE [Department] :
SUM(Sales)})
ELSE
    PREVIOUS_VALUE(0)
END
```

Finally, a few additional adjustments were made to the view: the field on **Size** is an ad hoc level of detail calculation with the code:

```
{EXCLUDE [Department] : SUM(Sales)}
```

Chapter 11

Figures

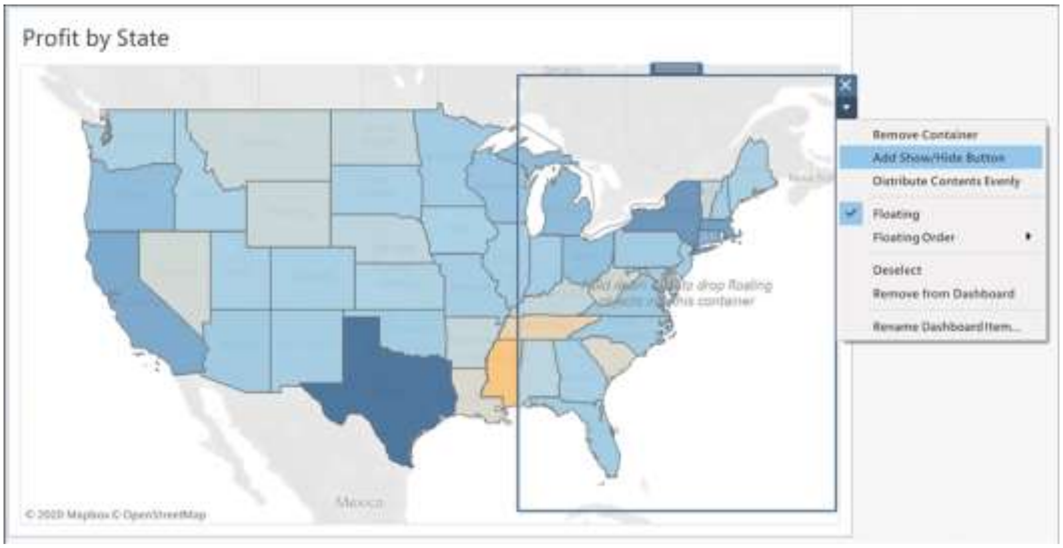


Figure 11.1: A dashboard with a single map view and a floating layout container

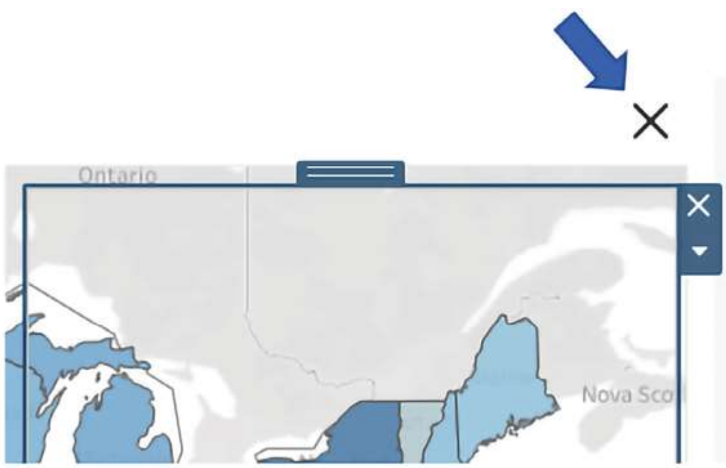


Figure 11.2: The Show-Hide button for the layout container

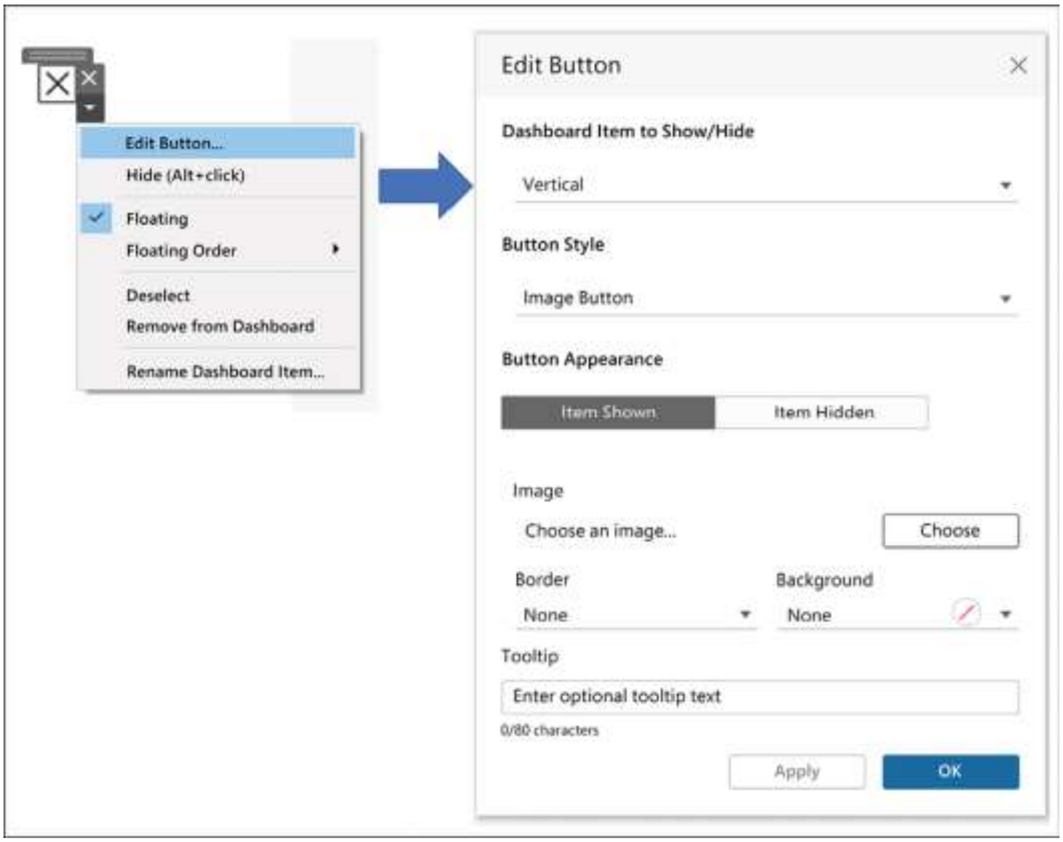


Figure 11.3: Selecting Edit Button… reveals many options for altering the button's behavior and appearance

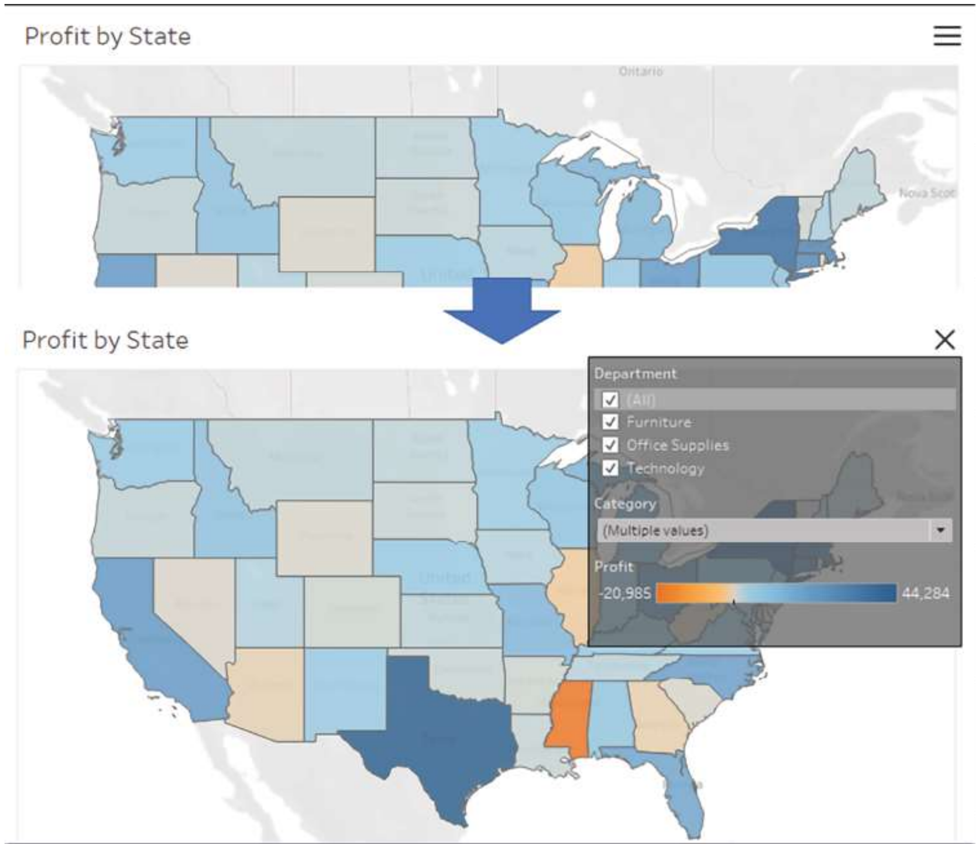


Figure 11.5: Here you can see the layout container's hidden and shown states along with the changing image of the button

Figure 11.6: Click the Presentation Mode button on the toolbar to enter presentation mode

Figure 11.7: Clicking the question mark button reveals instructions for how to use the dashboard

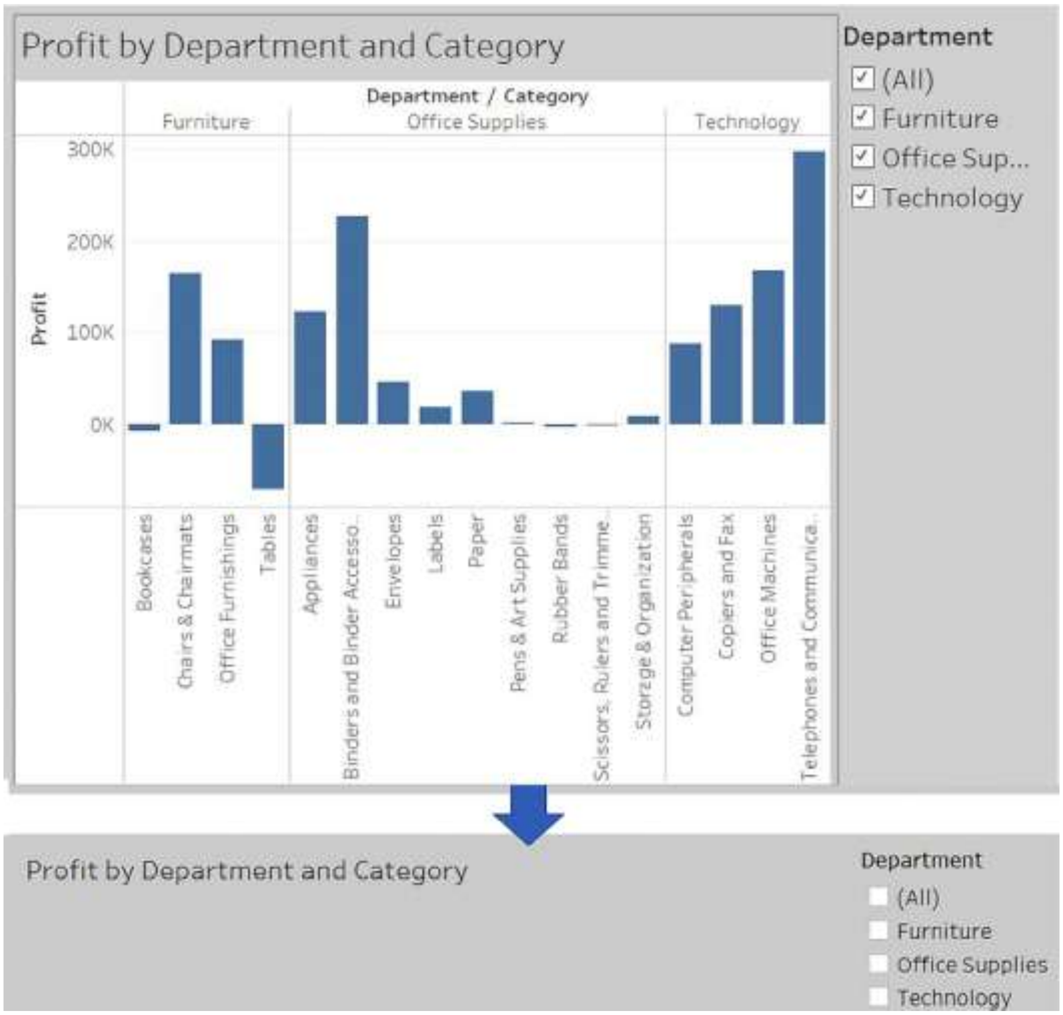


Figure 11.8: A demonstration of hiding a sheet in a dashboard

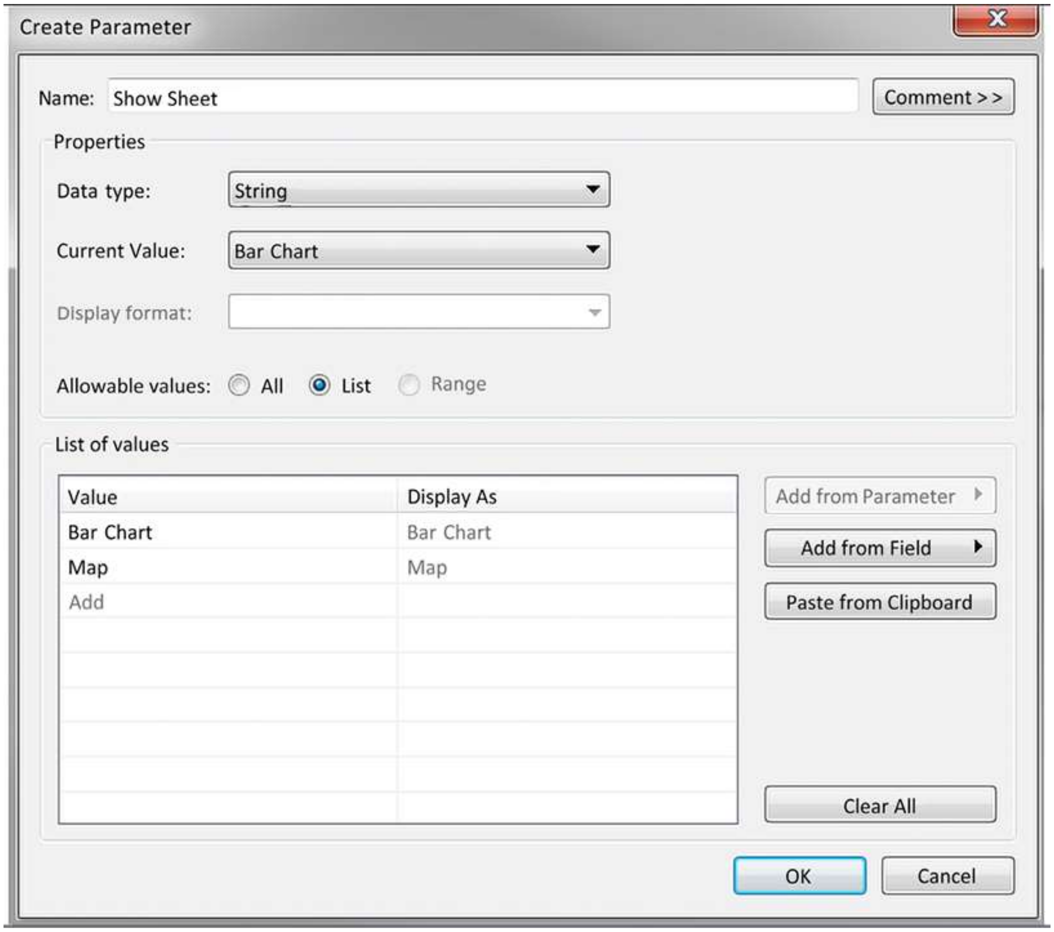


Figure 11.9: Creating a parameter to control which sheet is shown

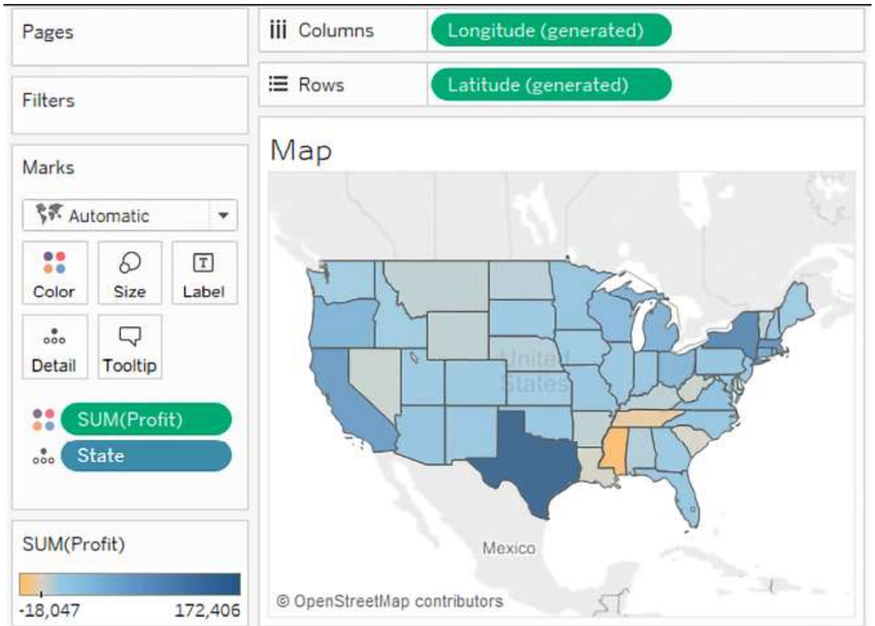


Figure 11.10: The Map view

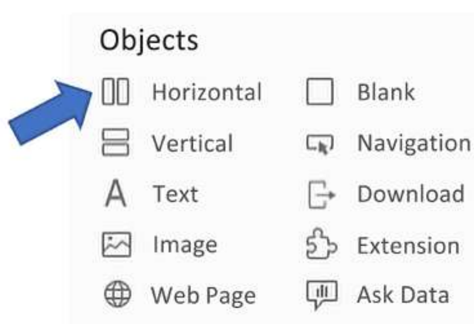


Figure 11.11: Insert a Horizontal layout container

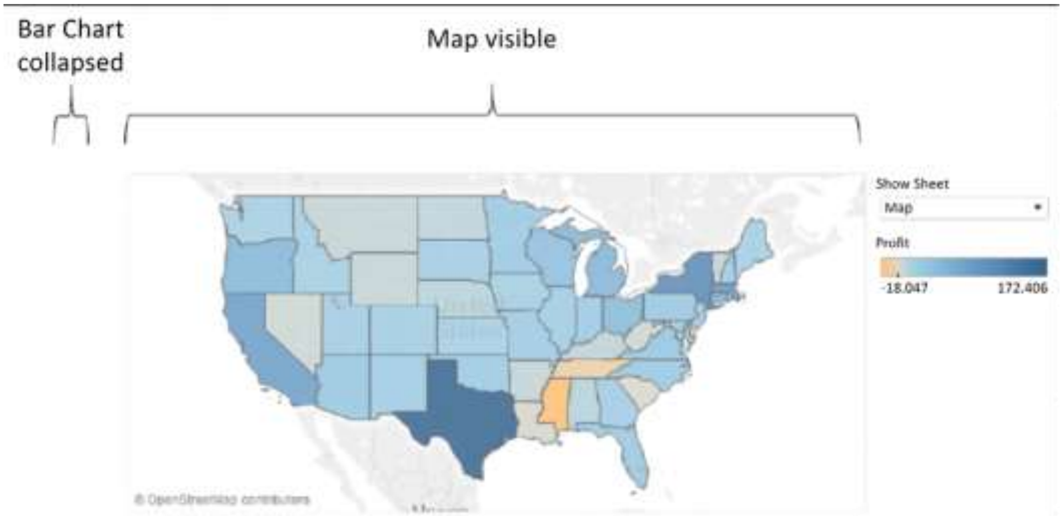


Figure 11.12: Map is visible while Bar Chart is collapsed

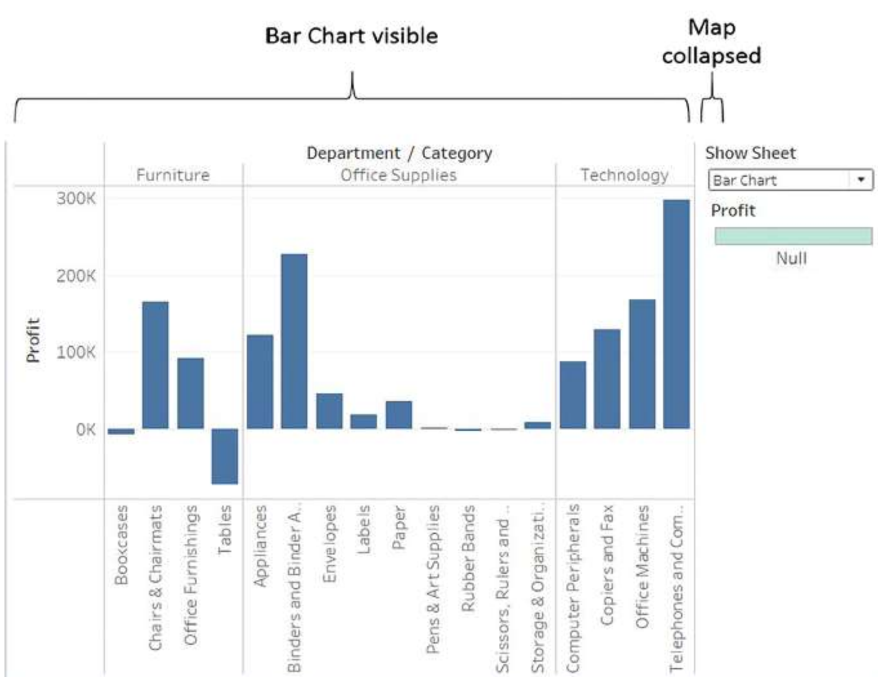


Figure 11.13: Bar Chart is visible while Map is collapsed

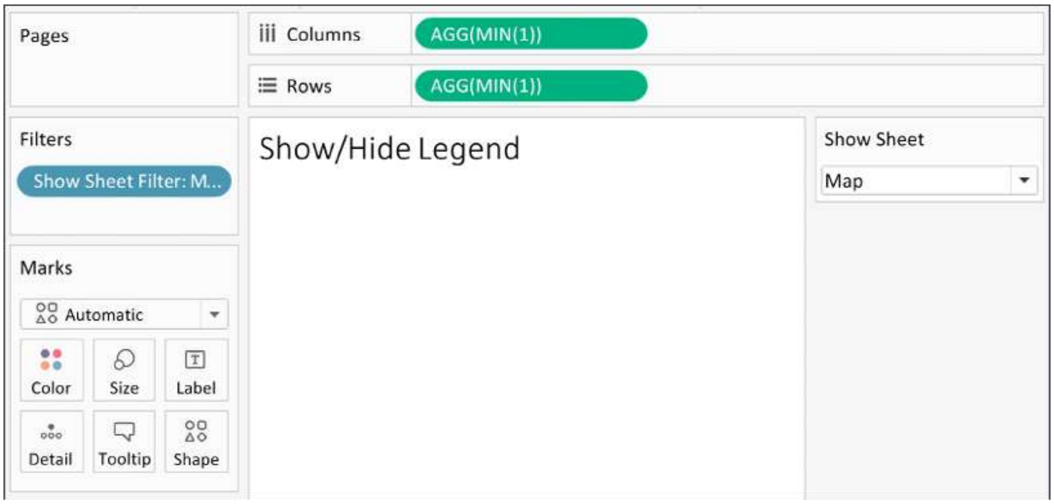


Figure 11.14: The Show-Hide Legend sheet with the Show Sheet Filter applied

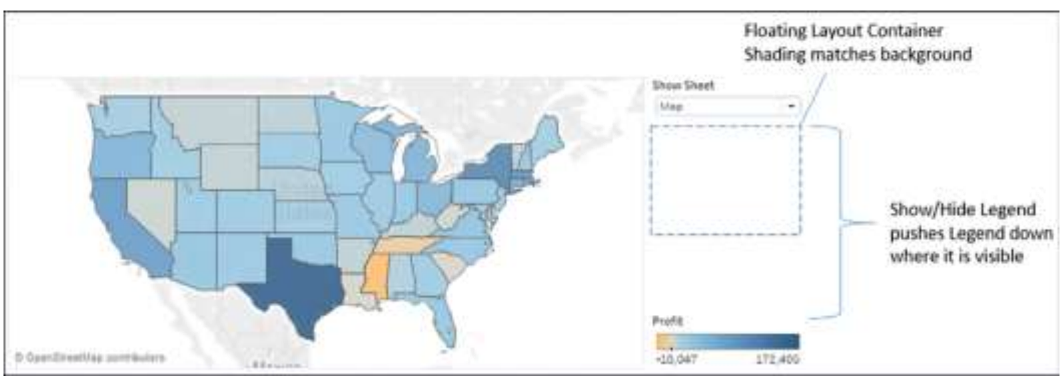


Figure 11.15: The Show-Hide Legend pushes the legend down past the floating object

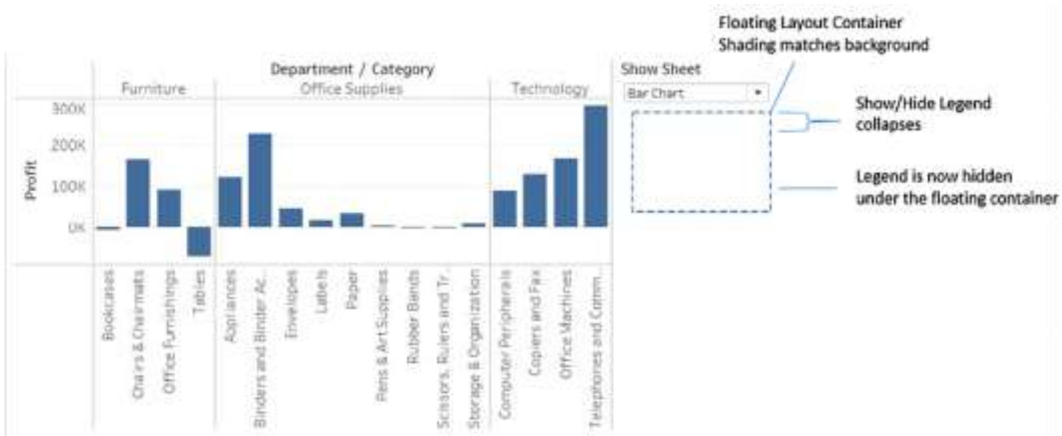


Figure 11.16: The Show-Hide Legend collapses, causing the legend to move under the floating object

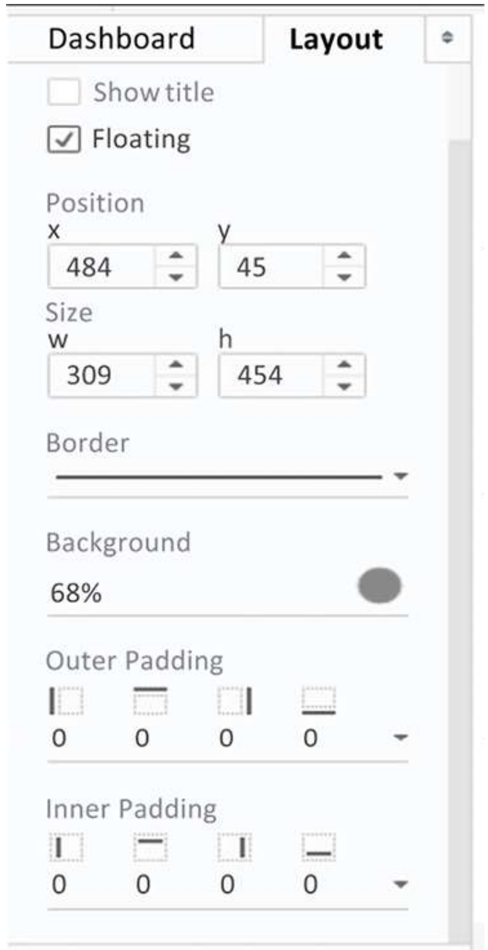


Figure 11.4: Use the Layout pane to adjust options for any selected dashboard object

Chapter 12

Figures

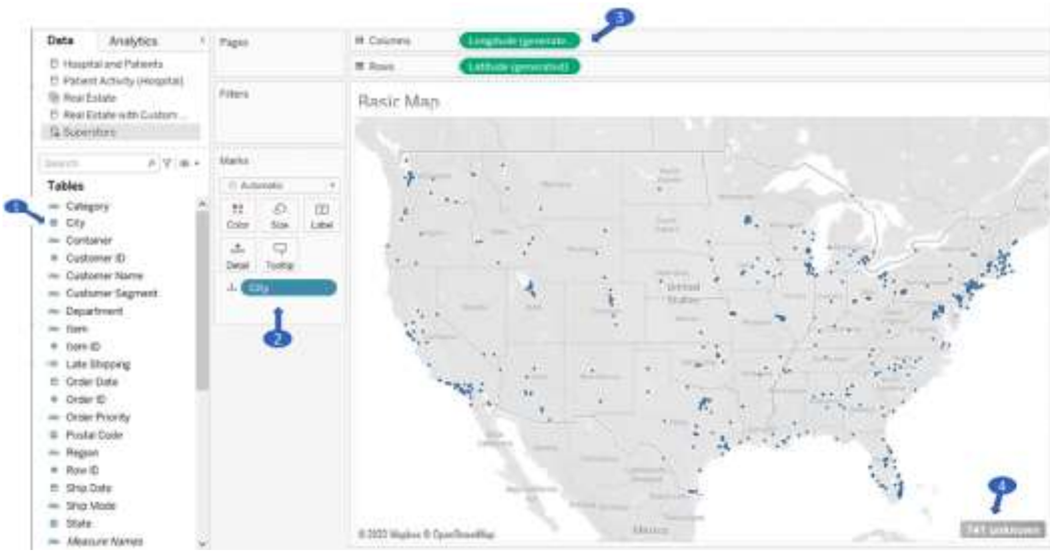


Figure 12.1: A basic geospatial rendering in Tableau

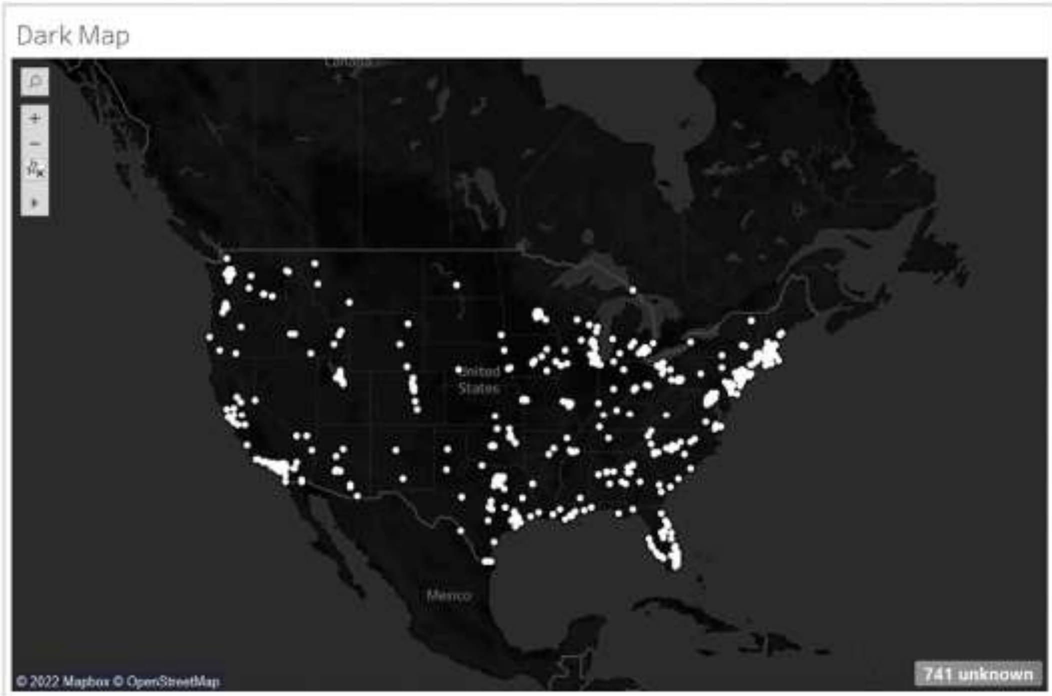


Figure 12.2: Dark Map is one of many options for map backgrounds

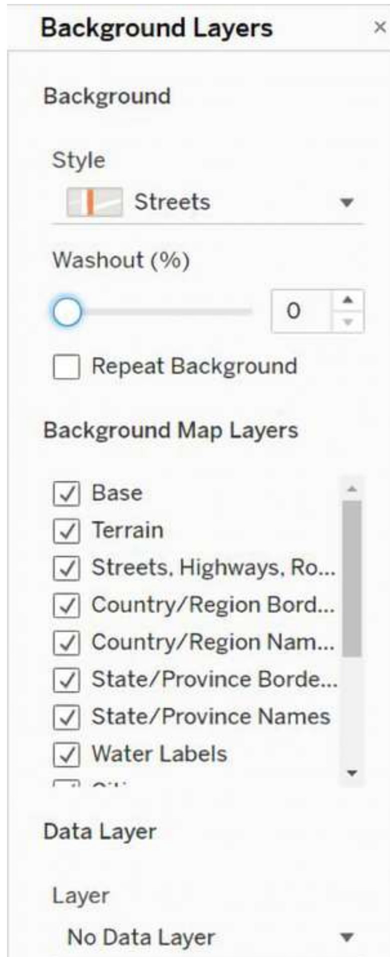


Figure 12.3: The Background Layers pane

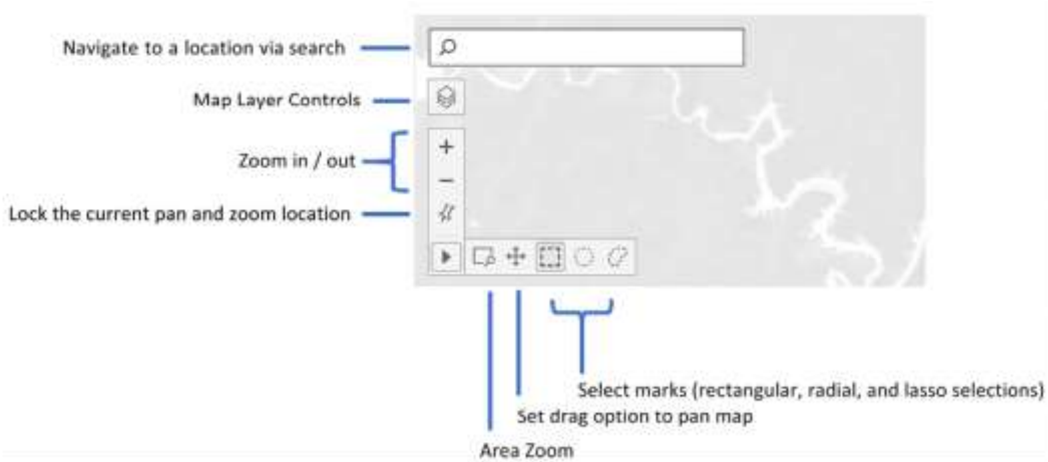


Figure 12.4: Available controls when customizing a map

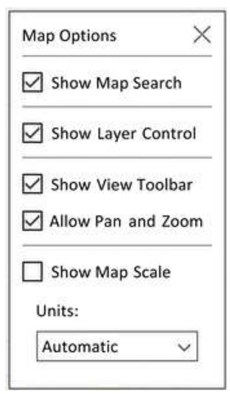


Figure 12.5: Map Options

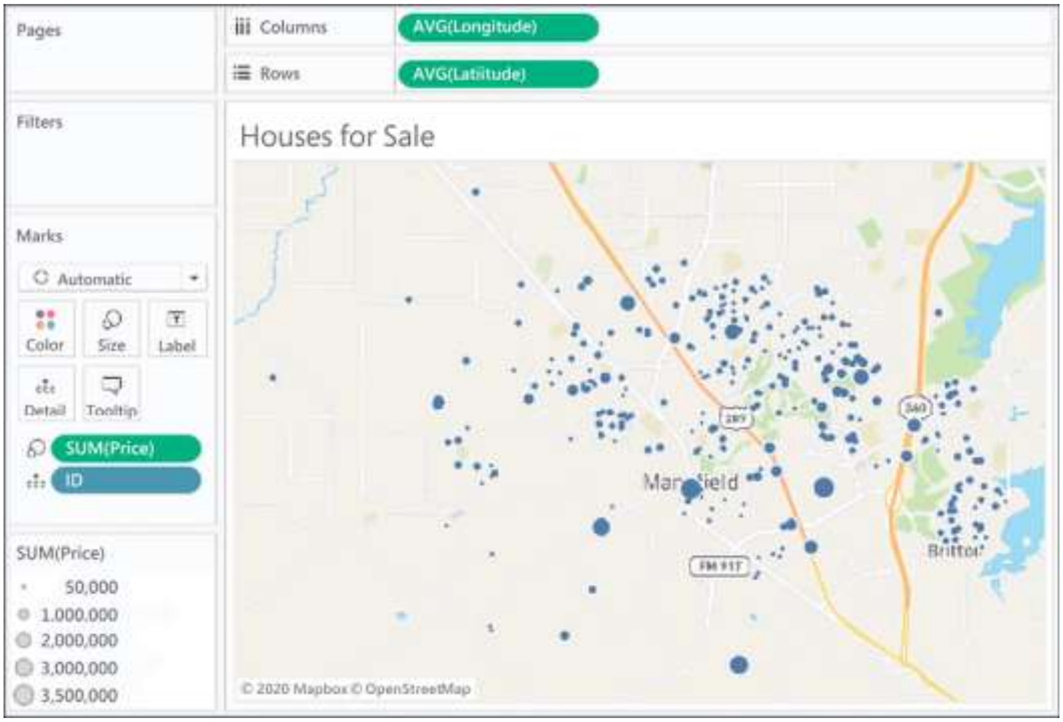


Figure 12.6: A map of houses for sale, sized by price

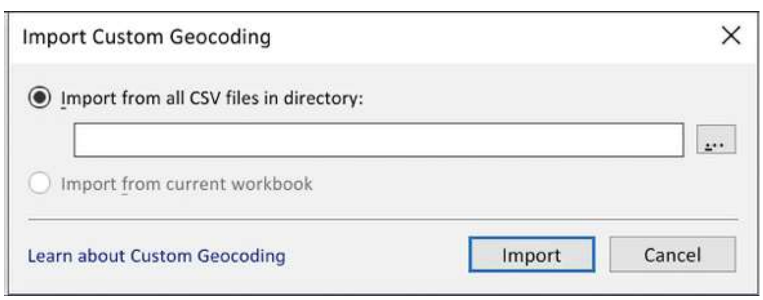


Figure 12.7: The Import Custom Geocoding dialog box

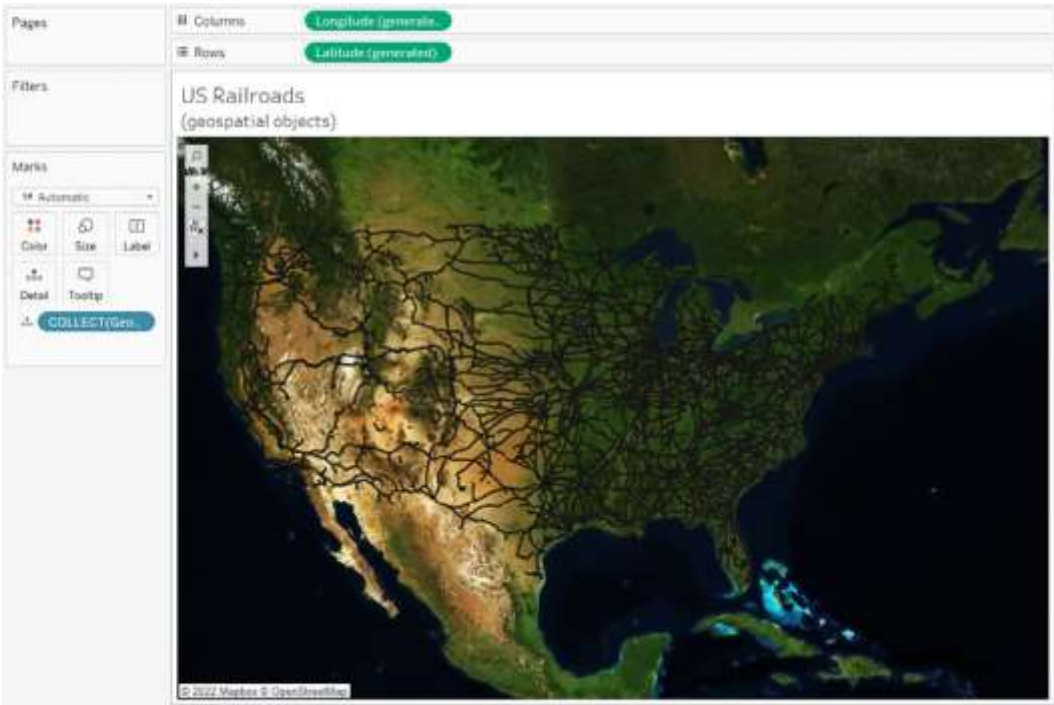


Figure 12.8: Map of US railroads

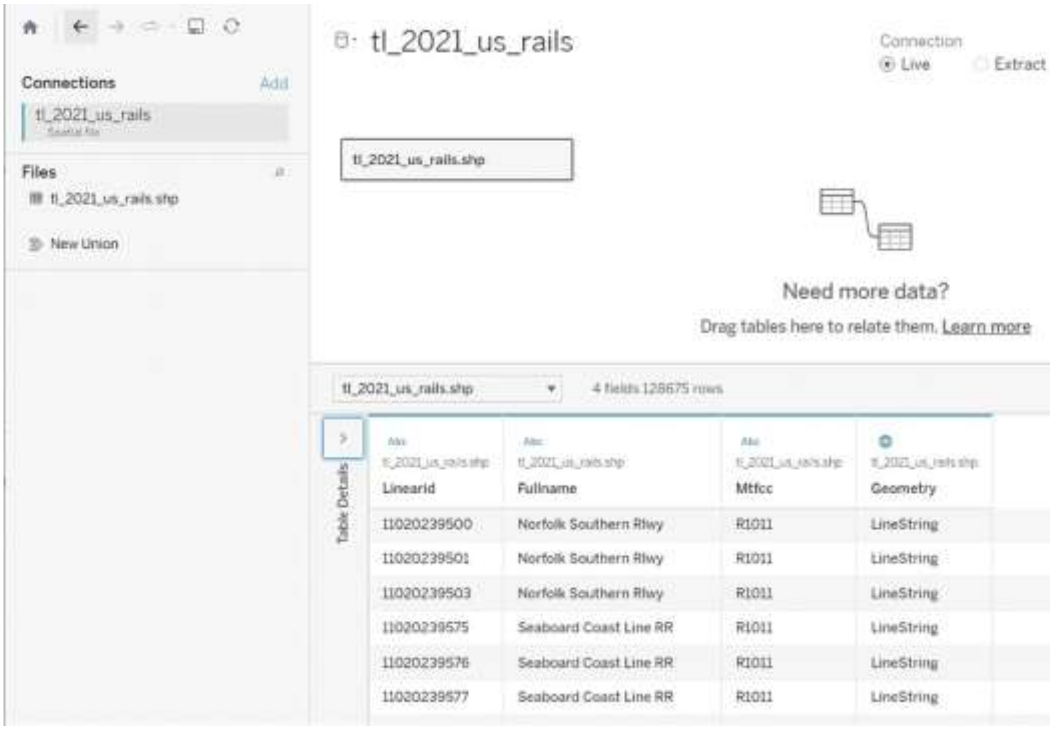


Figure 12.9: Map of US railroads preview

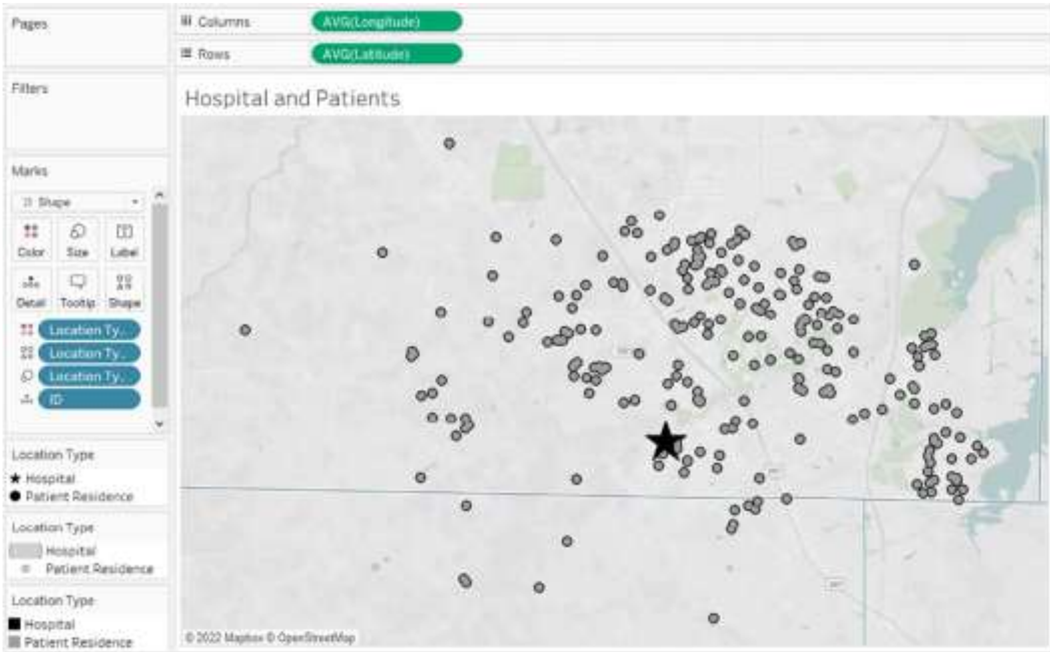


Figure 12.10: A hospital (represented by the star) surrounded by patients

Figure 12.11: Geography icon added to Line field

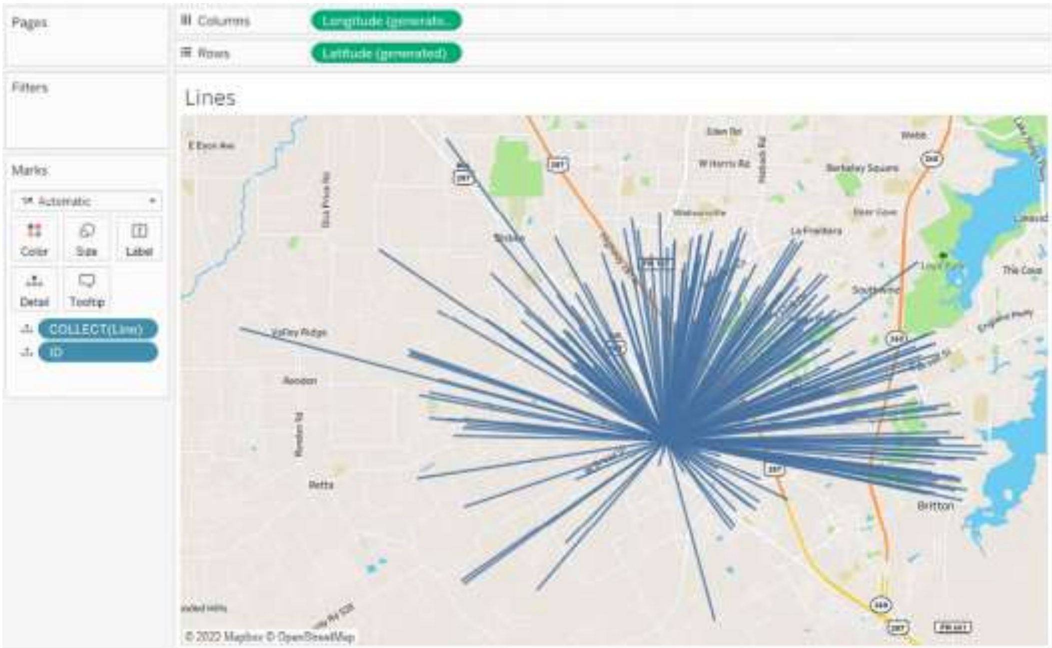


Figure 12.12: Each line originates at the hospital and is drawn to a patient

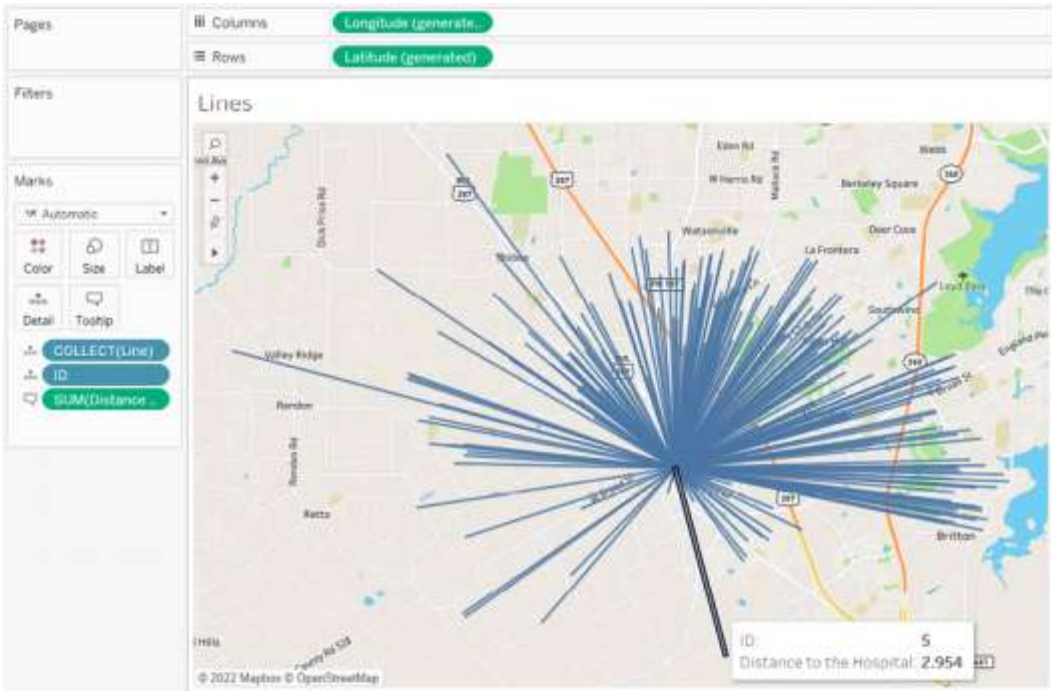


Figure 12.13: The tooltip now displays the distance from the hospital to the patient

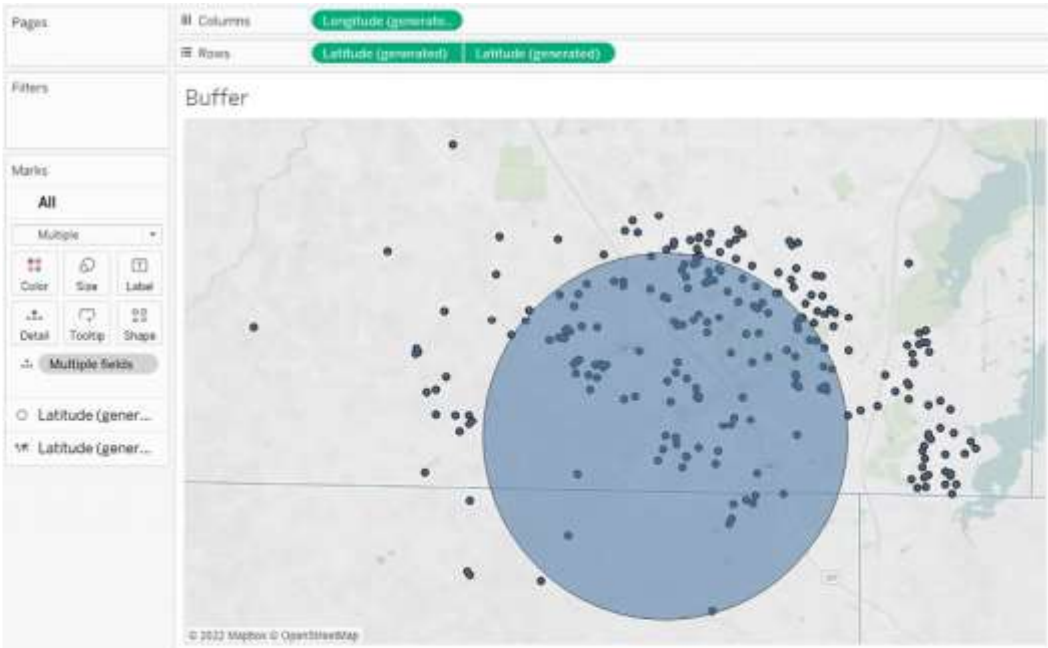


Figure 12.14: Patients who fall within a 3-mile radius of the hospital

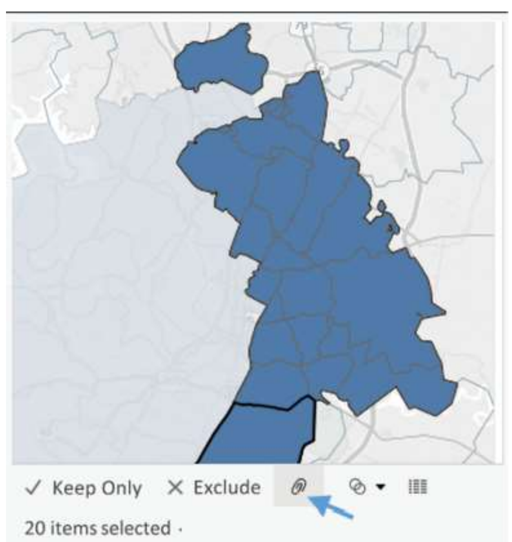


Figure 12.15: After selecting the filled regions to group as a new territory, use the paperclip icon to create the group

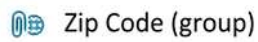


Figure 12.16: A group and geographic field

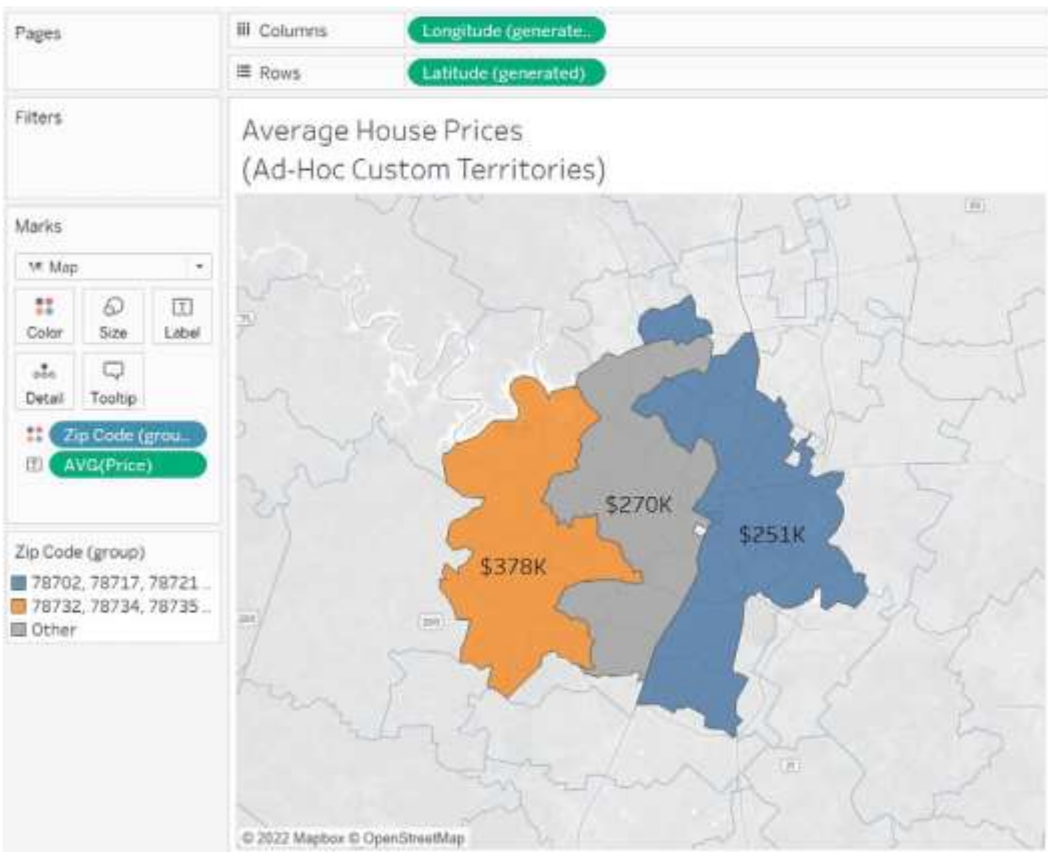


Figure 12.17: Grouping by Custom Territories

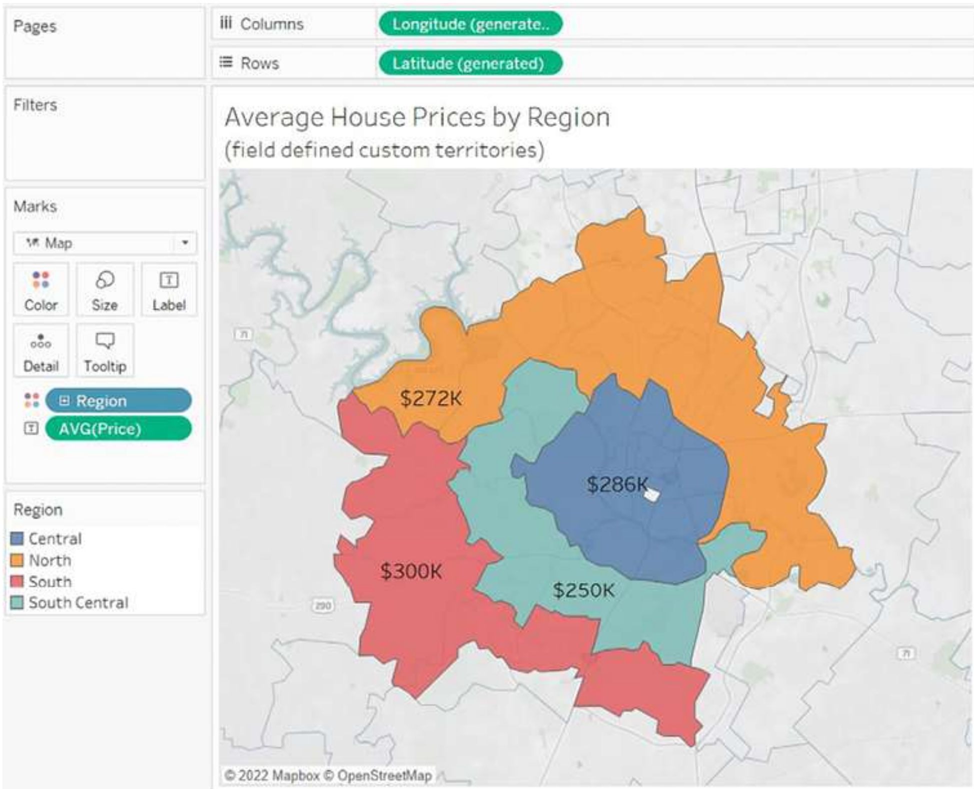


Figure 12.18: The custom regions here are defined by the Region field in the data

Figure 12.19: A filled map of zip codes

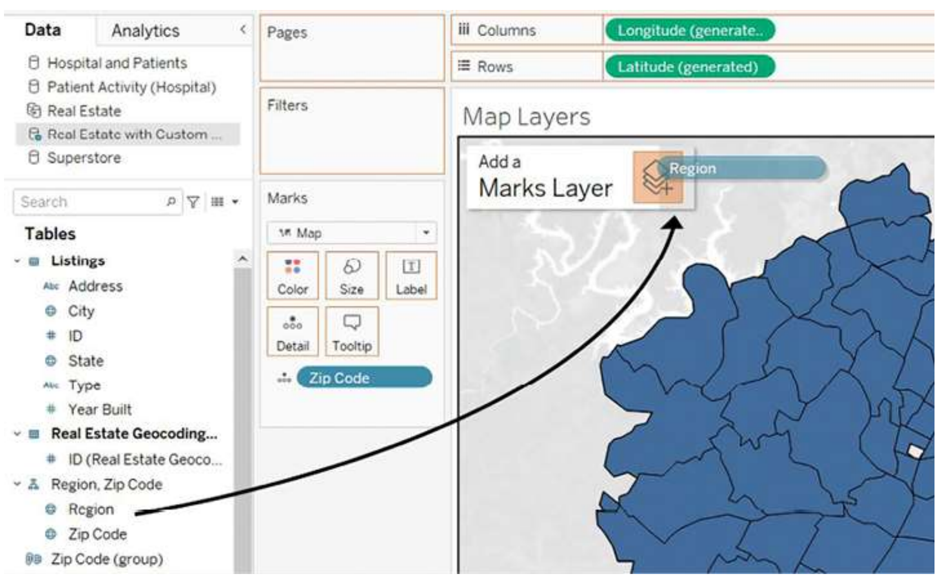


Figure 12.20: Dragging and dropping a geographic field onto the Add a Marks Layer interface adds a layer to the map

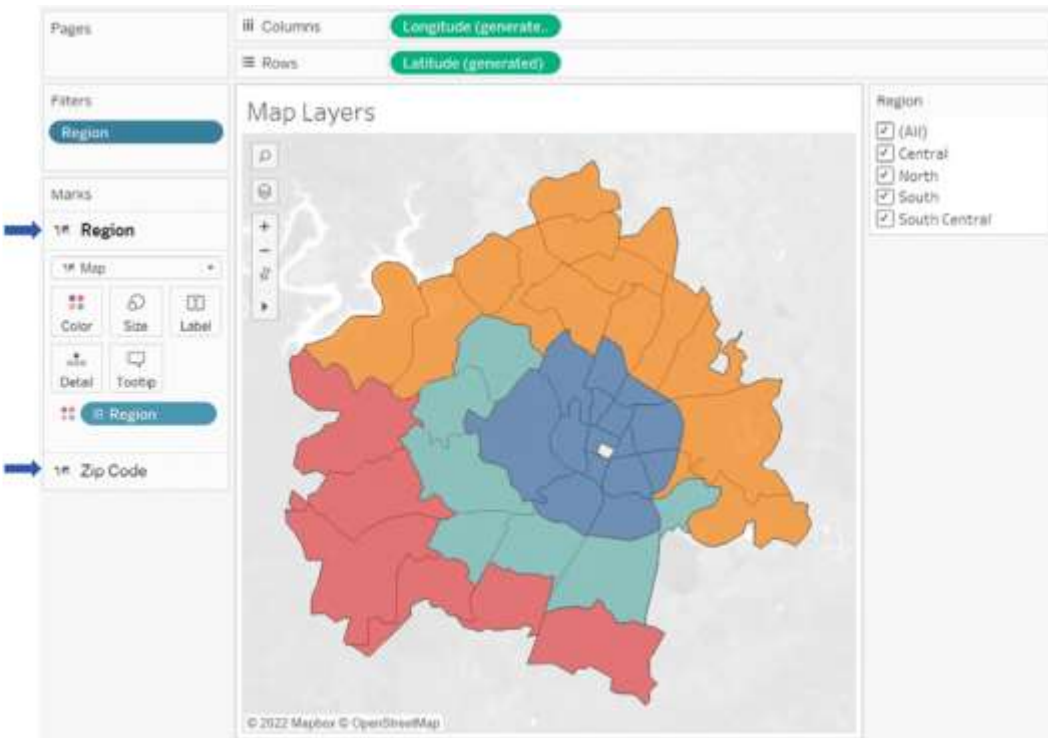


Figure 12.21: Region and Zip Code layers indicated as sections of the Marks card

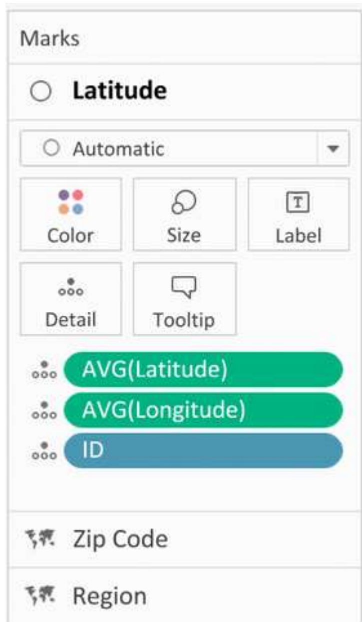


Figure 12.22: Latitude, Longitude, and ID are all required on the layer for Tableau to correctly plot each point

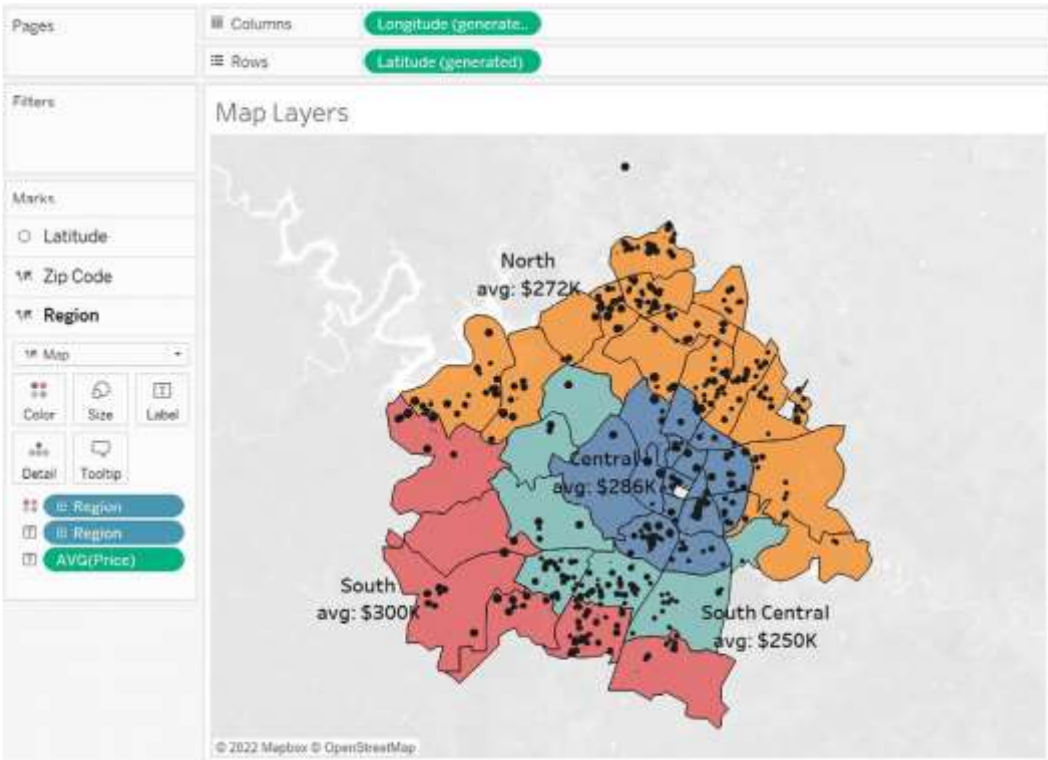


Figure 12.23: A map consisting of three layers

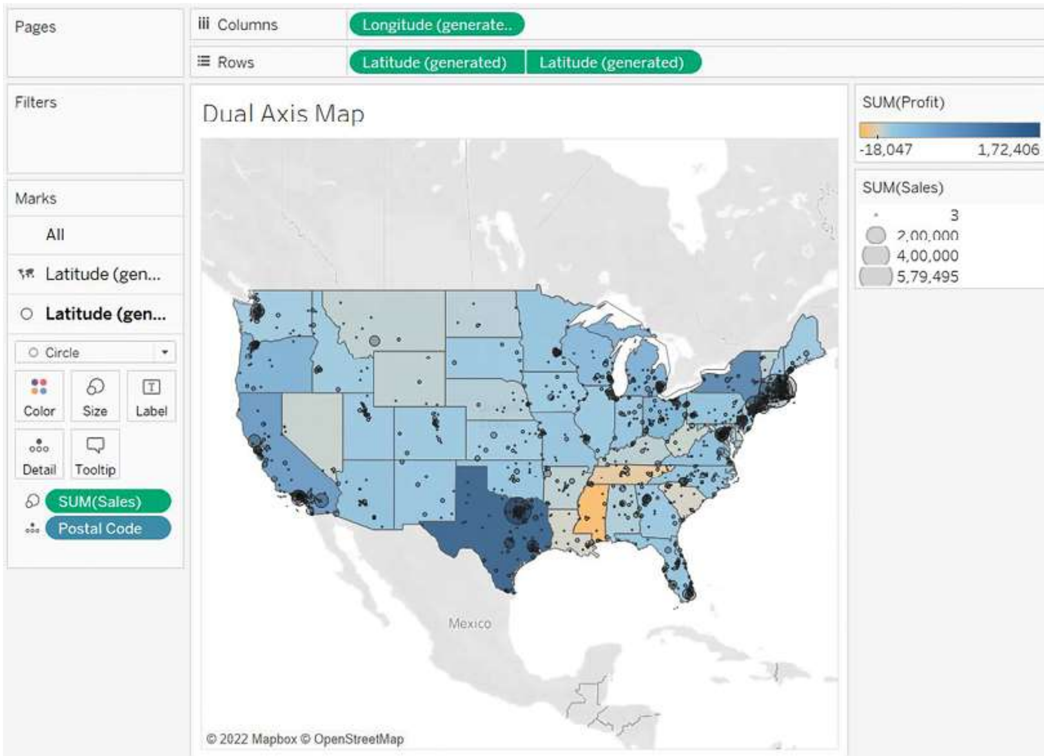


Figure 12.24: Dual Axis Map showing Profit at a state level and Sales at a Postal Code level

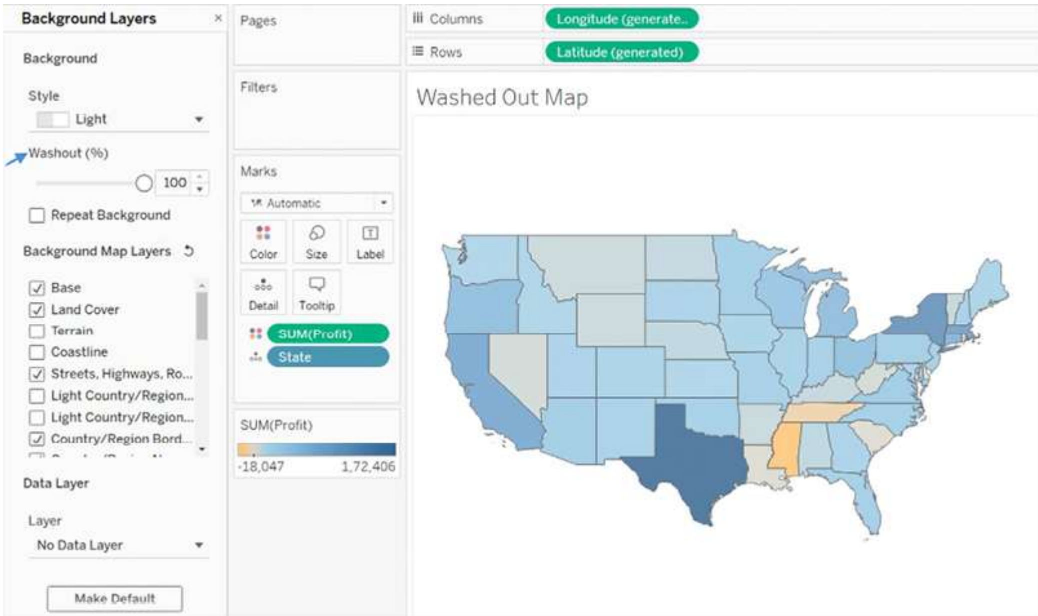


Figure 12.25: Washed Out Map

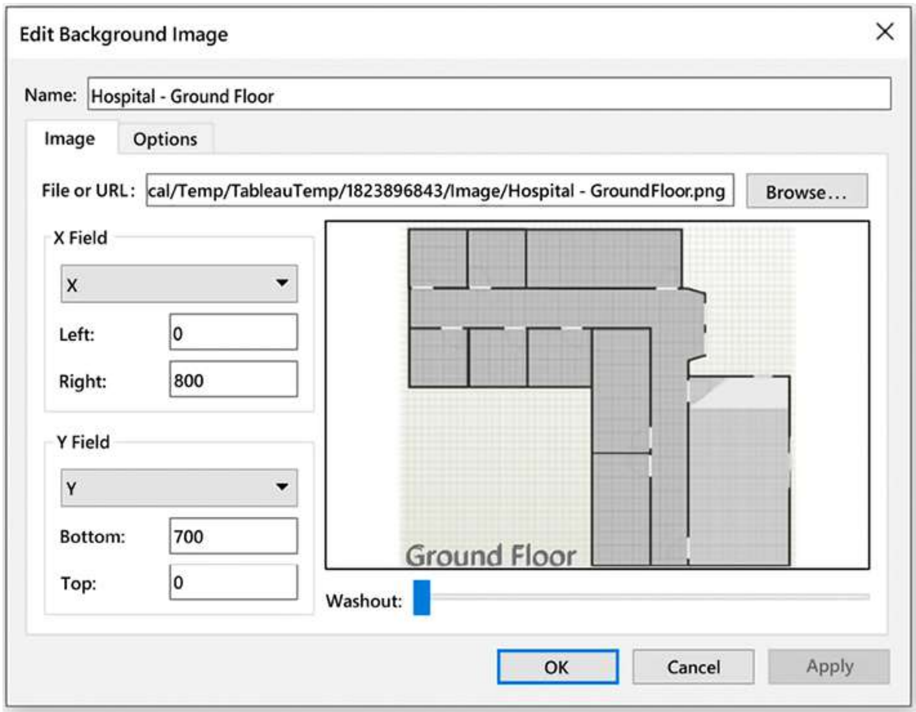


Figure 12.26: Add Background Image pane

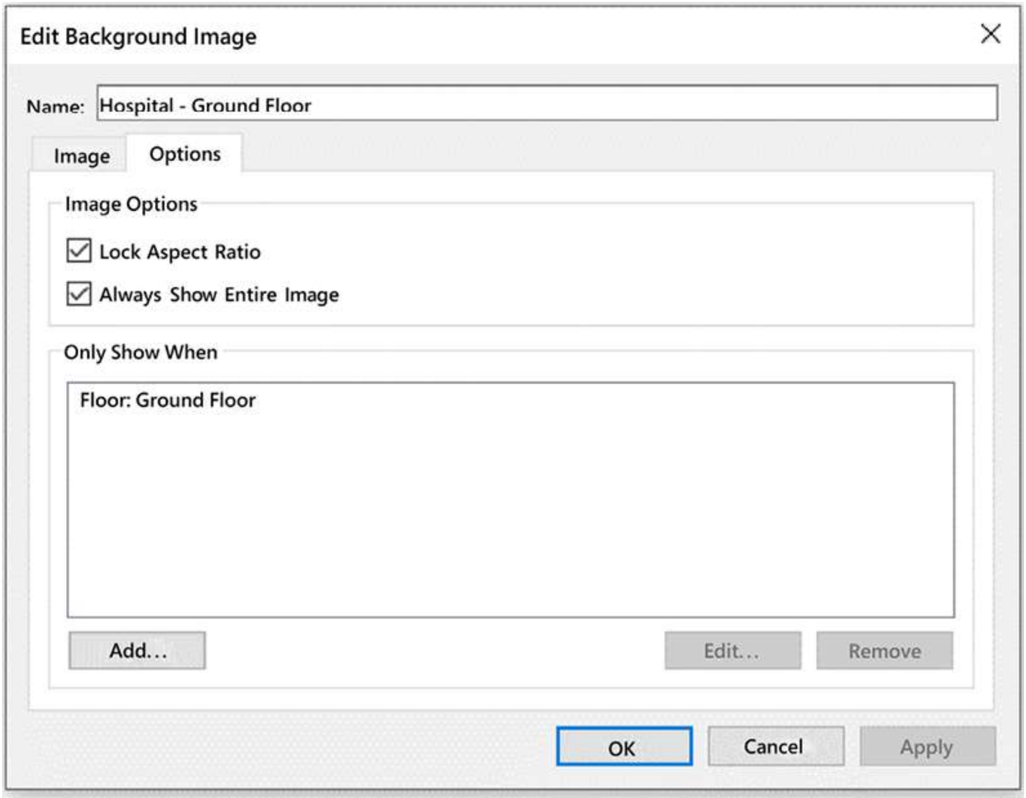


Figure 12.27: Edit Background Image pane

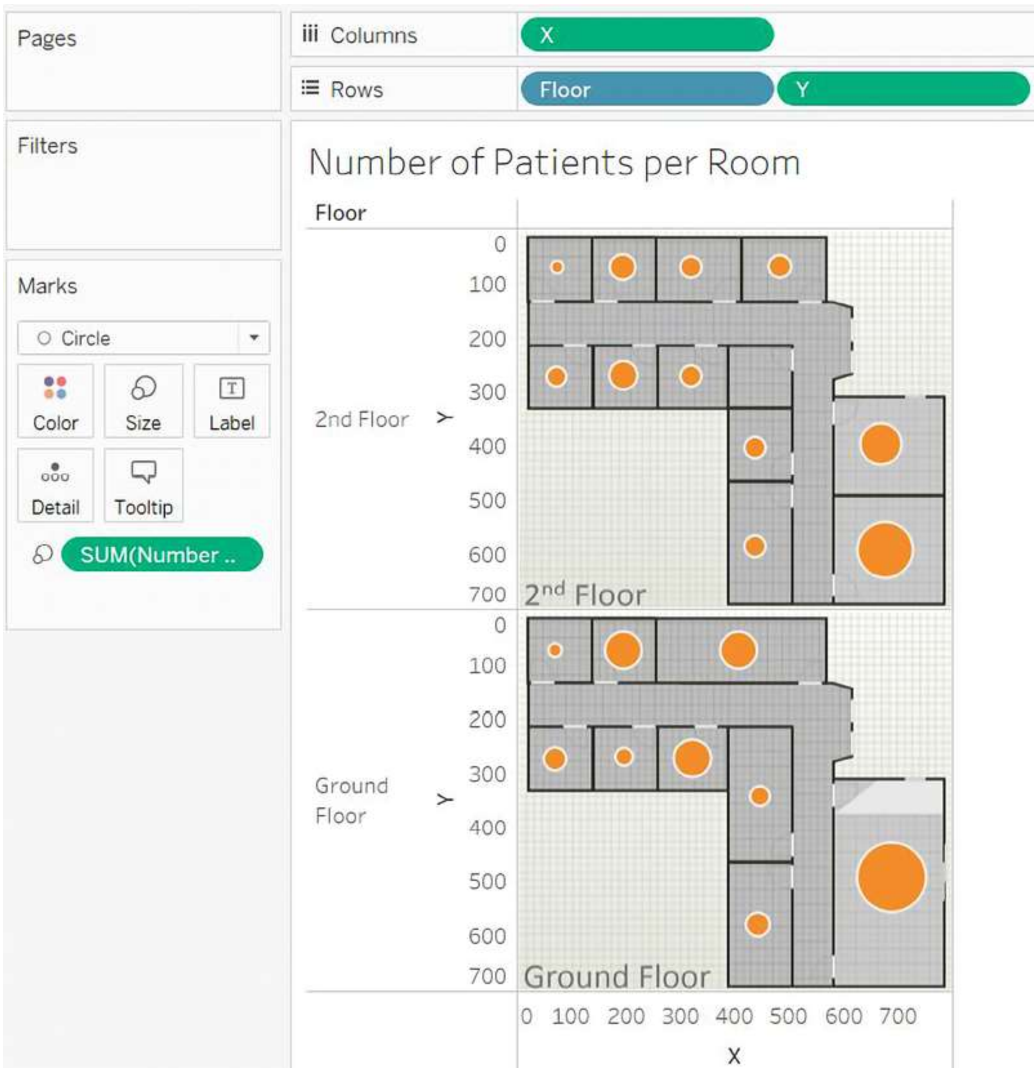


Figure 12.28: Plotting the number of patients per room on a floorplan image

Links

- The details map visualizations are outside the scope of this book; however, you'll find excellent documentation from Tableau at https://help.tableau.com/current/pro/desktop/en-us/maps_mapsources_wms.htm.

- To replicate the example in figure 12.8, download the shapefile from the United States Census Bureau here: https://catalog.data.gov/dataset/tiger-line-shapefile-2015-nation-u-s-rails-national-shapefile.

Code

Code 12.1 – We'll create a calculation called Hospital Latitude with the following code:

```
{FIXED : MIN(IF [Location Type] == "Hospital" THEN
[Latitude] END)}
```

Code 12.2 – And a corresponding calculation called Hospital Longitude with the following code:

```
{FIXED : MIN(IF [Location Type] == "Hospital" THEN
[Longitude] END)}
```

Code 12.3 – MAKELINE() and MAKEPOINT()

Create a calculated field to draw a line between the hospital and each patient. We'll name our calculation Line and write this code:

```
MAKELINE(
    MAKEPOINT([Hospital Latitude], [Hospital Longitude]),
    MAKEPOINT([Latitude], [Longitude])
)
```

Code 12.4 – DISTANCE()

We might create a calculated field named Distance to the Hospital with the following code:

```
DISTANCE(
    MAKEPOINT([Hospital Latitude], [Hospital Longitude]),
    MAKEPOINT([Latitude], [Longitude]),
    'mi'
)
```

Code 12.5 – BUFFER()

we'll create a calculated field named Hospital Radius, with the following code:

```
IF [Location Type] == "Hospital"
THEN BUFFER(MAKEPOINT([Latitude], [Longitude]), 3, 'mi')
END
```

Chapter 13

Figures

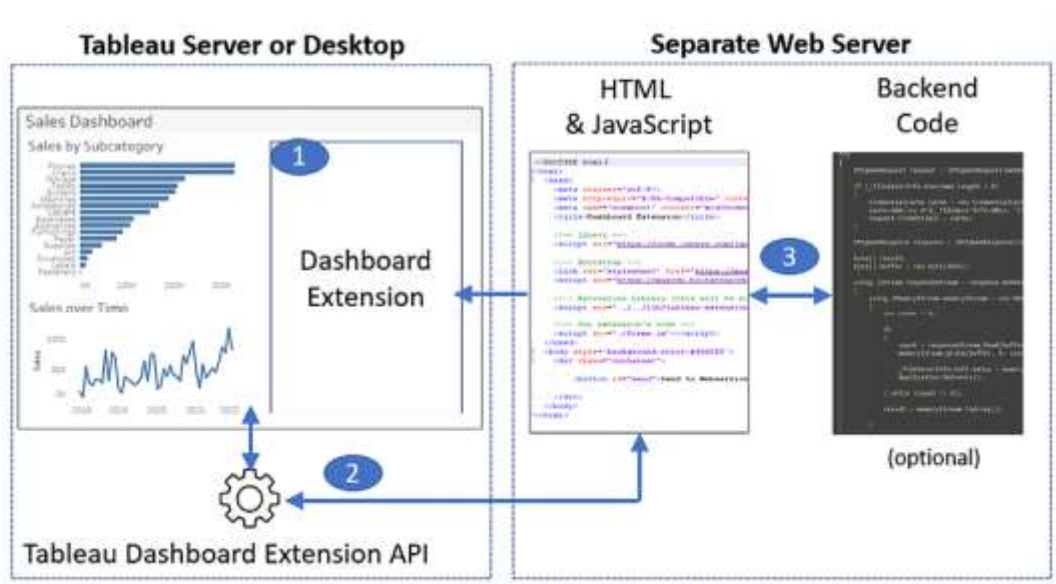


Figure 13.1: The basic architecture of a Tableau Dashboard Extension

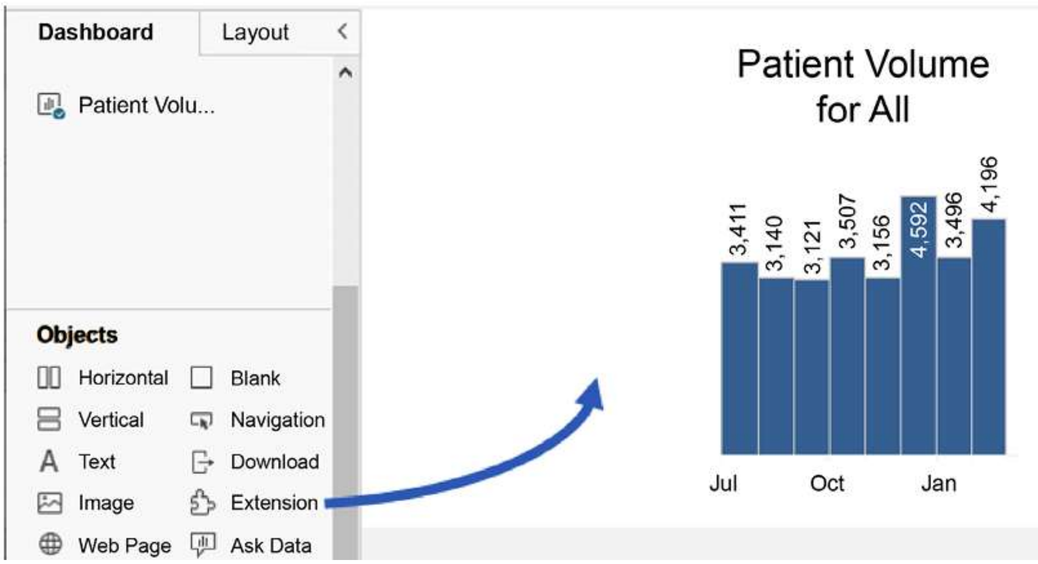


Figure 13.2: Drag and drop the Extension object into the dashboard

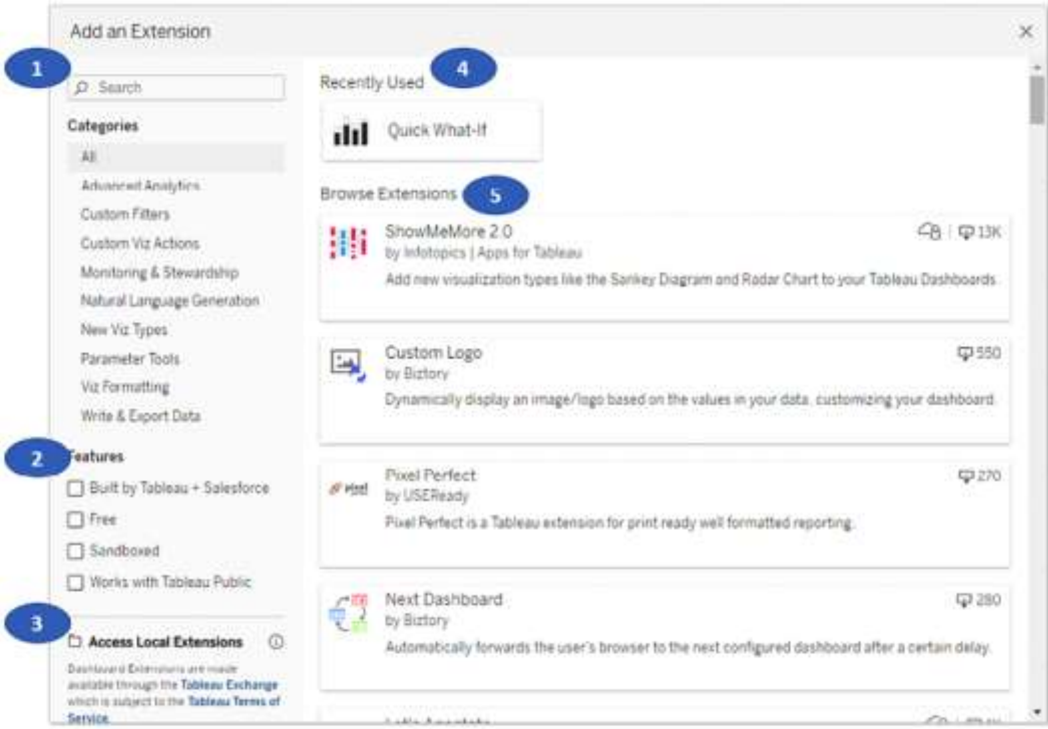


Figure 13.3: The Add an Extension dialog

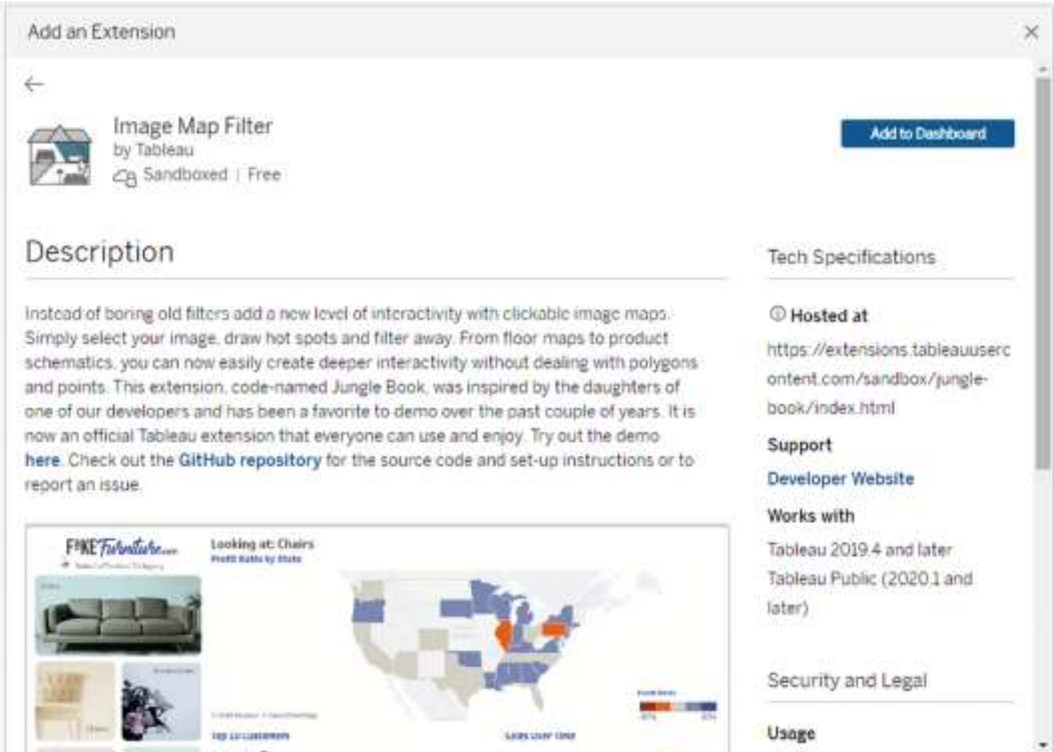


Figure 13.4: The description and specifications for the Image Map Filter dashboard extension

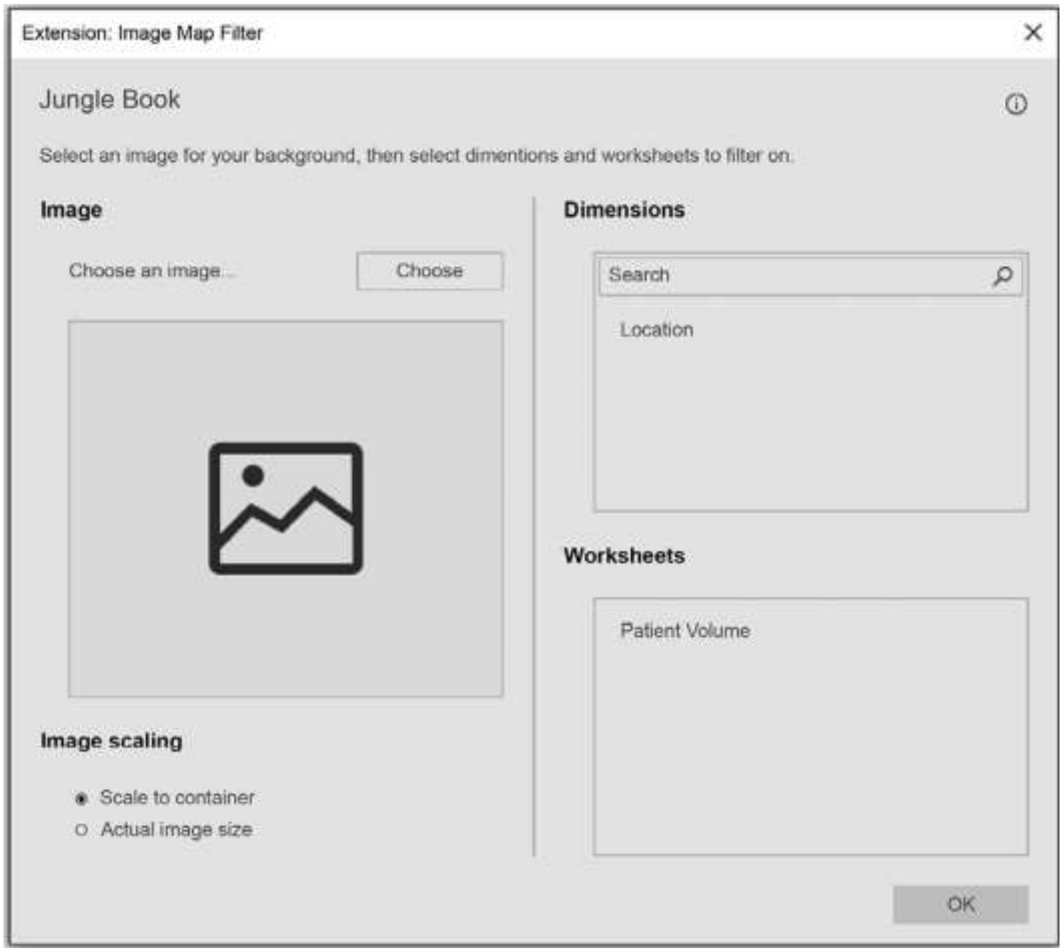


Figure 13.5: The description and specifications for the Image Map Filter dashboard extension

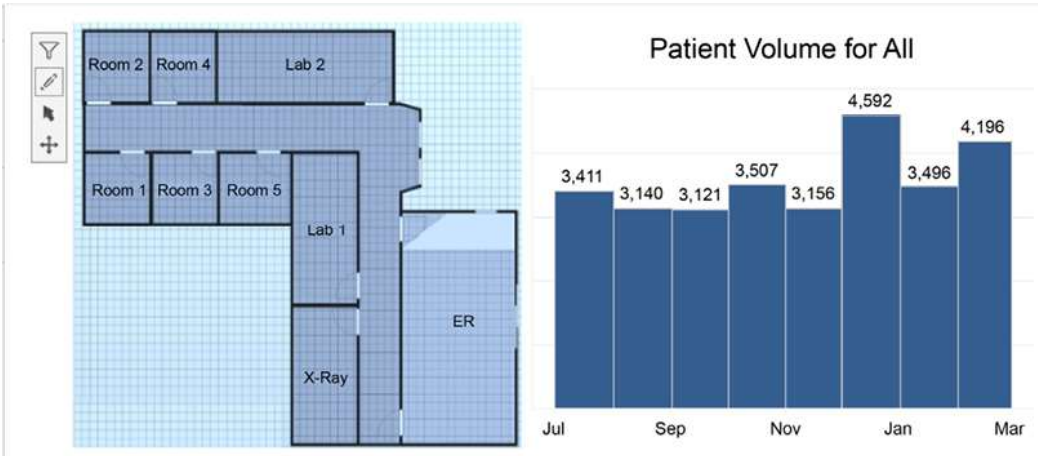


Figure 13.6: The dashboard with the configured dashboard extension

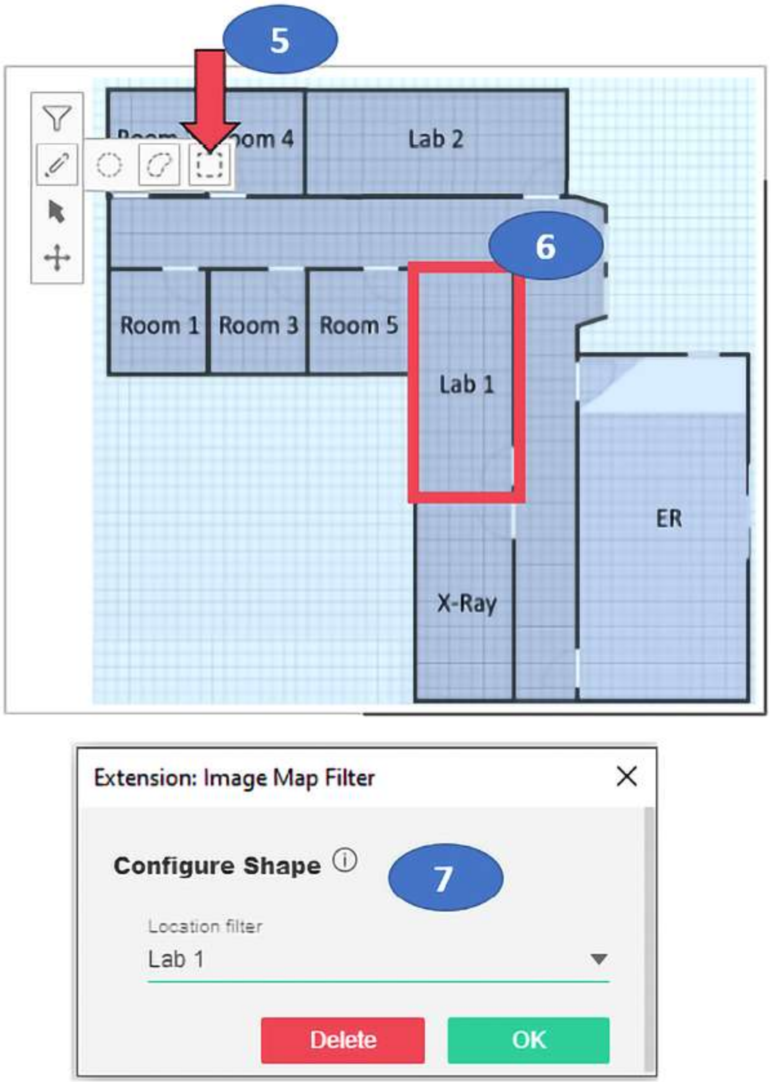


Figure 13.7: Drawing a shape around each room and matching them with the corresponding filter option

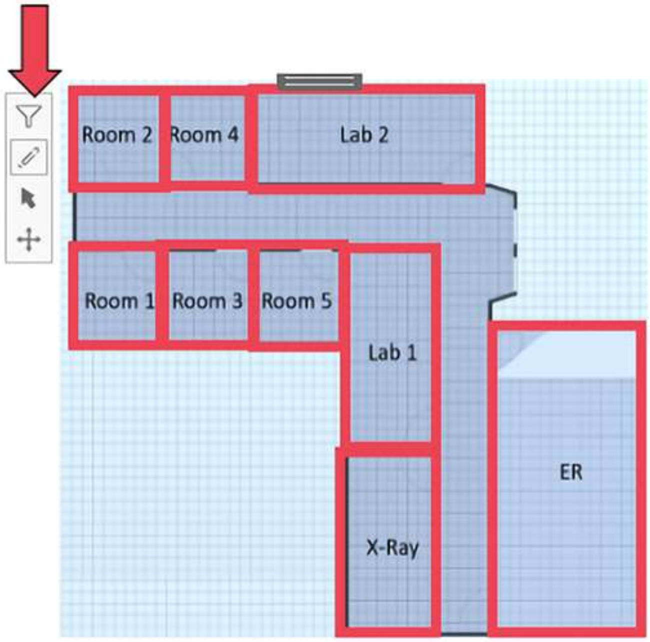


Figure 13.8: Switch to filtering once all shapes are mapped

Figure 13.9: Adding a local extension

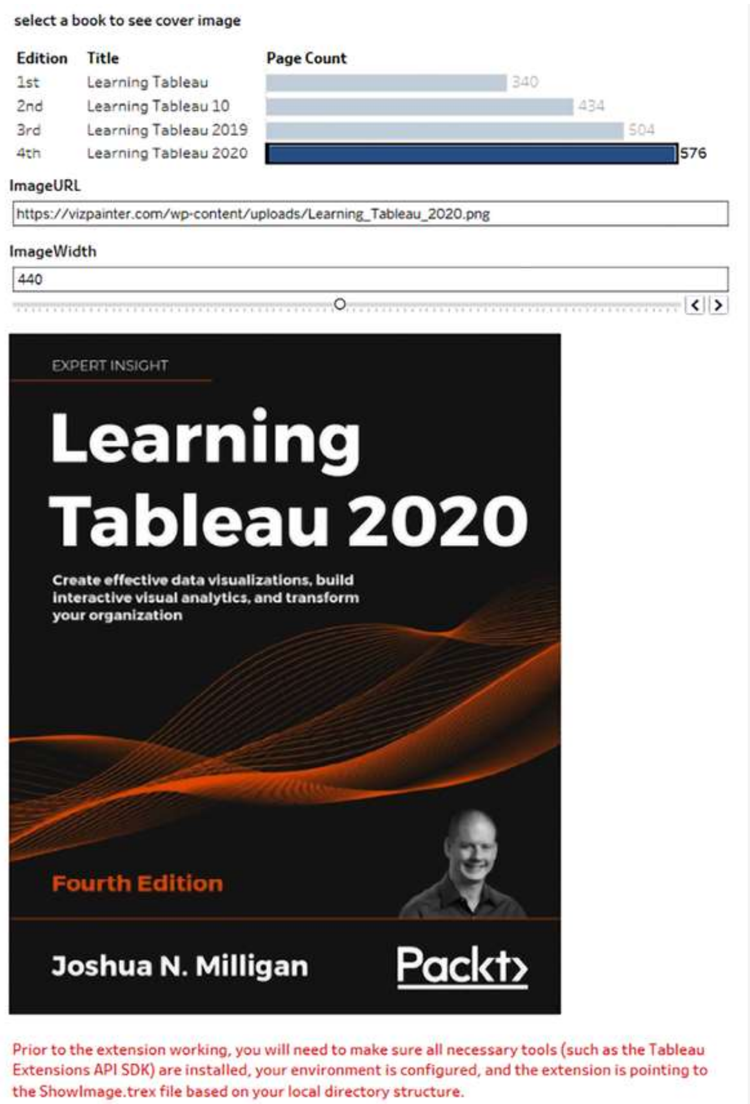


Figure 13.10: The dashboard with embedded ShowImage extension

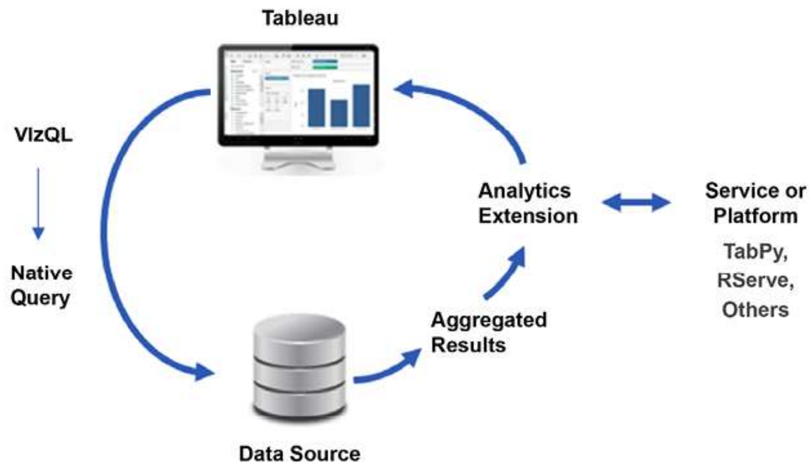


Figure 13.11: Analytics extensions are implemented based on aggregated results from the data source

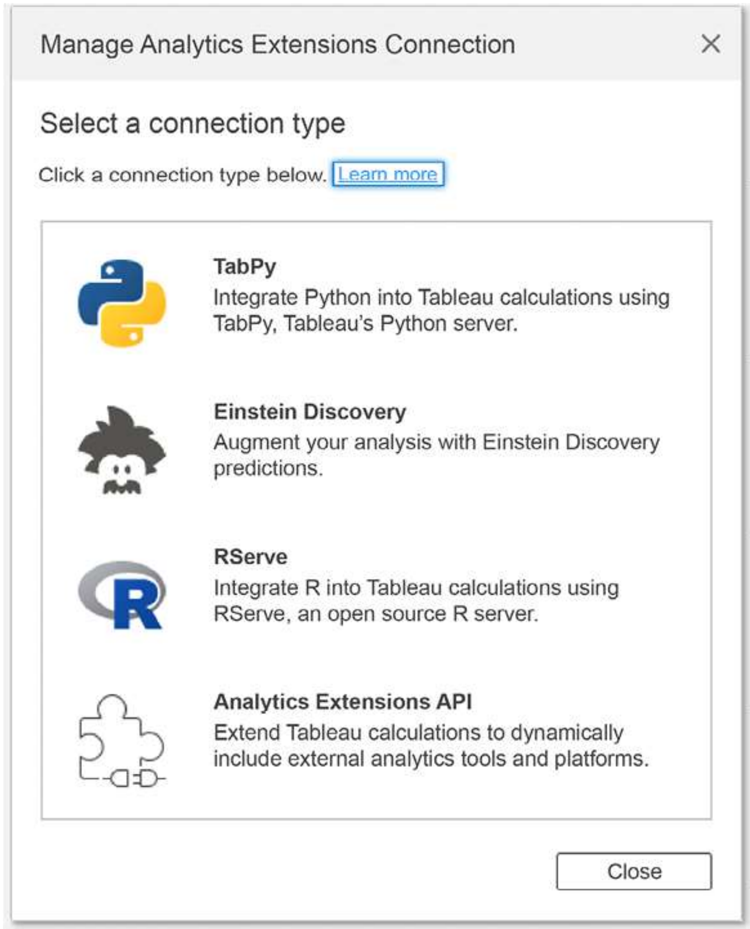


Figure 13.12: The Manage Analytics Extensions Connection dialog allows you to configure options for various analytics extension types

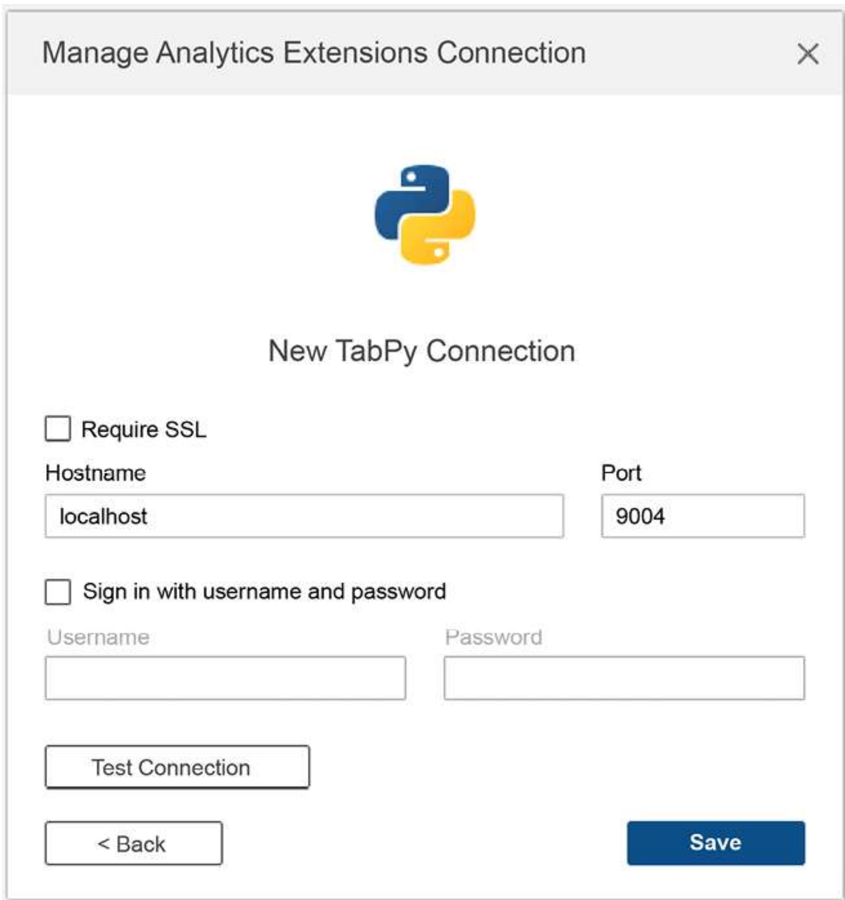


Figure 13.13: Setting up a new TabPy connection with hostname and port information

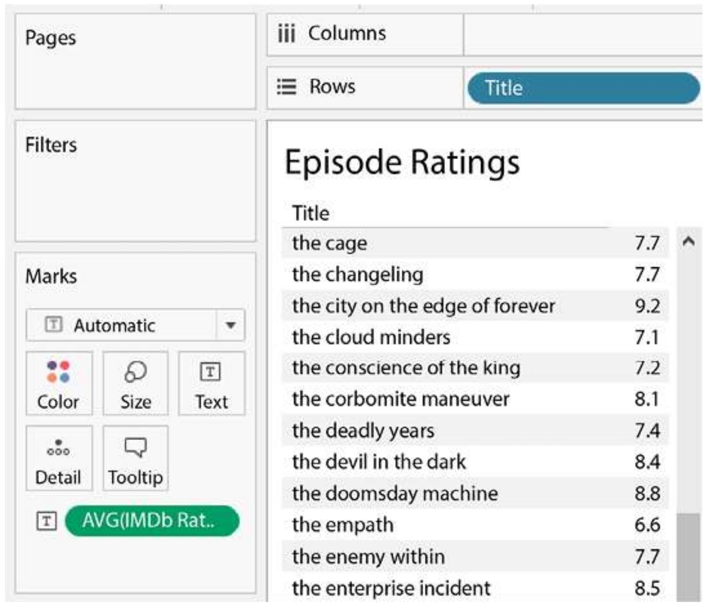


Figure 13.14: Episodes of Star Trek along with IMDb ratings

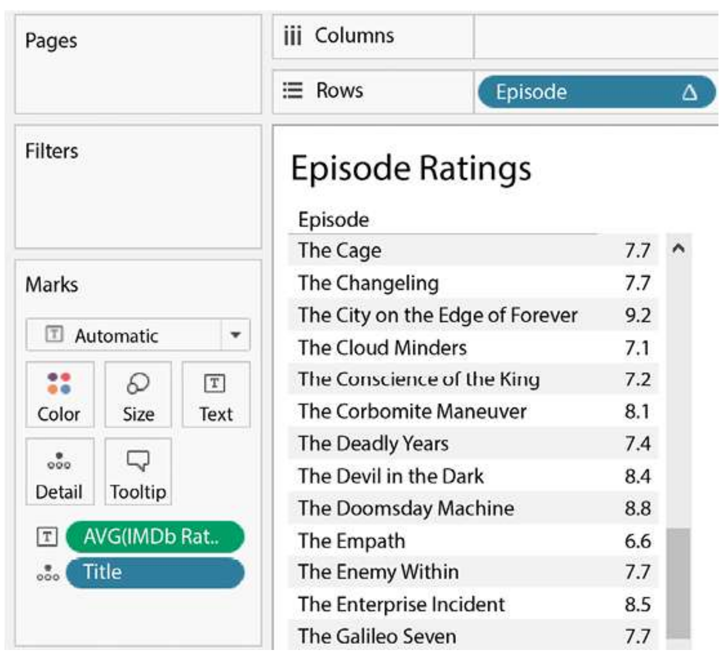


Figure 13.15: Episodes of Star Trek along with IMDb ratings

Figure 13.16: The Analytics tab in the Salesforce menu

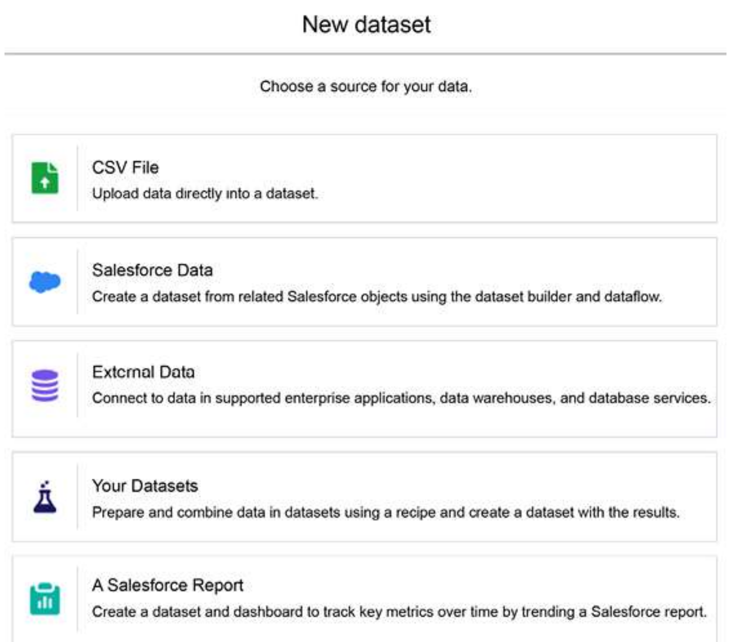


Figure 13.17: Selecting a new dataset for the analytics

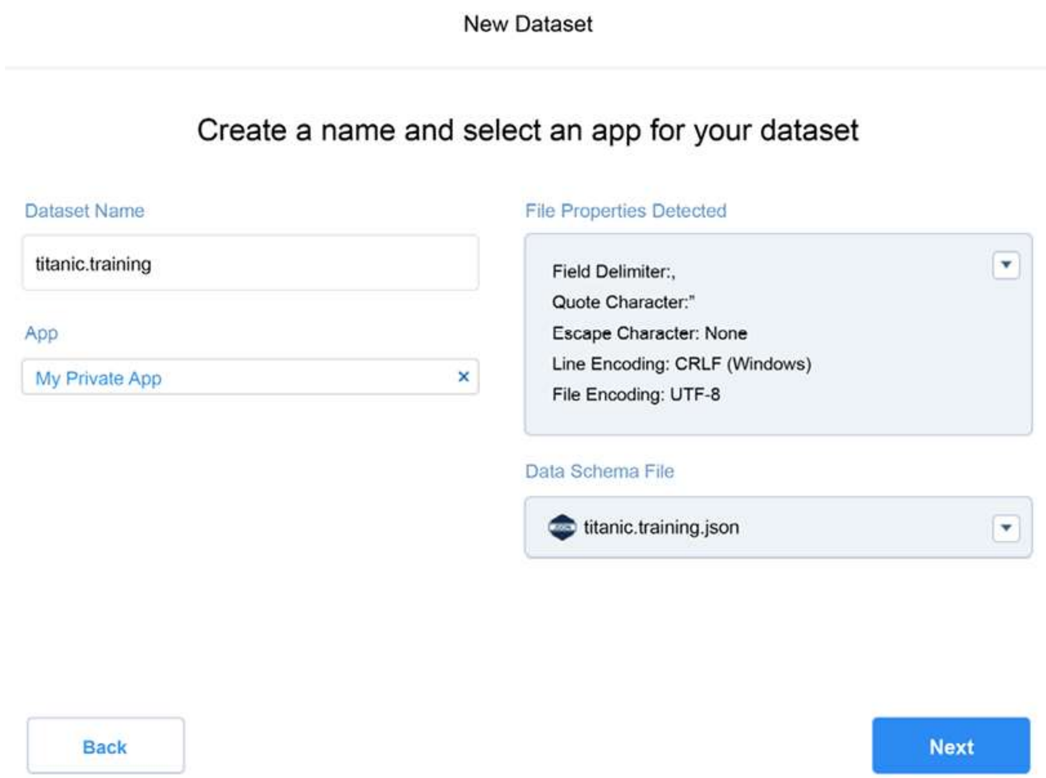


Figure 13.18: Specifying the options for your new dataset

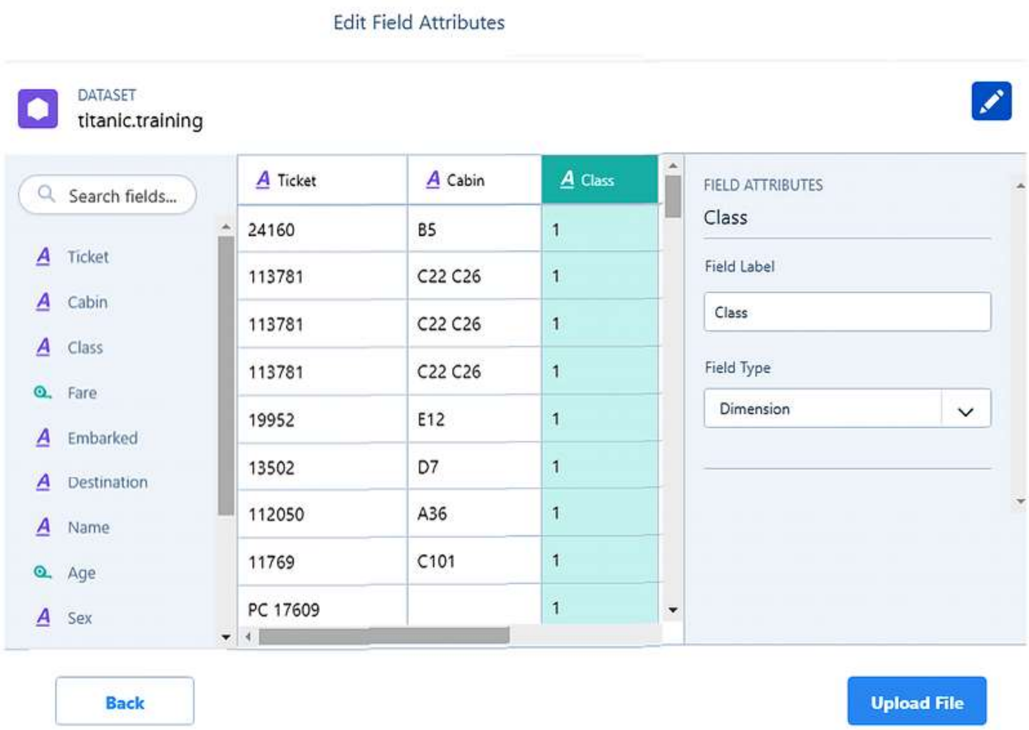


Figure 13.19: Editing field attributes

Figure 13.20: Dataset options

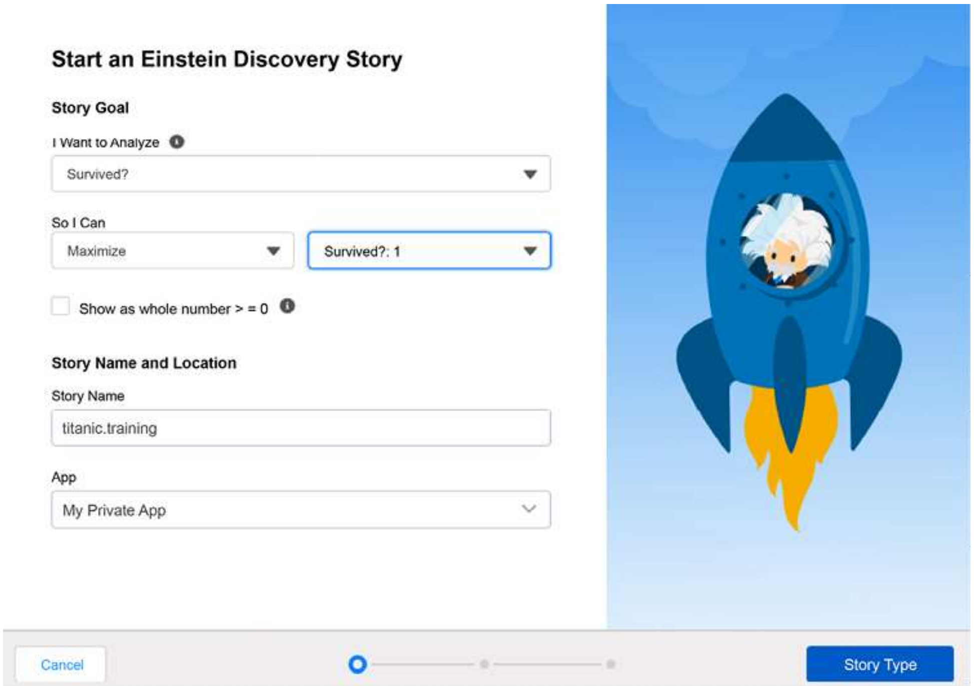


Figure 13.21: Starting and configuring our Einstein Discovery story

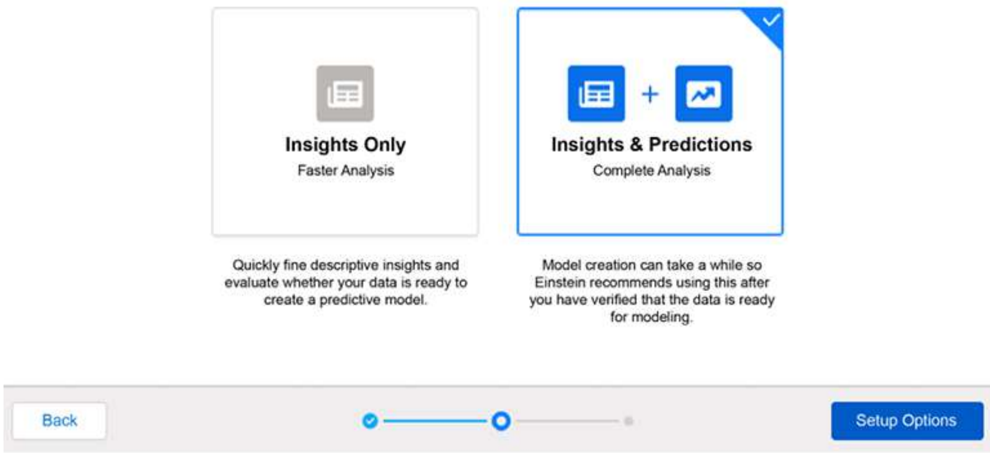


Figure 13.22: Selecting the story type

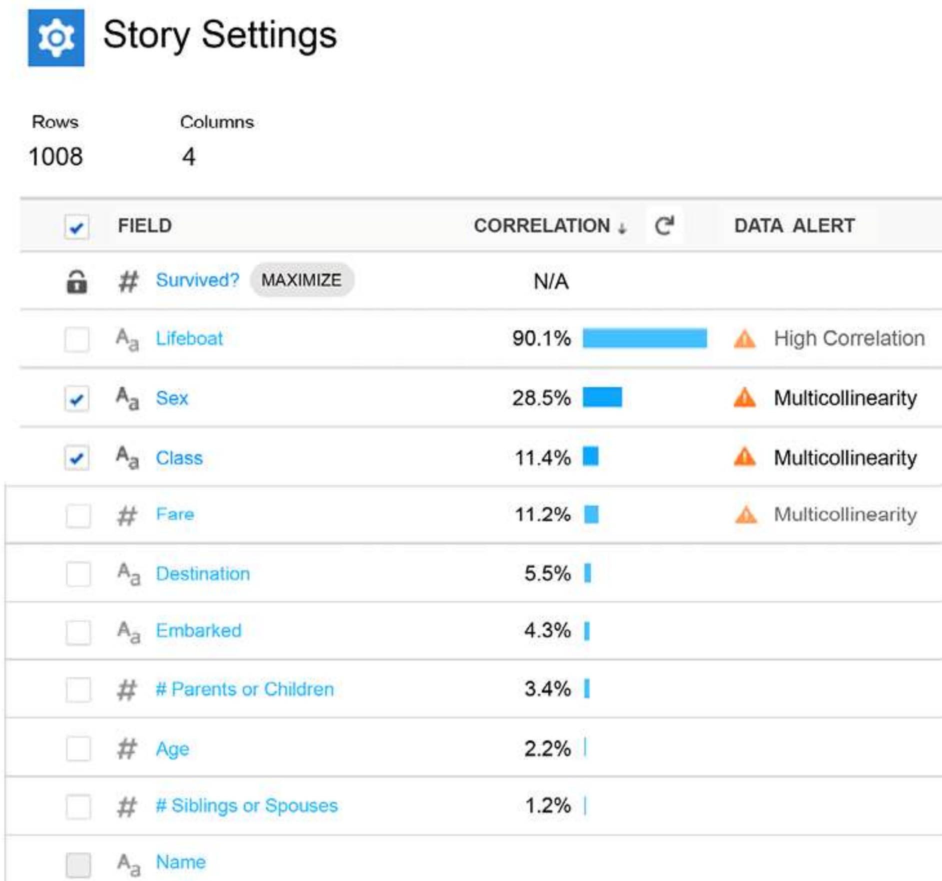


Figure 13.23: Story settings, correlation, and alerts

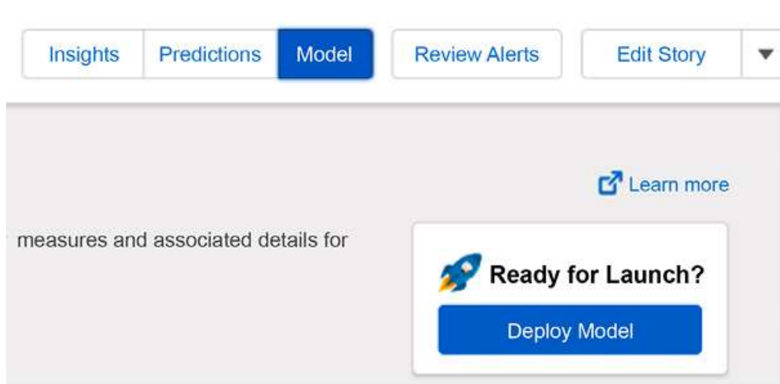


Figure 13.24: Deploying an Einstein Discovery model

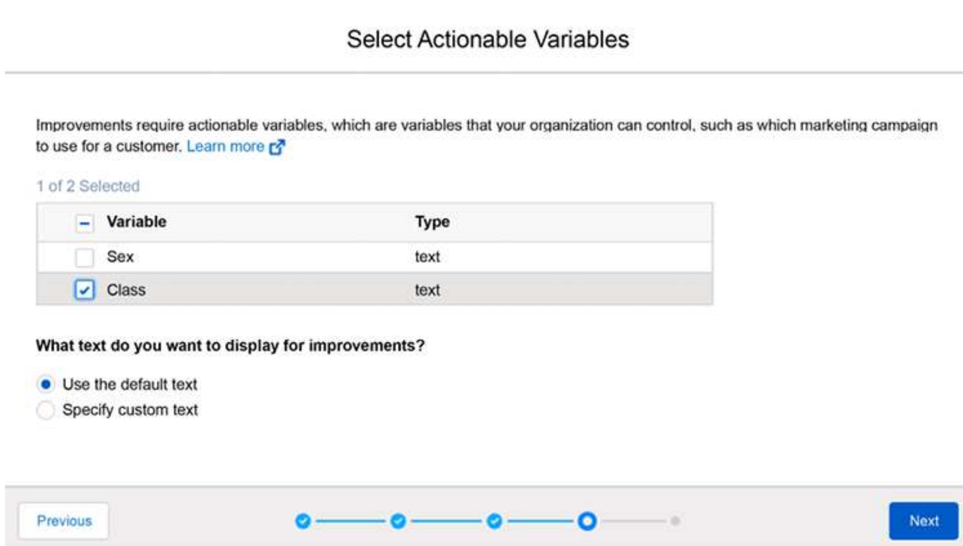


Figure 13.25: Selecting actionable variables

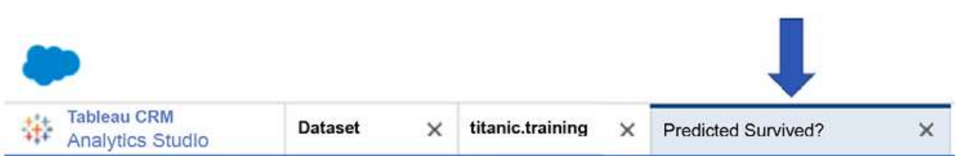


Figure 13.26: Selecting your Einstein model

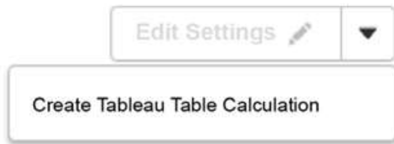
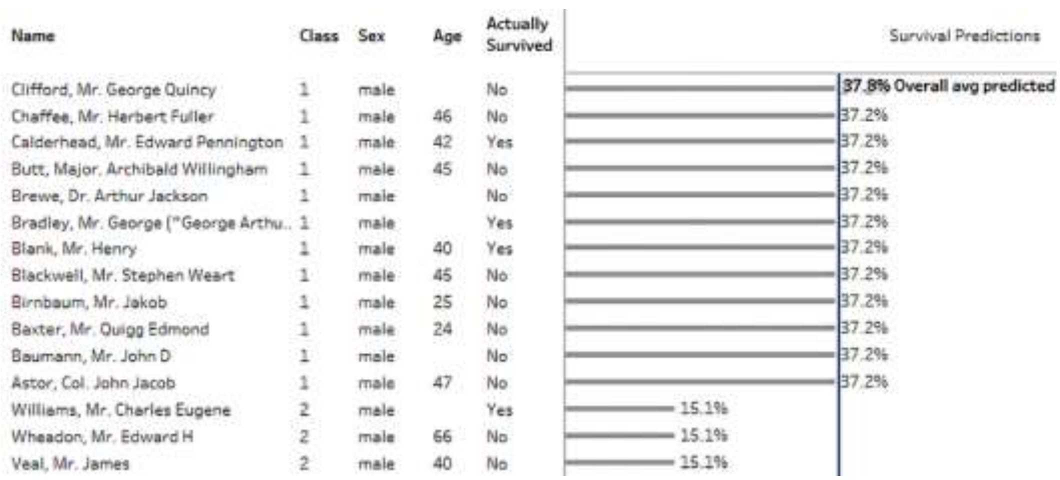


Figure 13.27: Generating the table calculation from within Salesforce



Name	Class	Sex	Age	Actually Survived	Survival Predictions
Clifford, Mr. George Quincy	1	male		No	37.8% Overall avg predicted
Chaffee, Mr. Herbert Fuller	1	male	46	No	37.2%
Calderhead, Mr. Edward Pennington	1	male	42	Yes	37.2%
Butt, Major. Archibald Willingham	1	male	45	No	37.2%
Brewe, Dr. Arthur Jackson	1	male		No	37.2%
Bradley, Mr. George ("George Arthu..	1	male		Yes	37.2%
Blank, Mr. Henry	1	male	40	Yes	37.2%
Blackwell, Mr. Stephen Weart	1	male	45	No	37.2%
Birnbaum, Mr. Jakob	1	male	25	No	37.2%
Baxter, Mr. Quigg Edmond	1	male	24	No	37.2%
Baumann, Mr. John D	1	male		No	37.2%
Astor, Col. John Jacob	1	male	47	No	37.2%
Williams, Mr. Charles Eugene	2	male		Yes	15.1%
Wheadon, Mr. Edward H	2	male	66	No	15.1%
Veal, Mr. James	2	male	40	No	15.1%

Figure 13.28: Einstein predictions of survival based on Class and Sex

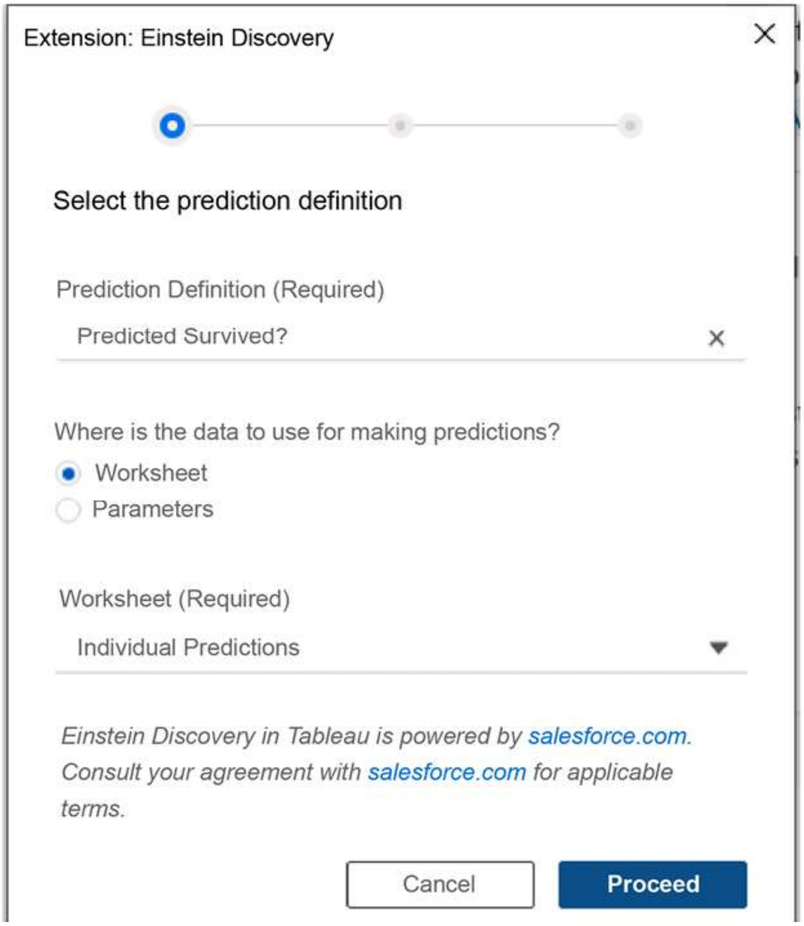


Figure 13.29: Configuration of Einstein Discovery dashboard extension

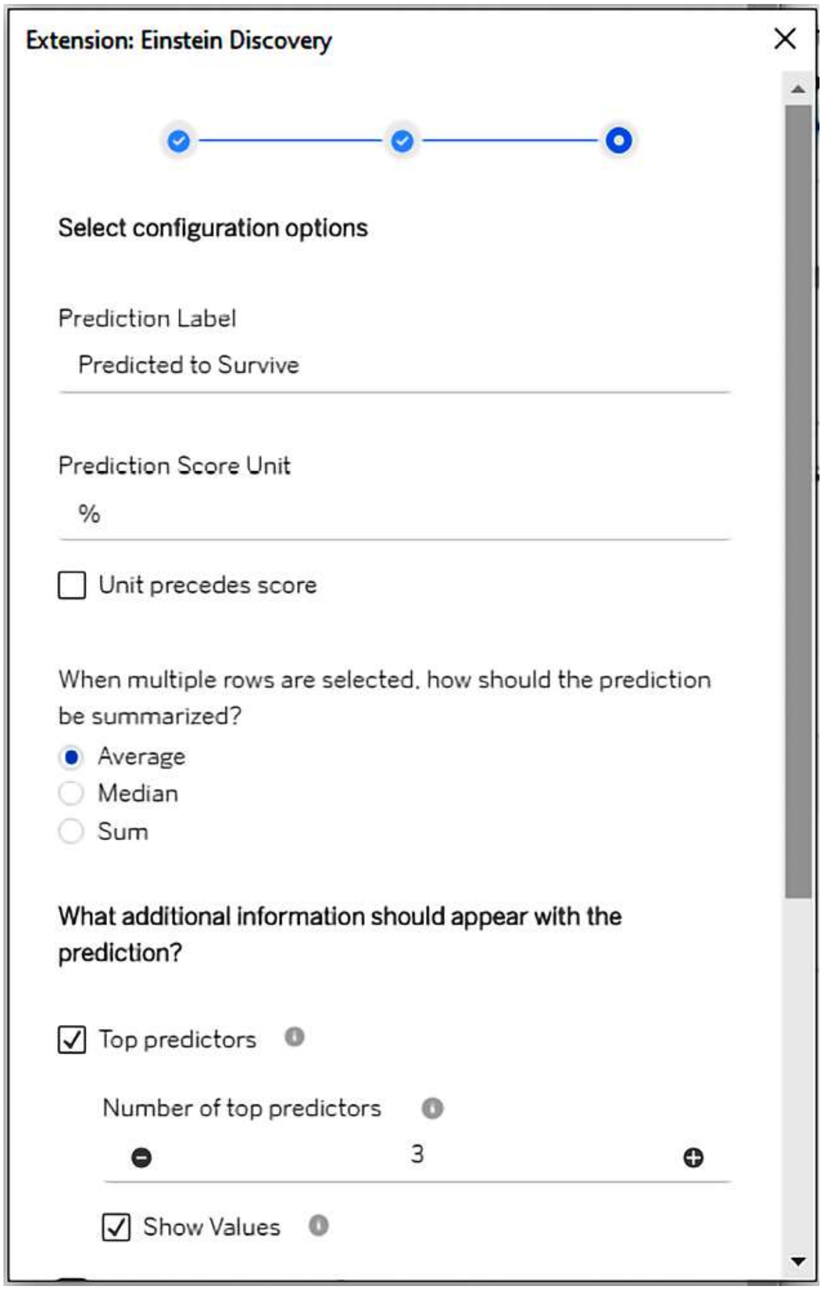


Figure 13.30: Setting label and units on final configuration screen

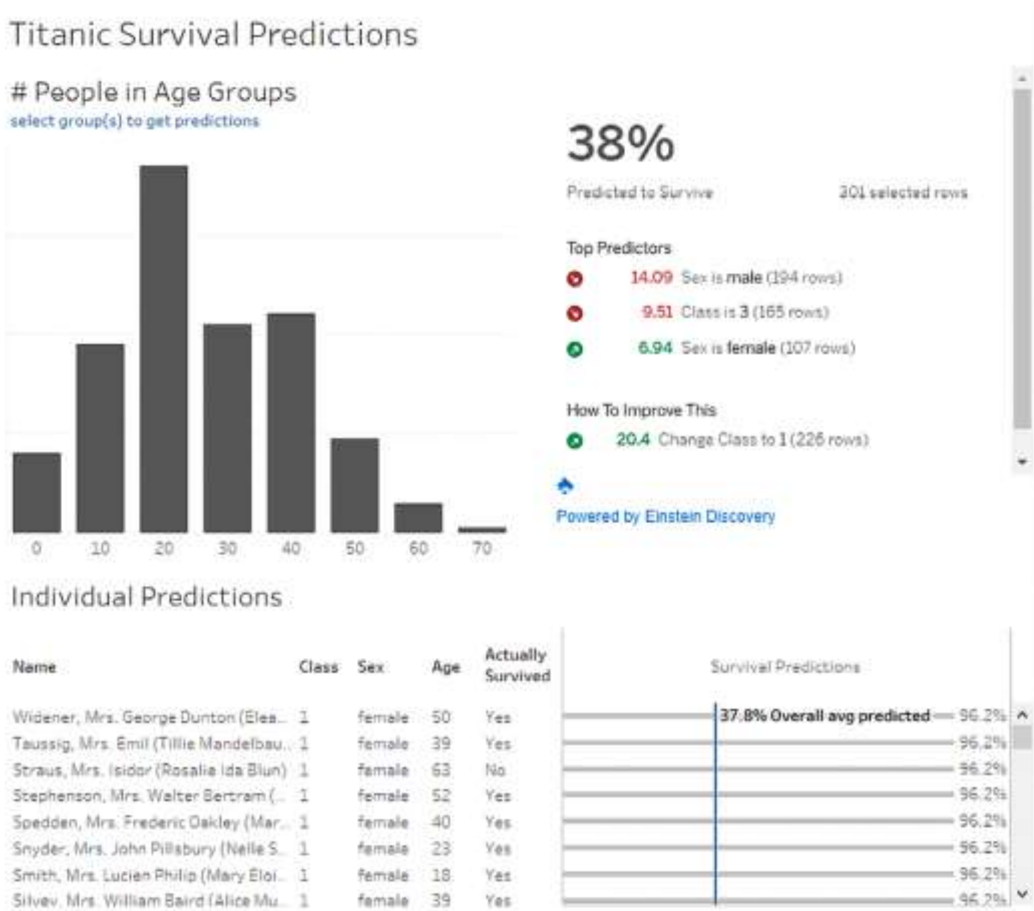


Figure 13.31: Final interactive dashboard that integrates Einstein predictions

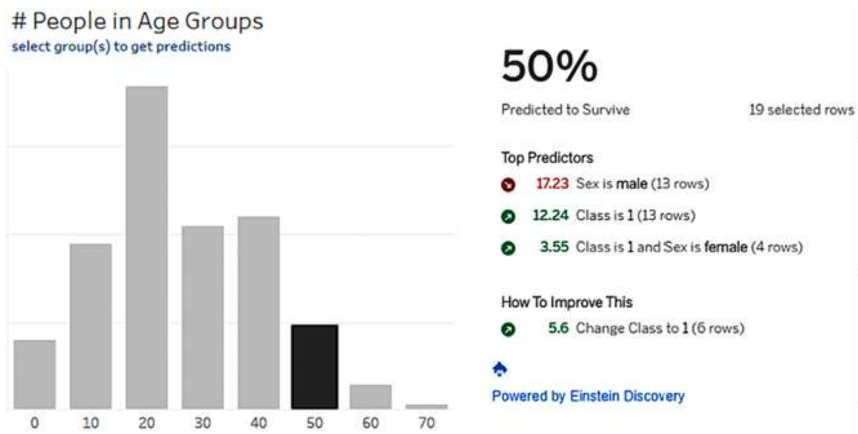


Figure 13.32: Interaction can be used to change the data evaluated by the model

Links

- It is highly recommended that you read through the online documentation completely and it is assumed that you will have installed and configured all the required components following that documentation, which is located here: https://tableau.github.io/extensions-api/docs/trex_getstarted.html.
- You will find documentation on installing and configuring TabPy here: https://tableau.github.io/TabPy/docs/server-install.html.
- You may complete the examples using a limited-time demo account, which you may sign up for here: https://www.salesforce.com/products/crm-analytics/overview/.

URLs

Tags of the Manifest File

First, notice the description and author tags:

```
<description>Show Image Based on Parameters</description>
<author name="Joshua N. Milligan"
email="jmilligan@teknionusa.com" organization="VizPainter"
website="https://www.vizpainter.com"/>
```

Next, consider the source-location tag:

```
<source-location>
<url>http://localhost:8765/Samples/ShowImage/ShowImage.html
</url>
</source-location>
```

Code

Code 13.1 – The HTML file

The most important snippets:

```
<script
src="../../lib/tableau.extensions.1.latest.js"></script>
```

This reference points to the latest version of the Extensions API. Assuming you configured and started the web server based on the location of your Tableau Extensions API download, your reference should not need to be altered.

Next, we have a reference to the JavaScript file we'll use for our code:

```
<script src="./ShowImage.js"></script>
```

This line references the ShowImage.js file that contains all the code we'll need to update the image based on parameter values in the dashboard.

Finally, within the body of the HTML, there is one final tag:

```
<img id="imgImage" src=''/>
```

This defines the image and initializes it with a blank source. We'll update this object in the JavaScript code based on parameter values in the dashboard.

Code 13.2 – The JavaScript file

First, we have a function to update the image based on the parameter values:

```
function updateImage (p) {
  if (p.name=='ImageURL'){
    document.getElementById("imgImage").src =
p.currentValue.formattedValue;
  }
  else if (p.name=='ImageWidth'){
  document.getElementById("imgImage").style.width =
p.currentValue.formattedValue +      "px";
  }
}
```

This function takes in a parameter object, p, and then checks the name of the object. If the parameter's name is ImageURL, then it sets the src attribute of the HTML image to the value of the parameter. If the parameter's name is ImageWidth, then it sets the width style to the parameter value with "px" (pixels) appended.

This example highlights the fact that many extensions will require certain naming conventions to be used in a dashboard. Specifically, this extension requires a parameter named ImageURL to set the URL for the image. The extension will function without a second parameter named ImageWidth, but only a parameter with that exact name can be used to set the image width.

Next, we have a function that is called whenever a parameter value changes:

```
function onParameterChange (parameterChangeEvent) {
  parameterChangeEvent.getParameterAsync().then(function
(param) {
    updateImage(param);
  });
}
```

We'll see next how the the onParameterChange function is set as an event listener. For now, understand that the function is called whenever a value of a parameter changes in the dashboard. When a parameter's value changes, the previously examined updateImage function is called with the changed parameter as an argument.

Finally, notice the document.ready function, which will be called after the HTML document has loaded, giving us a chance to initialize everything we need:

```
$(document).ready(function () {
 tableau.extensions.initializeAsync().then(function () {

tableau.extensions.dashboardContent.dashboard.getParameters
Async().then(function (parameters) {
   parameters.forEach(function (p) {

p.addEventListener(tableau.TableauEventType.ParameterChange
d, onParameterChange);
    updateImage(p);
   });
  });
 });
});
```

When the HTML document is loaded and ready, this block of code will be executed. First, the Tableau extensions initialization is called, after which tableau.extensions.dashboardContent.dashboard.getParametersAsync() gets an array of all parameters in the dashboard. For each of those parameters, the onParameterChange function is added as an event listener and the updateImage function is called, which will update either the image source or the width depending on which parameter is passed in.

All of these functions work together such that when the extension is first loaded in the dashboard and the image will be updated whenever a parameter value changes .

Code 13.3 – Using a Python script in Tableau

```
SCRIPT_STR(
"
import re
exceptions = ['a', 'an', 'the', 'at', 'by', 'for', 'in',
'of', 'on', 'to', 'up', 'and', 'as', 'but', 'or', 'nor',
'with', 'is']
if _arg1 is None:
```

```
 return None
title = _arg1[0]
word_list = re.split(' ', title.lower())
capitalized_title = [word_list[0].capitalize()]
for word in word_list[1:]:
 capitalized_title.append(word if word in exceptions else
word.capitalize())
return ' '.join(capitalized_title)
",
MIN([Title]) )
```

Code 13.4

```
SCRIPT_REAL('{
"modelOrPredictionDefinitionId": "1OR8c000000L1hcGAC",
"columns": ["Class", "Sex"]
}', ATTR([Class]), ATTR([Sex]))
```

Command

The basic commands to set up TabPy are:

To install pip, a package installer for Python, issue this command:

```
python -m pip install --upgrade pip
```

To install TabPy, issue this command:

```
pip install tabpy
```

To run TabPy, issue this command:

```
tabpy
```

Chapter 14

Figures

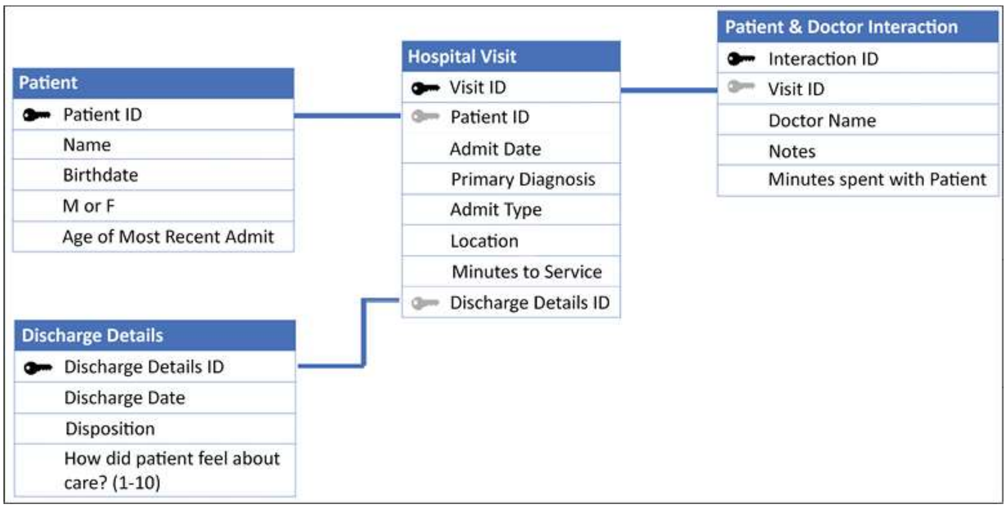


Figure 14.1: The four tabs of the Excel file illustrated as four tables with relationships

Figure 14.2: The Data Source screen lists the tabs in the Excel workbook and invites you to start a data model

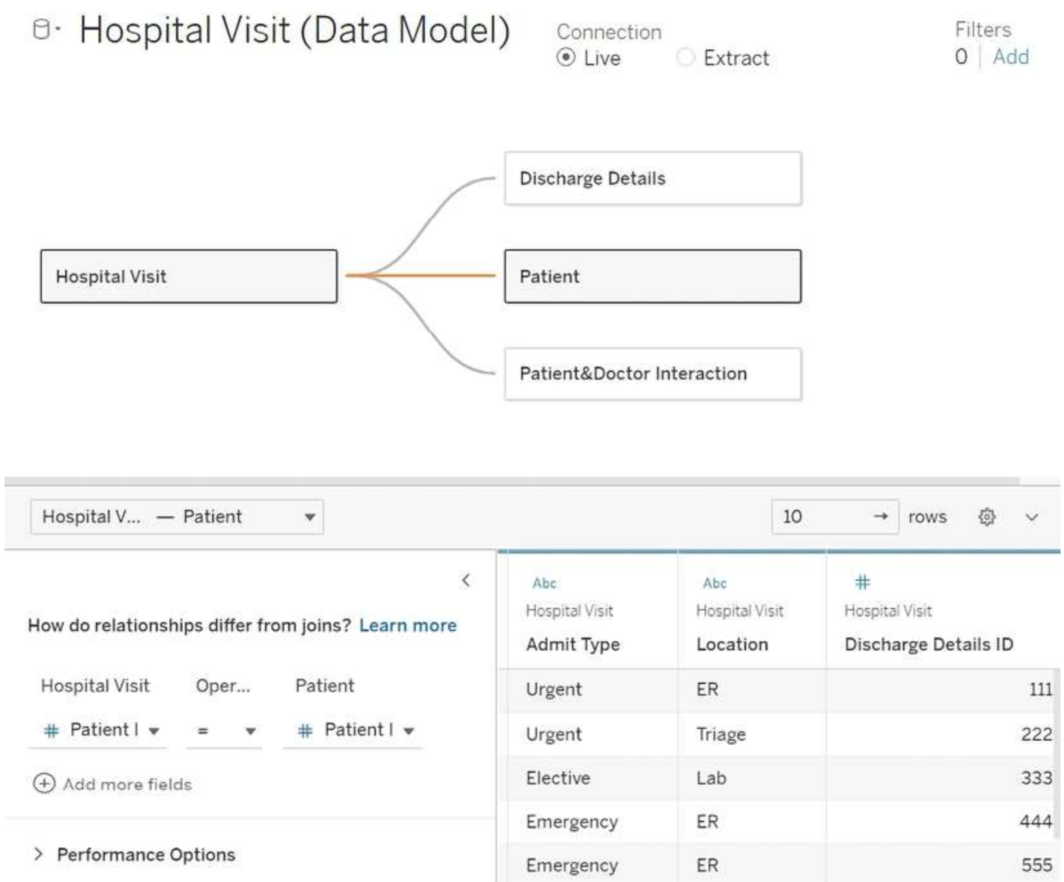


Figure 14.3: All tables have been added to the data model

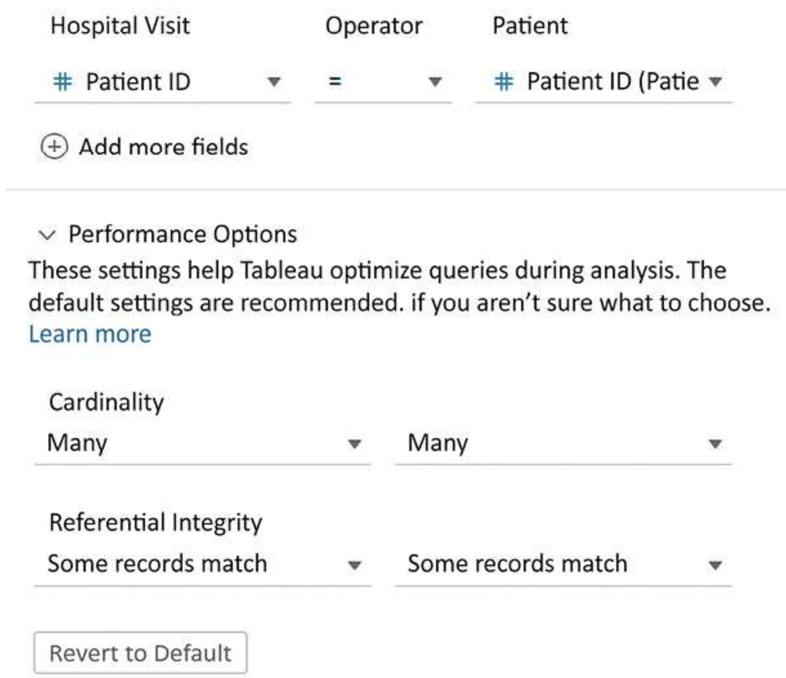


Figure 14.4: The Edit Relationship dialog box includes options to improve performance

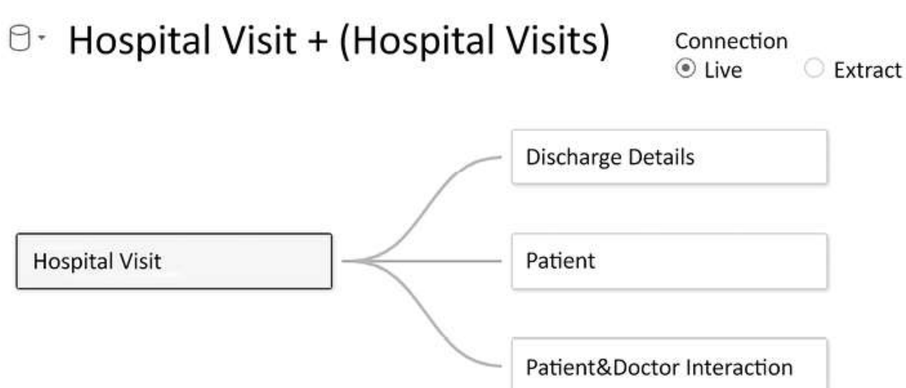


Figure 14.5: The logical layer of the data model

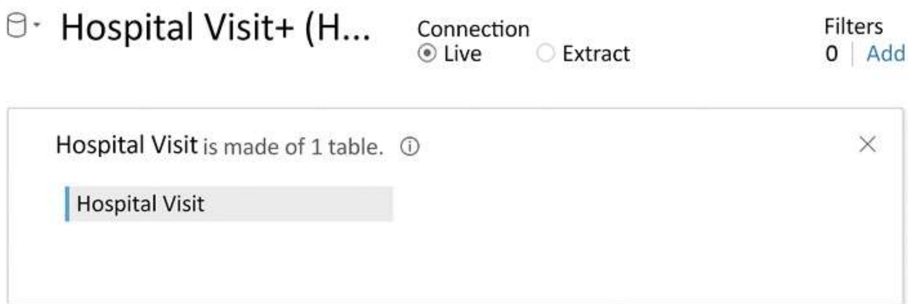


Figure 14.6: The physical layer of the physical tables that make up Hospital Visit

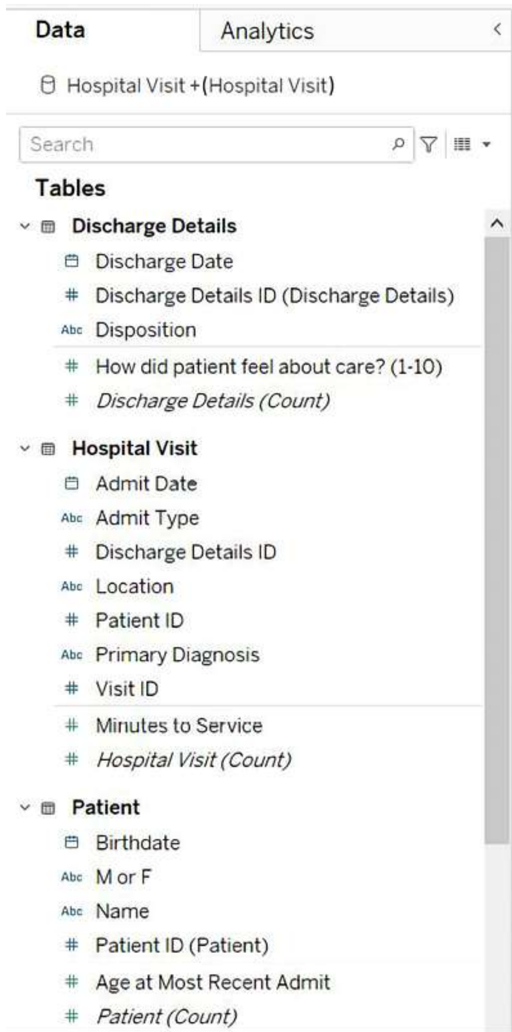


Figure 14.7: The Data pane is organized by logical tables and shows a separation of dimensions and measures per table

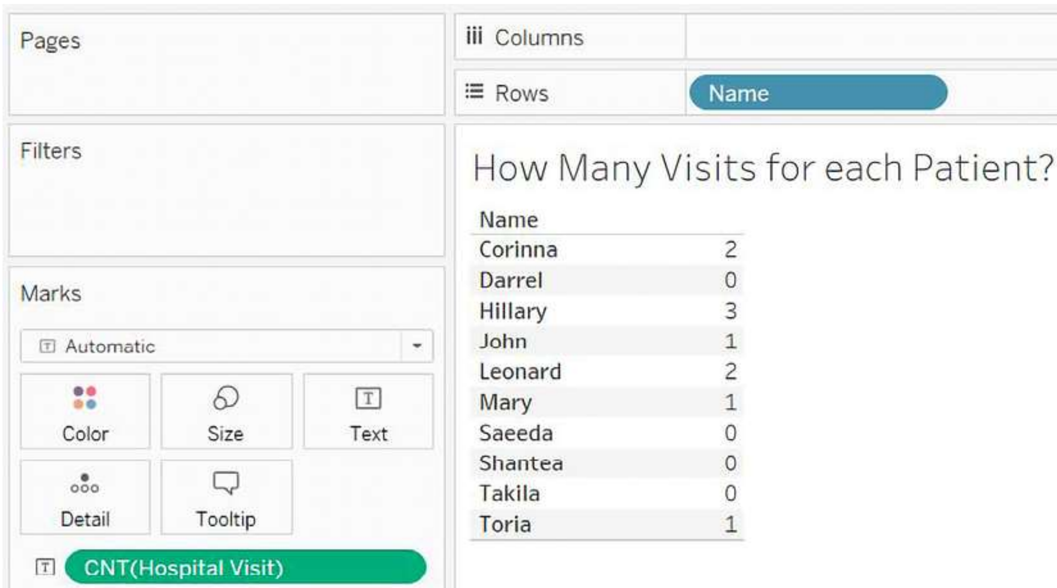


Figure 14.8: All patients are shown, even those with 0 visits

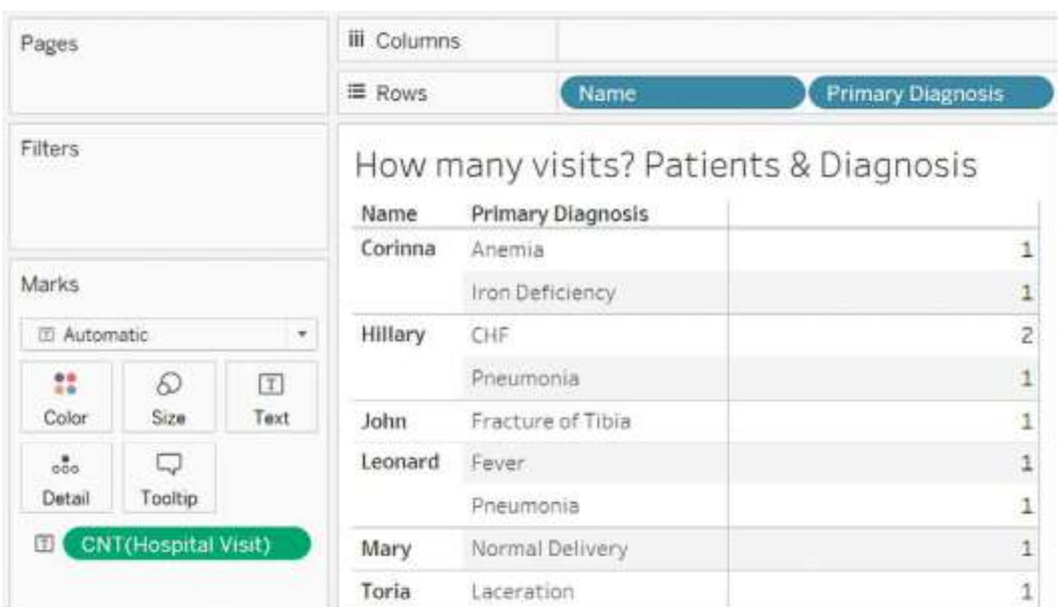


Figure 14.9: Only patients with visits are shown; most patients had a single visit with a given diagnosis, but one came in twice with the same diagnosis

Name	Primary Diagnosis	Age at Most Recent Admit	Count of Hospital Visit
Corinna	Anemia	41.00	1.00
	Iron Deficiency	41.00	1.00
Darrel	Null		0.00
Hillary	CHF	72.00	2.00
	Pneumonia	72.00	1.00
John	Fracture of Tibia	18.00	1.00
Leonard	Fever	22.00	1.00
	Pneumonia	22.00	1.00
Mary	Normal Delivery	30.00	1.00
Saeeda	Null		0.00
Shantea	Null		0.00
Takila	Null		0.00
Toria	Laceration	26.00	1.00

Figure 14.10: All patients are once again shown

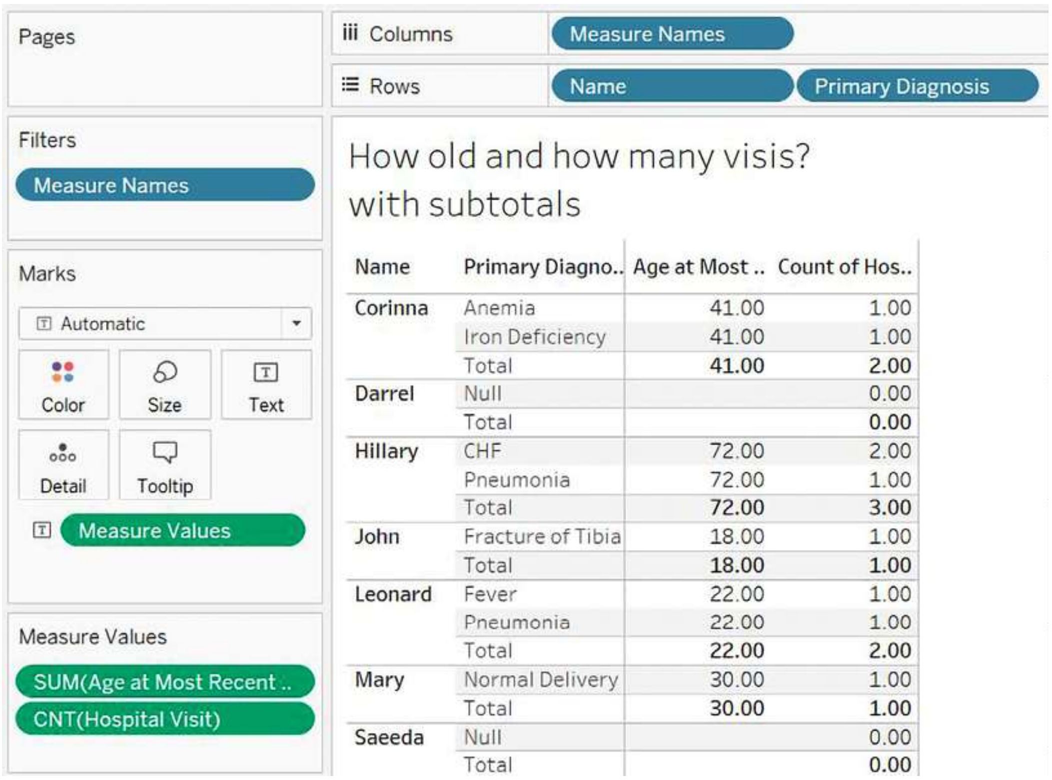


Figure 14.11: Tableau calculates the subtotals correctly, even though traditional join behavior would have duplicated the values

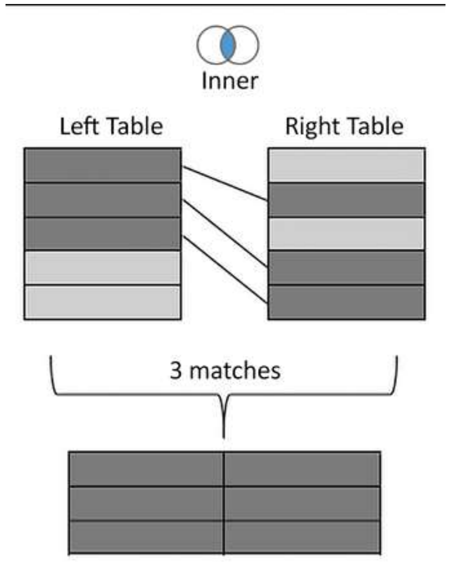


Figure 14.12: Inner join

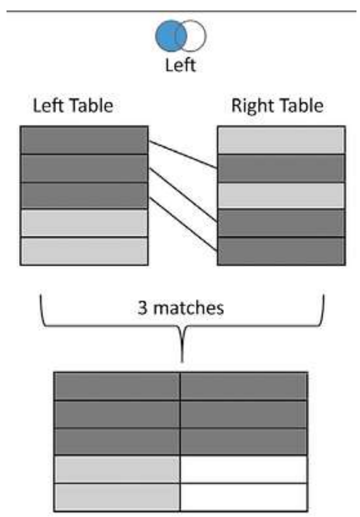


Figure 14.13: Left join

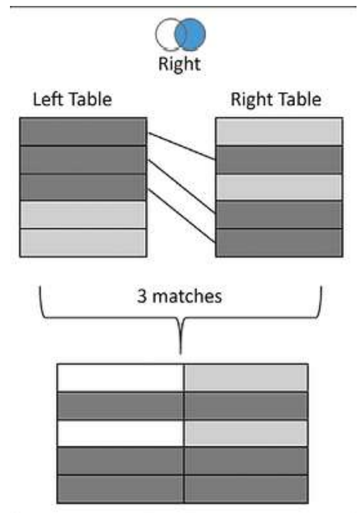


Figure 14.14: Right join

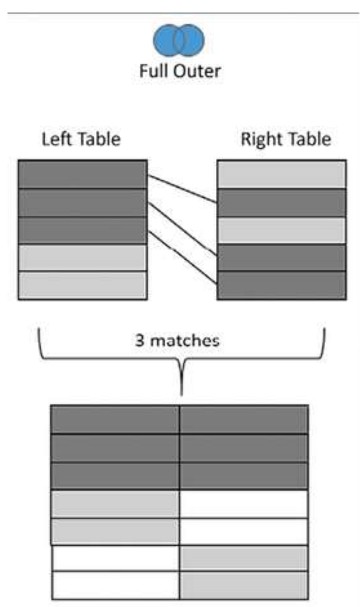


Figure 14.15: Full Outer join

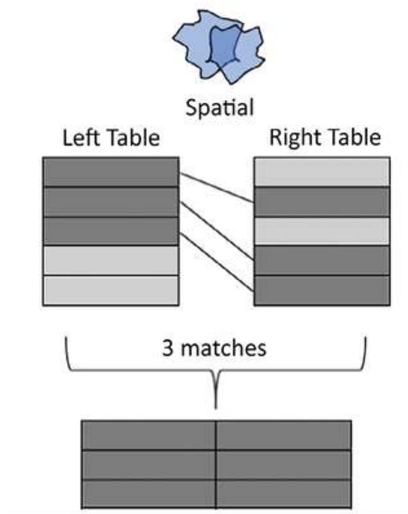


Figure 14.16: Spatial join

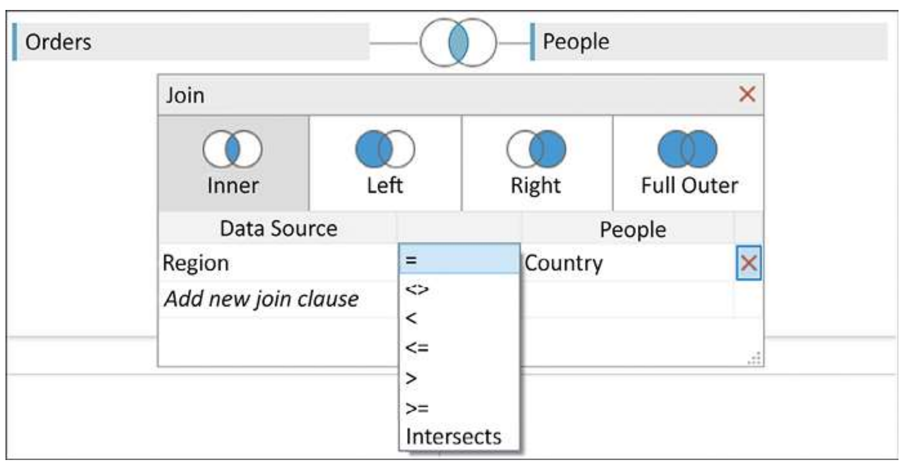


Figure 14.17: Assuming the two fields selected represent spatial objects, the Intersects option will be available

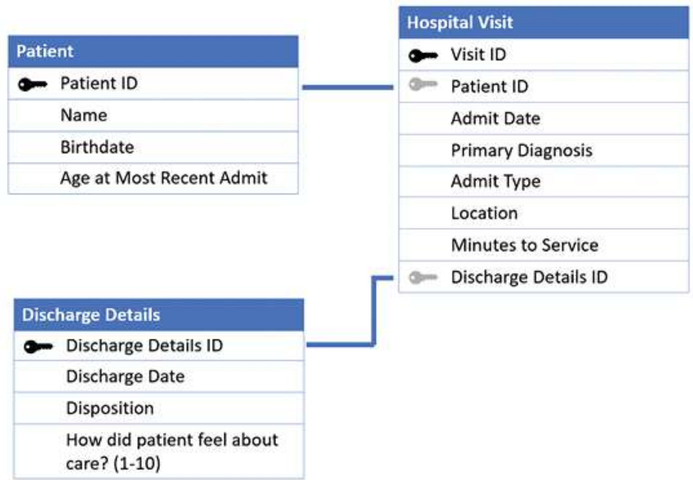


Figure 14.18: The primary Hospital Visit table with Patient and Discharge Details as they might exist in a relational database

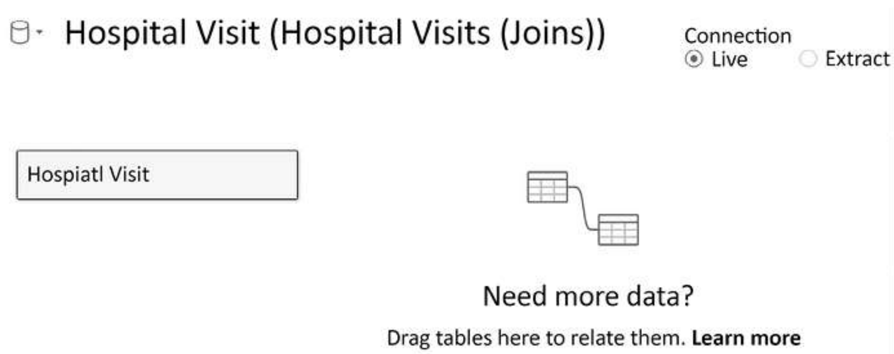


Figure 14.19: After dragging the table onto the canvas, the Hospital Visit object is created in the logical layer

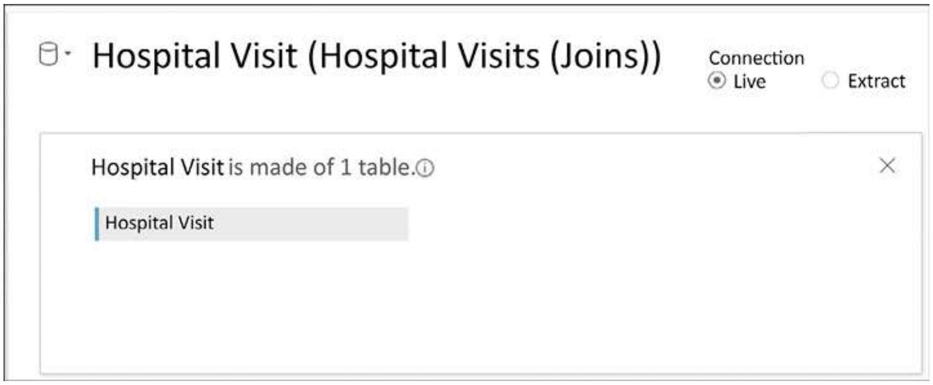


Figure 14.20: The physical layer, which currently consists of a single physical table

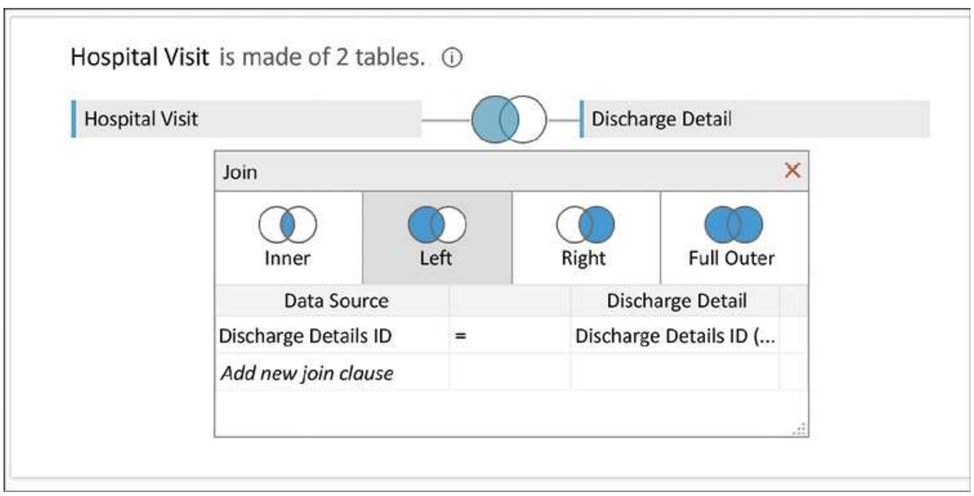


Figure 14.21: Joining Discharge Detail to Hospital Visit in the physical layer

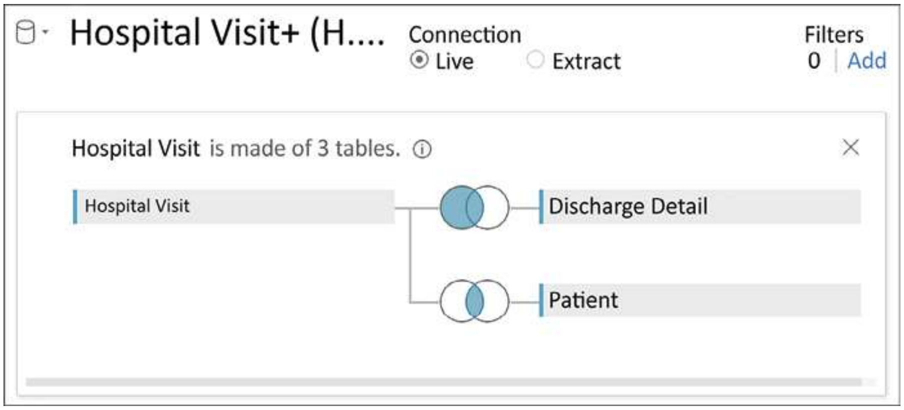


Figure 14.22: The physical layer is made up of three tables joined together

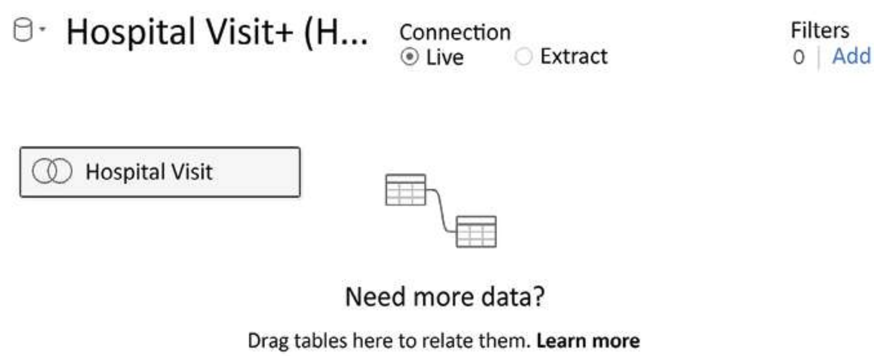


Figure 14.23: The logical layer contains a single object that is made up of three physical tables

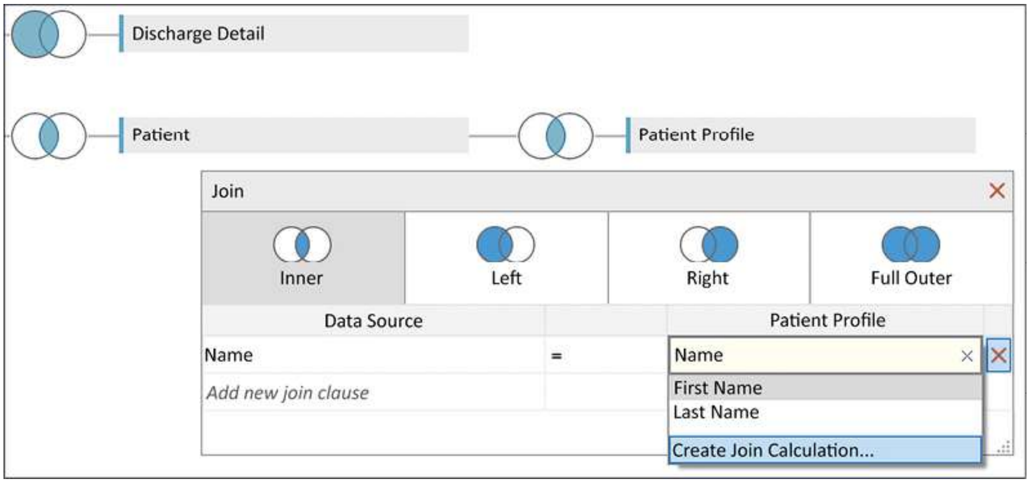


Figure 14.24: You can create a join calculation to aid in forming the correct joins

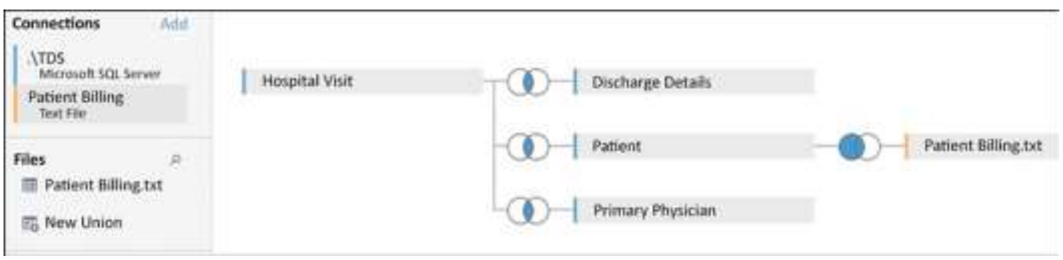


Figure 14.25: Joining tables or files based on separate data connections

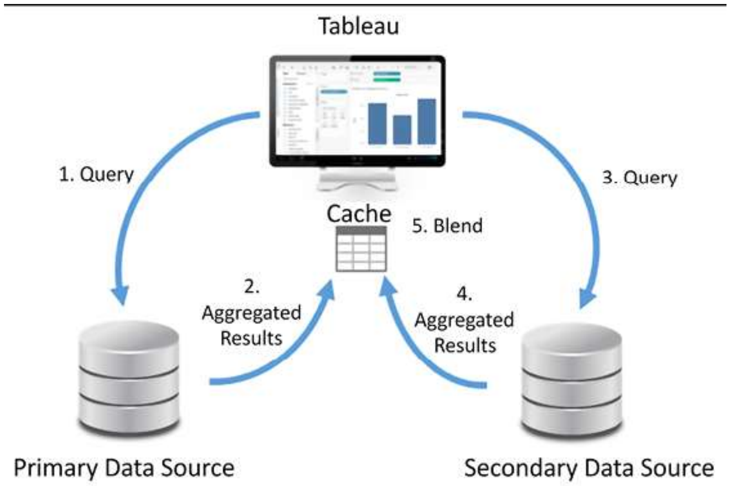


Figure 14.26: How Tableau accomplishes blending

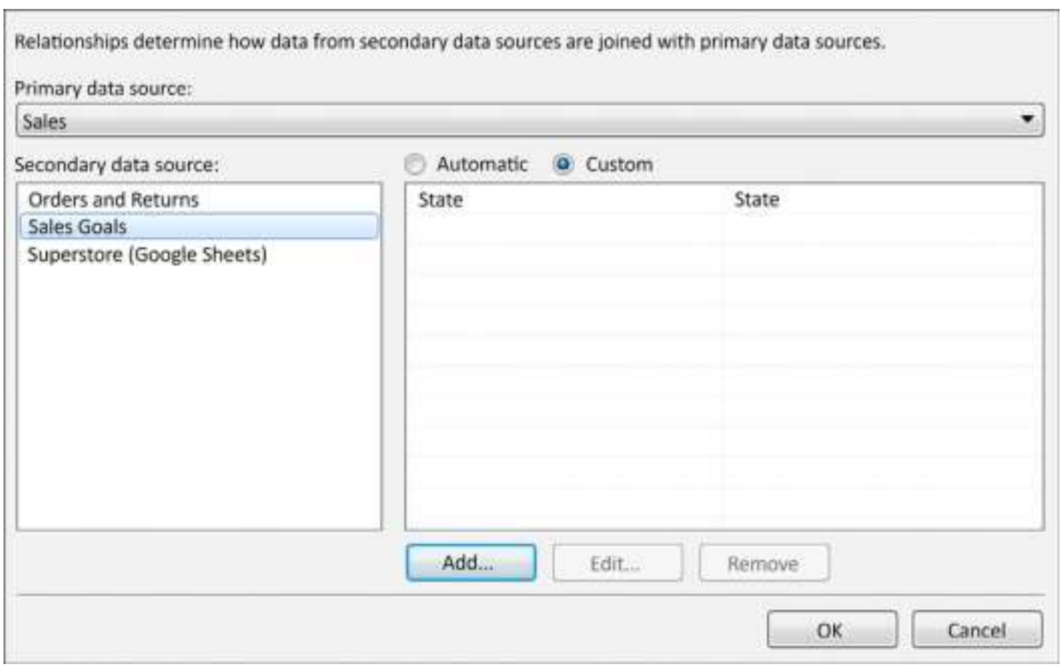


Figure 14.27: Defining blending relationships between data sources

Figure 14.28: Average Minutes to Service by Location

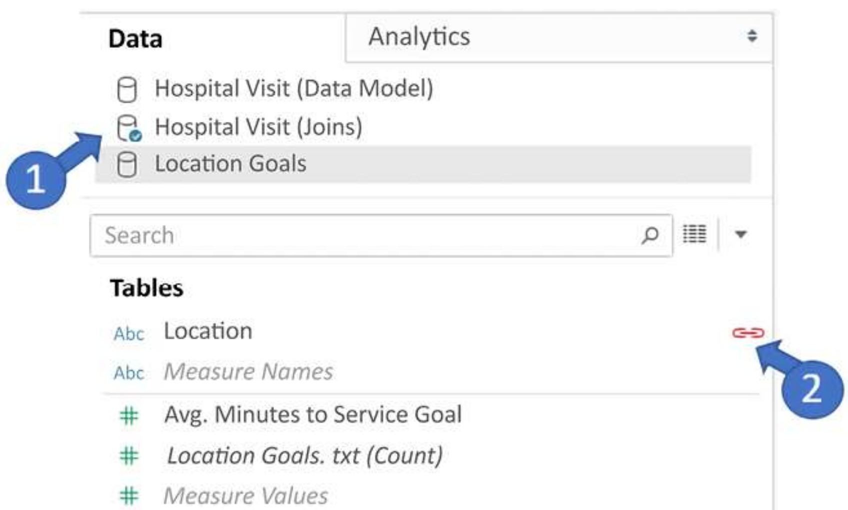


Figure 14.29: Hospital Visit (Joins) is shown as the Primary data source and Location in the Location Goals data source is indicated as a linking field

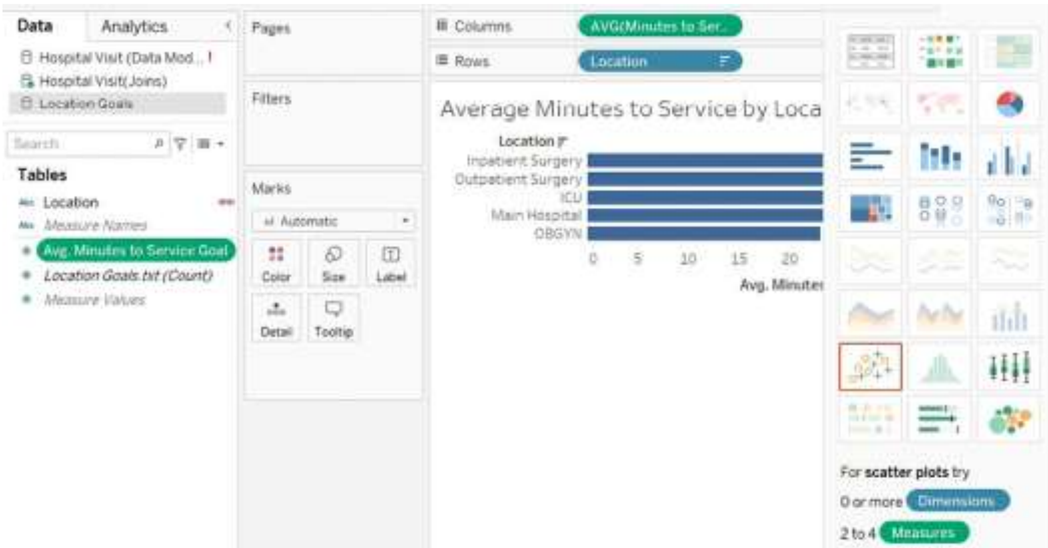


Figure 14.30: You may drag and drop fields from secondary sources into the view or use Show Me

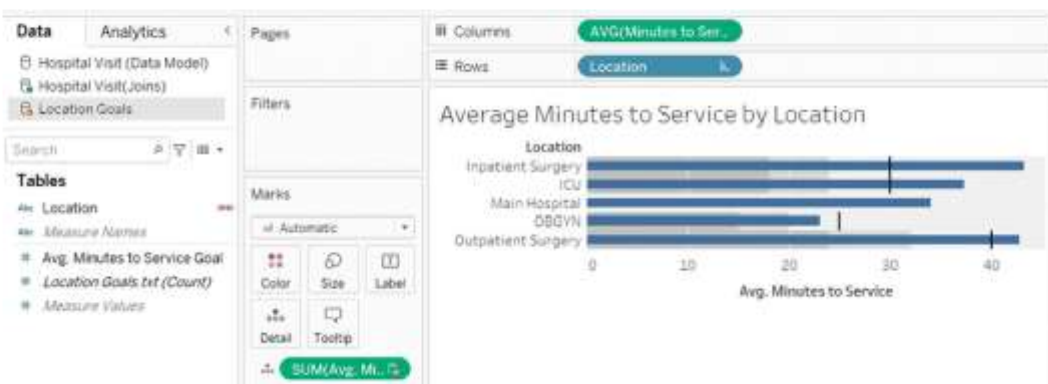


Figure 14.31: A view created from a primary source and a secondary source

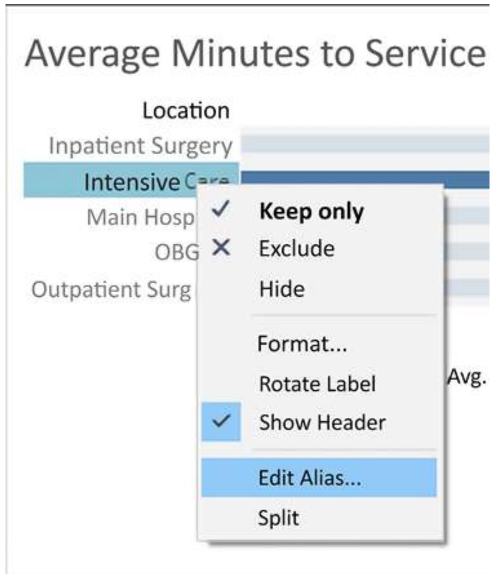


Figure 14.32: Using the Edit Alias... option

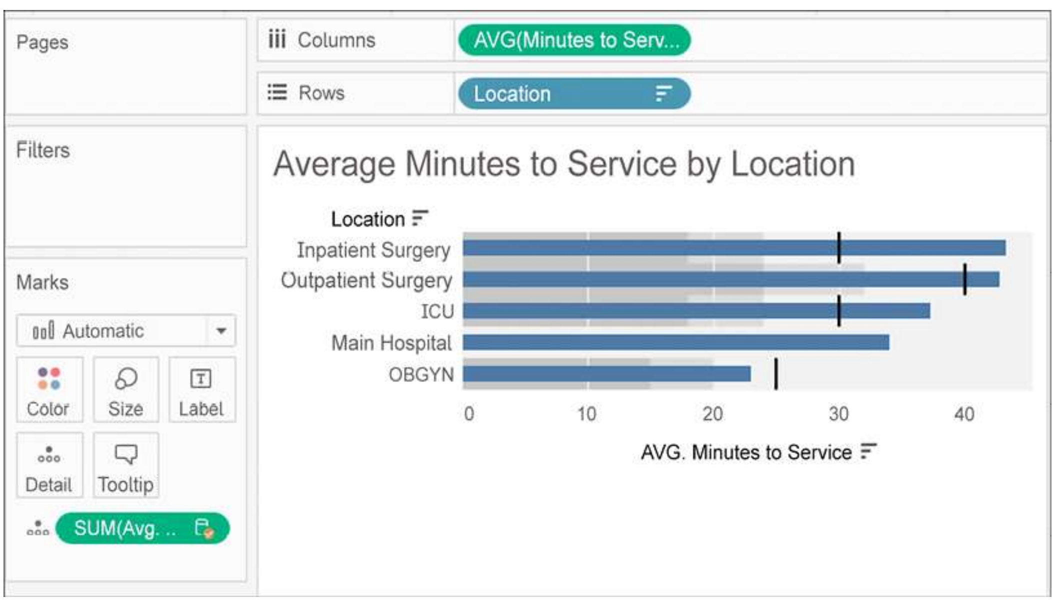


Figure 14.33: ICU now finds a match in the secondary source

Code

Code 14.1

```
MAKEPOINT([Latitude], [Longitude])
```

Tables

Visit ID	Patient Name	Doctor ID
1	Kirk	1
2	Picard	2
3	Sisko	3

Table 14.1 – Visit table

Doctor ID	Doctor Name
1	McCoy
2	Crusher
3	Bashir
2	Pulaski

Table 14.2 – Doctor table

Visit ID	Patient Name	Doctor ID	Doctor Name
1	Kirk	1	McCoy
2	Picard	2	Crusher
3	Sisko	3	Bashir
2	Picard	2	Pulaski

Table 14.3 – Resultant table

Location	Avg. Minutes to Service Goal
Inpatient Surgery	30
Outpatient Surgery	40
ICU	30
OBGYN	25
Lab	120

Table 14.4

Chapter 15

Tables

School	Classroom	Average GPA	Number of Students	Number of Students (School)
Pickaway Elementary	4th Grade	3.78	153	1,038
Pickaway Elementary	5th Grade	3.73	227	1,038
Pickaway Elementary	6th Grade	3.84	227	1,038
McCord Elementary	4th Grade	3.82	94	915
McCord Elementary	5th Grade	3.77	89	915
McCord Elementary	6th Grade	3.84	122	915

Table 15.1

Country Name	1960	1961	1962	1963	1964
Afghanistan	8,774,440	8,953,544	9,141,783	9,339,507	9,547,131
Australia	10,276,477	10,483,000	10,742,000	10,950,000	11,167,000

Table 15.2

Country Name	Year	Population
Afghanistan	1960	8,774,440
Afghanistan	1961	8,953,544
Afghanistan	1962	9,141,783
Afghanistan	1963	9,339,507
Afghanistan	1964	9,547,131
Australia	1960	10,276,477
Australia	1961	10,483,000

Australia	1962	10,742,000
Australia	1963	10,950,000
Australia	1964	11,167,000

Table 15.3

Patient ID	Patient Name	Admit Date	Discharge Date
1	David	12/1/2018	12/20/2018
2	Solomon	12/3/2018	12/7/2018
3	Asa	12/5/2018	12/22/2018
4	Jehoshaphat	12/5/2018	12/6/2018
5	Joash	12/9/2018	12/16/2018
6	Amaziah	12/10/2018	12/14/2018
7	Uzziah	12/12/2018	12/24/2018
8	Jotham	12/16/2018	12/29/2018
9	Hezekiah	12/18/2018	12/22/2018
10	Josiah	12/22/2018	12/23/2018

Table 15.4

Figures

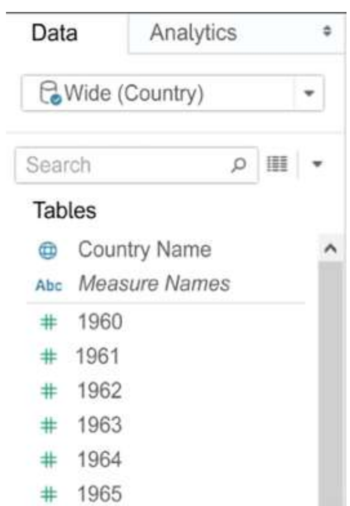


Figure 15.1: The wide data has a measure for every year

Figure 15.2: The wide data can still be used but in a complex and limited way

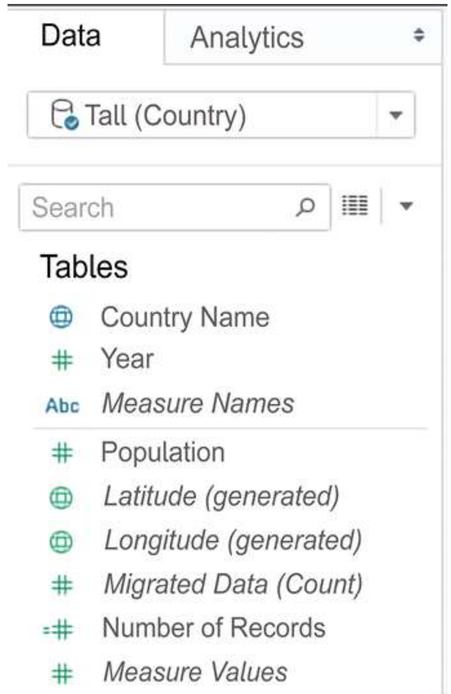


Figure 15.3: The tall data has a Year dimension and a single Population measure

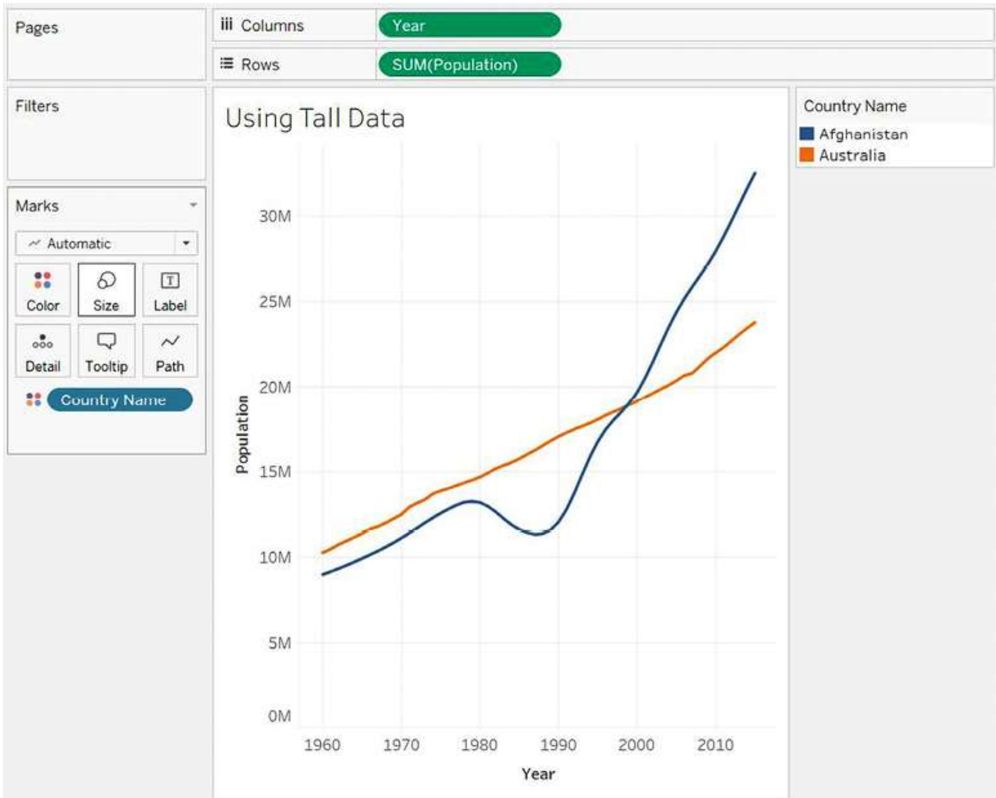


Figure 15.4: The view is much easier to create in Tableau with the tall data

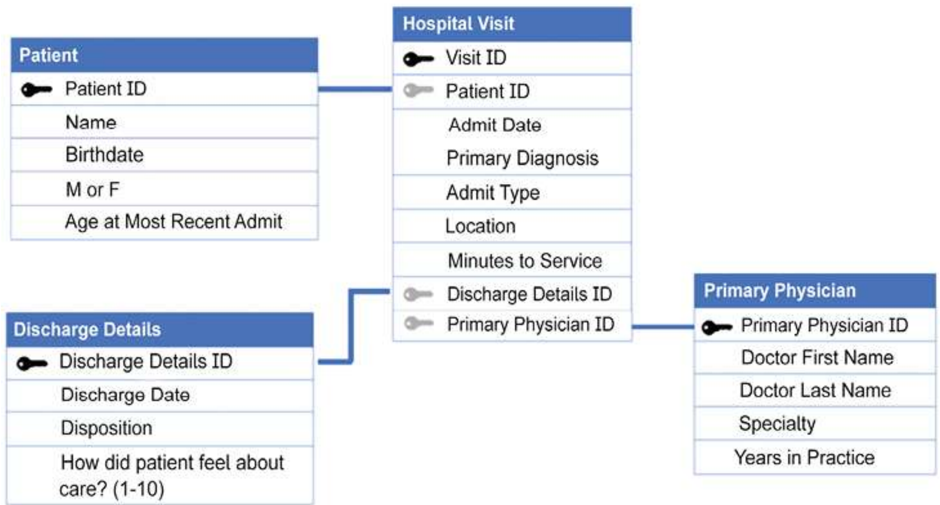


Figure 15.5: A simple star schema

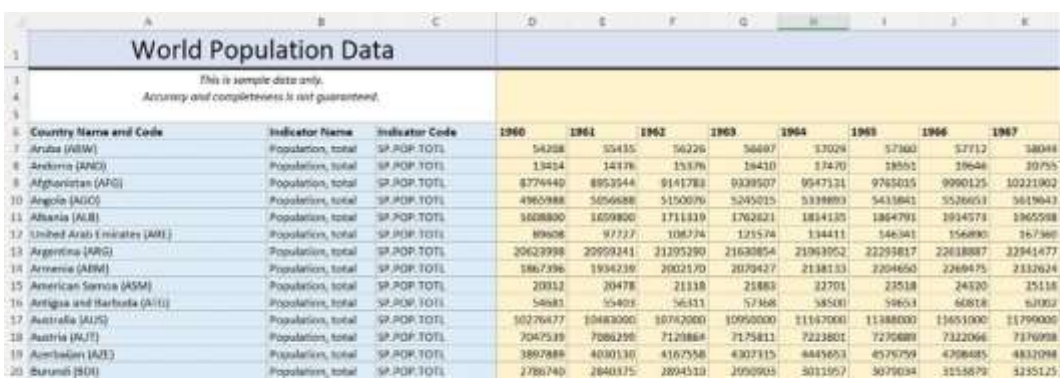


Figure 15.6: The World Population Data Excel file

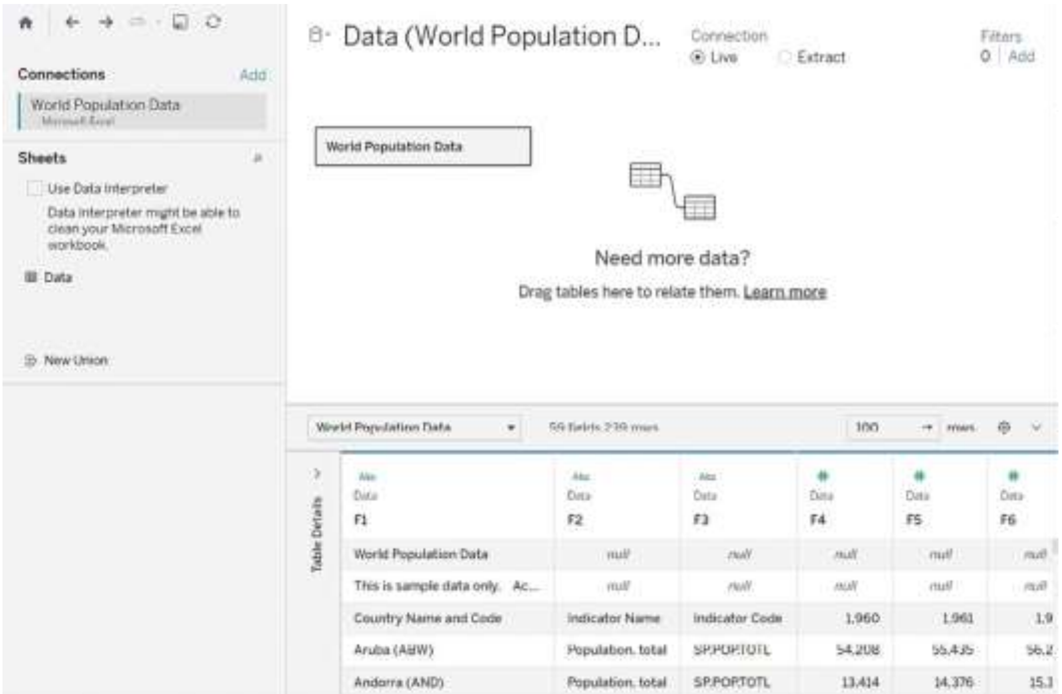


Figure 15.7: World Population Data.xlsx on Tableau's Data Source page

Country Name and Code	Indicator Name	Indicator Code	1960	1961	1962	1963	1964	1965
Aruba (ABW)	Population, total	SP.POP.TOTL	54,208	55,435	56,226	56,697	57,029	57,360
Andorra (AND)	Population, total	SP.POP.TOTL	13,414	14,376	15,376	16,410	17,470	18,551
Afghanistan (AFG)	Population, total	SP.POP.TOTL	8,774,440	8,953,544	9,141,783	9,339,507	9,547,131	9,765,015
Angola (AGO)	Population, total	SP.POP.TOTL	4,965,988	5,056,688	5,150,076	5,245,015	5,339,893	5,433,841
Albania (ALB)	Population, total	SP.POP.TOTL	1,608,800	1,659,800	1,711,319	1,762,621	1,814,135	1,864,791
United Arab Emirates (ARE)	Population, total	SP.POP.TOTL	89,608	97,727	108,774	121,574	134,411	146,341

Figure 15.8: Tableau Data Interpreter fixes many of the common issues found in Excel (and similar) data sources

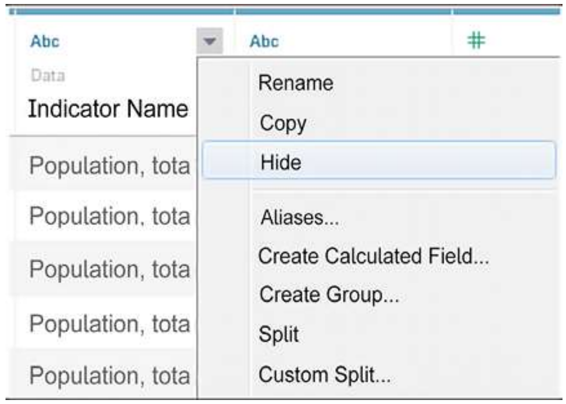


Figure 15.9: You can hide fields from the Data Source screen

Calculation Country Name	Calculation Country Code	Data 1960	Data 1961	Data 1962	Data 1963	Data 1964	Data 1965	Data 1966
Aruba	ABW	54,208	55,435	56,226	56,697	57,029	57,360	57,712
Andorra	AND	13,414	14,376	15,376	16,410	17,470	18,551	19,646
Afghanistan	AFG	8,774,440	8,953,544	9,141,783	9,339,507	9,547,131	9,765,015	9,990,125
Angola	AGO	4,965,988	5,056,688	5,150,076	5,245,015	5,339,893	5,433,841	5,526,653
Albania	ALB	1,608,800	1,659,800	1,711,319	1,762,621	1,814,135	1,864,791	1,914,573
United Arab Emir...	ARE	89,608	97,727	108,774	121,574	134,411	146,341	156,890

Figure 15.10: After some cleaning, the data is still in an undesirable wide structure

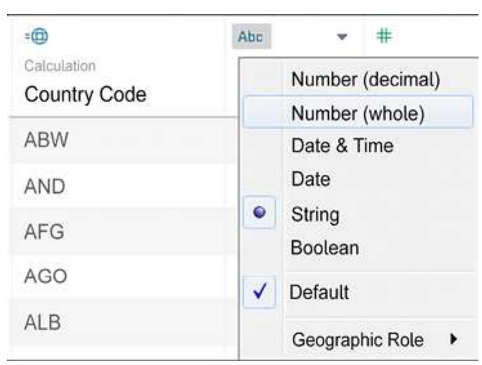


Figure 15.11: You can change the data types of the fields on the Data Source page

Calculation	Calculation	Pivot	Pivot
Country Name	Country Code	Year	Population
Aruba	ABW	1961	55,435
Andorra	AND	1961	14,376
Afghanistan	AFG	1961	8,953,544
Angola	AGO	1961	5,056,688
Albania	ALB	1961	1,659,800
United Arab Emirates	ARE	1961	97,727
Argentina	ARG	1961	20,959,241
Armenia	ARM	1961	1,934,239
American Samoa	ASM	1961	20,478
Antigua and Barbuda	ATG	1961	55,403
Australia	AUS	1961	10,483,000

Figure 15.12: The cleaned and pivoted dataset

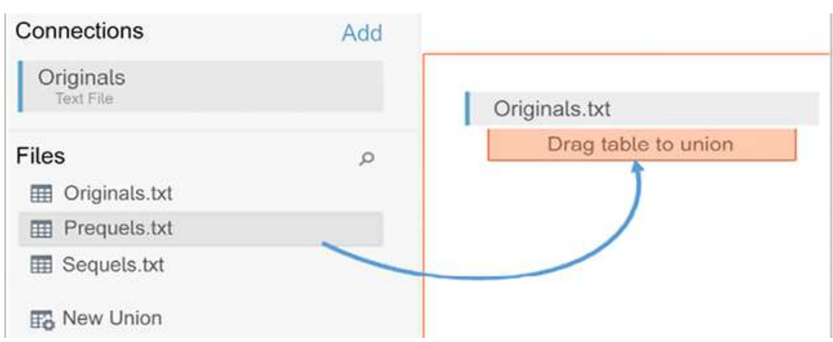


Figure 15.13: You may create unions by dragging and dropping tables or files directly under existing tables on the canvas

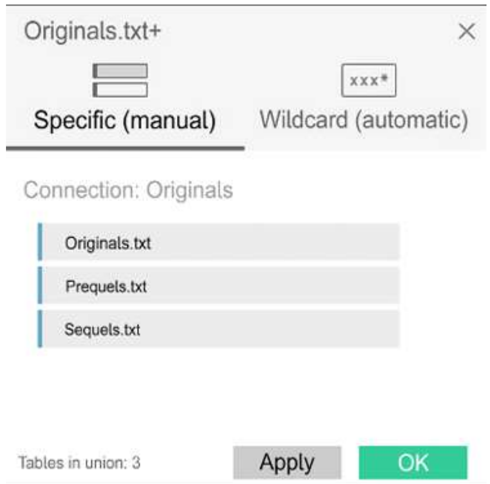


Figure 15.14: You may edit unions with these options

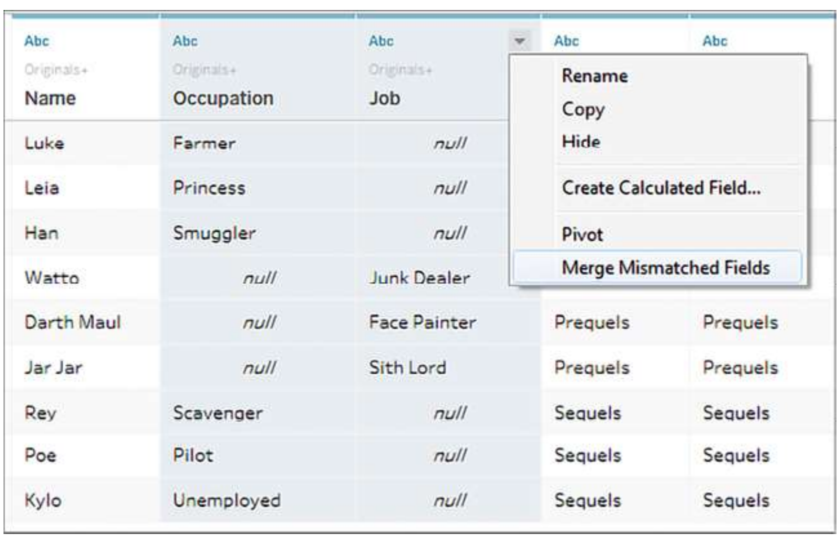
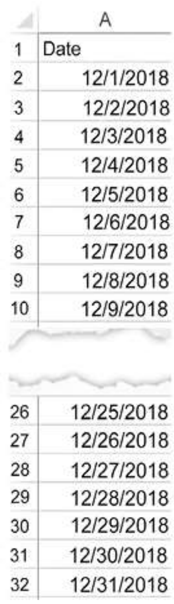


Figure 15.15: Use Merge Mismatched Fields to combine columns resulting from a union where the field names didn't match (this mismatch is not included in the example data)



	A
1	Date
2	12/1/2018
3	12/2/2018
4	12/3/2018
5	12/4/2018
6	12/5/2018
7	12/6/2018
8	12/7/2018
9	12/8/2018
10	12/9/2018
26	12/25/2018
27	12/26/2018
28	12/27/2018
29	12/28/2018
30	12/29/2018
31	12/30/2018
32	12/31/2018

Figure 15.16: An Excel file containing only a comprehensive list of dates

Figure 15.17: As we've seen, the join is created in the physical layer of the data model

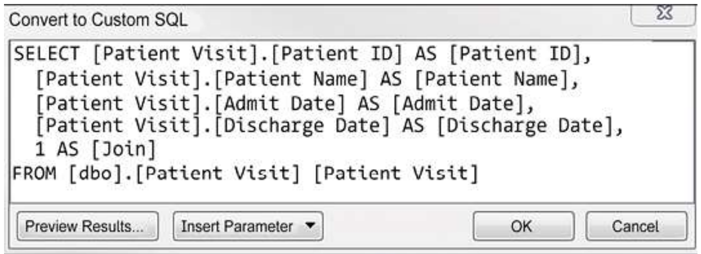


Figure 15.18: A sample script that could be used to create a value on which to join

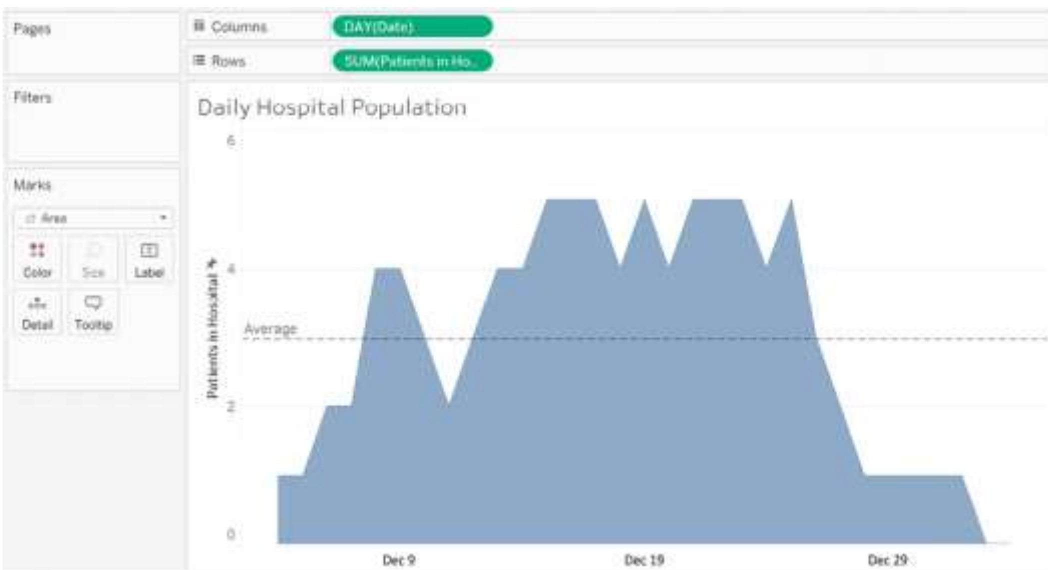


Figure 15.19: A visualization of the daily hospital population, made easy with some data restructuring

Apartment	Month	Rent Collected	Square Feet
A	Jan	$0	900
	Feb	$0	900
	Mar	$0	900
	Apr	$0	900
	May	$0	900
	Jun	$1,500	900
	Jul	$1,500	900
	Aug	$1,500	900
	Sep	$1,500	900
	Oct	$1,500	900
	Nov	$1,500	900
	Dec	$1,500	900
B	Jan	$1,200	750
	Feb	$1,200	750
	Mar	$1,200	750
	Apr	$1,200	750
	May	$1,200	750
	Jun	$1,200	750
	Jul	$0	750
	Aug	$0	750
	Sep	$0	750
	Oct	$0	750
	Nov	$0	750
	Dec	$0	750

Figure 15.20: The Apartment Rent data, which is poorly structured because the Square Feet measure is repeated for every month

Figure 15.21: An illustration of how various aggregations might be right or wrong depending on the level of detail

Apartment	Rent Collected	Sum of Square Feet	Avg. Square Feet	Min. Square Feet	Max. Square Feet
A	$10,500	10,800	900	900	900
B	$7,200	9,800	750	750	750
Grand Total	$17,700	19,800	825	750	900

Figure 15.22: None of the aggregations work to give us a grand total

Apartment	Rent Collected	Square Feet per Apartment
A	$10,500	900
B	$7,200	750
Grand Total	$17,700	1,650

Figure 15.23: An LOD calculation gives us the correct result at all levels of detail

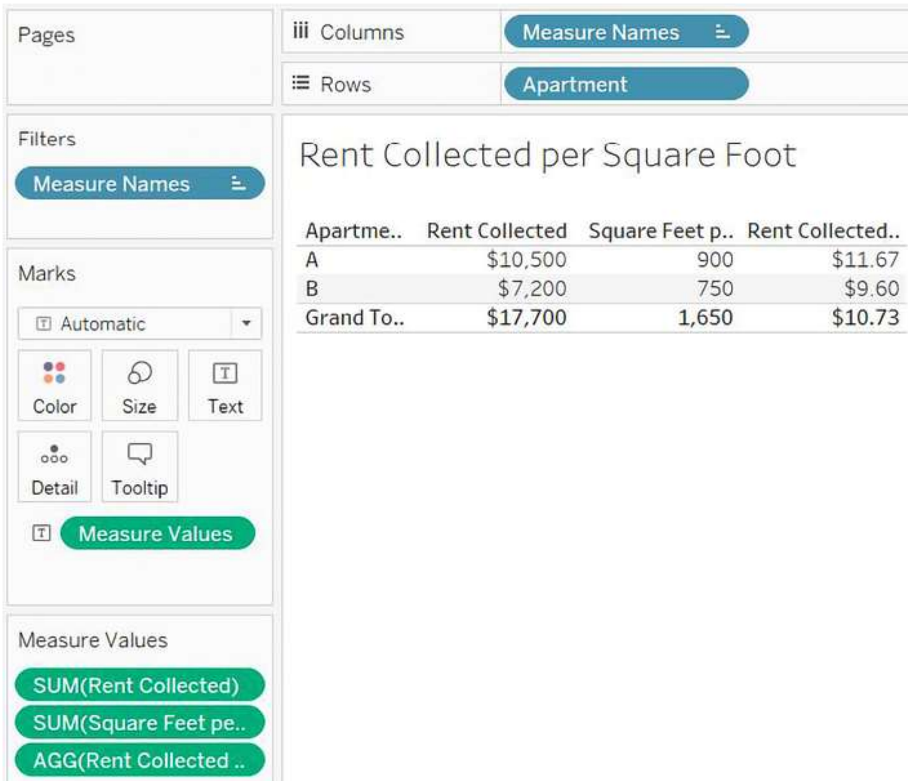


Figure 15.24: The LOD expression gives us the foundation for even more complexity, such as calculating the rent per area

Example of Unions

Table 15.1 – Originals:

Name	Occupation	Bank account balance
Luke	Farmer	$2,000
Leia	Princess	$50,000
Han	Smuggler	-$20,000

Table 15.1 – Prequels:

Name	Occupation	Bank account balance
Watto	Junk Dealer	$9,000
Darth Maul	Face Painter	$10,000
Jar Jar	Sith Lord	-$100,000

Table 15.1 – Sequels:

Name	Occupation	Bank account balance
Rey	Scavenger	$600
Poe	Pilot	$30,000
Kylo	Unemployed	$0

Table 15.1 – A union of these tables would give a single table containing the rows of each individual table:

Name	Occupation	Bank account balance
Luke	Farmer	$2,000
Leia	Princess	$50,000
Han	Smuggler	-$20,000
Watto	Junk Dealer	$9,000
Darth Maul	Face Painter	$10,000
Jar Jar	Sith Lord	-$100,000
Rey	Scavenger	$600
Poe	Pilot	$30,000
Kylo	Unemployed	$0

Code

Code 15.1 – Obtain the date using a calculated field with code such as the following:

```
DATEPARSE('yyyy-MM', [Table Name] )
```

Code 15.2

```
IF [Admit Date] <= [Date] AND [Discharge Date] >= [Date]
THEN 1
```

```
ELSE 0
END
```

Code 15.3

```
{ INCLUDE [Apartment] : MIN([Square Feet]) }
```

Code 15.4

```
SUM([Rent Collected])/SUM([Square Feet per Apartment])
```

Code 15.5

```
SELECT [Country Name],[1960] AS Population, 1960 AS Year
FROM Countries

UNION ALL

SELECT [Country Name],[1961] AS Population, 1961 AS Year
FROM Countries

UNION ALL

SELECT [Country Name],[1962] AS Population, 1962 AS Year
FROM Countries
...
...
```

Chapter 16

Figures

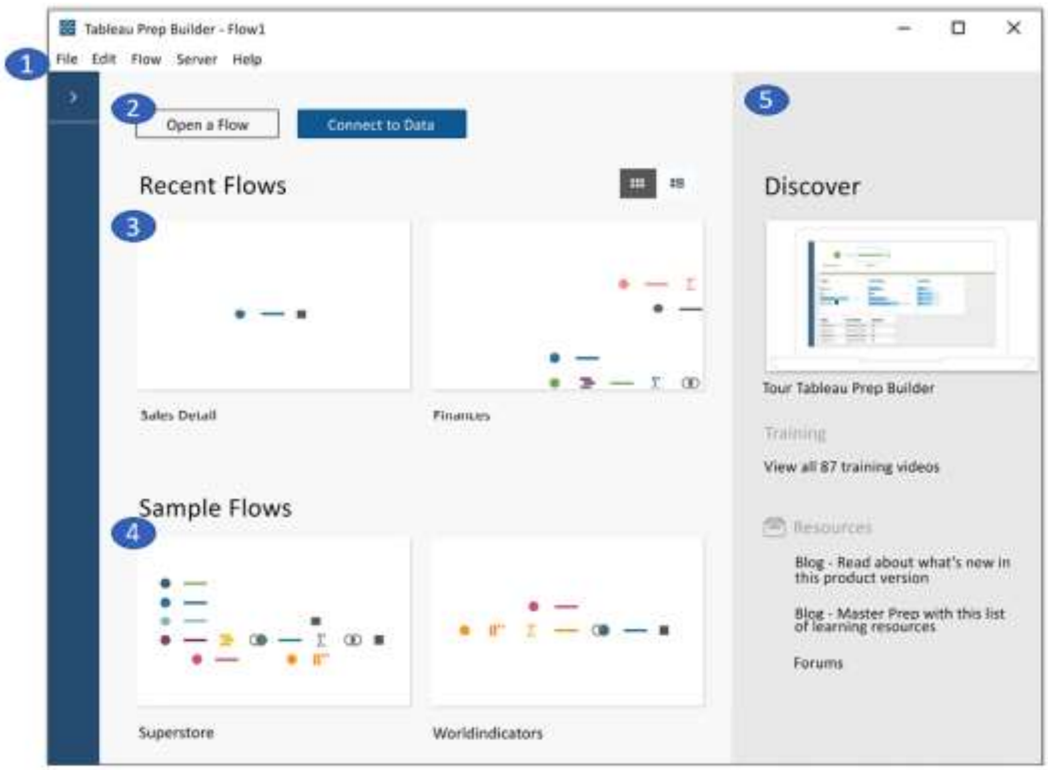


Figure 16.1: The Tableau Prep Builder welcome screen with numbering to identify key components of the UI

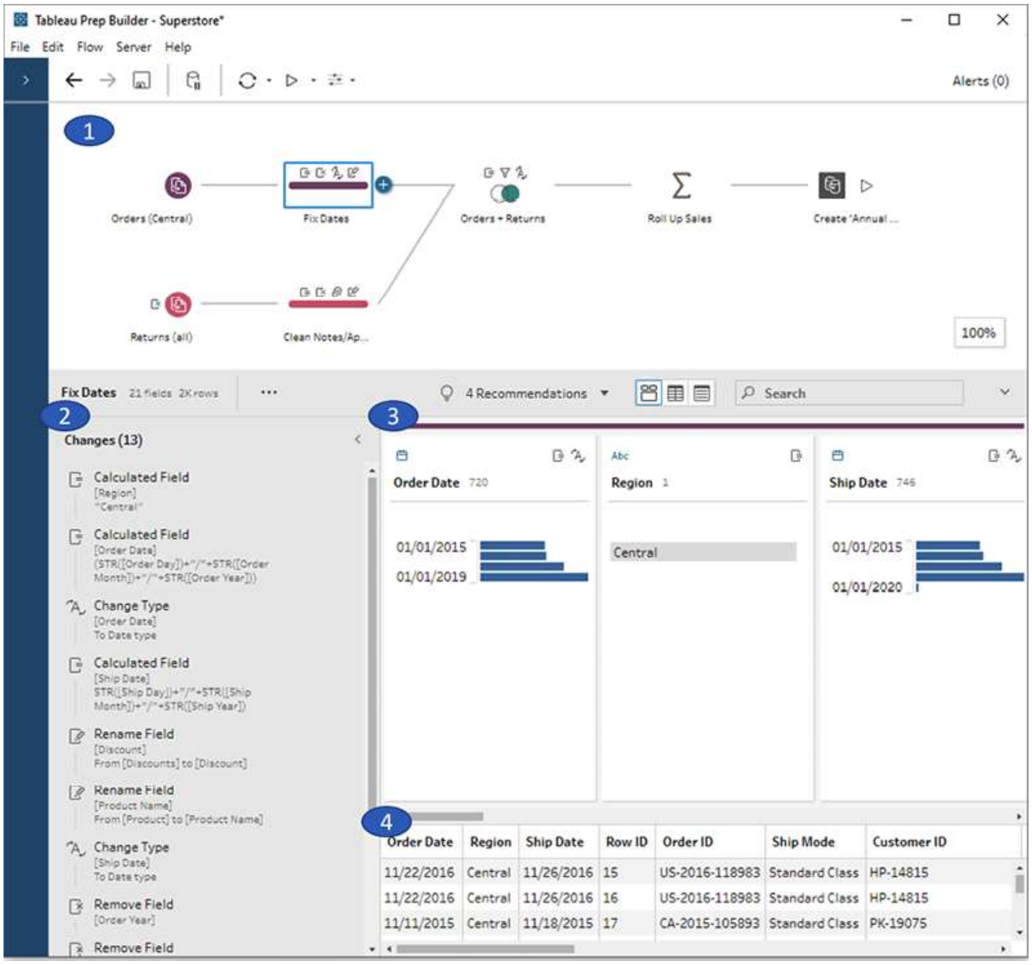


Figure 16.2: When designing a flow, you'll find an interface like this one. The major components are numbered and described as follows

Figure 16.3: An example flow in Tableau Prep

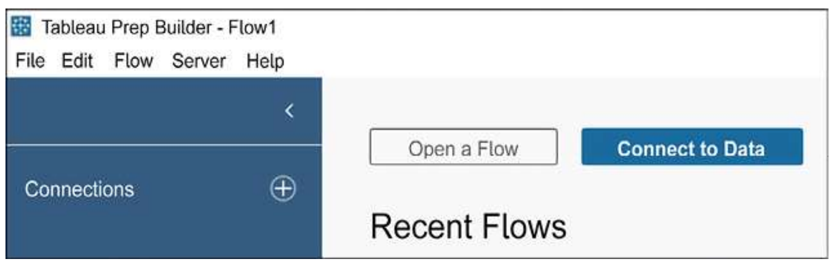
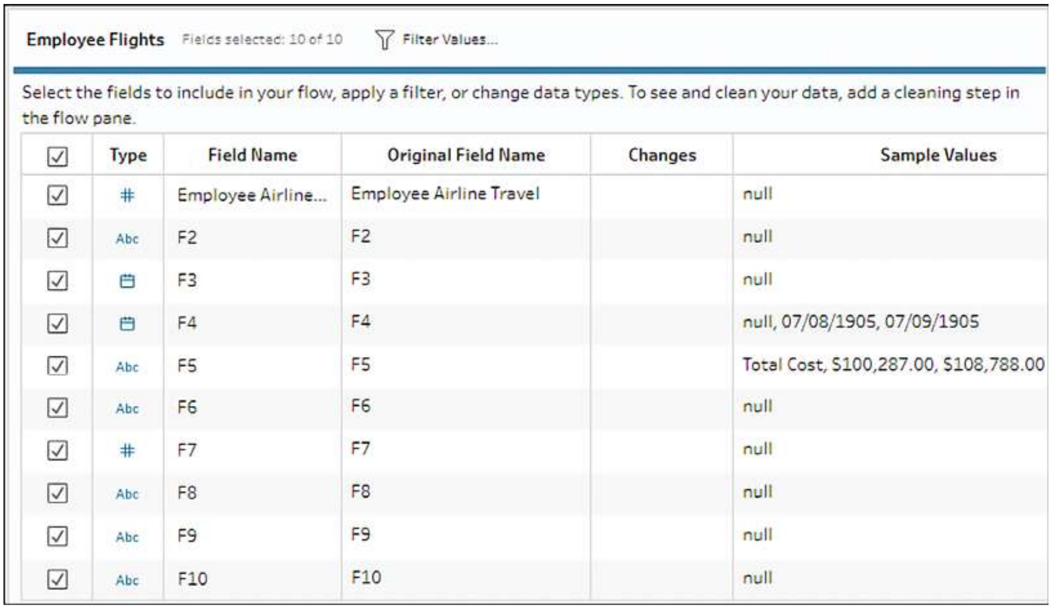


Figure 16.4: You can make a new data connection by clicking the + button or the Connect to Data button



Employee Flights Fields selected: 10 of 10 Filter Values...

Select the fields to include in your flow, apply a filter, or change data types. To see and clean your data, add a cleaning step in the flow pane.

☑	Type	Field Name	Original Field Name	Changes	Sample Values
☑	#	Employee Airline...	Employee Airline Travel		null
☑	Abc	F2	F2		null
☑		F3	F3		null
☑		F4	F4		null, 07/08/1905, 07/09/1905
☑	Abc	F5	F5		Total Cost, $100,287.00, $108,788.00
☑	Abc	F6	F6		null
☑	#	F7	F7		null
☑	Abc	F8	F8		null
☑	Abc	F9	F9		null
☑	Abc	F10	F10		null

Figure 16.5: The input preview allows you to select input fields to include in the flow, rename fields, and change data types

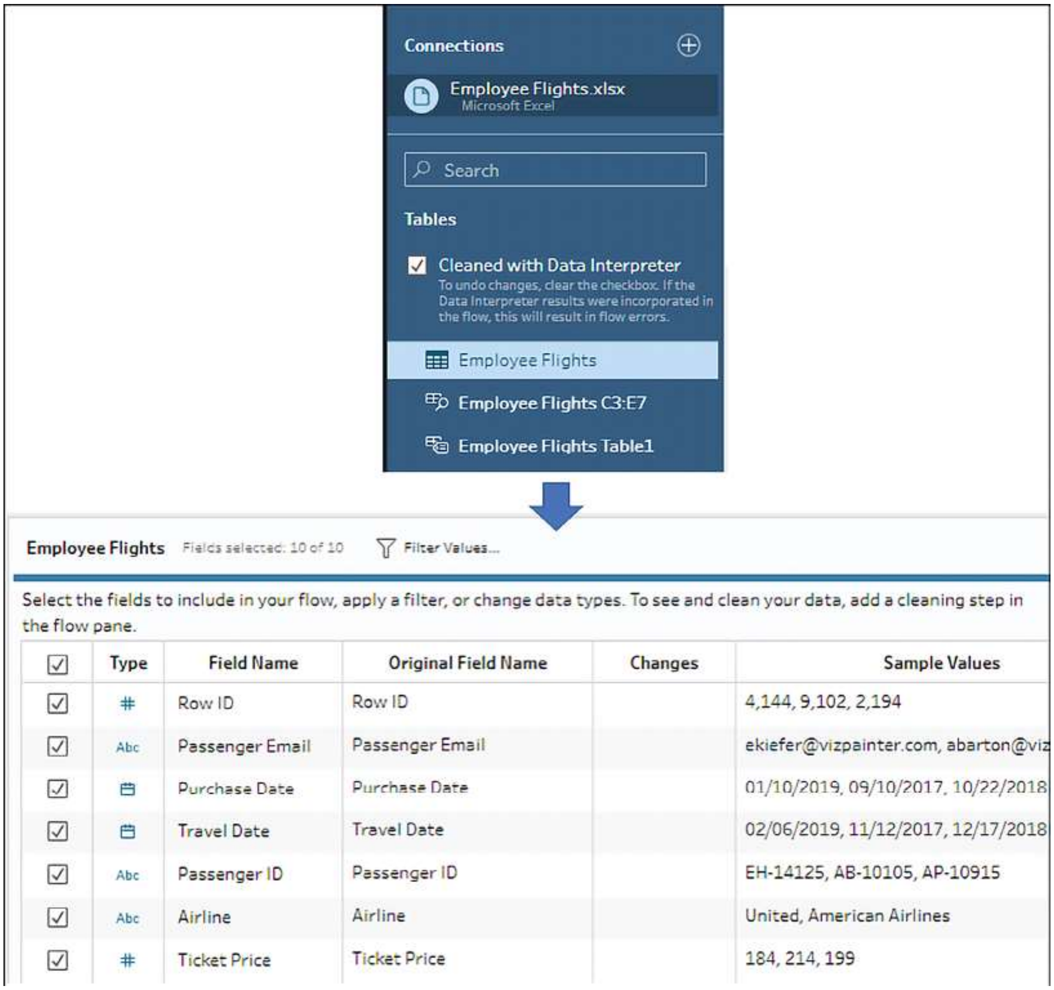


Figure 16.6: The data interpreter parses the file to fix common issues such as merged cells, empty headers, and sub-total lines

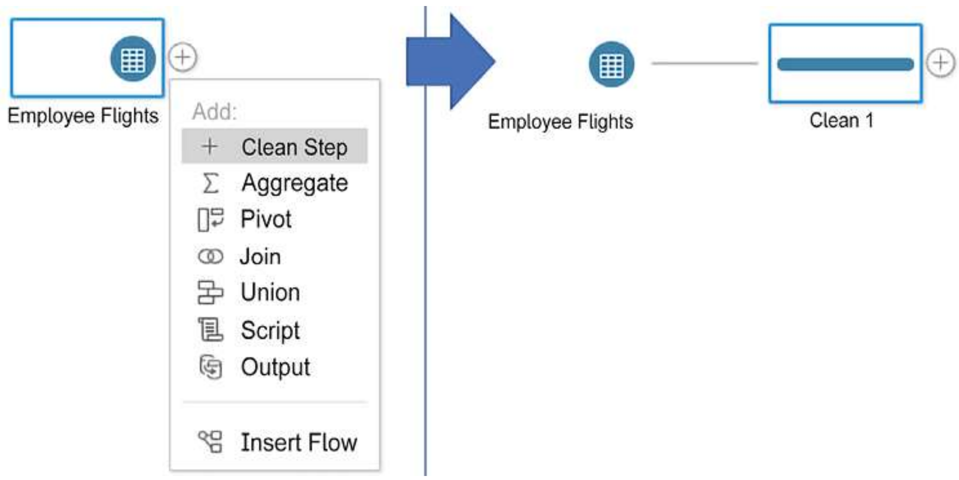


Figure 16.7: Adding a step extends the flow. Here, adding a clean step adds Clean 1

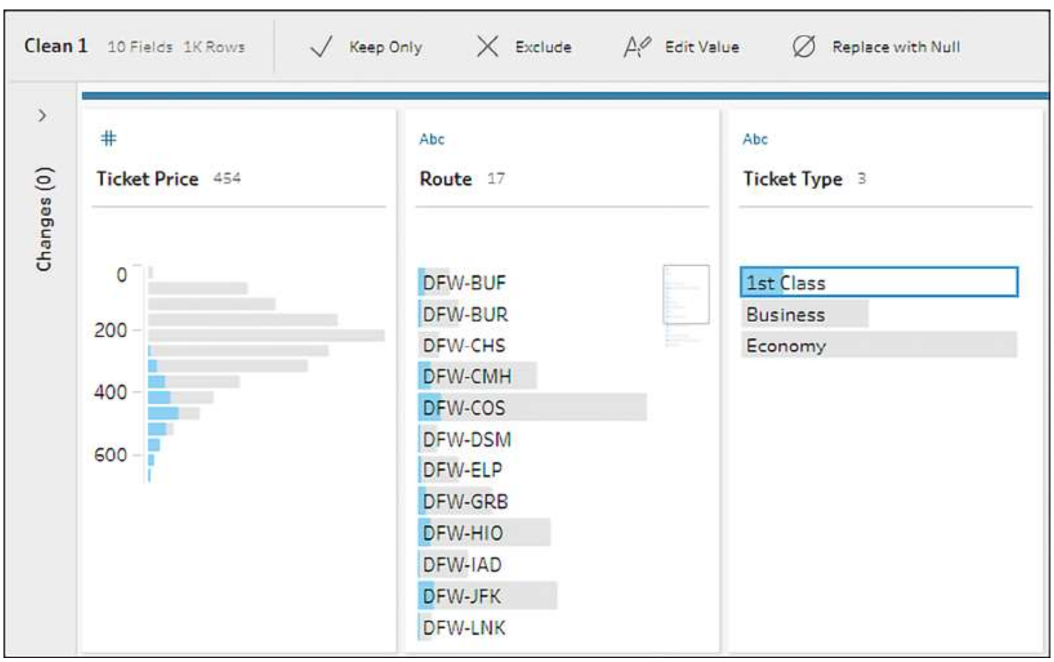


Figure 16.8: Selecting a value for a field in the profile pane highlights which values (and what proportion of those values) relate to the selected value

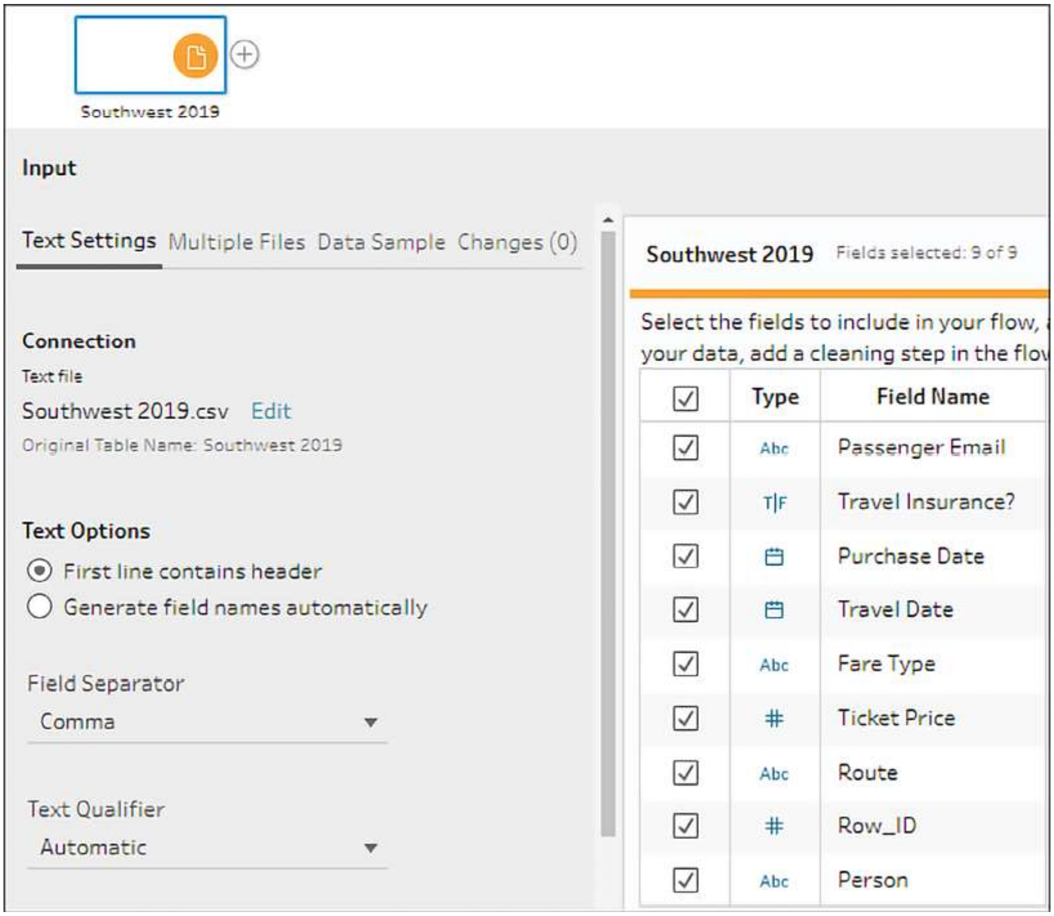


Figure 16.9: A text file includes options for headers, field separators, text qualifiers, character sets, and more. Notice also the tabs such as Multiple Files and Data Sample giving other options for the text input

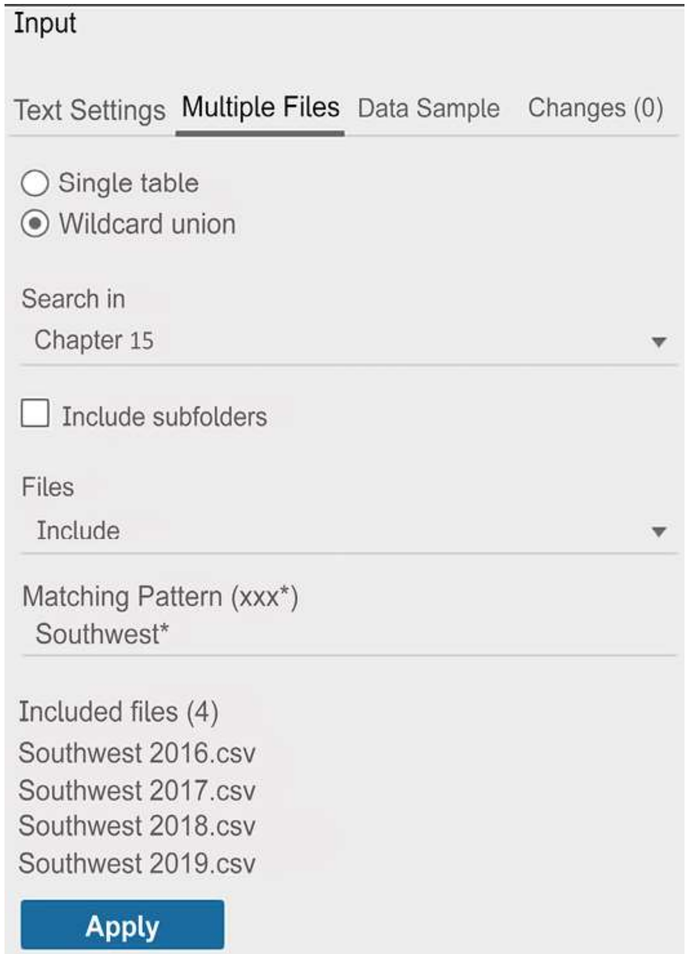


Figure 16.10: Using Matching Pattern tells Tableau Prep which files to union together. That way, when Southwest 2020.txt and future files are dropped into the directory, they will be automatically included

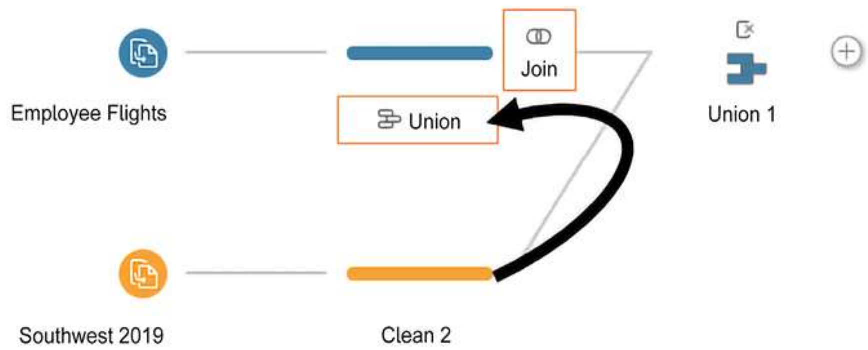


Figure 16.11: Dragging one step onto another in the flow reveals options for bringing the datasets together in the flow. Here, for example, there are options for creating a Union or Join

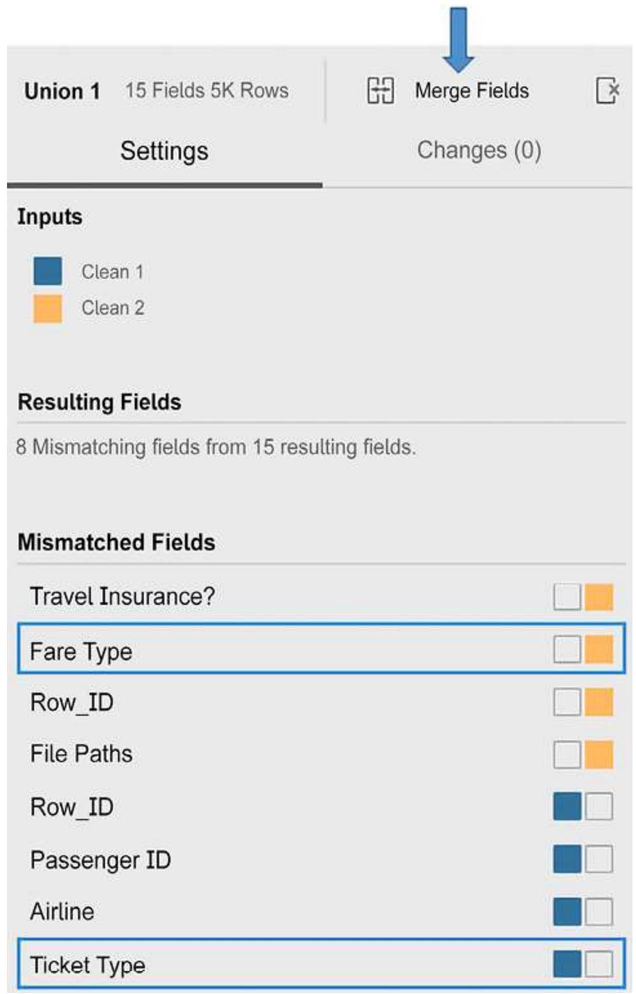


Figure 16.12: When you select a single field, Tableau Prep will highlight fields that are potentially the same data. Selecting both reveals the Merge Fields option

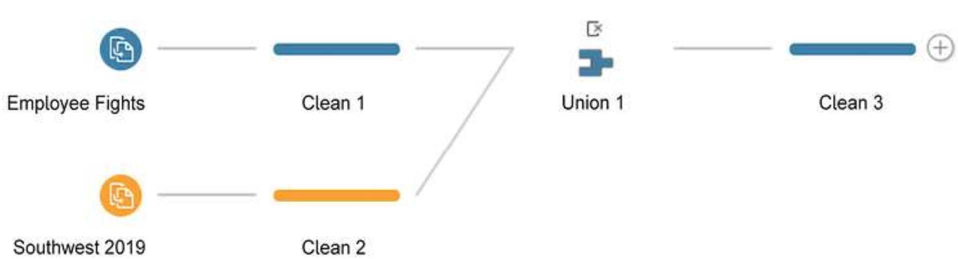


Figure 16.13: Your flow should look similar to this. You may notice some variation in the exact location of steps or color (you can change a step's color by right-clicking a step and selecting Edit Color)

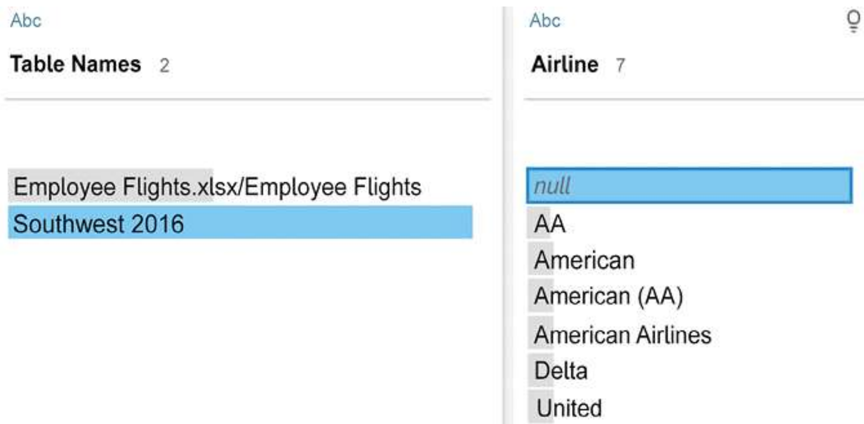


Figure 16.14: Every null value in the Airline field comes from the Southwest files. Fortunately, in this case, the source of the data indicates the airline

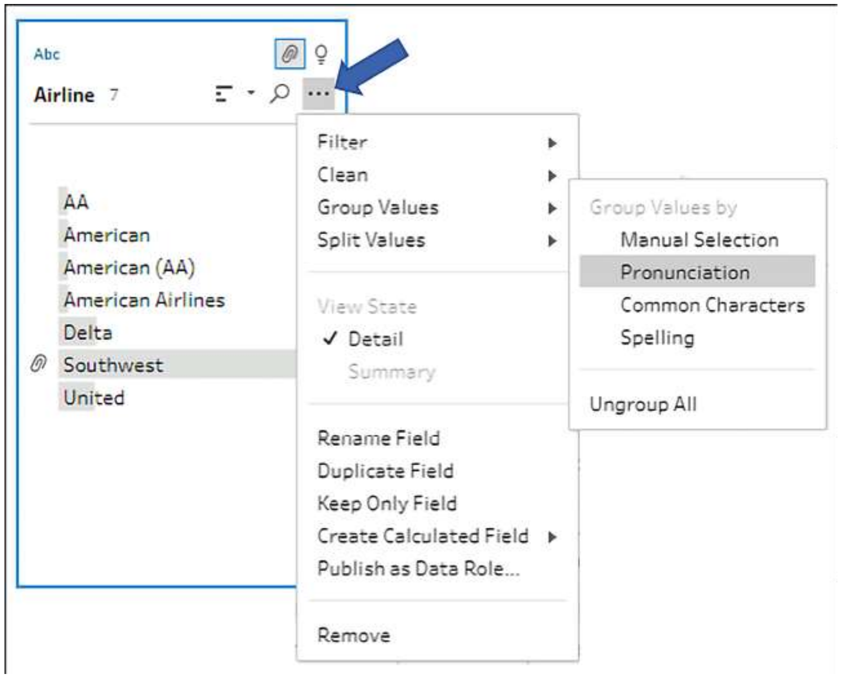


Figure 16.15: The ellipsis button on a field will reveal a plethora of options, from cleaning to filtering, to grouping, to creating calculations

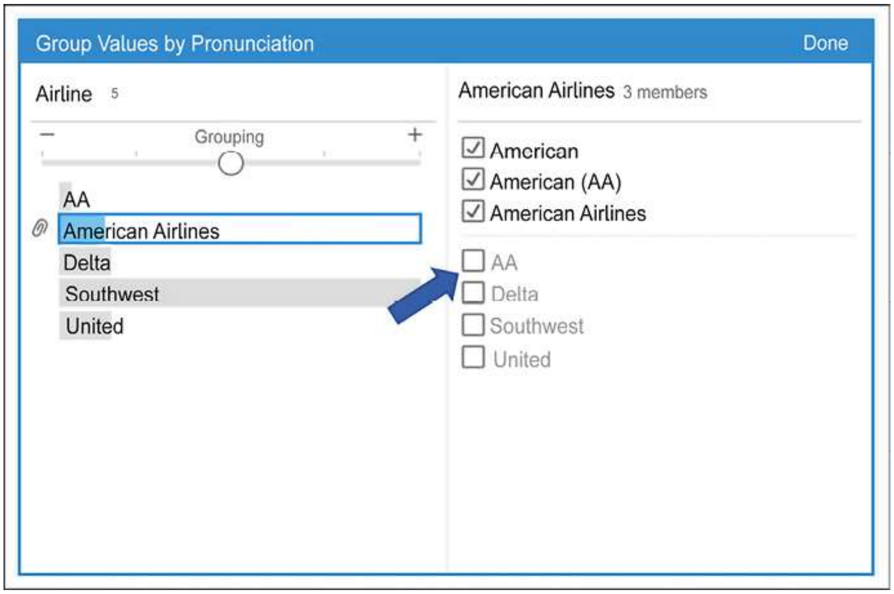


Figure 16.16: When grouping by pronunciation, you'll notice a slider allowing you control over the sensitivity of the grouping. You can also manually adjust groupings by selecting a field

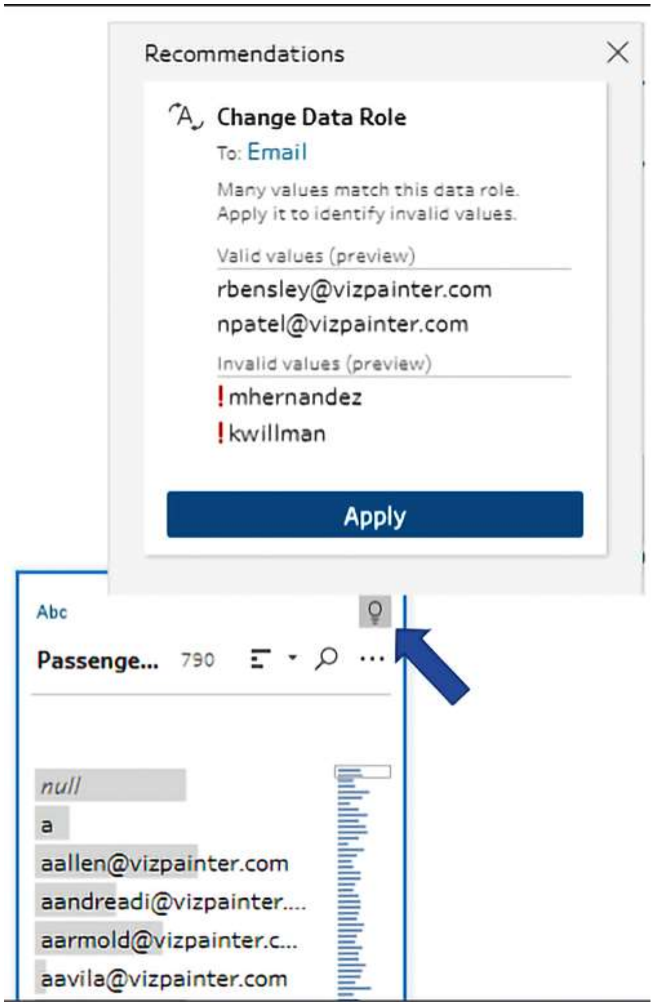


Figure 16.17: Recommendations will show when Tableau Prep has suggestions for cleaning a field

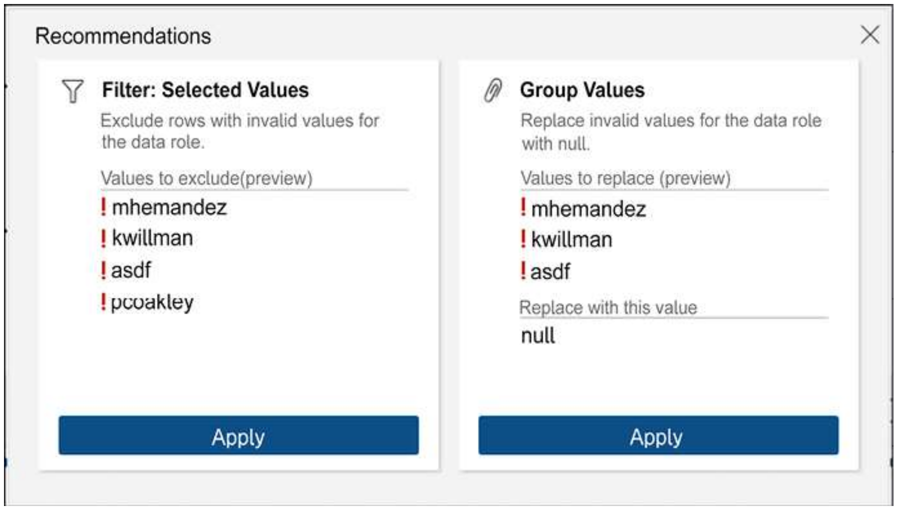


Figure 16.18: Here, Tableau Prep suggests either filtering out records with invalid values or replacing the invalid values with null. In this case, we don't want to filter out the entire record, but the invalid values themselves are useless and are best represented by null

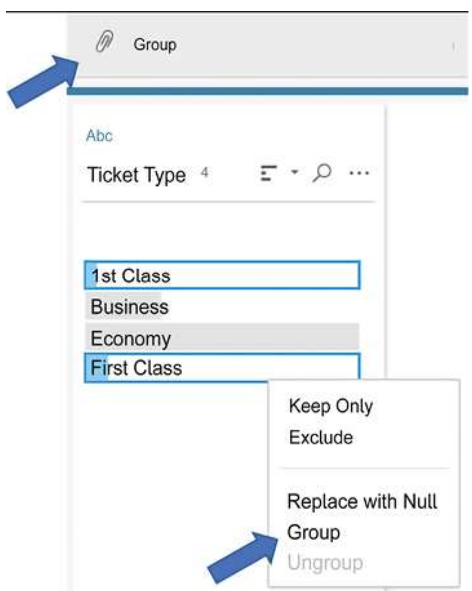


Figure 16.19: After selecting two or more values, you can group them together with the toolbar option or the right-click menu

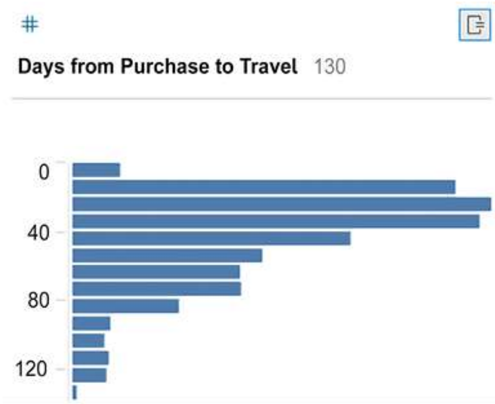


Figure 16.21: The calculated field shows up in the profile pane

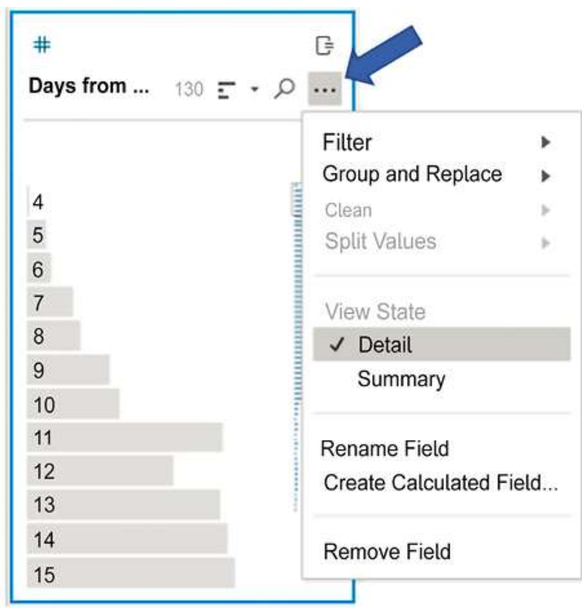


Figure 16.22: Numeric and date fields can be viewed in Summary or in Detail

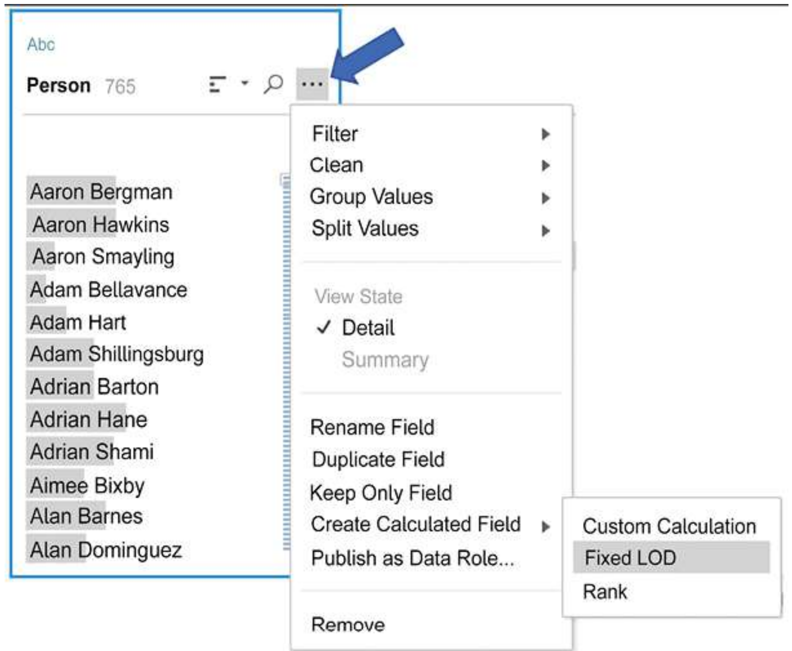


Figure 16.23: To create a Fixed LOD calculation, use the menu and select Create Calculated Field | Fixed LOD

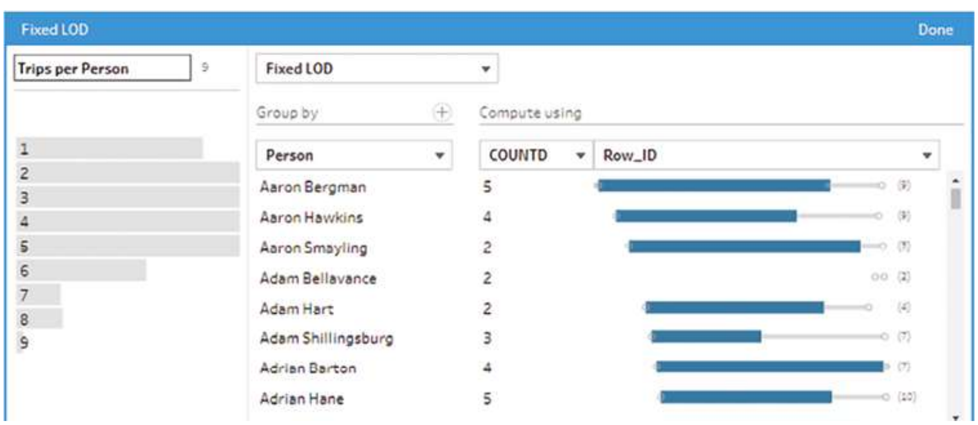


Figure 16.24: The Fixed LOD pane allows you to configure the LOD expression visually and get instant visual feedback concerning results

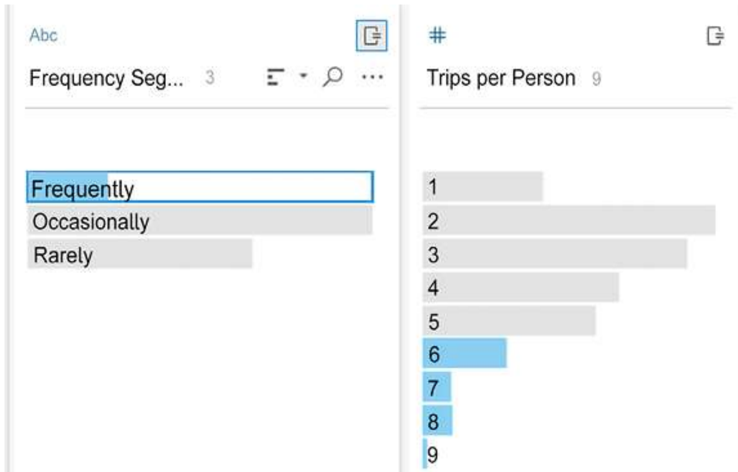


Figure 16.25: You can easily visualize how calculations relate to each other and other fields using the Profile pane

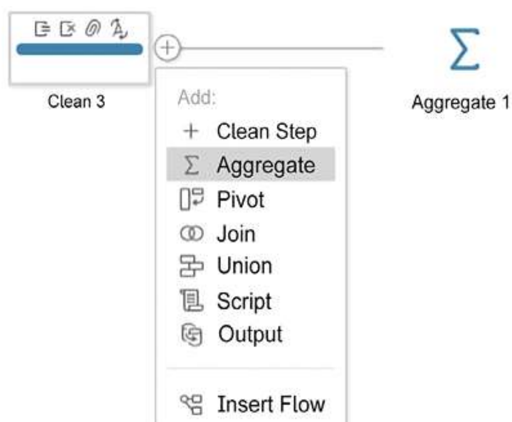


Figure 16.26: Adding an Aggregate step to the flow using the + symbol

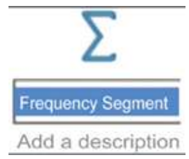


Figure 16.27: When editing the name of a step, you may also add a more verbose description to help document the step's purpose

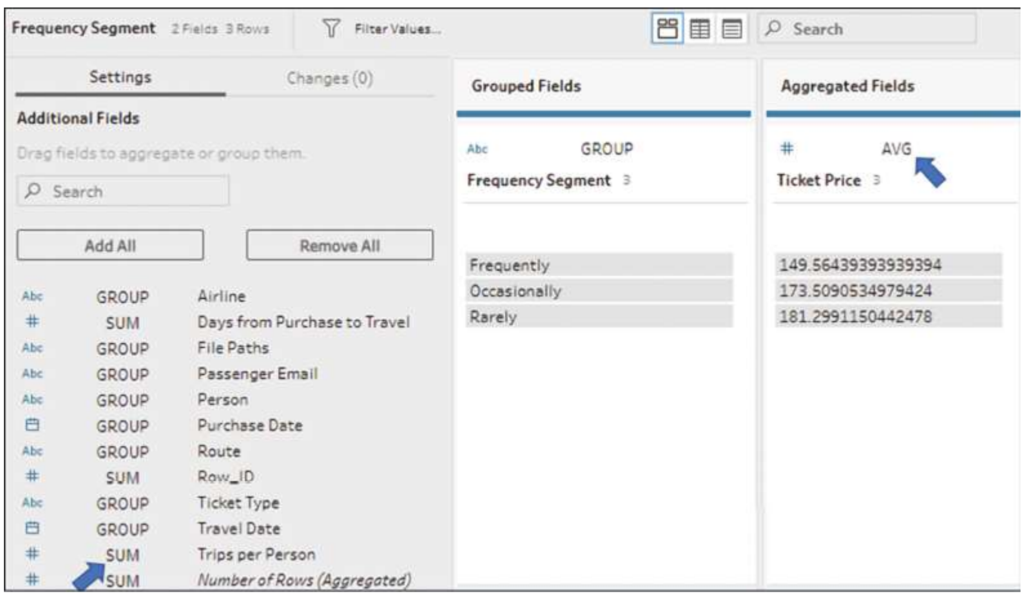


Figure 16.28: Adding an Aggregate step to the flow using the + symbol

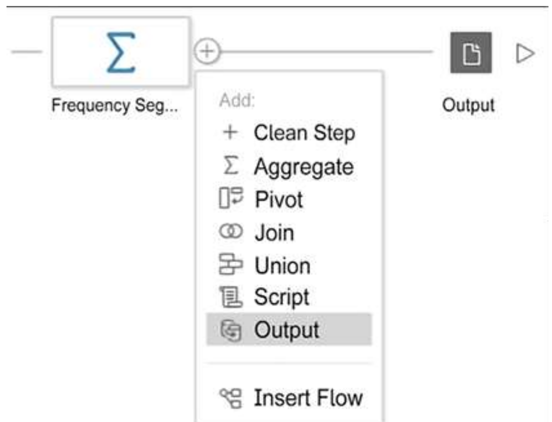


Figure 16.29: Adding an Output step to the flow using the + symbol

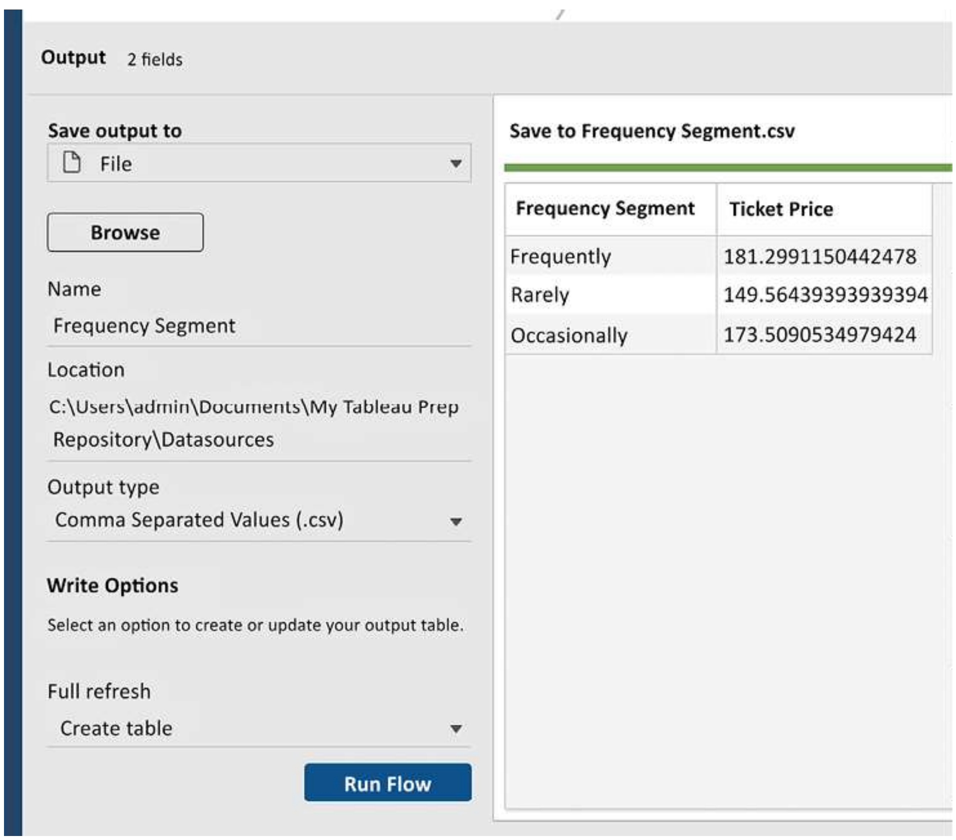


Figure 16.30: This output will contain exactly three rows of data

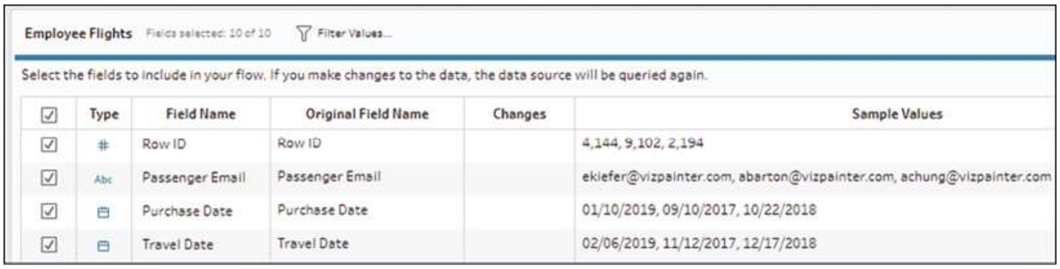


Figure 16.31: The Filter Values... option allows you to filter values on the input step. This could improve performance on large datasets or relational databases

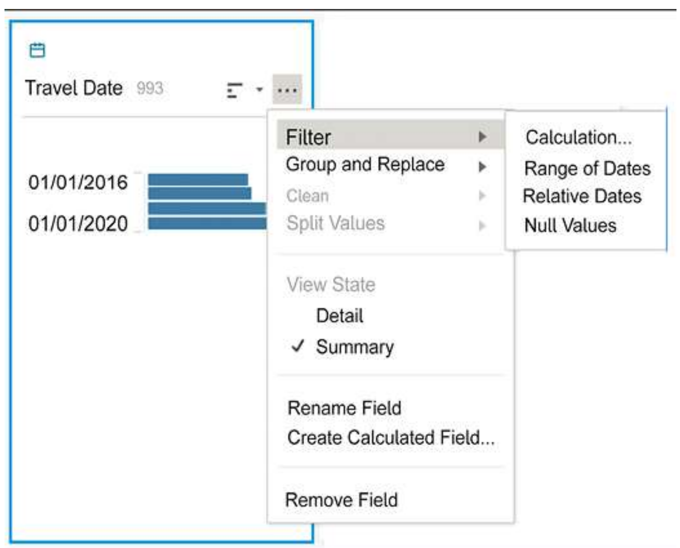


Figure 16.32: Filter options for a field include filtering by Calculation, Range of Dates, and Relative Dates, and keeping or excluding Null Values

Figure 16.33: Your flow should look similar to this (exact placement and colors of steps may vary)

Figure 16.34: Route uses airport codes for origin and destination separated by a dash

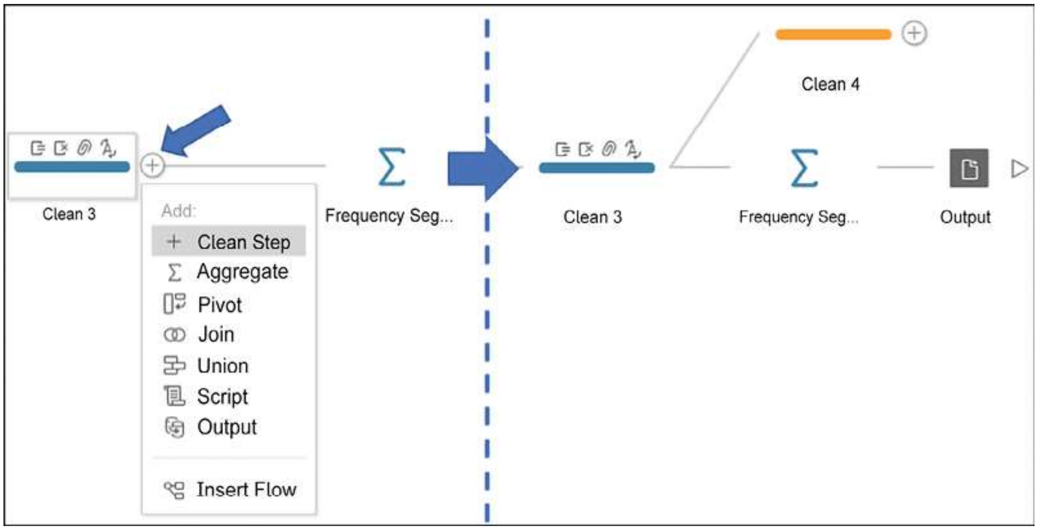


Figure 16.35: Adding to a step that already has an output adds a new branch to the flow

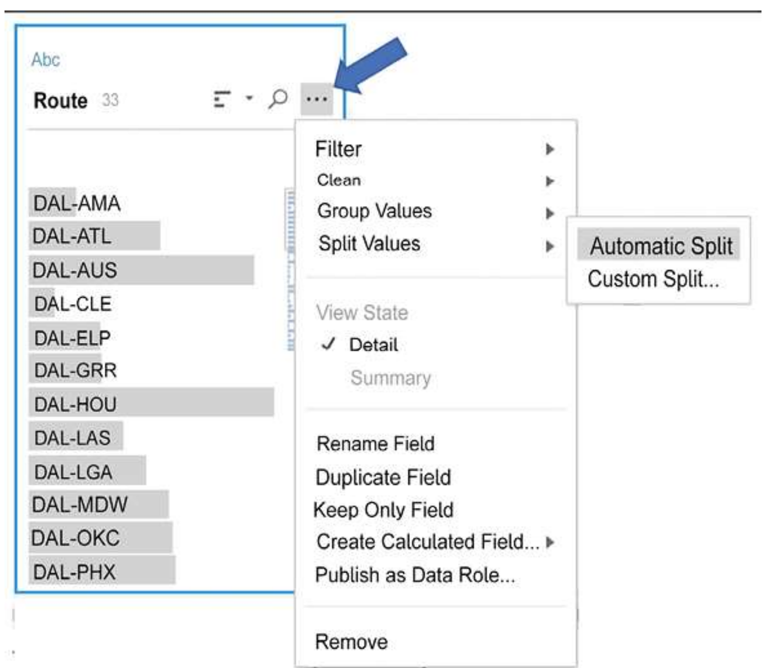


Figure 16.36: Split Values allows you to divide delimited strings into separate fields. Automatic Split attempts to determine the delimiter, while Custom Split… allows you greater options and flexibility

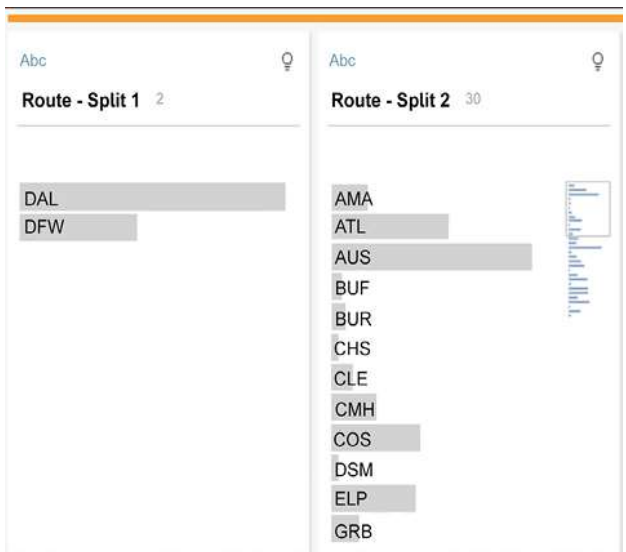


Figure 16.37: The results of the split will be new fields in the flow

Abc Extract Airport Code	Abc Extract Airport Name	Extract Latitude	Extract Longitude
BTI	Barter Island LRRS Airport	70.1340	-143.582
LUR	Cape Lisburne LRRS Airport	68.8751	-166.110
PIZ	Point Lay LRRS Airport	69.7329	-163.005
ITO	Hilo International Airport	19.7214	-155.048
ORL	Orlando Executive Airport	28.5455	-81.333

Figure 16.38: The hyper extract contains the data we'll need to supplement the flow with our own geospatial data

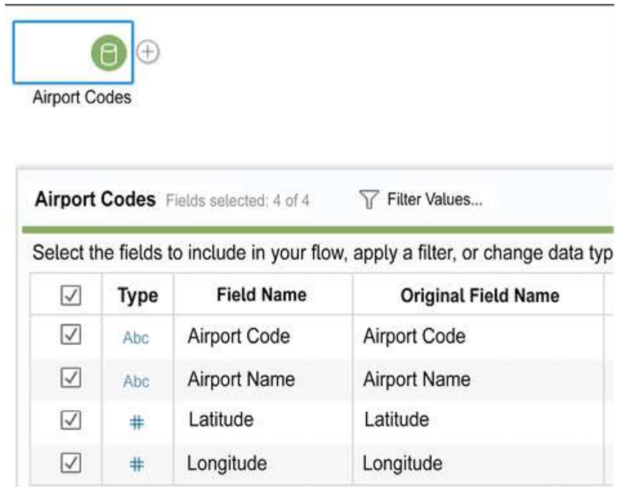


Figure 16.39: The input pane for the Airport Codes file

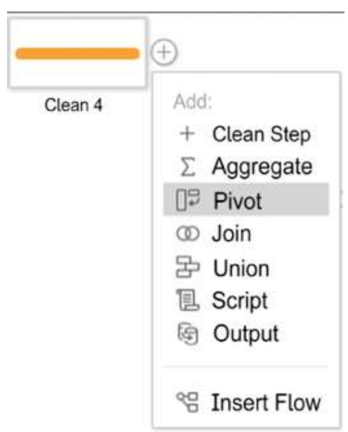


Figure 16.40: Adding a Pivot step from Clean 4

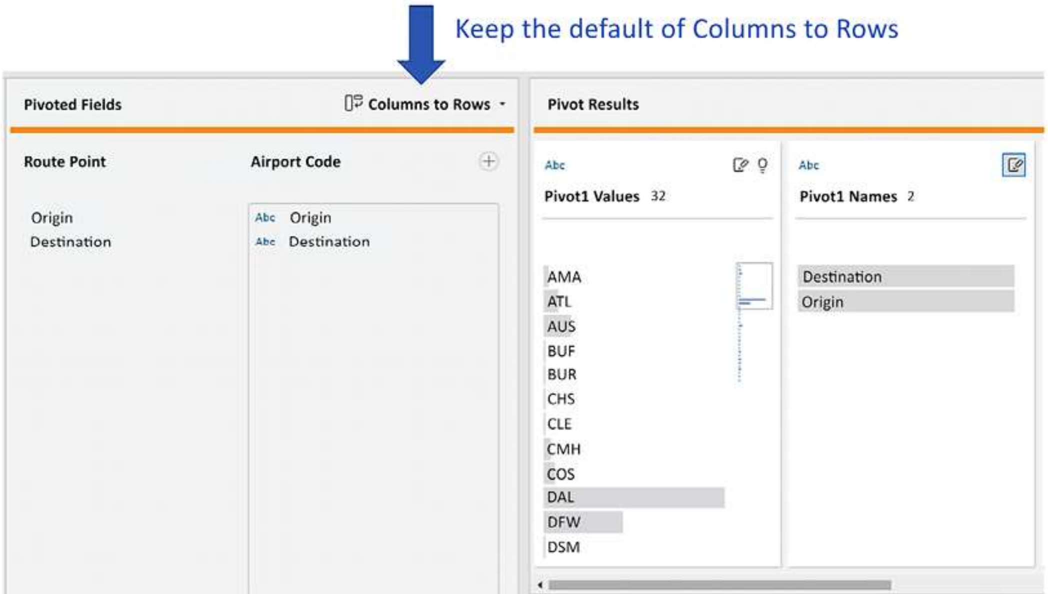


Figure 16.41: Pivot1 Names keeps values from the original column names, while Pivot1 Values contains the actual values from Origin and Destination

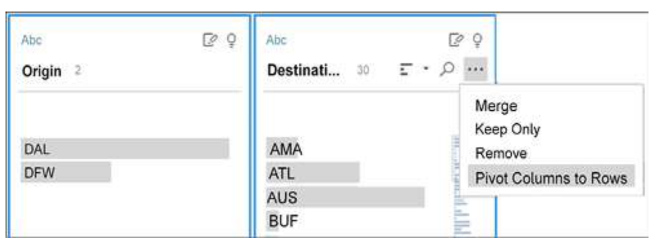


Figure 16.42: A shortcut for pivoting columns to rows

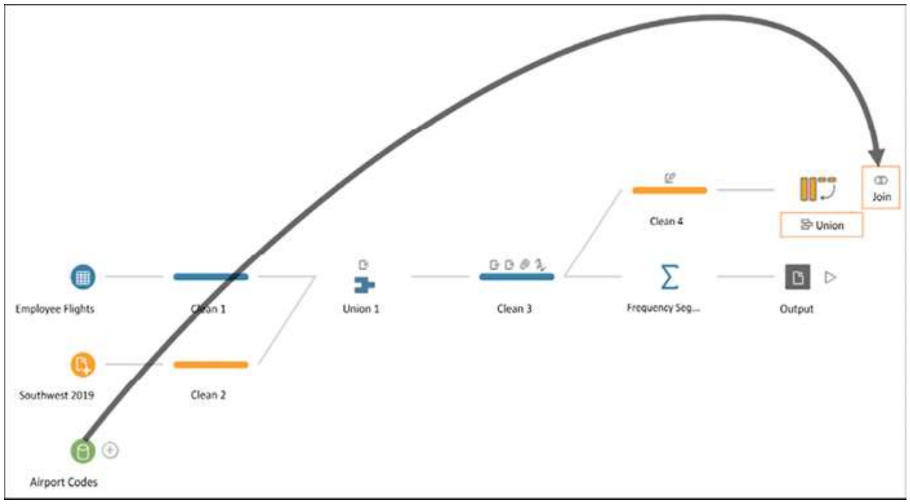


Figure 16.43: Dragging Airport Codes to the Join area of the Pivot step

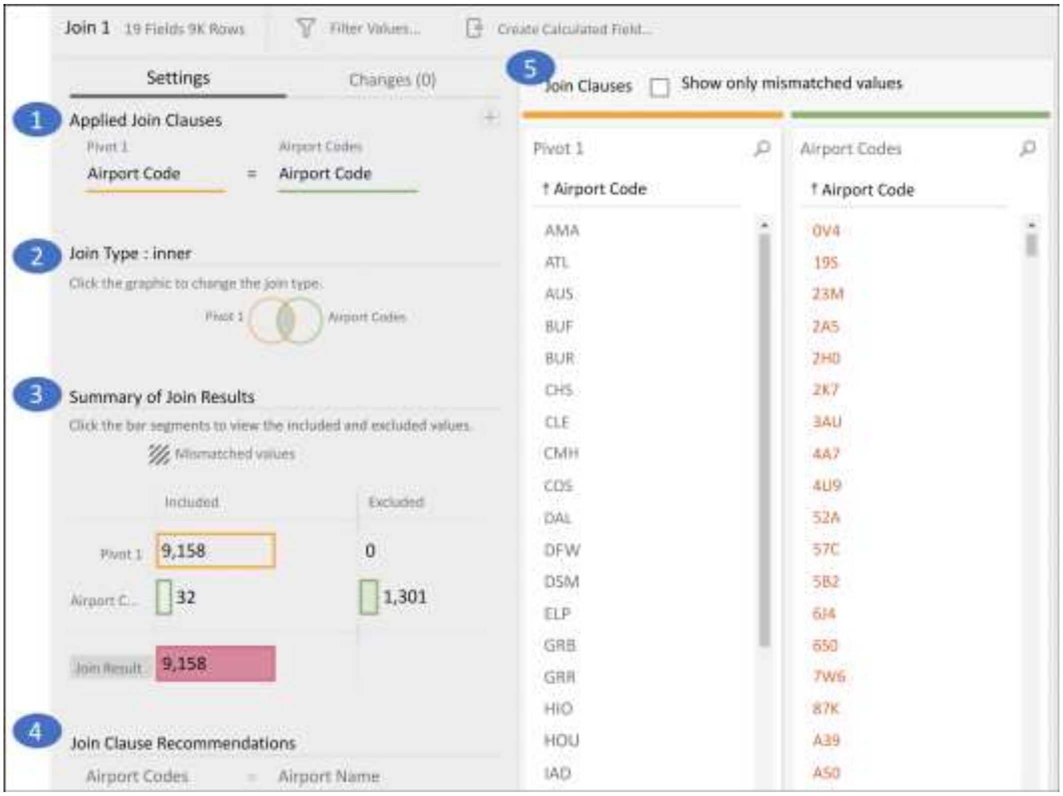


Figure 16.44: The join pane gives a lot of information and options for configuring the join and understanding the results. Important sections of the interface are numbered with descriptions below

Figure 16.45: The toolbar allows you to run the flow for all outputs or a single output, while the button on the output step will only run the flow for that output

Figure 16.46: Your final flow will resemble this, but may be slightly different in appearance

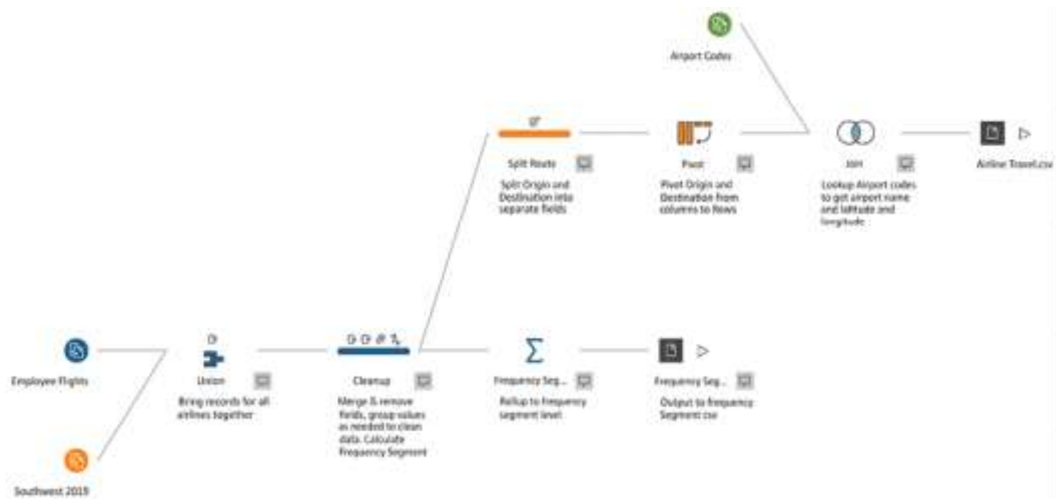


Figure 16.47: This flow is cleaned up and contains “self-documentation”

Figure 16.48: Exploring the data in the Airline Travel.twb workbook, which shows all flight destinations from Dallas Love Field (DAL) or Dallas Fort-Worth (DFW)

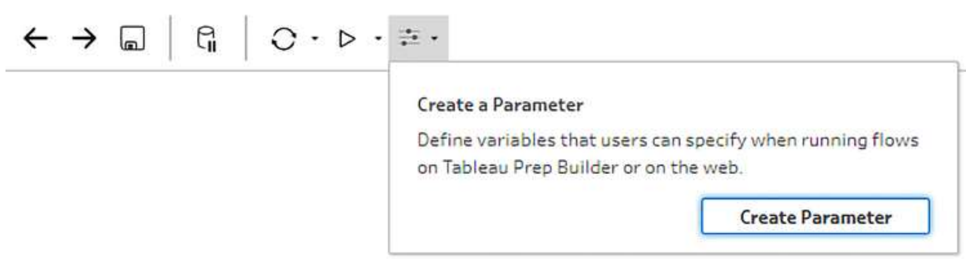


Figure 16.49: Clicking the Parameters button on the toolbar reveals the option to create a parameter or manage any existing parameters

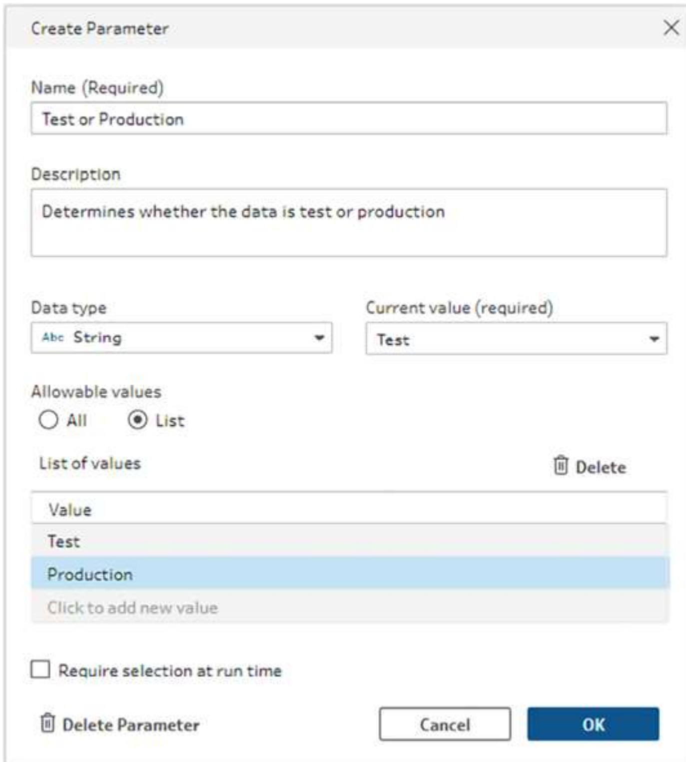


Figure 16.50: Creating a new parameter

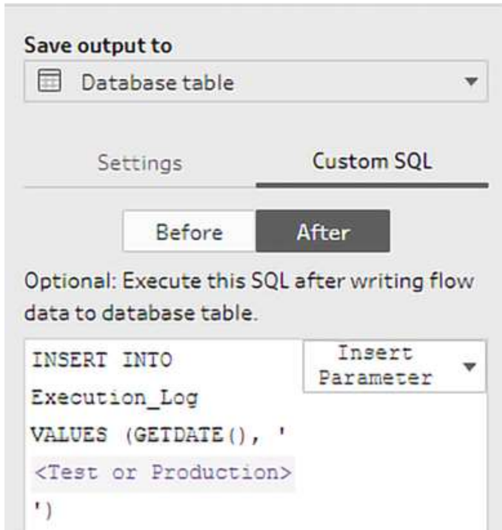


Figure 16.51: Using a parameter in a Output step Custom SQL statement

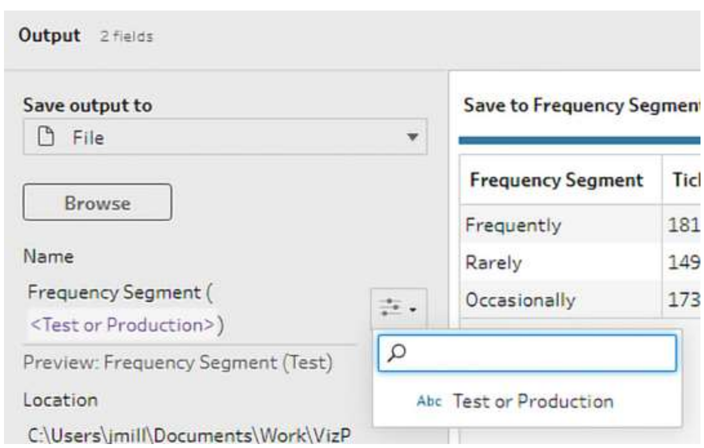


Figure 16.52: Using a parameter to define an output filename

Code

Code 16.1

```
DATEDIFF('day', [Purchase Date], [Travel Date])
```

Code 16.2

```
{FIXED [Person] : COUNTD([Row_ID])}
```

Code 16.3

```
{FIXED [Person] : SUM(1)}
```

Code 16.4

```
IF [Trips per Person] <= 2 THEN "Rarely"
ELSEIF [Trips per Person] <= 5 THEN "Occasionally"
ELSE "Frequently"
END
```

Code 16.5

```
[Travel Date] > MAKEDATE(2016, 1, 1)
```

Links

- Details available on Tableau Help: https://help.tableau.com/current/prep/en-us/prep_save_share.htm#refresh-output-files-from-the-command-line.

Chapter 17

Figures

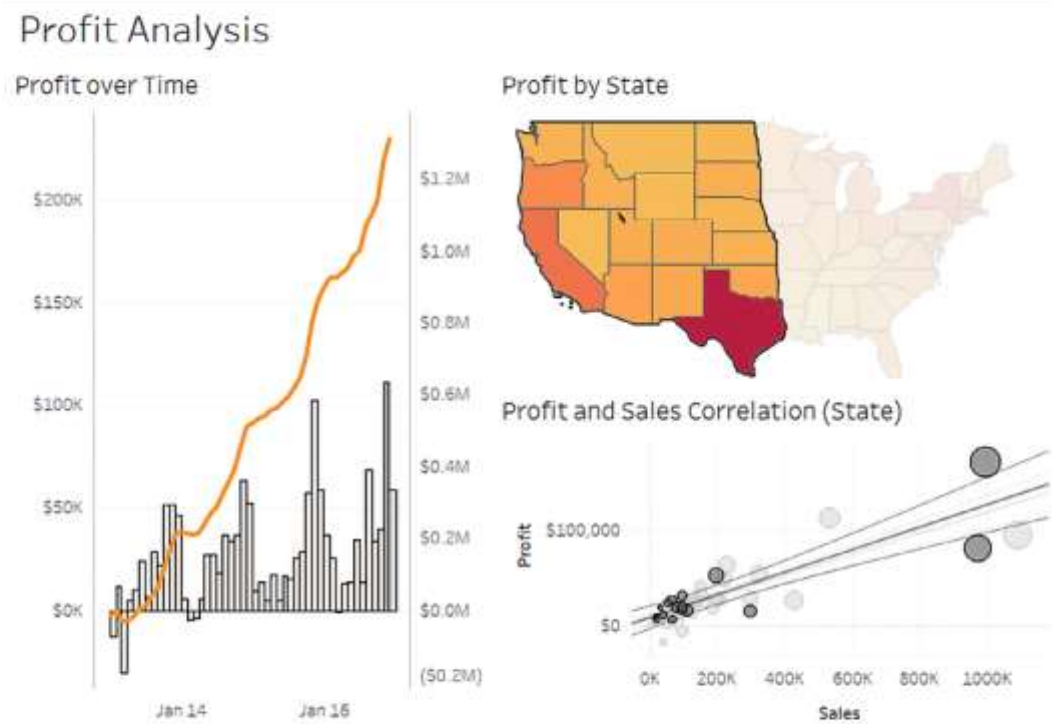


Figure 17.1: You can see states and circles that are selected in this screenshot. You may optionally print views with selections

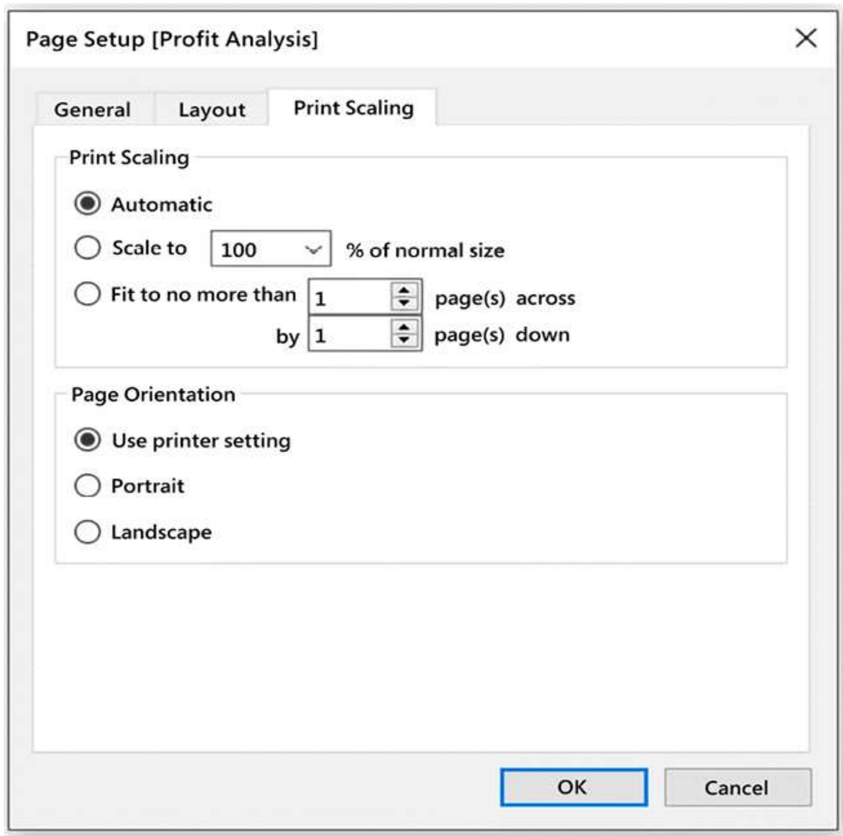


Figure 17.2: The Page Setup dialog contains options for layout and print scaling

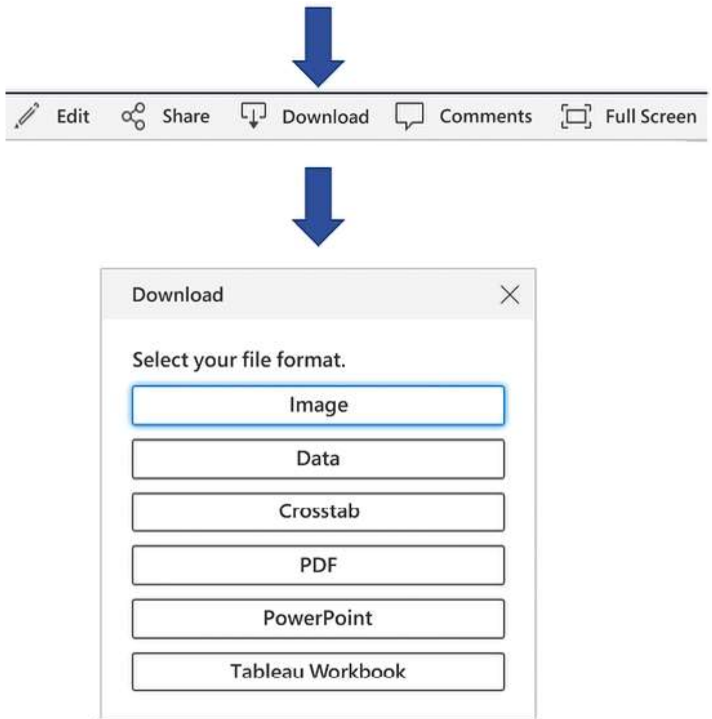


Figure 17.3: Exporting from the toolbar provides a similar experience for Server, Online, and Public

Figure 17.4: The sign-in screen for Tableau Cloud

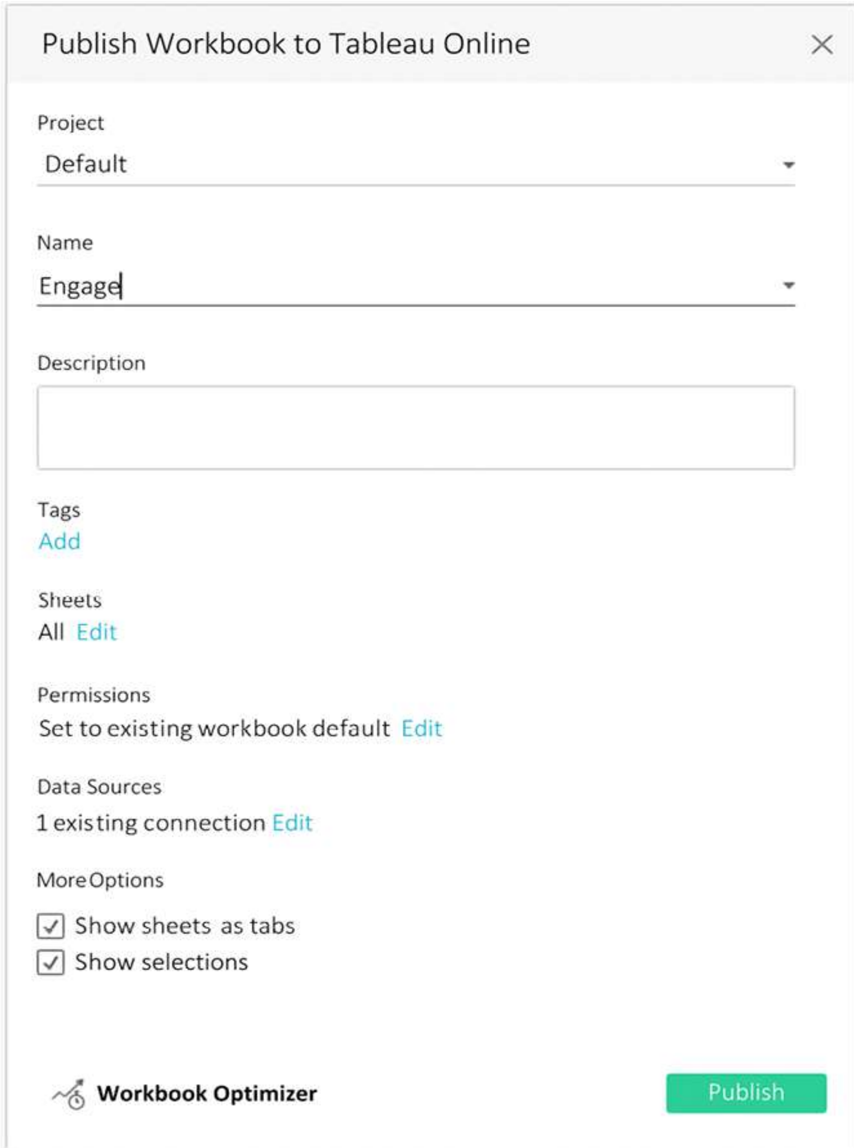


Figure 17.5: Publishing to Tableau Cloud

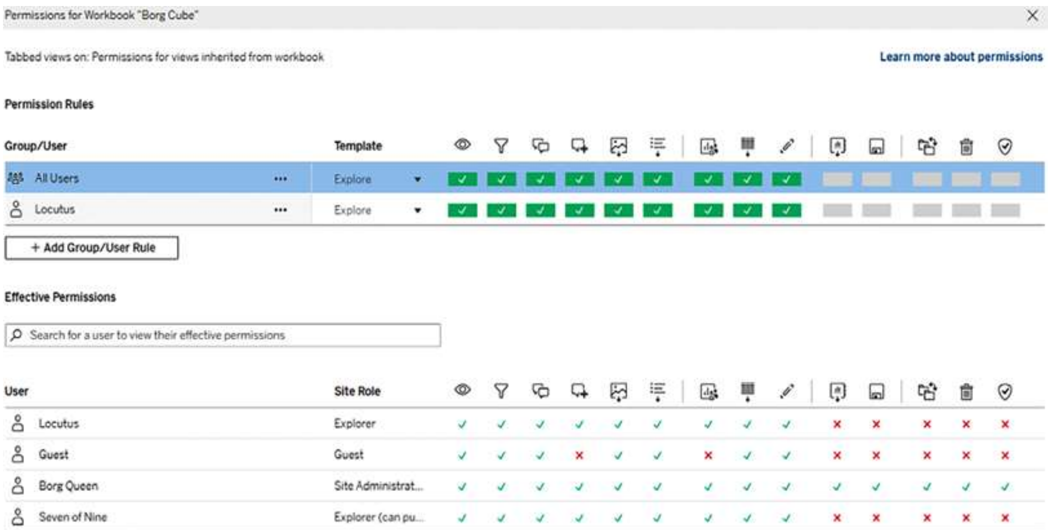


Figure 17.6: Tableau Server allows for a robust set of permissions. You can adjust individual and group permissions for viewing, filtering, commenting, editing, saving, and more

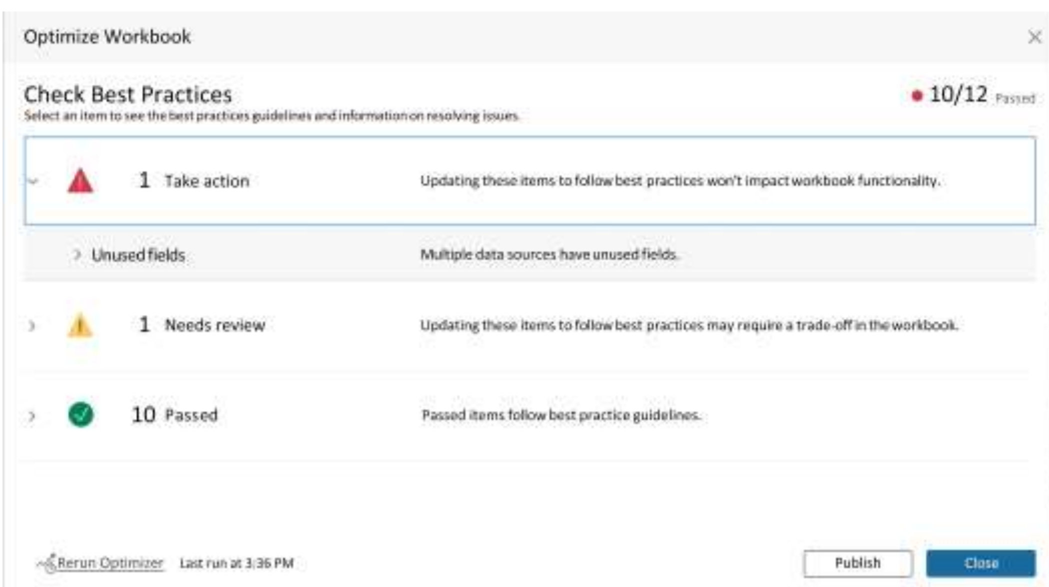


Figure 17.7: The Workbook Optimizer is a great option for increasing the efficiency of your workbook prior to publishing

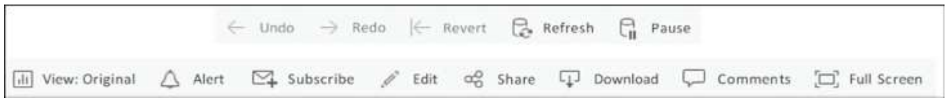


Figure 17.8: The Tableau Server toolbar

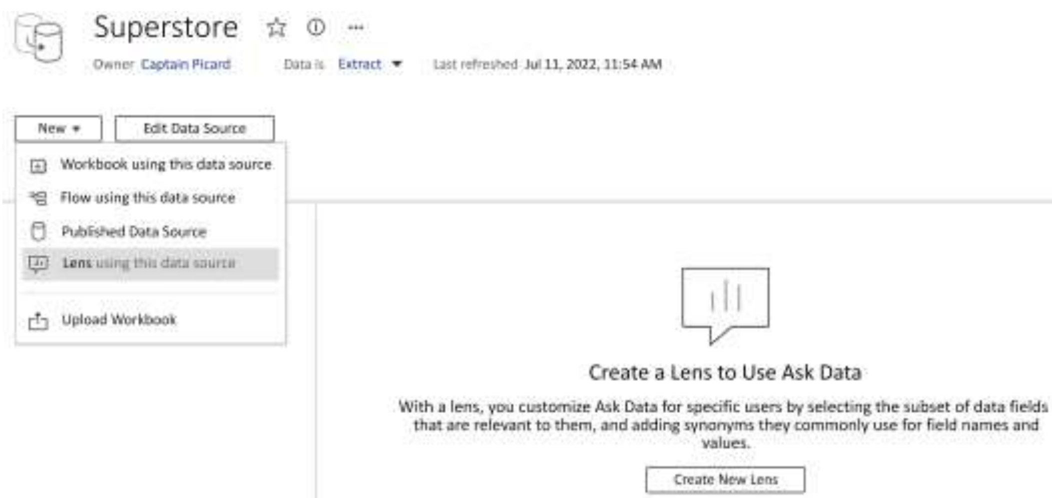


Figure 17.9: Creating an Ask Data Lens

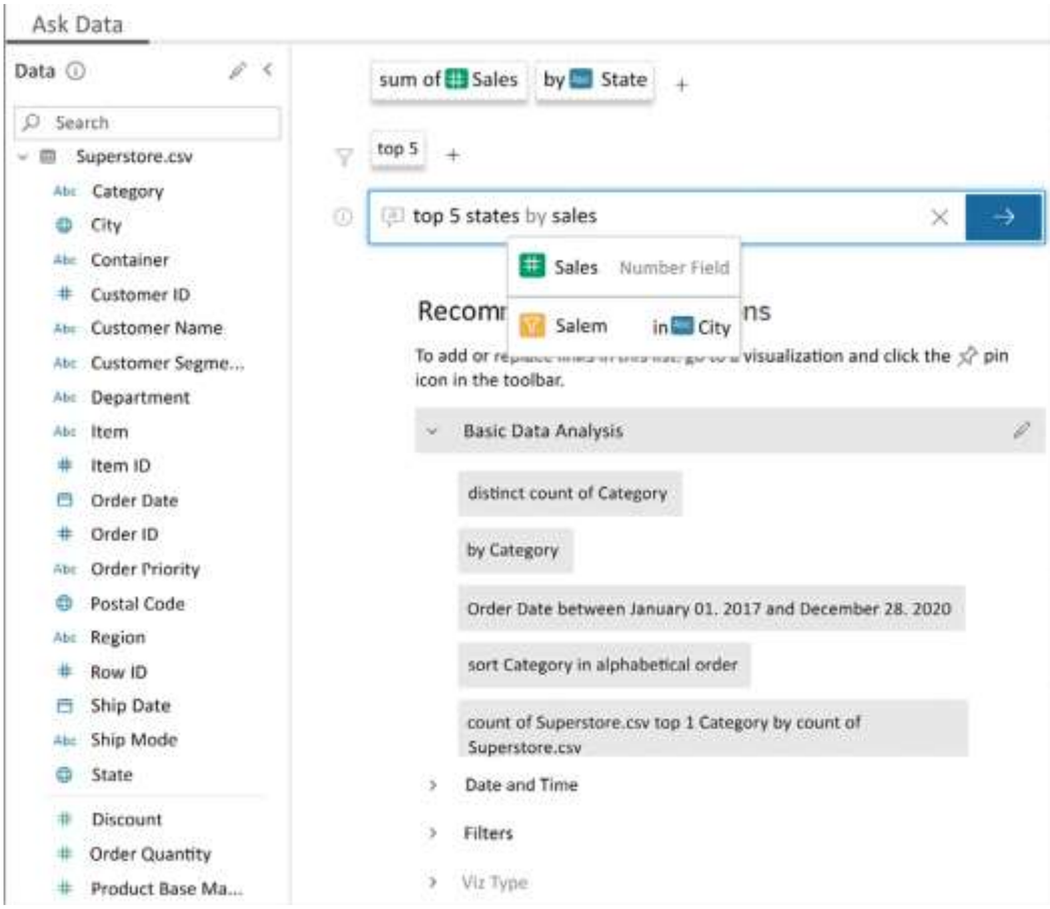


Figure 17.10: Typing a natural language phrase into Ask Data's search

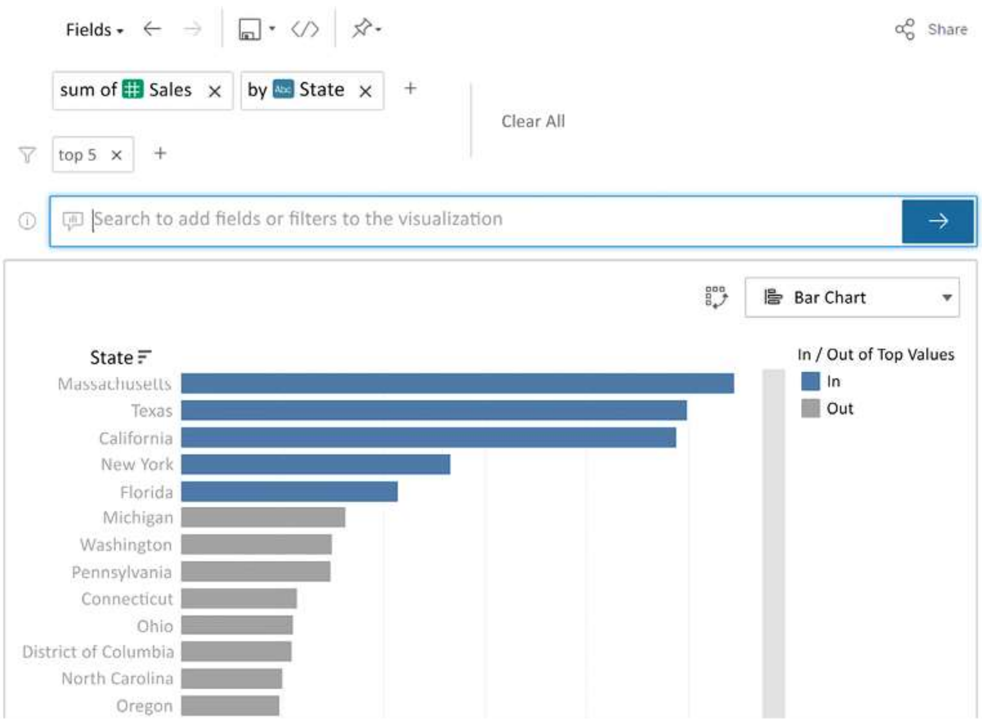


Figure 17.11: The visual response to the phrase "top 5 states by sales"

Tables

	Tableau Server	**Tableau Cloud**	**Tableau Public**
Description	A server application installed on one or more server machines that hosts views and dashboards created with Tableau Desktop.	A cloud-based service maintained by Tableau Software that hosts views and dashboards created with Tableau Desktop.	A cloud-based service maintained by Tableau Software that hosts views and dashboards created with Tableau Desktop or the free Tableau Public client.
Licensing cost	Yes	Yes	Free

Administration	Fully maintained, managed, and administered by the individual or organization that purchased the license.	Managed and maintained by Tableau Software with some options for project and user management by users.	Managed and maintained by Tableau Software. Users have some options for managing their content.
Authoring and publishing	Users of Tableau Desktop may author and publish workbooks to Tableau Server. Web Authoring allows Tableau Server users the capability to edit and create visualizations and dashboards in a web browser.	Users of Tableau Desktop may author and publish workbooks to Tableau Cloud. Web Authoring allows Tableau Cloud users the capability to edit and create visualizations and dashboards in a web browser.	Users of Tableau Desktop or the free Tableau Public client can publish workbooks to Tableau Public. Web authoring (currently in beta) allows users to create content with only a web browser.
Interaction	Licensed Tableau Server users may interact with hosted views. Views may also be embedded in intranet sites, SharePoint, and custom portals.	Licensed Tableau Cloud users may interact with hosted views. Views may also be embedded in intranet sites, SharePoint, and custom portals.	Everything is public-facing. Anyone may interact with hosted views. Views may be embedded in public websites and blogs.
	Tableau Server	**Tableau Cloud**	**Tableau Public**
Limitations	None except those based on licensing purchased and hardware used.	Most data sources must be extracted before workbooks can be published. Most non-cloud-based data sources must have extracts refreshed using Tableau	All data must be extracted and each data source is limited to 15 million rows.

		Desktop on a local machine or through the **Tableau Bridge**. Certain storage and load-time limits are also in place.	
Security	The Tableau Server administrator may create sites, projects, and users and adjust permissions for each. Access to the underlying data can be restricted, and downloading of the workbook or data can be restricted.	The Tableau Server administrator may create projects and users and adjust permissions for each. Access to the underlying data can be restricted, and downloading of the workbook or data can be restricted.	By default, anyone may download and view data; however, access to these options may be restricted by the author.
Good uses	Internal dashboards and analytics and/or use across departments/ divisions/clients through multi-tenant sites.	Internal dashboards and analytics, especially where most data sources are cloud-based. Sharing and collaboration with remote users.	Sharing visualizations and dashboards with the public. Embedding dashboards on public-facing websites or blogs.

Table 17.1