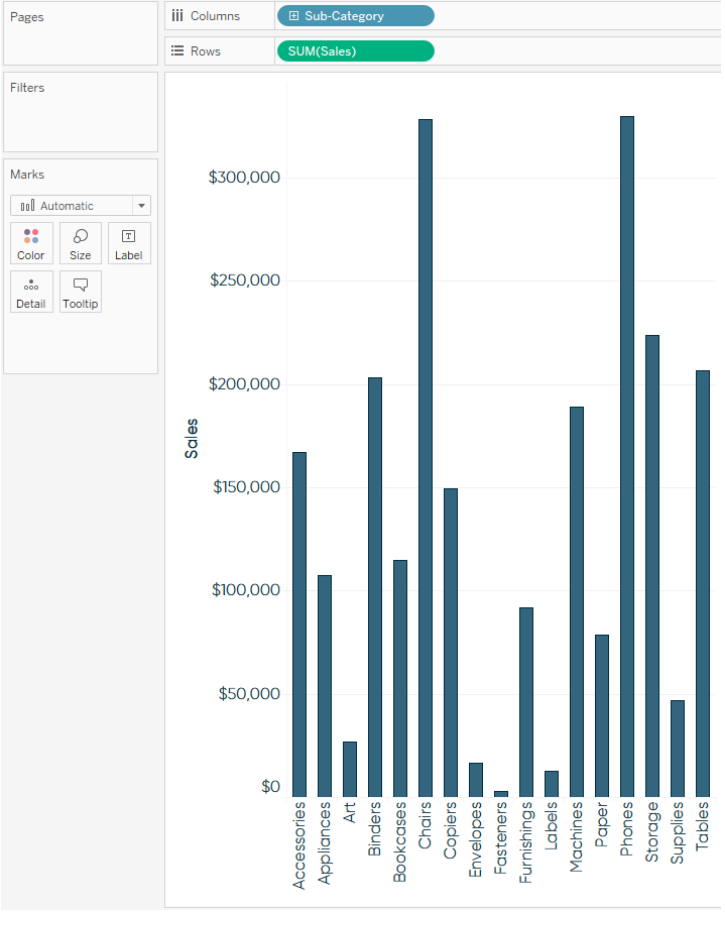
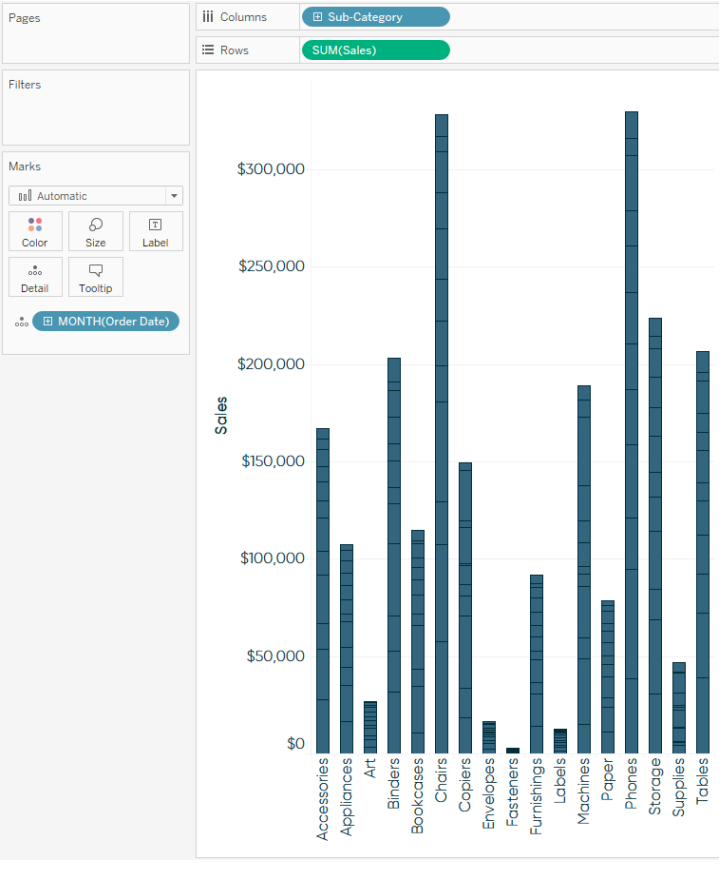
How to Make a Box-and-Whisker Plot in Tableau

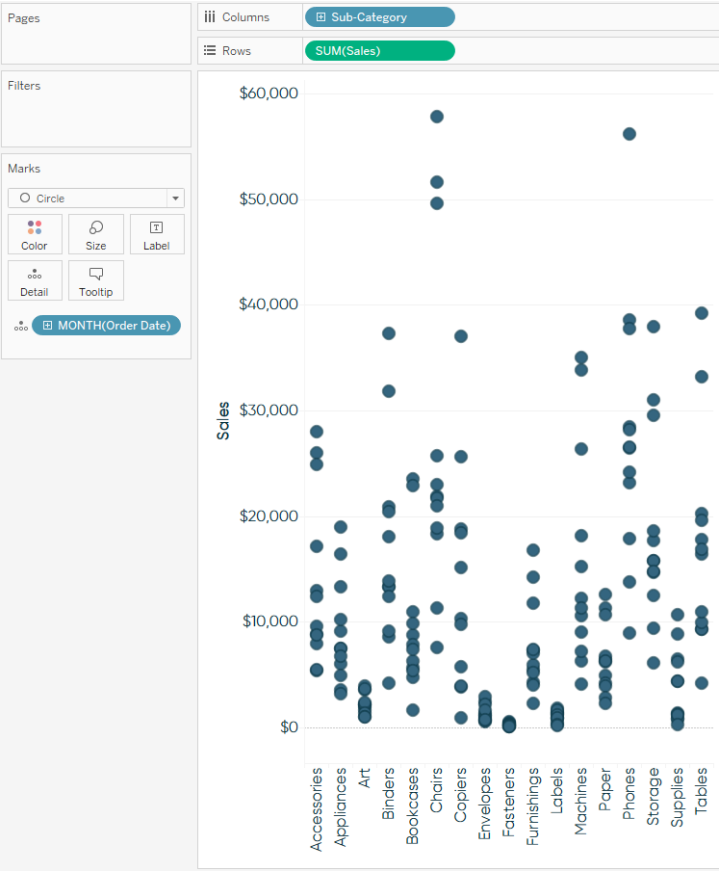
Box-and-whisker plots are one of the out-of-the-box Show Me options in Tableau, but they are actually created with reference lines—which is what I'll show here. To create a box-and-whisker plot, start by creating a bar chart with the dimension and measure of interest (in the preceding example, we are looking at Sales by Sub-Category):



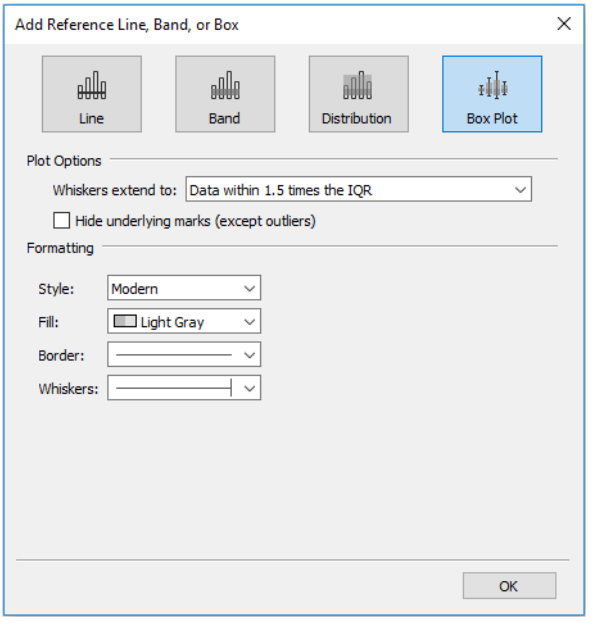
Next, add the distribution that you care about to the Detail Marks Card. In this case, we are looking at how Sales are distributed by Sub-Category, by Month of Order Date. So in this example, Month of Order Date is added to the Detail Marks Card:



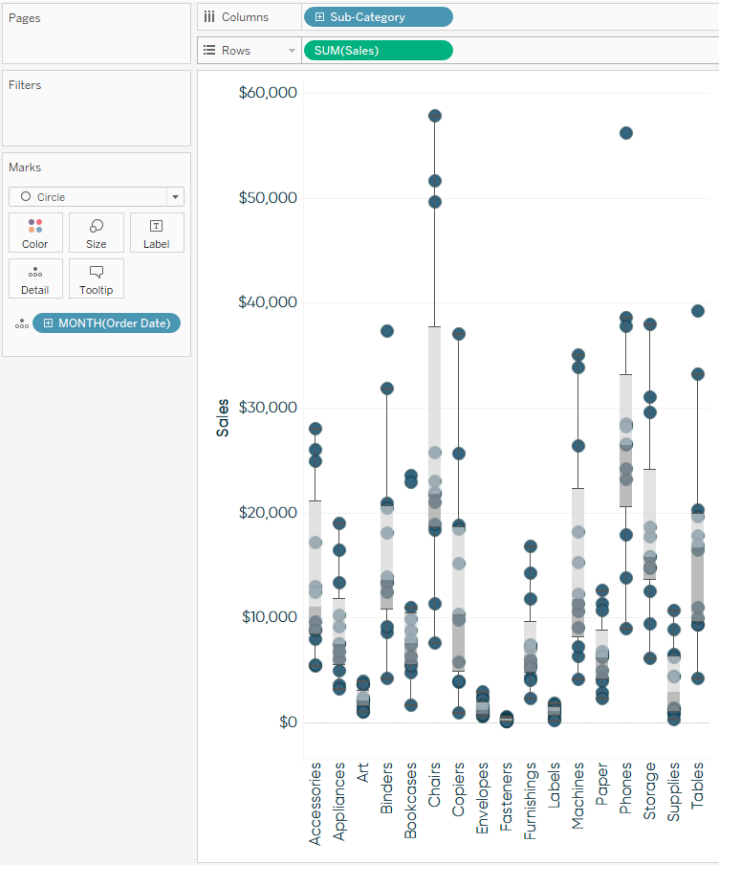
By increasing the level of detail, a stacked bar chart is created, with each stack per sub-category representing one of the twelve months of the year. To set the foundation of the box-and-whisker plot, convert this stacked bar chart to a dot plot by changing the mark type from Automatic (Bar), to Circle:



Lastly, to create a box-and-whisker plot, right-click the y-axis, and choose Add Reference Line. When the add reference line dialog box appears, click the choice for Box Plot. There are some formatting options available, but the default settings are usually best:



IQR stands for Interquartile Range, which are the data points between the first and third quartile. So the default options are telling Tableau to make all of the data points on the box-and-whisker plot fit into 1.5 times the IQR; anything outside of that range is an outlier. This sounds confusing and is probably why I don't see this chart type getting much traction, but this chart type provides a lot of helpful context and is easy to read once someone explains it to you. Let's take another look at the final box-and-whisker plot from this tutorial:



Each line on the box-and-whisker plot provides a piece of statistical context. The most important line is the one right in the middle of each "box," which represents median. With median displayed, you can quickly look across the dimension members and compare medians, regardless of how big or small the range of values is within each column.

That alone is very useful for an analysis, but the rest of the lines also have a meaning. Working out from the median, the next set of lines is showing plus or minus one quartile from the median.

Lastly, the upper whisker is 50% higher than the IQR, or "middle fifty," which are the data points within the first and third quartile. The lower whisker is 50% lower than the IQR. Any data points outside of the box-and-whisker are considered outliers.

Now not only can you make a box-and-whisker plot in Tableau, you know how to use them to get the most out of your analyses!

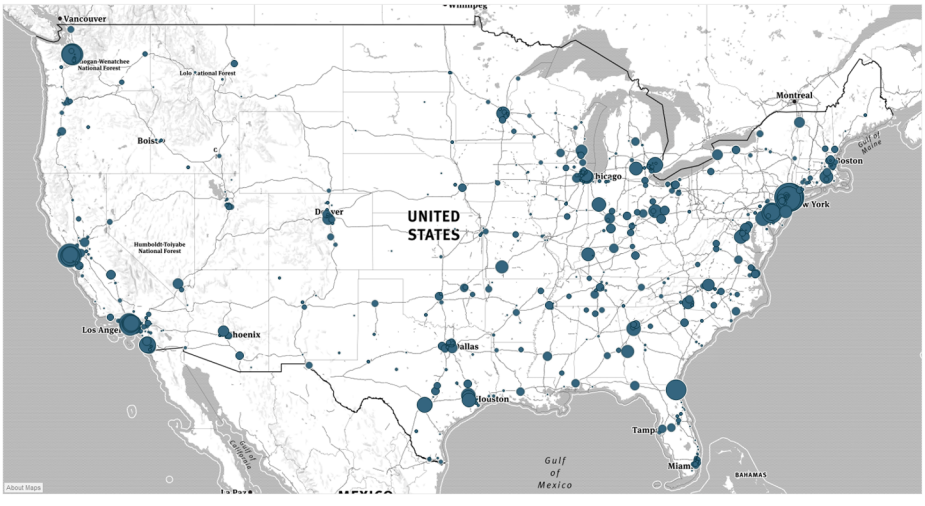
CHAPTER 30

How to Make a Symbol Map with Mapbox Integration

Maps, which were introduced with Tableau 4.0 in August 2008, are one of the most powerful visualization types available. The power of maps comes from their inherent ability to leverage schemas that your users have been building up for many years. In Chapter 98, I'll show how an image of a map helps you decode dozens of latitude/longitude pairs almost instantly.

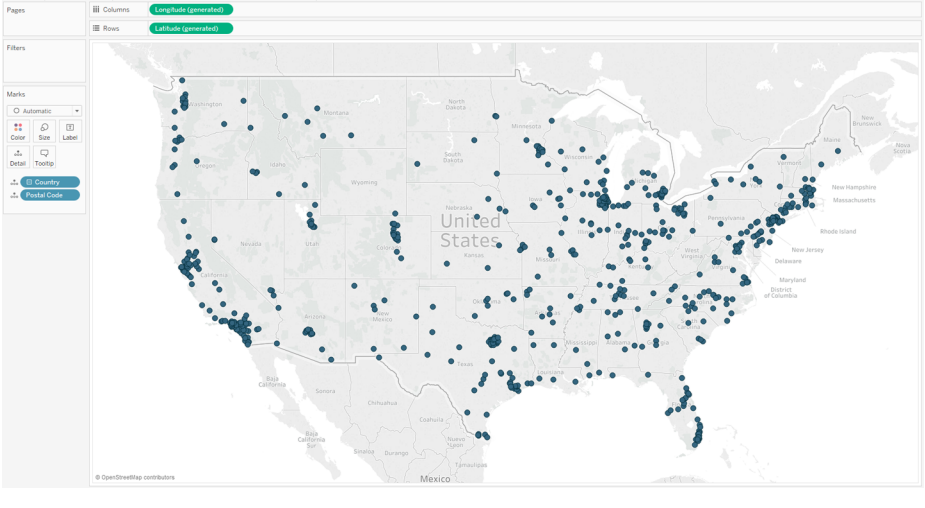
The next several chapters cover different map types you can create with Tableau and their purposes. The first shares how to harness this power by creating symbol maps and how to take your maps a step further by integrating Tableau with Mapbox maps.

First, let's take a look at how my final map example will look:



How to Make a Symbol Map in Tableau

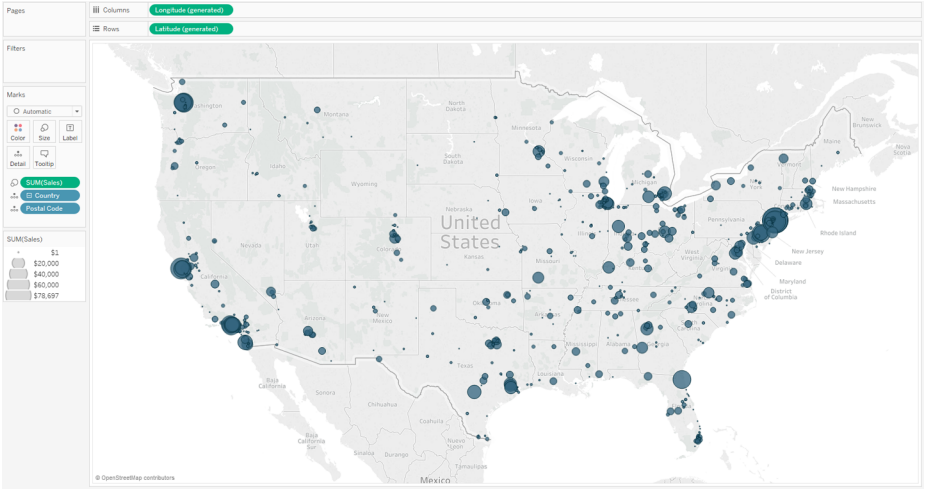
The easiest way to start a symbol map in Tableau is to double-click a geographic dimension from the Dimensions area of the Data pane on the left side of the interface. You know that Tableau recognizes your geographic fields as map-compatible if there is a globe icon next to the dimension. In my example, I am going to double-click Postal Code in the Sample – Superstore dataset to start the view:



Notice that Tableau put Longitude on the Columns Shelf and Latitude on the Rows Shelf, and each circle represents the intersection of each pair. At this point, I techni-

cally already have a symbol map, but there are several ways I can add value to the visualization. You may want to change the "symbol" from the default circle to a square or other shape. Perhaps you want to size and/or color the symbols by a measure such as sales by placing measures on the appropriate Marks Cards.

I personally like the default circles because you can add a border and they look nice with transparency (which is controlled on the Color Marks Card), so I am going to stick with that. I will change the color to something that pops more by clicking the Color Marks Card and size the bubbles by sales by placing that measure on the Size Marks Card. At this point, my view looks like this:



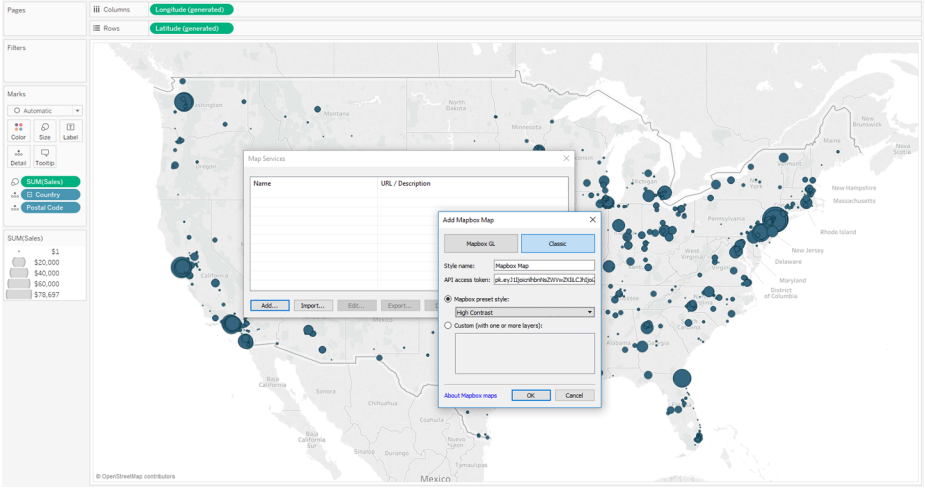
This is a solid symbol map at this point, but you can also add additional data layers or choose from three map styles by navigating to Map → Map Layers.

How to Add Mapbox Maps

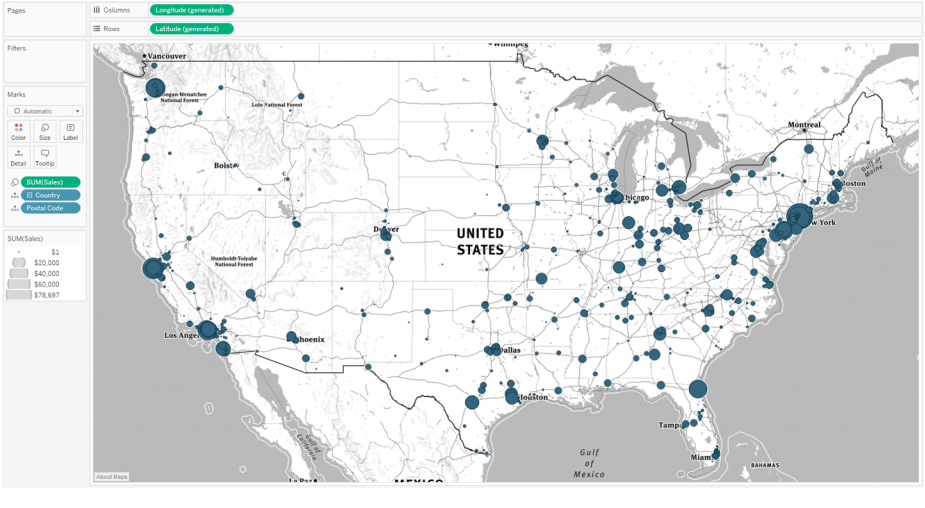
Tableau provides three map styles out of the box, but starting in Tableau version 9.2, it is very easy to integrate your Tableau workbooks with Mapbox, a custom map designing service, to access 14 additional map styles. Mapbox is a free service as long as you get fewer than 50,000 map impressions per month. Once you grow past that, there are paid plans available. To access the new styles, follow these simple steps:

1. Go to mapbox.com.
2. Sign up via the button in the upper-right corner.
3. Once you're signed up, navigate to the Home tab, and copy your access token on the right side.

4. From within your map view in Tableau, navigate to Map → Background Maps → and click Map Services.
5. From the new dialog box, click Add and select Mapbox Maps.
6. This is where you give your custom map a name, paste your access token, and choose one of the styles:



For my example, I chose the high contrast Mapbox style, but there are 13 additional styles to choose from! After clicking OK and closing the dialog box, I am left with a nice-looking Tableau symbol map with an integrated Mapbox style:

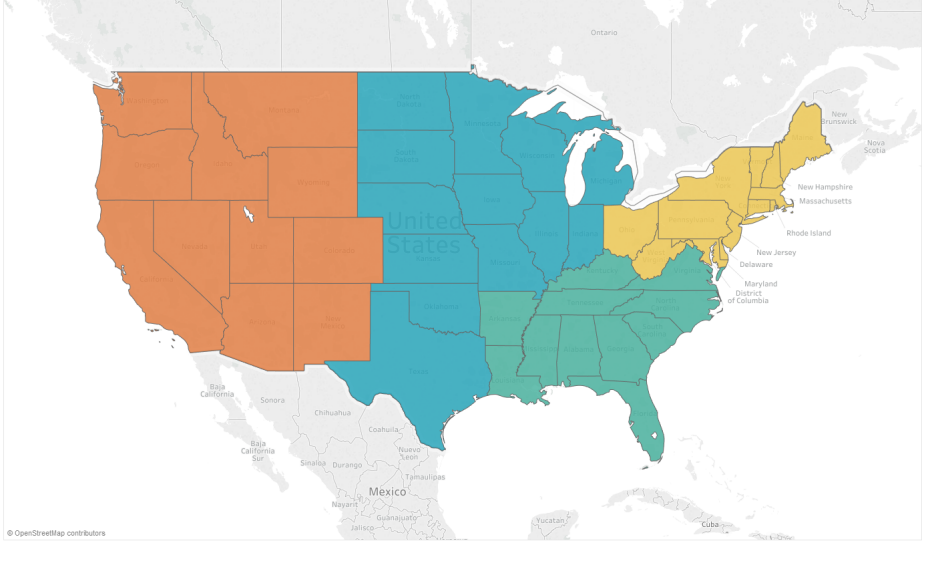


CHAPTER 31

How to Make a Filled Map

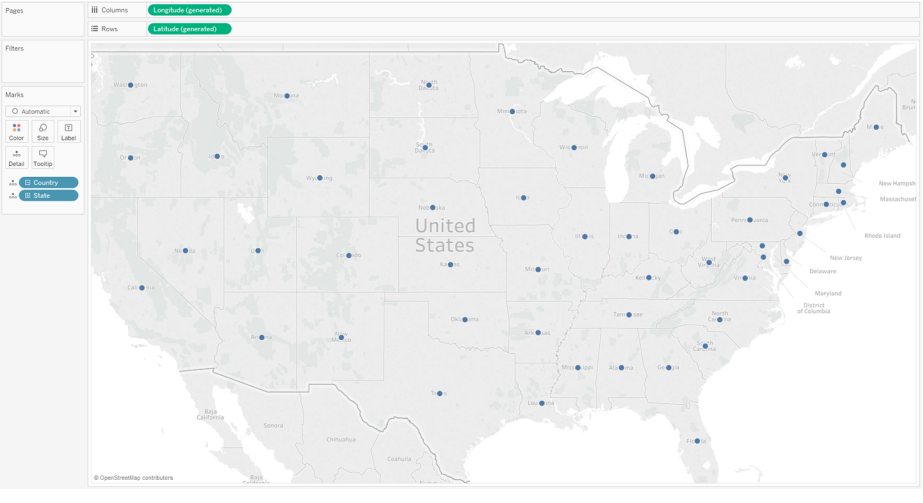
In the previous chapter, I shared how to create symbol maps in Tableau with custom Mapbox maps, and how mapping in Tableau leverages schemas that you are familiar with to help you rapidly decode latitude and longitude pairs. Maps in Tableau can be thought of as scatter plots, with a background image of a map to help you orient yourself almost instantly to the geographic territories being displayed.

Filled maps in Tableau are similar to symbol maps, but they include many more data points. While a symbol map draws a symbol at the intersection of each latitude and longitude pair, filled maps draw a polygon around the entire border. Here's one example of a filled map colored by region in Tableau:



How to Make Filled Maps in Tableau

Filled maps are one of the easier chart types to create in Tableau using Show Me. Just click a geographic dimension (identified by a globe icon) from the Dimensions area of the Data pane and choose "maps" under Show Me. That being said, in the spirit of learning, we will build a filled map from scratch. To create a filled map manually, I will first double-click the geographic dimension for State in the Sample – Superstore dataset:

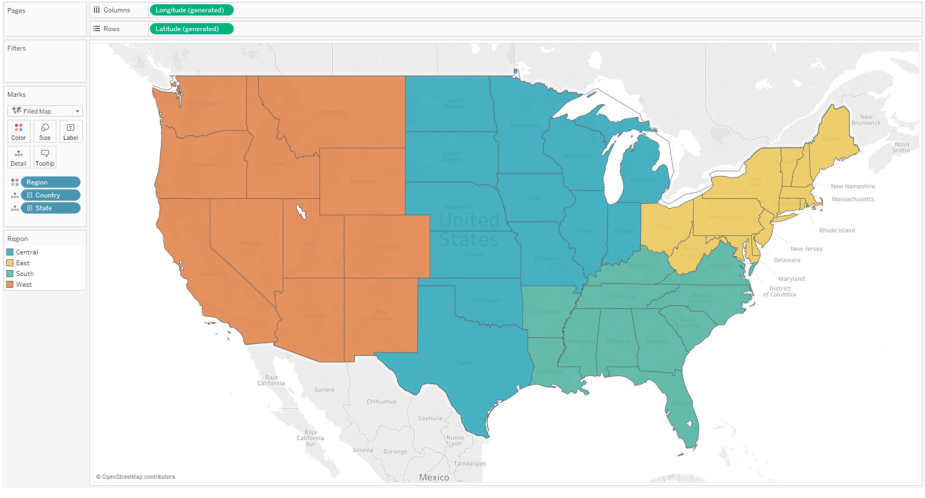


By default, Tableau generates a symbol map, placing a circle at the intersection of Longitude and Latitude for each state. In keeping with the "looking under the hood" theme of not relying on Show Me, take a look at the location of each dimension and measure. First, Longitude is on the Columns Shelf, which can also be thought of as the x-axis. Conversely, Latitude is on the Rows Shelf, or the y-axis. On the Marks Shelf, we can see that State is the most granular level of detail in the view. Even without seeing the map in the view, we should be able to guess what Tableau will display just by seeing the geographic fields on the Columns Shelf and Rows Shelf, the level of detail on the Marks Shelf, and the mark type of Automatic (which is Circle by default).

In order to change this from a symbol map to a filled map, change the mark type from Automatic to Map. By selecting this special mark type in Tableau, you will see that the single circles on each state have been converted to nice, smooth polygons that trace the entire border of each state.

From here, you can encode the filled map by color by placing a field on the Color Marks Card. Perhaps you want to color the territories by a measure such as Sales or Profit. You can also color the marks by a dimension, as is the case in this example,

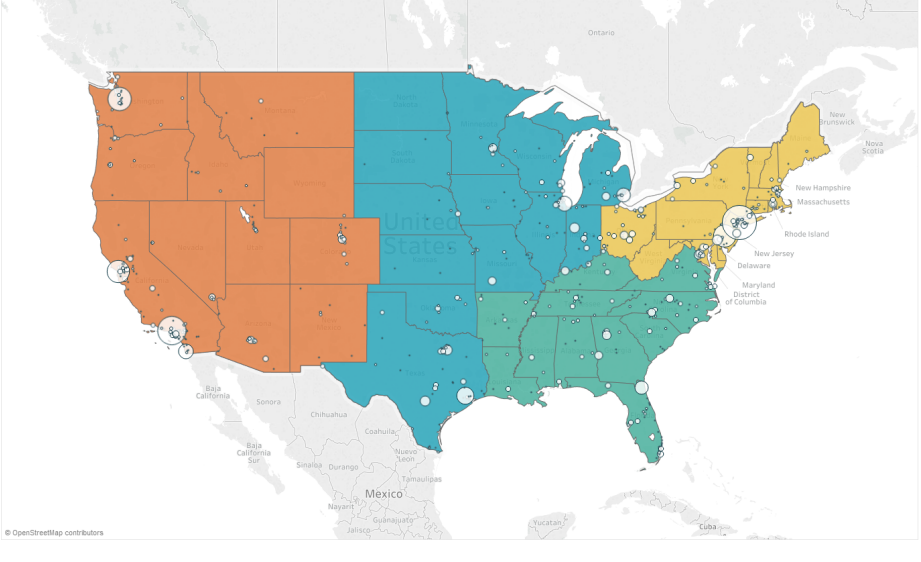
where the states are colored by the Region dimension. The map view ends up looking like this:



CHAPTER 32

How to Make a Dual-Axis Map

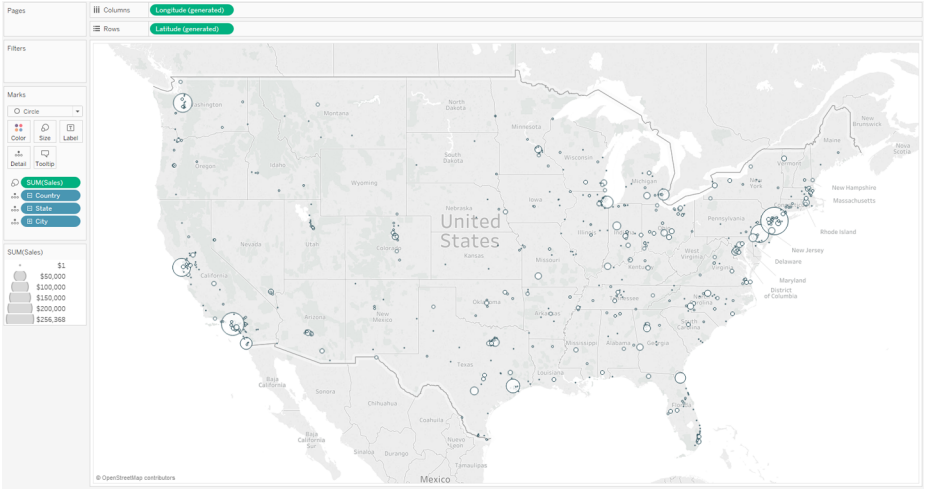
In the previous two chapters, you've read how to make a symbol map with Mapbox maps and how to make a filled map in Tableau. This chapter provides a step-by-step tutorial on how to combine the two into a single, dual-axis map. The final view in this example will display sales by city as a symbol map and US States colored by region as a filled map:



This type of map is useful any time you want to combine two maps into one to help provide additional context that would not be available when showing only one map or the other. One of the best real-life use cases I have seen for a dual-axis map in

Tableau was when an analytics partner wanted to see how the sales of their product by territory compared to the brick and mortar locations of a competitor.

To create a dual-axis map in Tableau, start by creating one of the two maps that you want to combine. I will start this example by creating a symbol map for sales by city:



Once the first map is created, you can create a duplicate of that map in one of two ways:

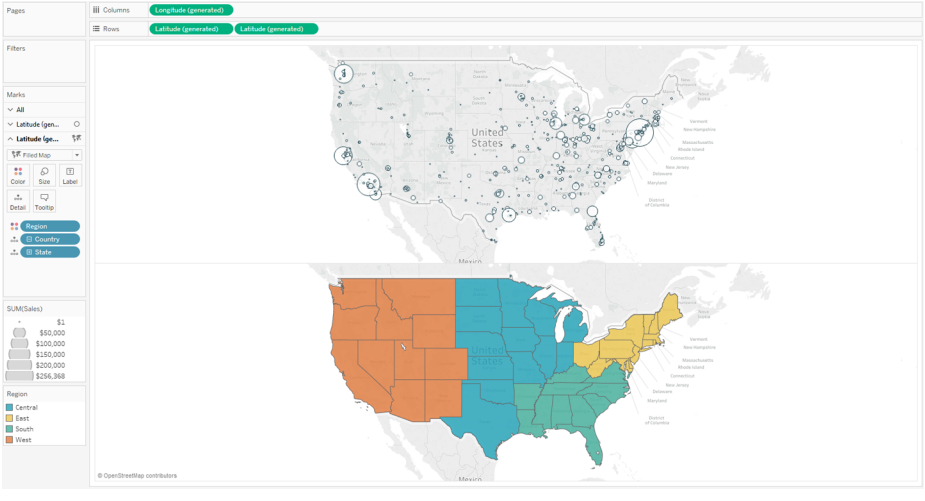
- While holding the Control key, left-click and drag the Longitude field on the Columns Shelf or the Latitude field on the Rows Shelf right next to itself. This is a shortcut for duplicating fields that works with any blue or green pill in Tableau.
- Drag Longitude from the Measures area of the Data pane to the second position on the Columns Shelf, or drag Latitude from the Measures area of the Data pane to the second position on the Rows Shelf.

Whichever method you select, the key is that by placing a second pill on the Columns or Rows Shelf, a second Marks Shelf for the geographic field is generated. Now that there are two separate Marks Shelves, one for each map, the maps can be edited independently. That means a symbol map can exist on the first Marks Shelf, and the second Marks Shelf can be changed to a filled map. In this example, I will leave the first map as is, but change the second map to be a filled map by region. To format the map for my example, follow these steps:

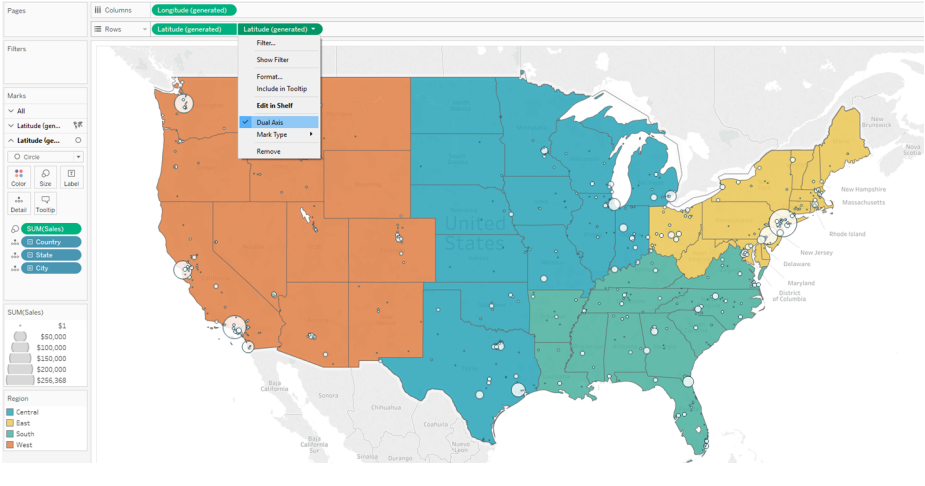
- Navigate to the second Marks Shelf by clicking it or clicking the pill associated with it.
- Remove the Sales measure.

- Remove the City dimension.
- Add the Region dimension from the Dimensions area of the Data pane to the Color Marks Card.

I've now got two distinct maps in my view:



To combine the maps, click the second geographic measure on the Rows or Columns Shelf (depending on whether you duplicated Longitude or Latitude) and click Dual-Axis. If you want to change the float order of the maps, drag the second geographic measure on the Rows or Columns Shelf to the first position. My final map looks like this:



CHAPTER 33

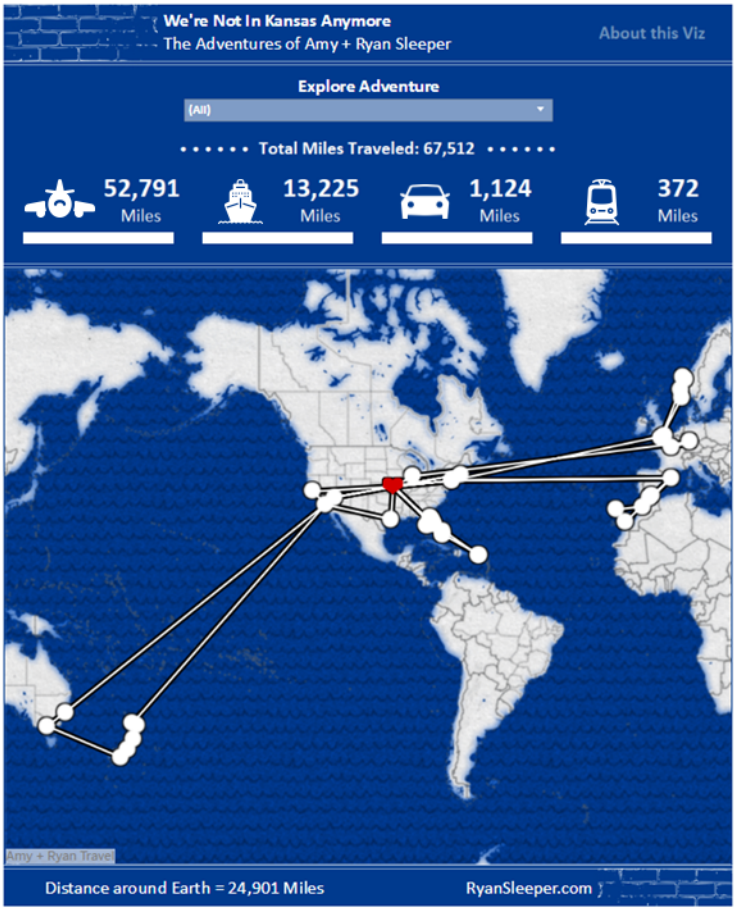
How to Map a Sequential Path

As with most features in Tableau, maps are flexible enough to meet many different analysis needs. One of the most popular uses of maps in Tableau that I have seen involves evaluating traffic through a certain travel hub, such as an airport or train station. These powerful visualizations are created using hub and spoke paths, with the airport or train station acting as the hub, and the paths to the destinations acting as the spokes. See the Tableau Path Mapping article on Knowledge Base if you are interested in creating this type of map.

This is a great application of path maps in Tableau, but I had a need to display multiple destinations in sequential order instead of from a single origin to multiple destinations. My wife and I make it a priority to travel as much as we can, and I wanted to create a visualization to document our travels between stops. This tutorial walks you through how to create sequential paths on a map in Tableau.

How to Map Paths in Tableau

Before we get started, let's take a look at the visualization that inspired this chapter:



Notice that within each trip, a path is displayed not only from our hometown of Kansas City to each destination, but from each stop in between. Here are the steps required to get this effect:

1. Prepare your data.

   Create your dataset so that each stop has a latitude, longitude, and path order. By looking up the latitude and longitude sets, you will have more flexibility when you create maps than you would if you use the automatically generated coordinates that Tableau applies to geographic fields. The path order is what tells Tableau how to connect the dots. Here's how my underlying data looks for one of our trips ("Stop" = Path Order):

Trip Name	Stop	Latitude	Longitude	City	Country
A. DeVos SBM Reunion in Orlando - October 2011	1	39.0997	-94.5783	Kansas City	United States
A. DeVos SBM Reunion in Orlando - October 2011	2	28.538	-81.379	Orlando	United States
A. DeVos SBM Reunion in Orlando - October 2011	3	29.8947	-81.3144	St. Augustine	United States
A. DeVos SBM Reunion in Orlando - October 2011	4	30.3369	-81.6614	Jacksonville	United States
A. DeVos SBM Reunion in Orlando - October 2011	5	28.538	-81.379	Orlando	United States
A. DeVos SBM Reunion in Orlando - October 2011	6	39.0997	-94.5783	Kansas City	United States

When you open this data for the first time in Tableau, your Stop field will be classified as a measure because it is quantitative. Make sure you change this field to a dimension by either dragging and dropping it from the Measures area of the Data pane to the Dimensions area of the Data pane, or by right-clicking the measure and choosing Convert to Dimension.

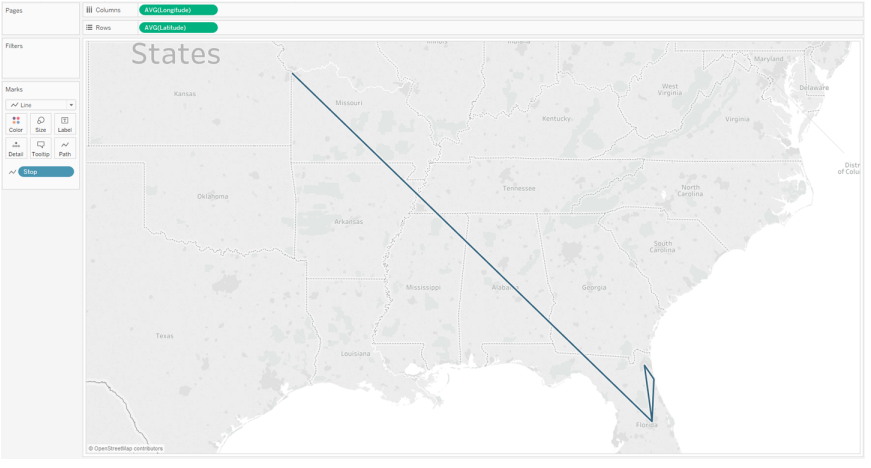
2. Start your map.

   Using the dataset from step 1, begin your map in Tableau by placing your Longitude field on the Columns Shelf and your Latitude field on the Rows Shelf. It is important in this step to use the coordinates that you added to your dataset instead of the generated latitude and longitude in Tableau. You can always tell the difference because generated fields in Tableau are italicized.

3. Change the map type.

   At this point, you should see a map with a single point. That single point represents the average for all of the coordinates in your data. Change the mark type from Automatic, which is set to circles by default, to Line. This change will reveal a new Marks Card called Path. The final step is to add the Stop (or path order) dimension to the Path Marks Card. This should connect the dots in the proper order, creating a sequential path.

   Using the sample dataset from before, your final product should look like this:



Remember, many Tableau Public authors allow you to download their workbooks by clicking in the lower-right corner of an embedded Tableau Public viz. This is a great way to reverse engineer techniques that you want to incorporate into your own work.

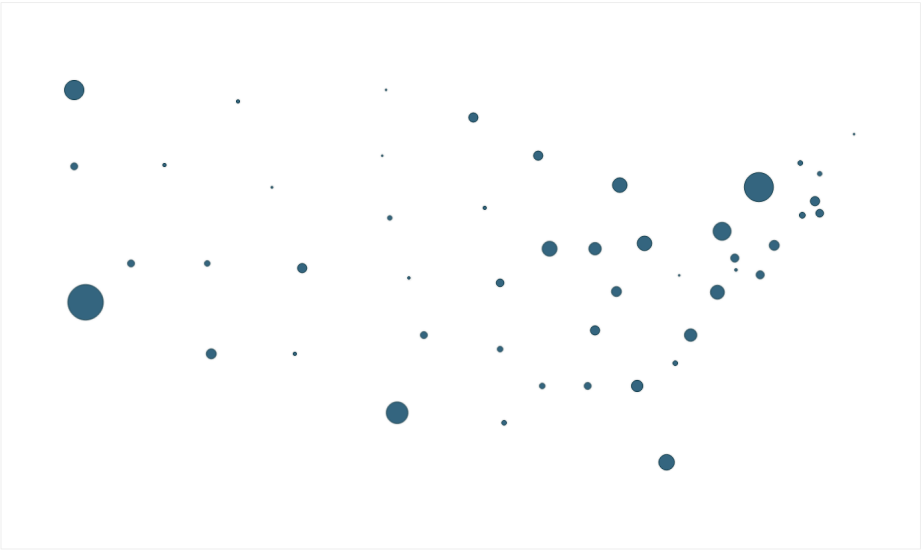
CHAPTER 34

How to Map Anything in Tableau

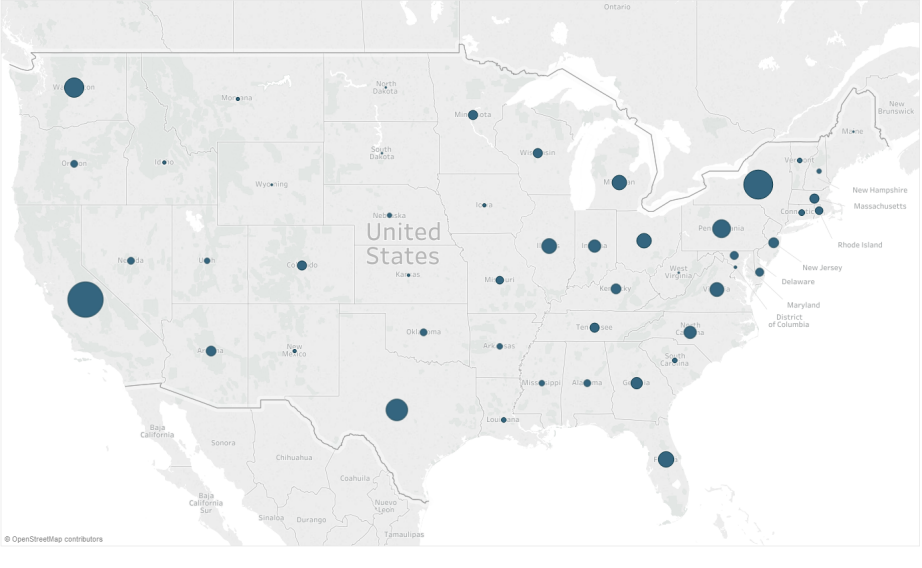
The out-of-the-box Tableau symbol and filled maps are some of the most powerful visualization types available in the software. As illustrated in Chapter 98, maps provide a means of decoding hundreds of latitude/longitude pairs almost instantly—all with a single image. Not only that, Tableau does a lot of the heavy lifting for us by providing the geographic coordinates of locations all over the world—this way we get to enjoy making maps instead of looking up longitudes and latitudes. Maps became even better in Tableau 9 with lasso and radial selections.

But wait—there's more! You can create custom maps in Tableau for just about anything. This tutorial will walk you through how to map *any* background image in Tableau.

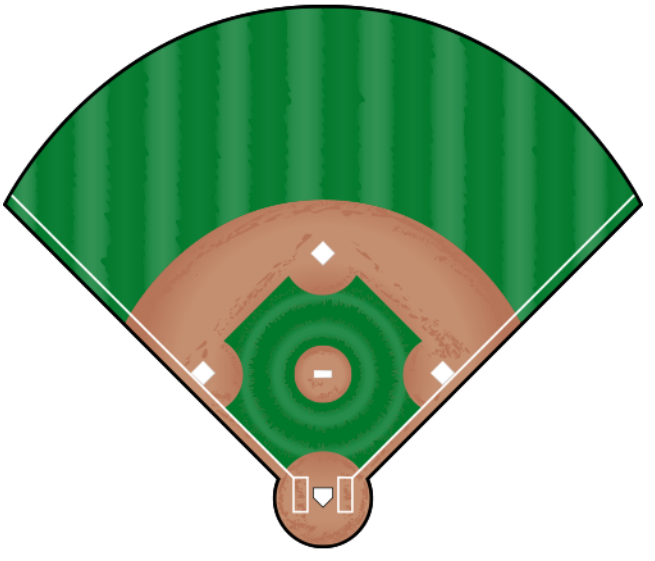
Maps in Tableau can be thought of as scatter plots with a background image. Tableau will plot a point at each combination of the latitude on the y-axis and longitude on the x-axis. To help illustrate the scatter plot concept, take a look at the following two images. First, the latitude and longitude plotted with no background image:



And here, the same data points plotted with a background image of the United States:



Due to the way maps function in Tableau, as long as you know the coordinates of a point on the vertical y-axis and horizontal x-axis, you can map anything you would like! This provides some unique possibilities. So let's build one of these. For this tutorial, I will use this image of a baseball diamond:



Building Custom Background Maps in Tableau

Let's imagine that we want to visualize different baseball statistics by player position. To create this visualization, we first need to plot the nine positions on the field.

1. Locate image and image dimensions.

   First we will need an image. If you want to follow along, download an image of a baseball diamond. For best results, pre-size the image in the desired dimensions and take note of the height and width of the image in pixels. The baseball diamond image just shown is 500 pixels wide by 500 pixels high.

2. Create dataset with fields for *x* and *y* coordinates.

   The next step is to create a dataset that has at least three fields:

   a. Each point you want to plot (in our case it's player position)

   b. *x* position

   c. *y* position.

      For the first row, just put an arbitrary name in the cell for the point you want to plot, such as Lookup. The *x* value will be the width of the image, and the *y* value will be the height of the image. At this point, my dataset looks like this (the player position coordinates still need to be looked up):

	A	B	C
1	Position	X	Y
2	Lookup	500	500
3	Pitcher		
4	Catcher		
5	First Base		
6	Second Base		
7	Shortstop		
8	Third Base		
9	Left Field		
10	Center Field		
11	Right Field		

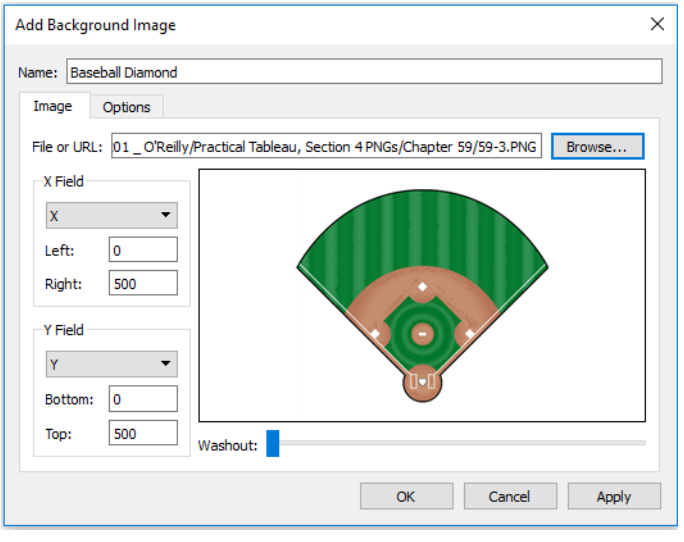
3. Add the background image.

   Start a new workbook in Tableau and connect to the data from step 2. Navigate to Map → Background Images, and click the data source. This will open a new dialog box where you can choose Add Image in the lower-left corner. This is where you can choose which image will act as the background. If you're following along with the baseball example, navigate to the image that you downloaded earlier.

   All that is left in this step is to put the maximum value for the *x* and *y* coordinates. If our image is 500 px wide by 500 px high, we would make the following entries:

   - X Position: 0 Left, 500 Right
   - Y Position: 0 Bottom, 500 Top

     Also be sure to change the Y Field from X to Y. At this point, your screen should look like this:



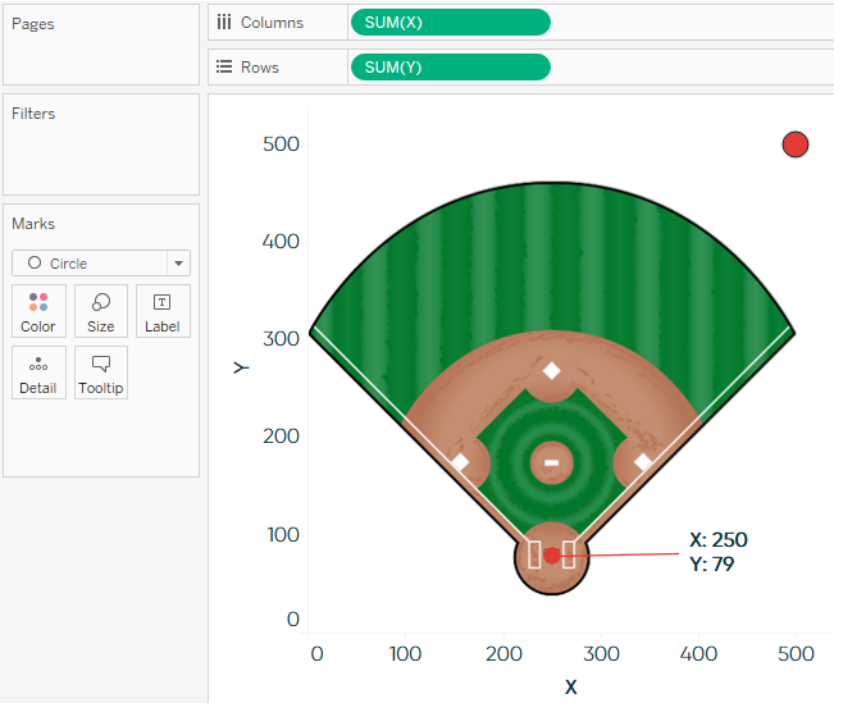
Click OK to apply your changes.

4. Look up *x* and *y* position for each coordinate.

   Start your new custom map by placing the *x* measure on Columns Shelf and the *y* measure on the Rows Shelf. Ensure both axes are fixed at the maximum X and Y values; which are 500 for both in our case. If you need to change an axis, simply right click the axis, choose Edit Axis, and fix the range at 0 to 500.

   In order to look up the *x* and *y* coordinates for each position player (or whatever you are plotting), right-click the view, hover over Annotate, and click Point. If the X and Y measures are the only fields on your view, the annotation will show you the X and Y values by default. This is what we want, but you may want to make the font larger so that it's easier to read.

   Now drag the end point of the annotation to each location you want to plot, and see how the X and Y values change. For best results, format the annotation so the line end is a circle; this way you can precisely drag the circle to the exact location you want plotted. Here is one example showing the coordinates for the Catcher:



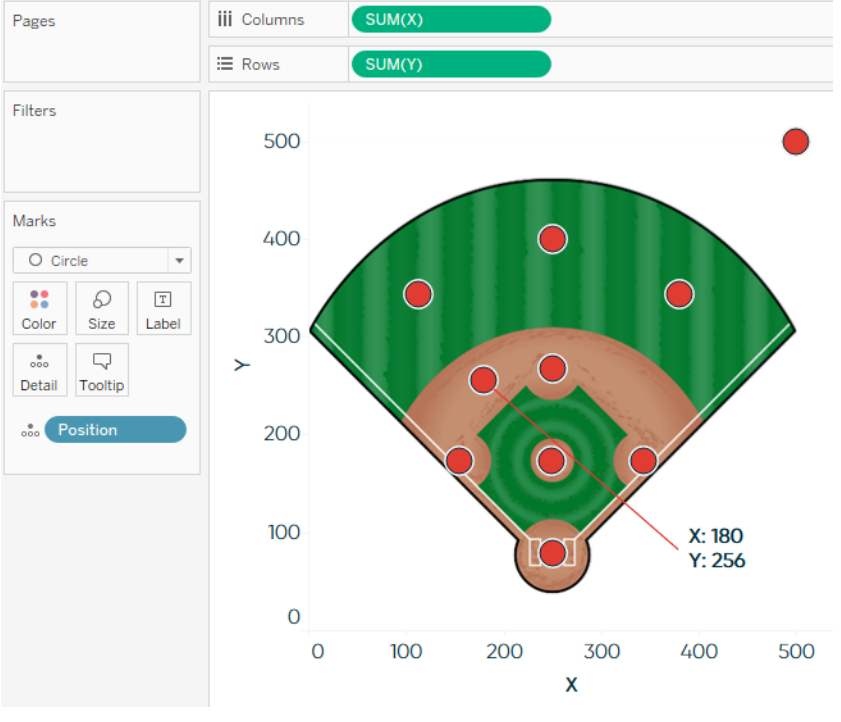
Drag the annotation to each point you want to plot, and record the coordinates in your dataset. Note that you don't need to record the decimals. Here is a look at my data with the *x* and *y* coordinates for each player position:

	A	B	C
1	Position	X	Y
2	Lookup	500	500
3	Pitcher	249	173
4	Catcher	250	79
5	First Base	344	173
6	Second Base	250	268
7	Shortstop	180	256
8	Third Base	155	173
9	Left Field	113	344
10	Center Field	250	400
11	Right Field	380	344

5. Refresh your map.

   After all coordinates have been recorded in your dataset, save your data and refresh the data source in Tableau.

   Drag the field you are plotting, such as position in the baseball example, to the Detail Marks Card. You should now see a circle at each position, like this:



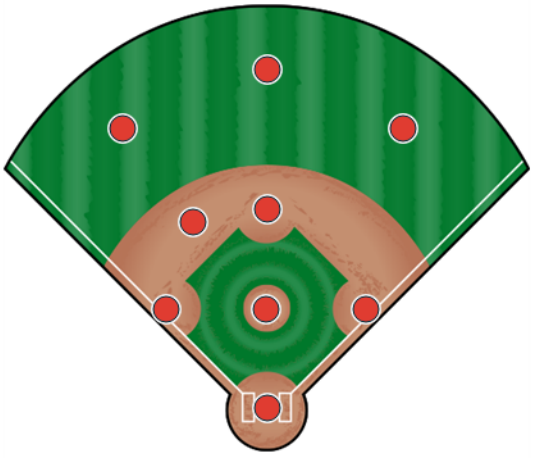
6. Finalize your map and/or add metrics.

   There are just a few items you can do to finalize your custom Tableau map:

   - Remove the annotation by right-clicking it and selecting Remove.
   - Hide the axes by right-clicking each one and deselecting Show Header.
   - Format your marks; my favorite is a filled circle with a custom color and white border.
   - Filter out the Lookup *x* and *y* coordinates. You will notice a mark in the upper-right corner of your new custom map. You can filter this out by either right-clicking it and choosing Exclude, or for better longterm results, remove this row from your underlying data and refresh your view.

- Optional: You can now add measures to your underlying data for each position to visualize different metrics on your customized view (i.e., size the circles by fielding %).

The final custom Tableau map will look like this:



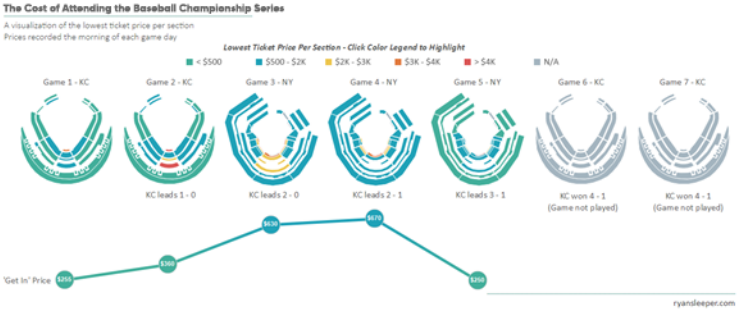
CHAPTER 35

How to Make Custom Polygon Maps

In the last two tutorials about mapping, we have discussed path maps (in Chapter 33) and custom symbol maps (in Chapter 34). There is a third type of map in Tableau called a polygon map that allows you to map custom shapes. These types of visualizations are what we're making anytime we're making a filled map. Imagine a map of sales by US state where each state is colored by their respective sales volumes, something we've done before. With these filled maps, Tableau is essentially looking up the latitude and longitude coordinates all the way around the border of each state, and plotting a custom polygon for each territory. With custom polygons, we're not limited to a prepared set of polygons like state borders—we can define shapes for anything we can imagine from custom geographic dimensions, to your favorite theme park, to your local dog park, to grocery store shelves, or anything else!

This tutorial will use one of my most asked-about visualizations, The Cost of Attending the Baseball Championship Series, to illustrate how you can create custom polygon maps with any shapes—including stadiums!

First, let's take a look at the screenshot of the original viz:



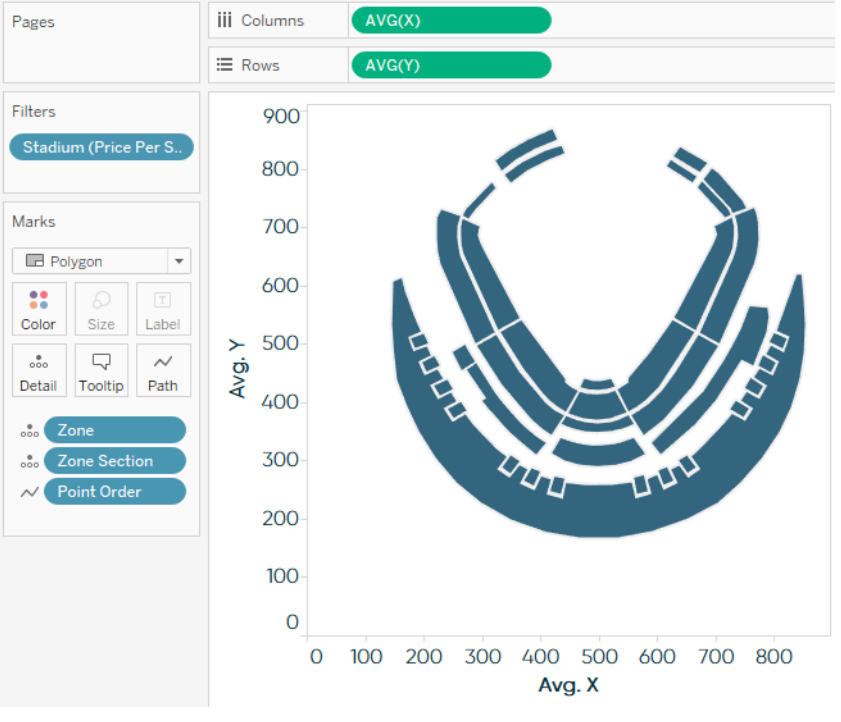
Polygon maps in Tableau are created by looking up the coordinates of the shape you want to draw, and then connecting the dots by drawing a path between them. I encourage you to read the first two chapters in this series mentioned in the opening paragraph if you are not familiar with these concepts. The main difference between a polygon map and a path map is that you "close the loop" around the dots by choosing a polygon mark type instead of a line mark type. Here are the steps for making custom polygon maps in Tableau:

1. Find an image of what you want to draw in Tableau. Being a visualization about the 2015 Baseball Championship Series, I used Kauffman Stadium in Kansas City and Citi Field in New York.
2. Follow the steps in Chapter 34 to set up your map in Tableau and record the coordinates for your shapes.
3. Similar to mapping a sequential path (see Chapter 33), each combination of coordinates should be given a *point order*. This is a field in your underlying data that tells Tableau what order the dots are connected. At this point, my underlying data for one section of Kauffman Stadium looks like this:

	A	B	C	D	E
1	Stadium	Zone	Point Order	X	Y
2	Blank	None		1000	1000
3	Kauffman Stadium	State Farm Neighborhood	1	268	720
4	Kauffman Stadium	State Farm Neighborhood	2	273	731
5	Kauffman Stadium	State Farm Neighborhood	3	317	776
6	Kauffman Stadium	State Farm Neighborhood	4	324	767
7	Kauffman Stadium	State Farm Neighborhood	5	289	731
8	Kauffman Stadium	State Farm Neighborhood	6	277	715

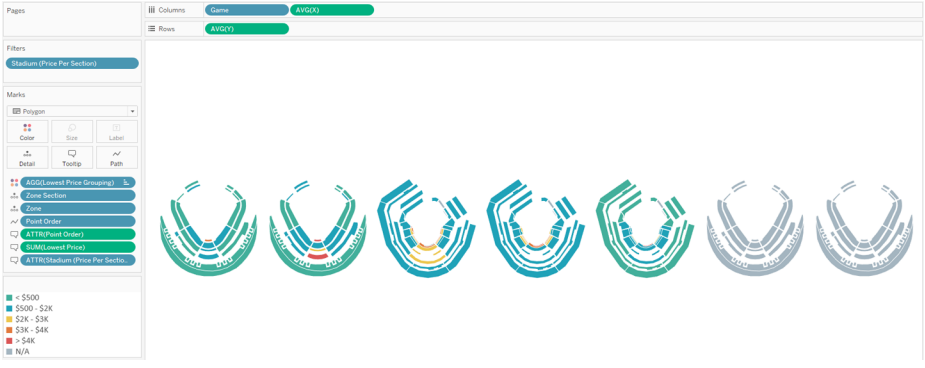
4. Once you have the *x* and *y* coordinates for each point of each shape you want to plot, we are ready to build the polygon map in Tableau. To start, put your *x* measure on the Columns Shelf and *y* measure on the Rows Shelf; both with an aggregation of AVG.
5. Change the mark type from Automatic to Polygon, and place your Point Order dimension on the Path Marks Card.
6. Place the Section dimension (and/or the dimension with the most granular level of detail) on the Detail Marks Card. Note that my maps are actually plotted by Zone and Zone Section (which was required when the same zone name existed in two places in the stadium) so I had to add both of these to the Detail Marks Card. If your granularity isn't reflected on both sides of your polygon, you would just

place the most granular dimension on the Detail Marks Card. At this point, my map looks like this:



7. This can now be made into a small multiples view by *slicing* the map coordinate measures by a dimension. In my case, I was looking at the maps for each game of the series, so I put "Game" on the Columns Shelf to create a column for each game with a stadium in each cell.

From here, you can hide the axes and encode each section by putting a measure on the Color Marks Card. Here's what my final product looks like under the hood:



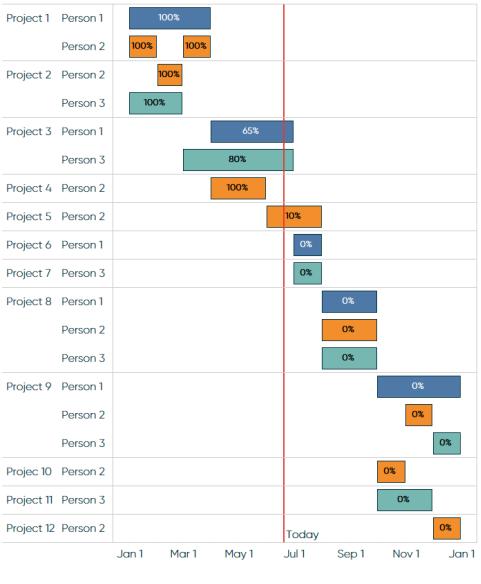
I admit that this map creation process requires painstaking attention to detail, but this step only has to be done once. The map coordinates can then be used over and over in future analyses and the results are great!

The approach provided in this chapter requires no additional software or resources (outside of Excel and some elbow grease). I would be remiss if I did not mention that there are other paid programs including Adobe Illustrator and Alteryx that can help make this process more efficient.

CHAPTER 36

How to Make a Gantt Chart

Gantt charts are traditionally used for visualizing project schedules. They are effective for illustrating task durations and dependencies in context of the larger business operation. As with several visualizations that can be built with Tableau, the fact that "Gantt Bar" is a mark type option unlocks some flexibility that allows you to use this chart type in many other ways. To start demonstrating the usefulness of the Gantt Bar mark type, we will build a Gantt chart with a traditional application of illustrating project schedules. The final product will look like this:



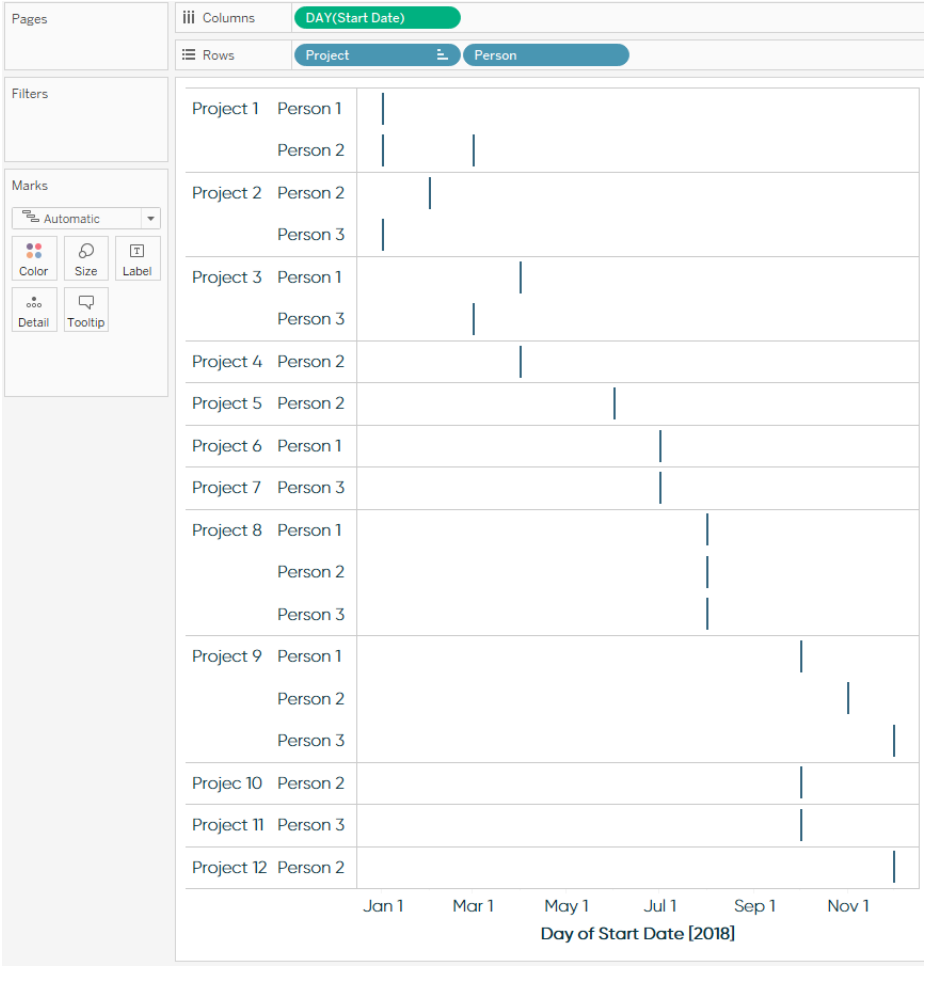
This is one of the few examples that cannot be illustrated using the Sample – Superstore data source, so I will be using the following dataset:

	A	B	C	D	E
1	Person	Project	Start Date	End Date	Percent Complete
2	Person 1	Project 1	1/1/2018	3/31/2018	1
3	Person 2	Project 1	1/1/2018	1/31/2018	1
4	Person 3	Project 2	1/1/2018	2/28/2018	1
5	Person 2	Project 2	2/1/2018	2/28/2018	1
6	Person 2	Project 1	3/1/2018	3/31/2018	1
7	Person 3	Project 3	3/1/2018	6/30/2018	0.8
8	Person 1	Project 3	4/1/2018	6/30/2018	0.65
9	Person 2	Project 4	4/1/2018	5/31/2018	1
10	Person 2	Project 5	6/1/2018	7/31/2018	0.1
11	Person 1	Project 6	7/1/2018	7/31/2018	0
12	Person 3	Project 7	7/1/2018	7/31/2018	0
13	Person 1	Project 8	8/1/2018	9/30/2018	0
14	Person 2	Project 8	8/1/2018	9/30/2018	0
15	Person 3	Project 8	8/1/2018	9/30/2018	0
16	Person 1	Project 9	10/1/2018	12/31/2018	0
17	Person 2	Project 10	10/1/2018	10/31/2018	0
18	Person 3	Project 11	10/1/2018	11/30/2018	0
19	Person 2	Project 9	11/1/2018	11/30/2018	0
20	Person 3	Project 9	12/1/2018	12/31/2018	0
21	Person 2	Project 12	12/1/2018	12/31/2018	0

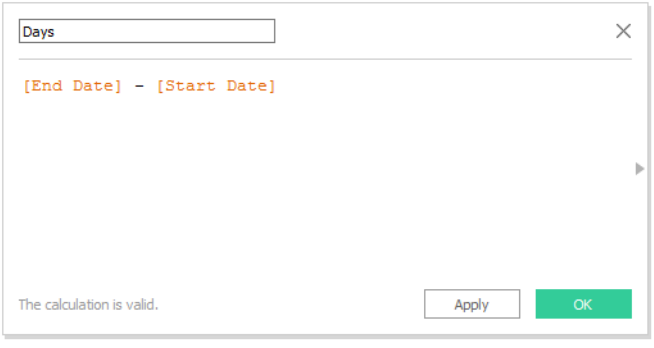
For the purposes of this illustration, let's pretend that we are a manager with three employees and we are plotting out their schedules across twelve projects for the year. We have created the schedules in Excel, but want to leverage data visualization to (a) get a better sense of how our projects and employees line up, (b) track progress to completion for each project, and (c) provide a visual schedule for our employees to use. All three of these features can be provided through a Gantt chart.

Gantt charts are created with one date, one or more dimensions, and zero to two measures. The date provides the axis; the dimensions provide the breakdowns we want to visualize and/or the encoding of the Gantt bars; the measures create the length of the Gantt bars and/or their encoding.

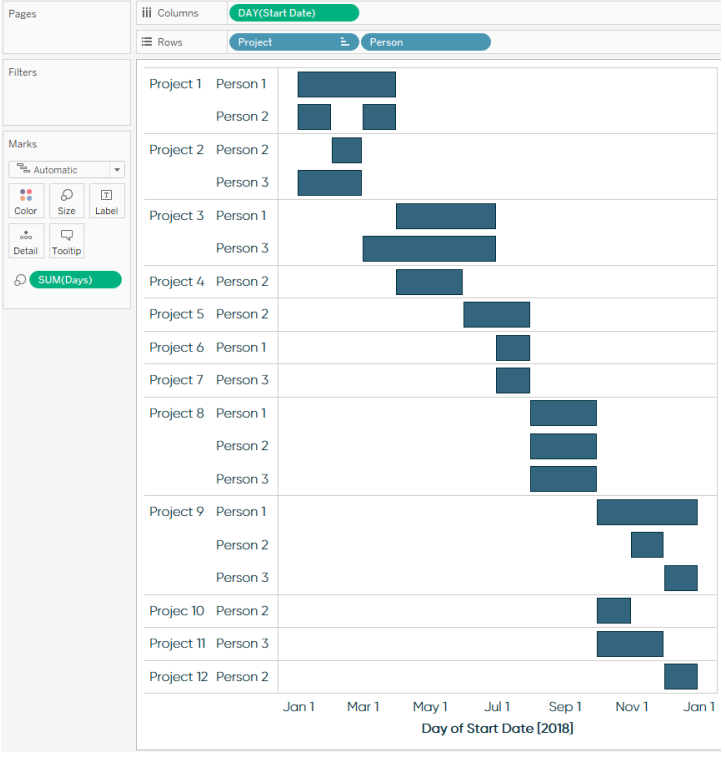
For this Gantt chart, I would like to have a continuous axis running left to right, so I will put the Date dimension (being used as a continuous field) on the Columns Shelf. Looking at the final product shown before, there are rows for each combination of Project and Person, which tells us those breakdowns reside on the Rows Shelf. Gantt Bar is the default mark type in Tableau for this combination of dimensions and measures, so placing the fields on the view results in this foundation of a Gantt chart:



What Tableau has done is place a Gantt mark at the start date of each combination of Project and Person. To extend the Gantt bars to illustrate task duration, we need to size each mark by the number of days in each respective project/person combination. You may have a field for duration in the underlying dataset, but it can also be created in Tableau with a calculated field. In this case, duration simply equals [End Date] – [Start Date]:



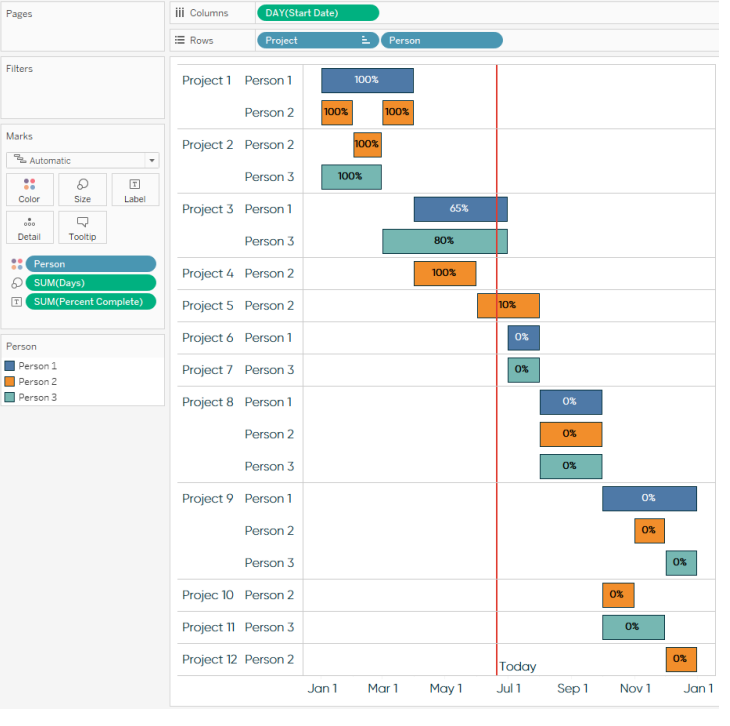
Placing the duration on the Size Marks Card will extend the Gantt marks so that project/person combinations with longer durations will be longer bars and project/person combinations with shorter durations will be shorter bars:



At this point, we have a usable Gantt chart, but there is still room to add a lot of value in Tableau through formatting, encoding, and reference lines. This step is flexible based on your own requirements, but as just one example, I have done the following:

- Colored the bars by employee by adding the Person dimension to the Color Marks Card. One of my goals was to provide a visual schedule to my team and by doing this, they can quickly view their own schedule and even highlight themselves in Tableau by clicking their name on the color legend.
- Added a reference line for "Today" (pretending it's June 20, 2018) to show each project in context of today's date.
- Added percent complete to the Label Marks Card to help determine if we are ahead or behind pace for each project. This number was hardcoded in my underlying dataset but in the real world it may also come from a time-tracking system. Another good use of color would have been to color the Gantt bars by progress to completion.

Here is my final Gantt chart after making these changes and cleaning up the formatting:

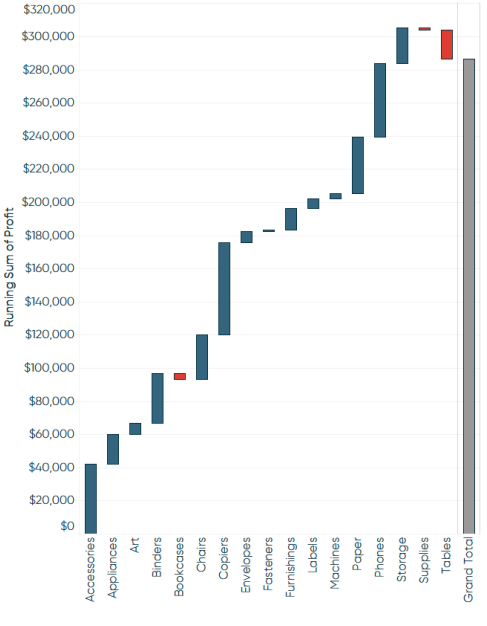


In the next chapter, I will illustrate one special use of the Gantt mark type: waterfall charts.

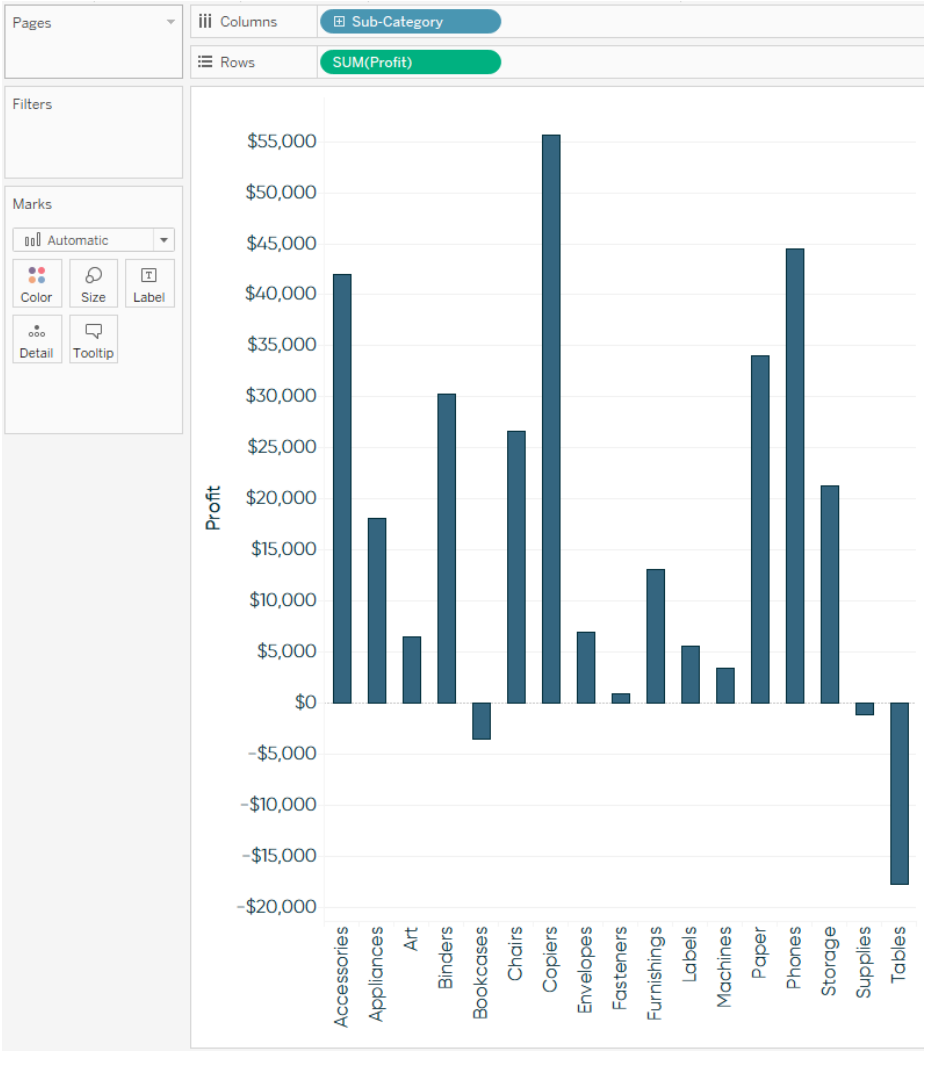
CHAPTER 37

How to Make a Waterfall Chart

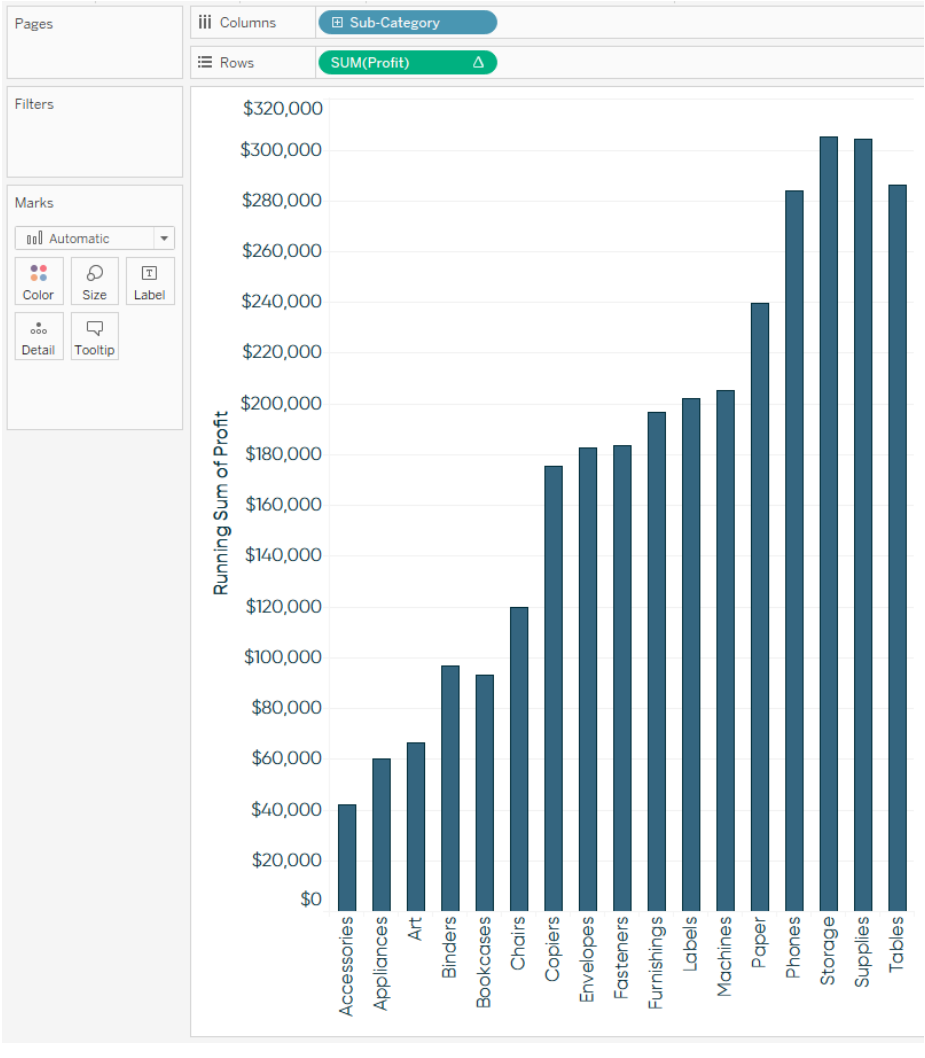
This chapter shares how to make a waterfall chart in Tableau—a visualization that helps understand how positive and negative values of dimension members are contributing to a cumulative total. What makes waterfall charts different from a simple running total calculation is that they illustrate how each dimension member with a positive value adds to a running total and each dimension member with a negative value detracts from a running total. For this tutorial, we will build the following waterfall chart in Tableau, which visualizes how each Sub-Category in the Sample – Superstore dataset is contributing to total profit:



To start a waterfall chart in Tableau, create a vertical bar chart showing the measure, Profit, by the dimension, Sub-Category:

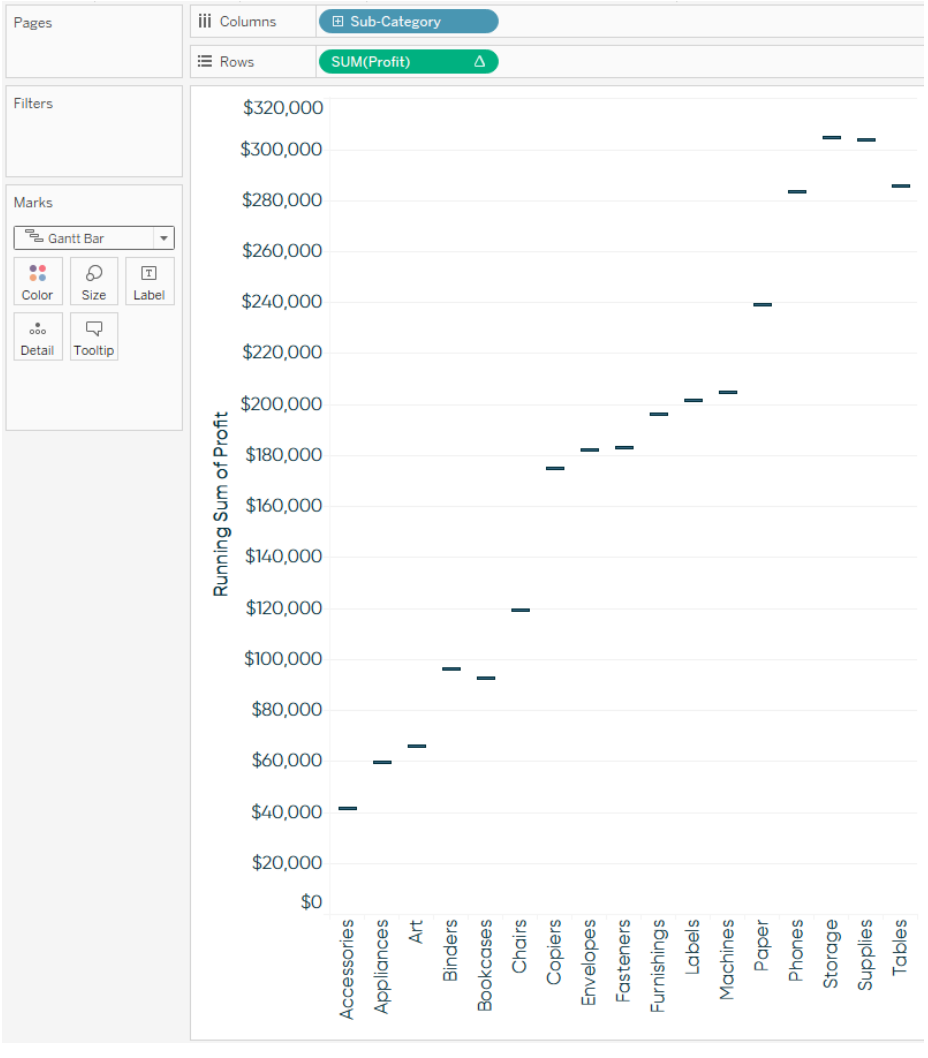


Next, add a table calculation to the Profit measure so that it calculates a "Running total" on Table (Across). For a refresher on table calculations, see Chapter 13. After adding the quick table calculation, the view looks like this:

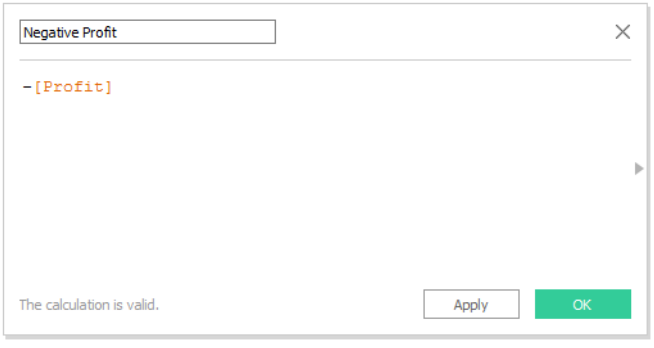


As mentioned in the introduction, while at this point we see how the running total has accumulated across our different sub-categories, it is not easy to determine the positive or negative contribution of each individual dimension member. To make this easier, we will convert this bar chart showing running total to a waterfall chart with a couple of additional steps.

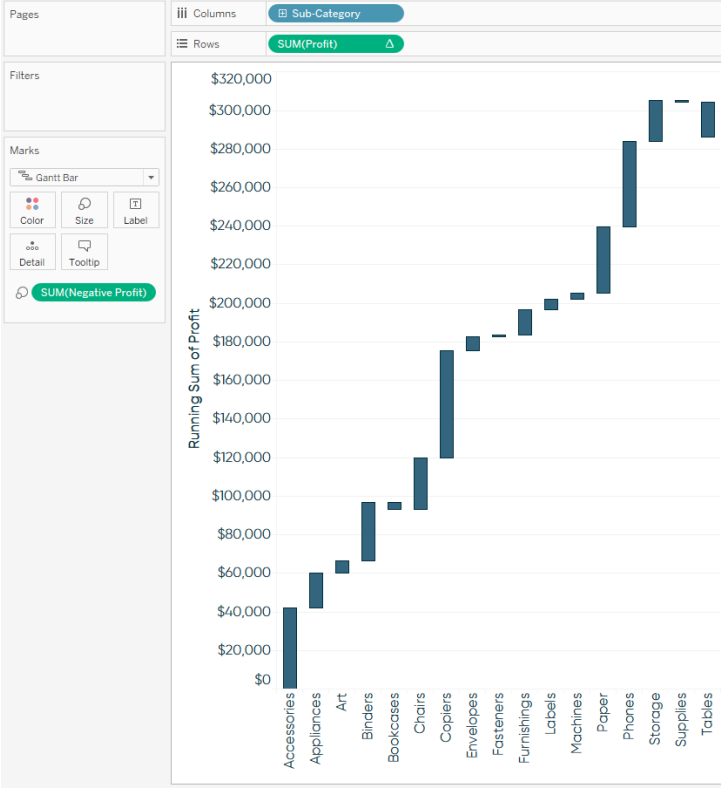
First, change the mark type from Automatic, which is currently Bar, to the Gantt Bar mark type:



To get this view to look like the first image in this chapter, we need the tops or bottoms of each bar to line up at the same points on the y-axis. To accomplish this, we have to size the Gantt bars by something in order to extend them. While your first instinct may be to size the Gantt bars by the Profit measure or even the Profit measure on the view that includes a table calculation for running total, there is a trick involved with this step to get the desired effect. In order to get the Gantt bars for each dimension member to properly line up, you first have to create a new calculated field that takes the measure in the waterfall chart multiplied by negative one. This example is using the Profit measure, so I will create a new calculated field that equals –[Profit]:



Once this calculated field has been created, this is the measure that you drag to the Size Marks Card to create the waterfall effect:



In Chapter 13, I mentioned that double-clicking a measure on the view gives you a kind of "x-ray vision" showing the underlying syntax of the measure. This technique can also be used to change a measure in the flow of your analysis. Another way to convert the Profit measure to negative is to drag SUM([Profit]) to the Size Marks Card, double-click the Profit pill once it is on the view, type a negative sign (-) at the beginning, and hit Enter.

At this point, we have an effective waterfall chart, but there are a few things I did to polish the final product as shown at the beginning of this chapter:

- Cleaned up the axis formatting.
- Colored the Gantt bars by Profit by dragging the Profit measure to the Color Marks Card; this created the blue and red color coding, which represents positive and negative values, respectively.
- Added a total to the far right side of the visualization by navigating to Analysis → Totals in the top navigation and choosing Show Row Grand Totals.

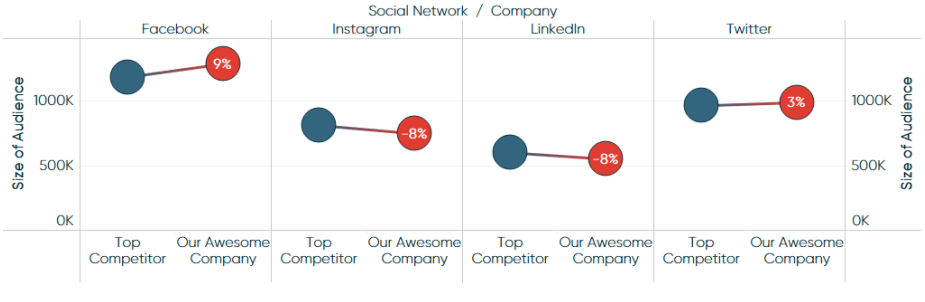
Lastly, you may choose to sort the dimension members by the sequence in which they were introduced or their values by ascending or descending order. As with many uses of Tableau, there is a great deal of inherent flexibility. These types of choices will depend on your analysis, business requirements, and business questions. Now that you know how waterfall charts are constructed in Tableau, experiment with the sort order of the dimension members to get the visualization that works best for you.

CHAPTER 38

How to Make Dual-Axis Slope Graphs

Slope graphs, or essentially line graphs between two points, are one of my favorite Tableau charts when my analysis requires a comparison between two data points. They work so well, in fact, that they are the one chart I will use to connect lines between discrete categorical variables. In almost every scenario, lines should only be used to connect points in time, but with slope graphs, I am OK drawing a line between "Thing 1" and "Thing 2," as well as "Time 1" and "Time 2." I like the following approach so much because you can easily view changes of individual dimension members (i.e., sub-categories) in context of each other, and the dual-axis mark provides extra real estate to share additional context in your analysis.

This chapter shares not only how to make slope graphs in Tableau, but how to enhance them by leveraging a second axis to provide additional context. By the end of the chapter, we will have re-created this dual-axis slope graph:



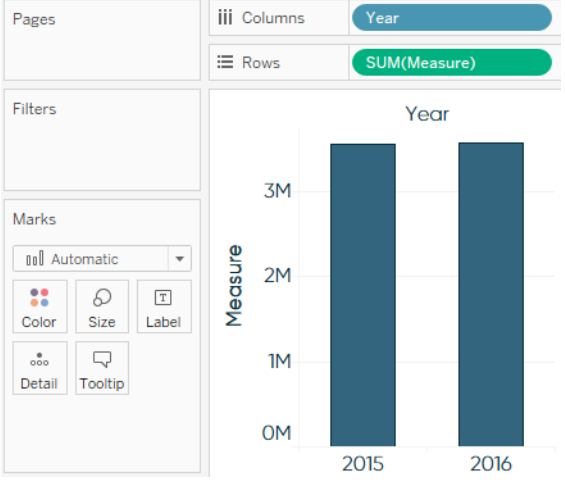
How to Make Slope Graphs in Tableau

First, let's knock out the traditional slope graphs using this data:

	A	B	C
1	Year	Category	Measure
2	2016	Category 1	1289632
3	2015	Category 1	1186461
4	2016	Category 2	989511
5	2015	Category 2	965035
6	2016	Category 3	551117
7	2015	Category 3	600718
8	2016	Category 4	748338
9	2015	Category 4	812562

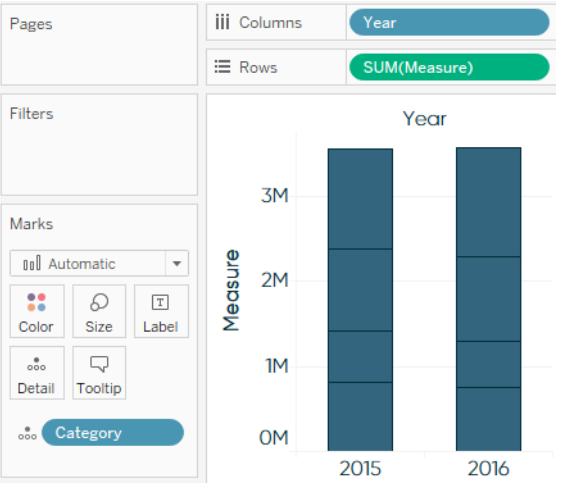
1. Create a bar chart.

   Create a bar chart with your two-point time comparison on the Columns Shelf as a discrete dimension and your measure on the Rows Shelf:



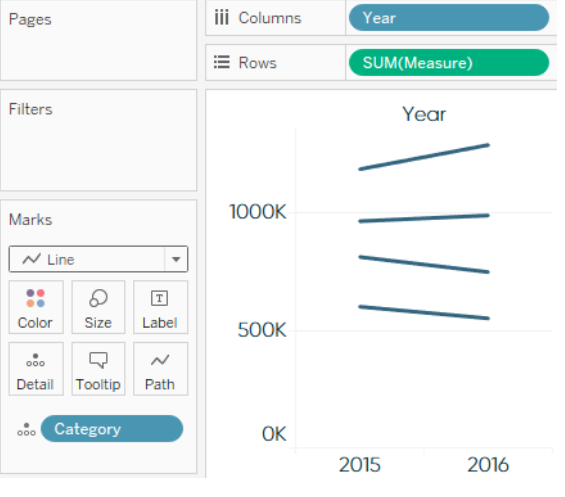
2. Add dimension to level of detail.

   Add the dimension you are wanting to compare to the Detail Marks Card. This will eventually create one line per categorical variable:



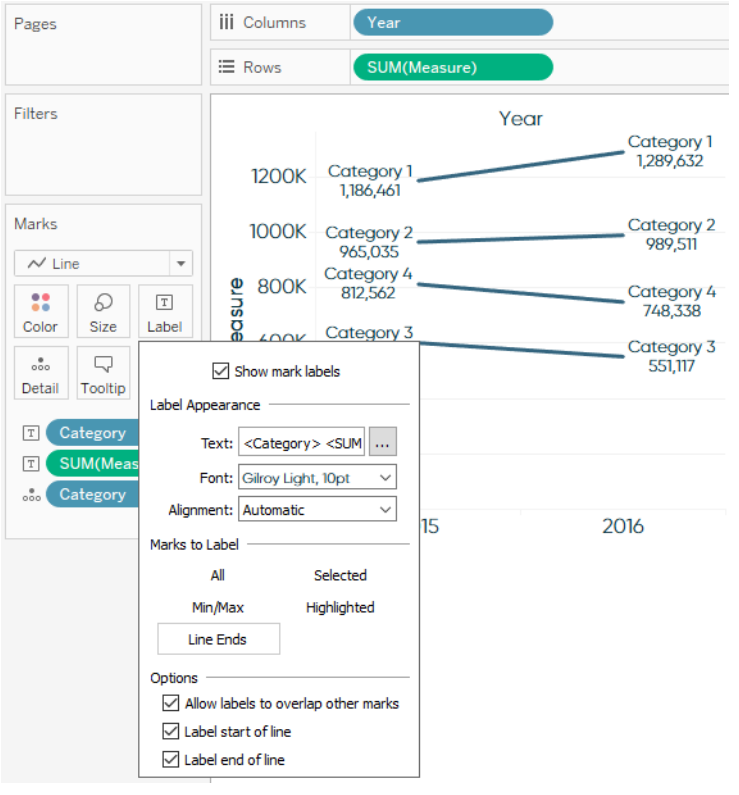
3. Change the mark type to Line.

   Change the mark type from Automatic (currently set to Bar) to Line to create slope graphs:



4. Format the lines.

   To finalize the view, add labels and markers to the line ends. To update the labels, place the fields you want on the view to the Label Marks Card, and edit the Label Marks Card to show labels on the line ends. The markers are added by editing the Color Marks Card:



How to Make Dual-Axis Slope Graphs in Tableau

For the purposes of this exercise, the underlying data looks like this:

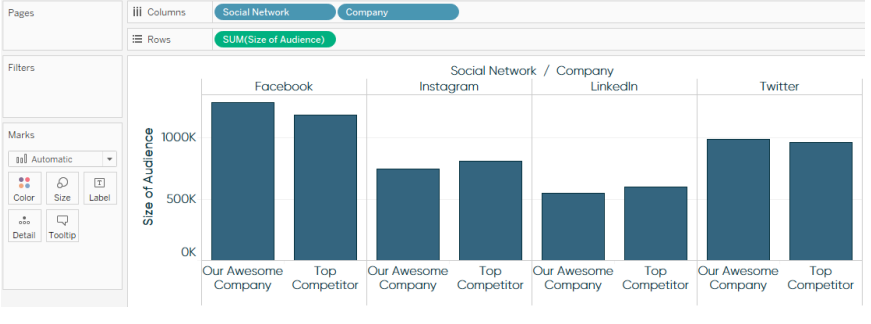
	A	B	C
1	Company	Social Network	Size of Audience
2	Our Awesome Company	Facebook	1289632
3	Top Competitor	Facebook	1186461
4	Our Awesome Company	Twitter	989511
5	Top Competitor	Twitter	965035
6	Our Awesome Company	LinkedIn	551117
7	Top Competitor	LinkedIn	600718
8	Our Awesome Company	Instagram	748338
9	Top Competitor	Instagram	812562

I'm using a Company A versus Company B scenario just to show a slightly different use case, but this same exact process is applicable if you are comparing two points in time.

Disclaimer before I let the genie out of the bottle: in the Thing 1 versus Thing 2 scenario I'm about to share, I only think this works well if you are comparing exactly two things. Once you get beyond two things, the visualization officially becomes a line graph, and should never be used to connect dimensions that are not elements of time. The best practice with more than two things is a simple bar chart.

1. Create a bar chart.

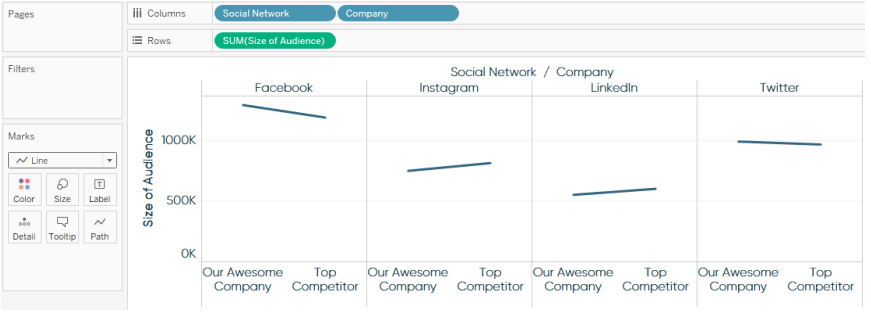
   As already mentioned, the first step to creating slope graphs is to get the columns set up with whatever dimensions we are comparing. Sometimes with traditional slope graphs, there can be overlapping points and labels if data points are too close together. In that case, and when comparing two things (versus times), I prefer to place the more granular dimension (in our case, social network) on the Columns Shelf first, followed by the dimension that includes "Thing 1" and "Thing 2" or "Time 1" and "Time 2." The measure we are analyzing goes on the Rows Shelf, to end up with a bar chart view like this:



   Again, this chart type is best practice when we are comparing two discrete categorical variables such as Company A versus Company B, so stop here if you are feeling any discomfort with looking at this in a different way.

2. Change mark type to Line.

   From here, a bar chart is converted into a slope graph by simply changing the mark type on the view from Bar to Line:



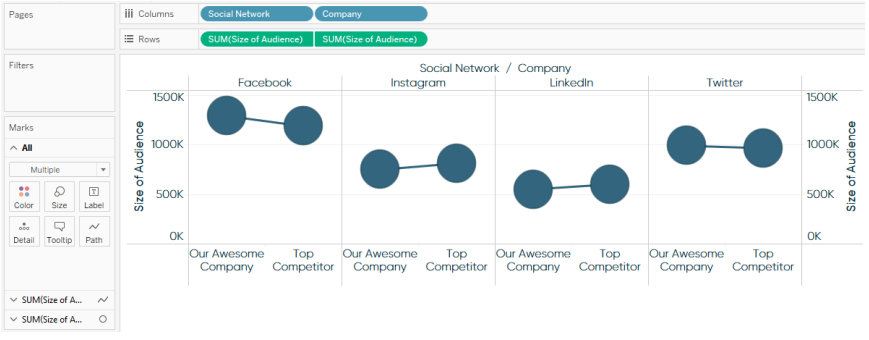
You can get a nice look from here by adding markers to the line ends:

At this point, you already have slope graphs in Tableau. This is an awesome chart type. If you want an incredibly awesome chart type, keep reading.

3. Create a dual-axis combo chart.

   Create a dual-axis combo chart by dragging the measure you're interested in (in our case, Size of Audience) to the opposite axis, and change the mark type on the second Marks Shelf for SUM(Size of Audience) to Circle. If you are not familiar with building dual-axis combo charts, be sure to check out Chapter 21.

   For best results, also be sure to synchronize the second axis by right-clicking the right-axis and choosing Synchronize Axis. At this point, my chart looks like this:



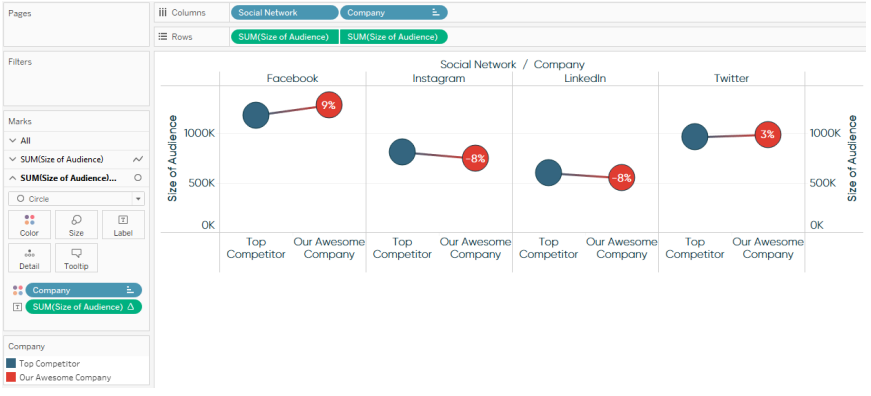
4. Customize the second view.

   Now that we have two measures on the Rows Shelf, we have two Marks Shelves for those measures that we can edit independently of each other. This means we can keep the slope graph on one Marks Shelf, but change the size, colors, and context on the circles in the secondary Marks Shelf.

As just one example, I'll use the circles to display the delta between ourselves and our main competitor. I will also color both the slope graphs and circles to make our brand clear on the view. Here are the steps:

1. Change the sort order so the top competitor is listed first and Our Awesome Company is listed second. This is more intuitive when you are comparing things across a table. The sort order can be changed by clicking the Company dimension on the Columns Shelf.
2. Add the Size of Audience measure to the Label Marks Card.
3. Add a quick table calculation to the Size of Audience measure for Percent Difference.
4. Edit the table calculation to compute the percent difference along Pane (Across). This will execute the table calculation for each social network column.
5. Add the Company dimension to the Color Marks Card of both the Slope Graph and the Circles (which is actually a dot plot) Marks Shelves.

   After some basic formatting, I end up with this:



This is just one example, but you can use this approach with other mark types, to show different KPIs, and/or to color based on performance such as a stoplight index (discussed in Chapter 61). Slope graphs are an excellent choice for comparing two things—especially changes over two points in time—and they are made even stronger when additional context is added by leveraging a dual-axis in Tableau.

CHAPTER 39

How to Make Donut Charts

When used properly, donut charts can be an effective way to communicate comparisons in a unique way. When used improperly, they are the butt of jokes mentioned in the same vein as pie charts. In fact, donut charts are essentially pie charts with a circle in the middle, and in Chapter 93, I explain why you shouldn't use pie charts. The big reason charts like bar charts work better than pie charts is that the viewer is comparing the length of the bars; not comparing the area of the wedges of the pie—and our brains are much better at comparing length than area.

So what's the difference with donut charts?

One of my recommendations for practitioners that cannot quite let go of pie charts is to use five slices or fewer. Personally, I would not use more than two. When used for the specific purpose of showing a metric's progress to goal, with one "slice" being the current state of the KPI and one "slice" being the remainder to goal, I think a donut chart works well.

I admit, a bullet graph would be the most efficient way to communicate the progress to goal scenario. A bullet graph would be processed faster by end users than a donut chart and also take up less real estate on a dashboard. It also works better for showing progress above goal because a donut chart stops once you get to 100%. If you think a bullet graph may be a more appropriate choice for your data visualization, see Chapter 26.

That being said, I don't mind the extra real estate that a donut chart takes up because that space can be used to provide additional context in a visually appealing way. For example, you can place an icon that denotes what the donut chart applies to or a callout number that shows the percent your KPI has progressed to goal. Donut charts are an acceptable alternative chart type that can make your data visualization stand out. For more on making your work "remarkable," see Chapter 97.

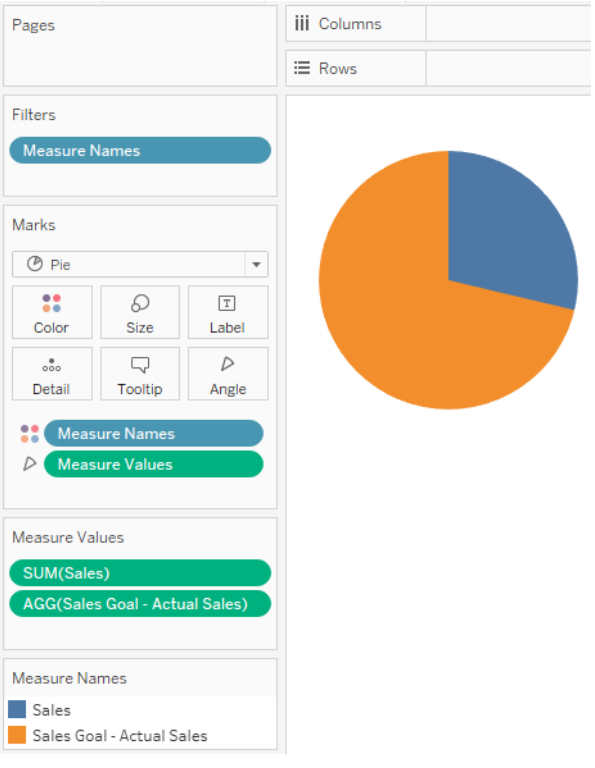
How to Make Donut Charts in Tableau

For the purposes of this exercise, we will make a gauge showing how our sales are progressing toward goal.

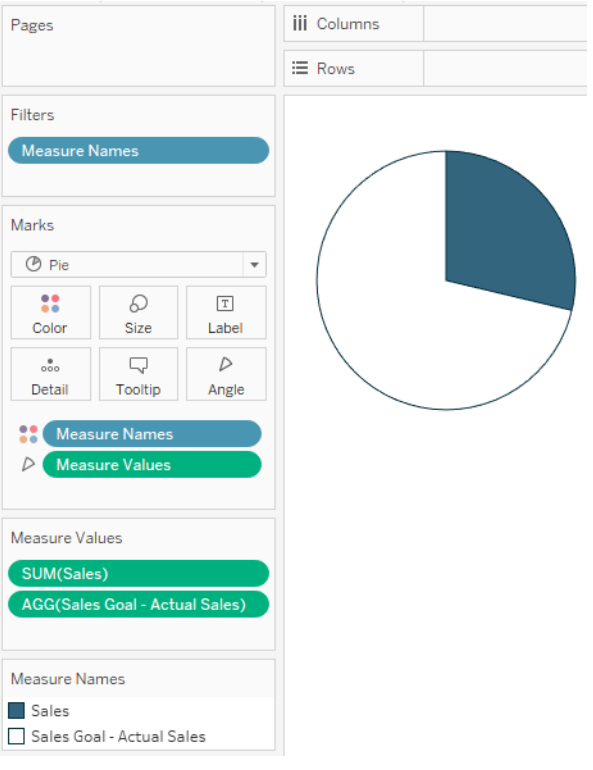
1. The Sample – Superstore dataset contains a measure for Sales, but we will need to create one more field that calculates the sales goal minus the actual sales. The data you are working with may contain a measure for Sales Goal, but I am going to hardcode a goal of $8,000,000 into my calculated field: $8,000,000 – SUM([Sales]):

   The reason we make this calculated field is so the two slices in our donut chart will always add up to the total goal, and actual sales will always by a correctly sized portion of the total goal.

2. You will now create a pie chart using two measures: Sales and the newly created Sales Goal – Actual Sales. Do this by first changing the mark type on your view to Pie. Now place Measure Names on the Color Marks Card and Measure Values on the Angle Marks Card. Finally, filter the view on Measure Names to only include Sales and Sales Goal – Actual Sales. At this point, your view should look like this:



3. Format the donut chart (currently a pie) by increasing the size, adding a border, and most importantly, coloring the Sales measure to a color of your choice and the Sales Goal – Actual Sales measure white. This will eventually create the filling gauge effect we are going for:

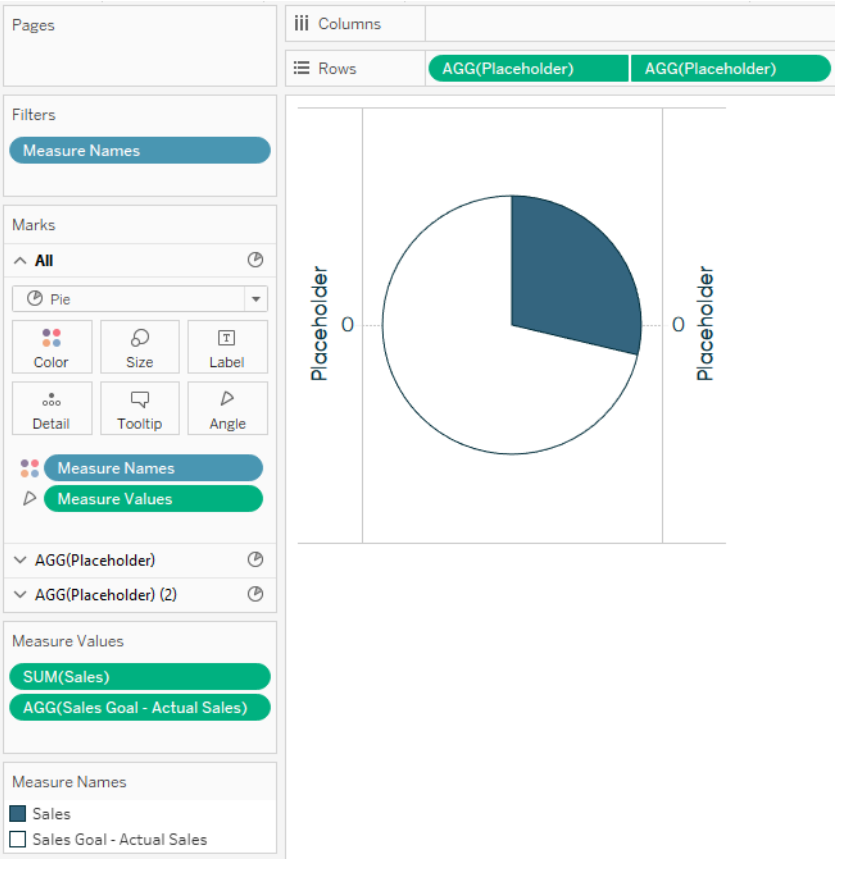


4. The final step in creating a true donut chart is to "poke a hole" in the chart we have created so far by adding a circle to the middle. I have two approaches for this final step:

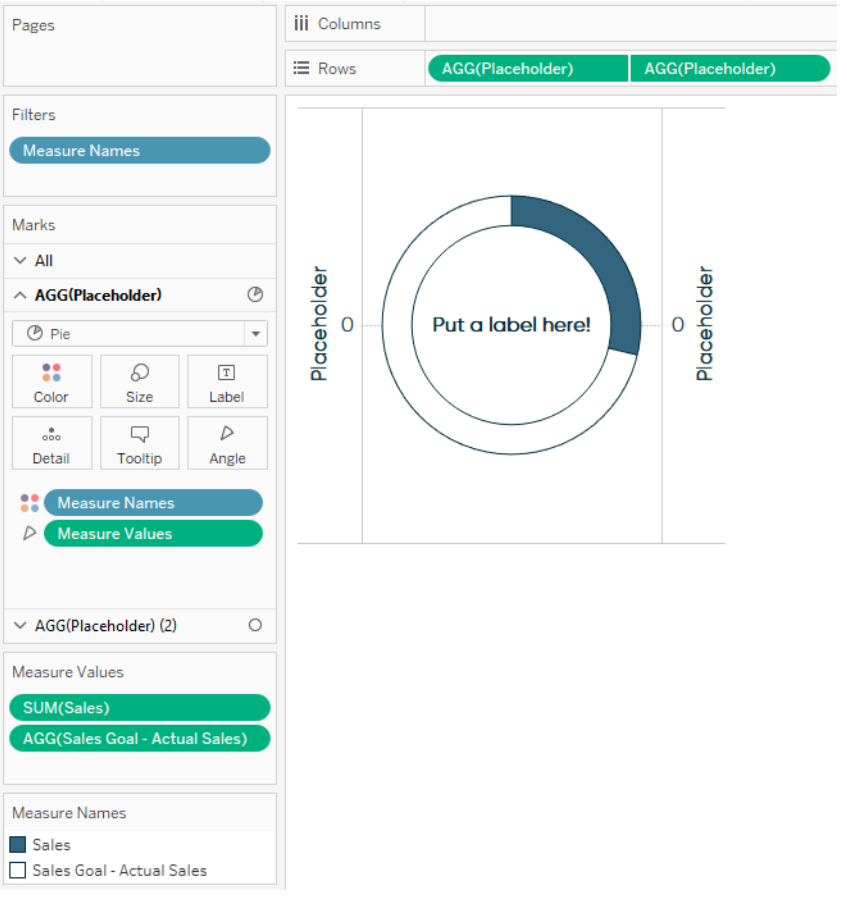
   Approach 1: Single Sheet

   For most situations, this first approach will be the most elegant solution.

   - Start by creating a calculated field called "Placeholder" that has this formula: MIN(0)
   - Create a Dual-Axis Chart by placing this new calculated field on the Rows Shelf, then adding it again to the opposite axis. At this point, my view looks like this:



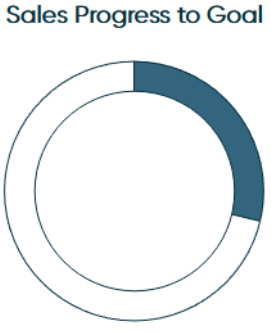
Notice that there are now different Marks Shelves for each of my Placeholder fields. That means that you can now edit the marks for each field independently of each other. On the Marks Shelf for the first placeholder, leave the settings as is with a mark type of Pie—this is our original chart and we don't want to change it. On the Marks Shelf for the second placeholder field, change the mark type to Circle, which will become the hole in the donut chart. Reduce the size and change the color of the circle as desired. You can also modify the marks for the second placeholder further by adding a label to provide additional information. At this point, my view looks like this:



Finally, clean up the view by doing some simple formatting such as hiding zero lines and headers. You now have a donut chart that is comparing Sales to Sales Goal. You can take this a step further by adding a dimension to the Columns Shelf to create a small multiples view that compares performance across several different dimension members (i.e., Region, Segment, etc.):

Approach 2: Floating Sheets

If your visualization requires icons or additional graphic design, as is the case in my Your Salary vs. a MLB Player's Salary viz, I recommend you float a *.png* image over the chart. It is very important that you use a *.png* file so that your corners are transparent and you will be able to see the underlying chart:



Finally, I mentioned earlier that I don't mind donut charts because the real estate available in the inner circle can be used to communicate additional context either through icons or text.

If you're using approach 1, you can create a calculated field that determines the percentage of sales goal obtained, and then add this calculated field to the label of the placeholder 2 Marks Shelf.

If you're using approach 2, you can enhance the *.png* image being used with your own icons or other design elements. If you would like to use a calculated field to show progress to goal, you can create a sheet with that metric and float it over the chart and *.png* image.

In either approach, your final product will end up looking like this:

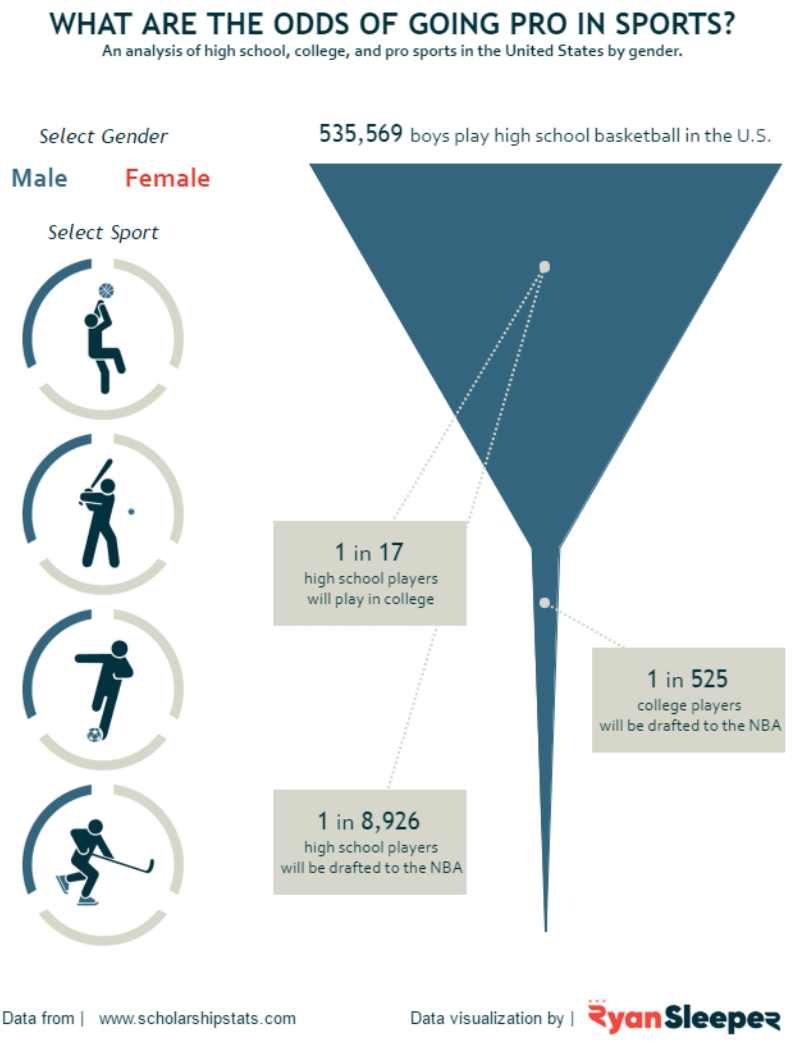
CHAPTER 40

How to Make Funnel Charts

The Odds of Going Pro In Sports viz has generated more questions around how it was created than any other viz I have put together during my career with Tableau. With its one dominant funnel chart and icon-based navigation, the viz tells the story about the share of high school athletes progressing to the college and pro levels across several sports for each gender.

The most common question I receive: *That was made in Tableau?* I would be lying if I said that question doesn't make me want to stand up a little taller, but the secret is, the viz was one of the easiest dashboards I have ever put together. In fact, I put it together in a couple of hours on a Sunday afternoon. Funnel charts are one of the simplest chart types you can create, but they have proved to be incredibly effective in a corporate setting—think conversion rates and customer flows. This chapter will walk you through multiple approaches to creating funnel charts.

First, let's take a look at the full version of *What are the odds of going pro in sports?*



How to Make Funnel Charts in Tableau

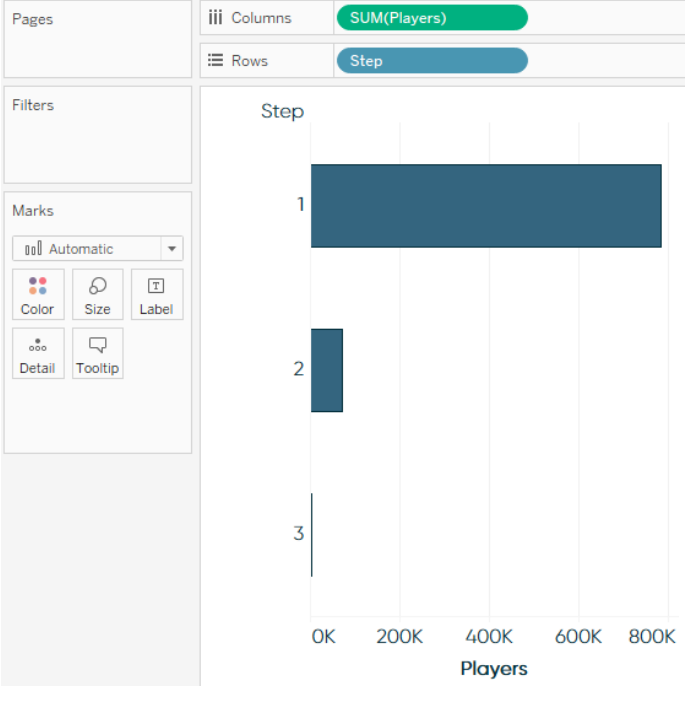
Option 1: The Step Dimension

Going into this project, I knew that I wanted the main view to be a funnel chart. For this reason, I was able to plan ahead and put the data in an optimal format for Tableau. Here is a sample of the underlying data:

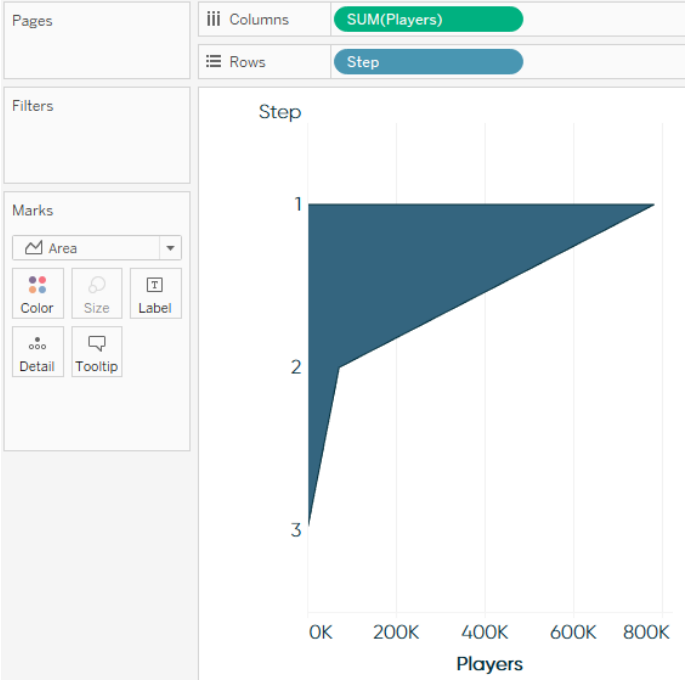
	A	B	C
1	Gender	Step	Players
2	Female 1	1	371393
3	Female 2	2	35490
4	Female 3	3	36
5	Male 1	1	412351
6	Male 2	2	36741
7	Male 3	3	77

Note that there is a dimension called Step, which is just an arbitrary, intuitive name for each of the three levels of high school, college, and pro. The measure is then the corresponding number of players that reached each step. This is the easiest way to build a funnel chart. Here are the steps:

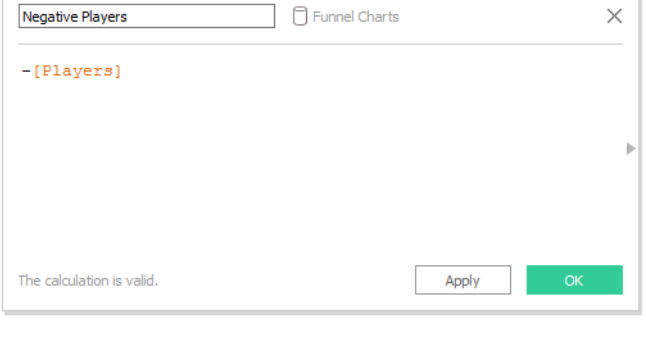
1. Create a horizontal bar chart by dragging your Step dimension onto the Rows Shelf and your Players measure to the Rows Shelf:



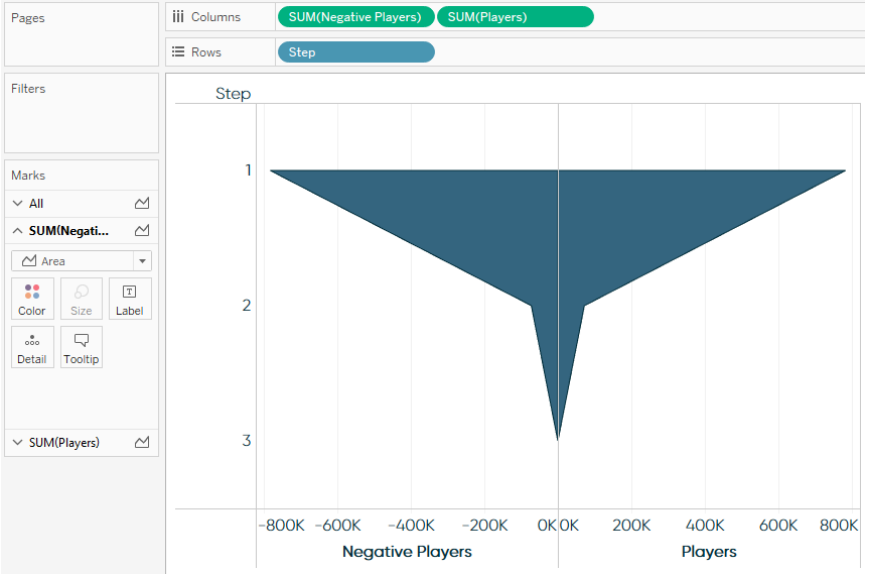
2. Here is where the magic happens. Simply change the mark type from Automatic (which is currently Bar) to Area. You are left with the right side of the funnel:



3. If your data is in a format outlined in the table just shown, there is a simple solution to mirroring the right side of the funnel onto the left side. To do this, you will create a calculated field by right-clicking the Players measure and selecting Create Calculated Field. Name your calculated field Negative Players and add a "-" in front of [Players]. Your entire formula will look like this:



4. The final step in creating a symmetrical funnel chart is to drag and drop your new Negative Players measure in front of the Players measure on the Columns Shelf:



From here, your funnel chart can be used like any other chart in Tableau. Encode it by color (in my case, it's colored by gender), add filters, and add tooltips/labels/annotations to make the data in the chart clear to the end user.

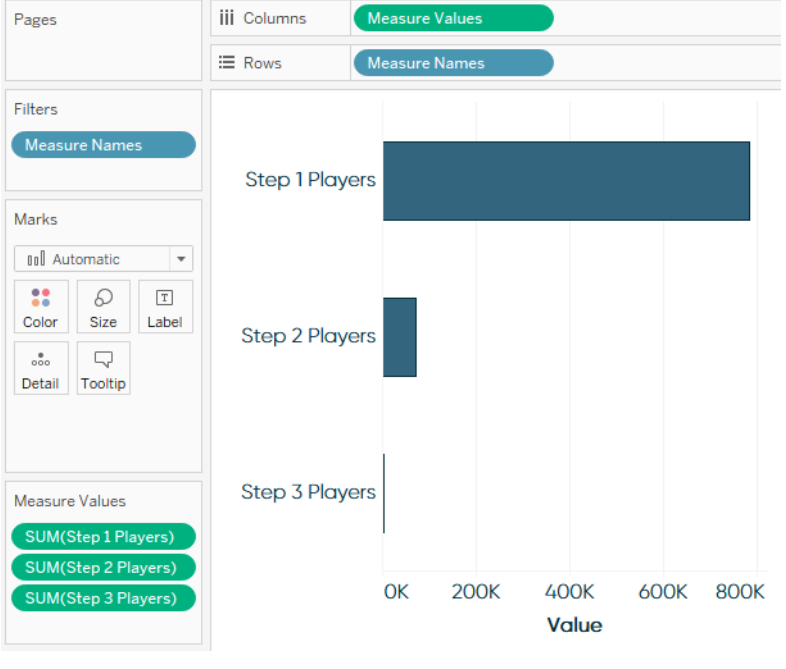
Option 2: Separate Measures

It is likely that your corporate data is not in the layout of the preceding table by default, and you may not be in a position to easily change this on your own. This solution is not as flexible or optimal for Tableau, but it is functional. Let's assume that instead of the layout in the table from before, your data looks more like this, with each Players measure broken out by level, causing your layout to be more horizontal instead of vertical:

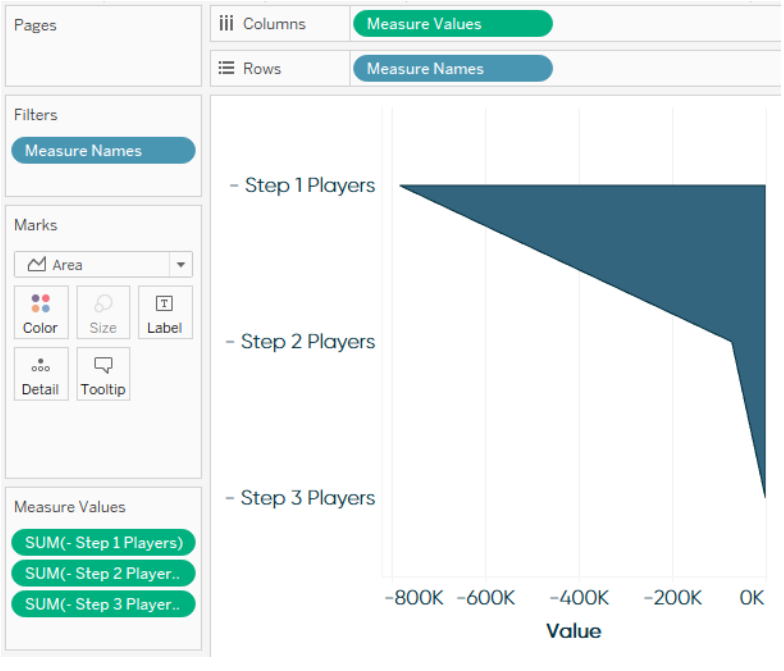
	A	B	C	D
1	Gender	Step 1 Players	Step 2 Players	Step 3 Players
2	Female	371393	35490	36
3	Male	412351	36741	77

In this case, because you no longer have a dimension of Step, the approach is slightly different:

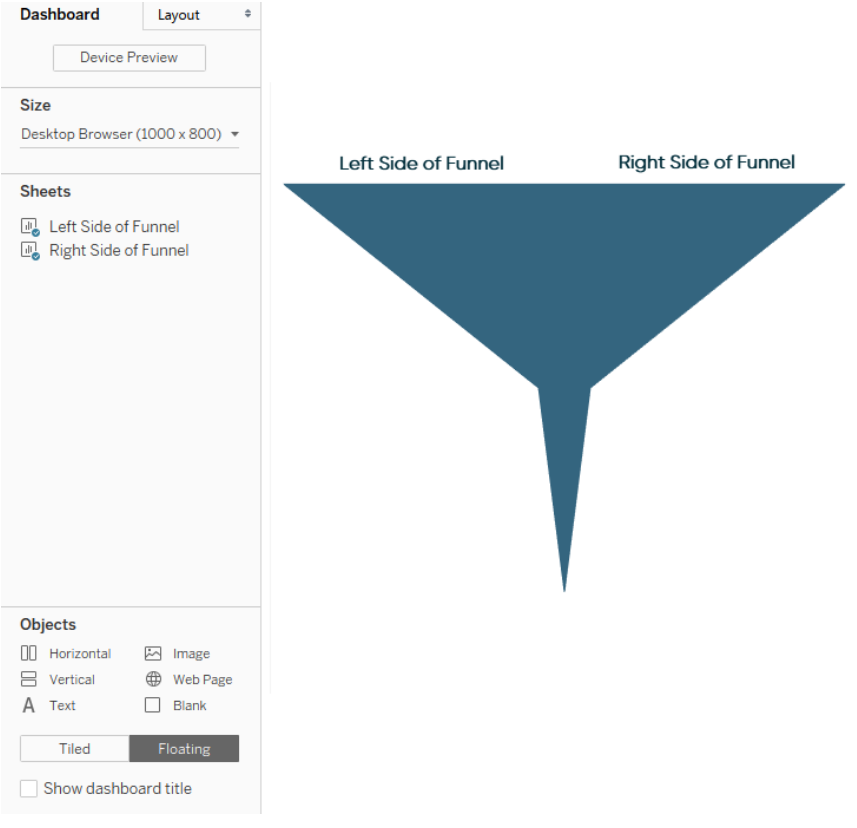
1. To create the right side of the funnel, you will need to add the measures to the view individually. The easiest way to do this is to double-click each of your three measures (Step 1 Players, Step 2 Players, and Step 3 Players) to add them to the view. Double-clicking each measure will create a small multiples chart, so you will have to use the Show Me button to change the chart type to a bar chart. While the setup is different, you should now see the beginnings of the right side of your funnel, just like in Step 1 from the first approach:



2. Follow step 2 from the first approach, changing your mark type from Automatic to Area.
3. This is where the second approach gets tricky. The first thing you need to do is create a negative version of each of your three measures, much like creating a negative version of the Players measure in step 3 from the first approach. Once you have three new calculated fields, one negative version for each of your three measures (i.e., Negative Step 1 Players, Negative Step 2 Players, and Negative Step 3 Players), open a new worksheet and follow steps 1 and 2—this time using your newly created negative versions of each measure. You should end up with the left side of your funnel:



4. The final step to this approach is joining the left and right side of your funnel. This can be accomplished by floating each sheet on a dashboard and lining them up:



Admittedly, this second approach requires some hackish creativity, but it works with most data out of box without any reformatting.

So there you have it! Funnel charts in four steps, regardless of the layout of your funnel data.

CHAPTER 41

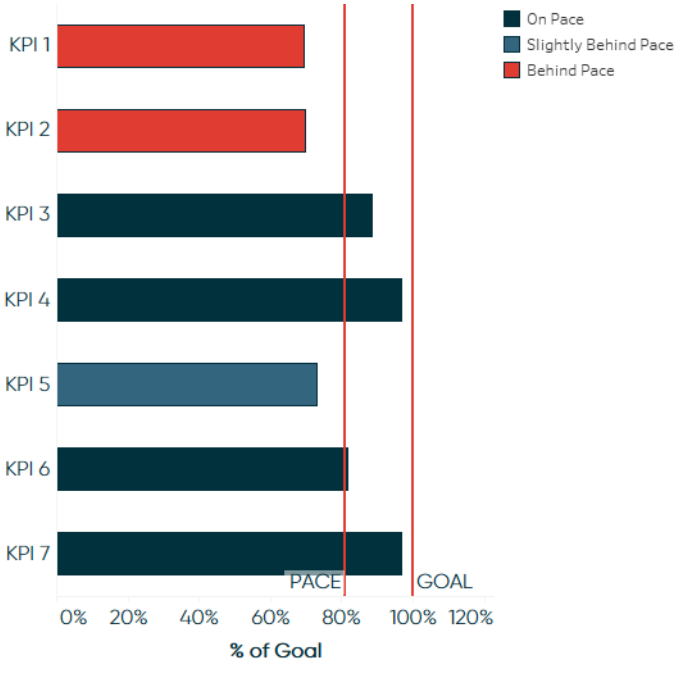
Introducing Pace Charts in Tableau

Pace charts are an alternative bullet graph design that normalize progress to goal visualizations across KPIs, even if the KPIs have different data formats, scales, and/or seasonal trends. They are useful for providing an "apples to apples" pace to goal comparison in businesses that have KPIs that span different categories such as revenue, social media followers, attendance, and so on.

With KPIs such as revenue, social media followers, and attendance, not only are the metrics in different formats, but they are often on very different scales and have varying seasonality. For example, you may gain social media followers throughout an entire year, but if you are an NFL team, your attendance won't start until August. Further, your growth on social media might be on a scale of thousands, while revenue may be on a scale of millions.

Pace charts normalize KPIs by comparing them all on an axis that ends at 100% (the goal). In addition to showing how much progress each KPI has made toward the goal, a linear or seasonal pace is displayed to illustrate whether progress to goal is on pace to reach goal. To enhance the illustration, the marks can be colored to show how current progress to goal for each respective KPI compares to its pace to goal.

To illustrate how to create a pace chart in Tableau, I will start by re-creating this pace chart showing a variety of KPIs that are on a linear pace (i.e., they should all be at the same progress to goal at this point in the year):

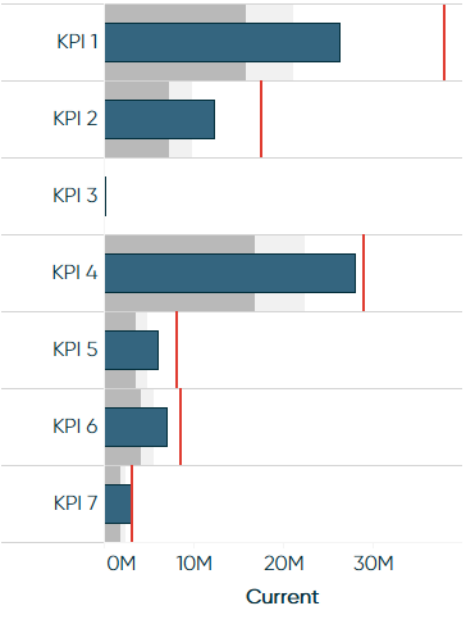


The underlying dataset used to create this pace chart looks like this:

	A	B	C
1	Measure	Current	Goal
2	KPI 1	26308469	37900000
3	KPI 2	12235114	17550000
4	KPI 3	7959	9000
5	KPI 4	28005548	29000000
6	KPI 5	6000000	8200000
7	KPI 6	7000000	8588000
8	KPI 7	3100000	3200000

Even though the KPIs are on very different scales, it is easy to compare them to determine which are on pace, slightly behind pace, or behind pace.

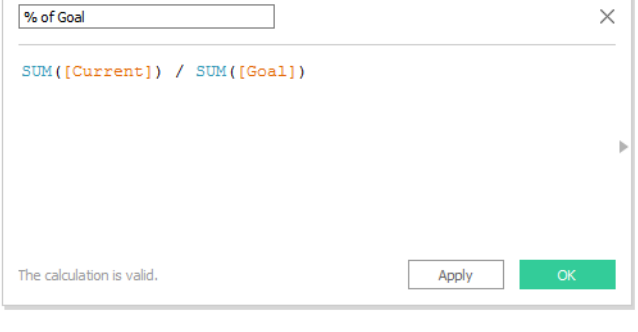
To affirm this point, here is the same data using a traditional bullet graph:



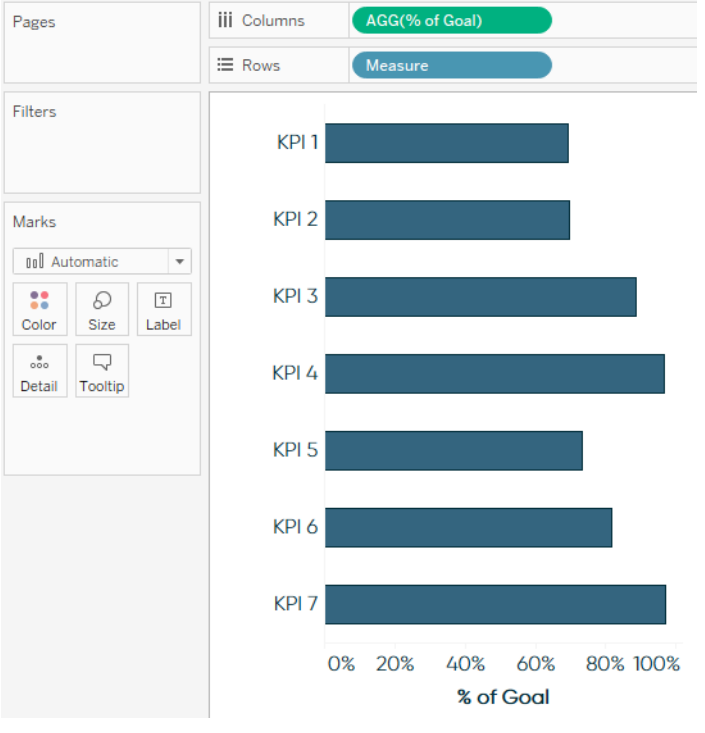
With this visualization, it is impossible to compare the progress to goal of KPI 3, which is on a much smaller scale than the other KPIs. It is also challenging to determine the progress to goal of KPI 7, because it is on the same scale as KPI 1, which has the largest goal and is extending the x-axis. While you could break up this graph into seven different parts to fix the scaling, there is a better way to normalize the data.

How to Create a Pace Chart with a Linear Pace in Tableau

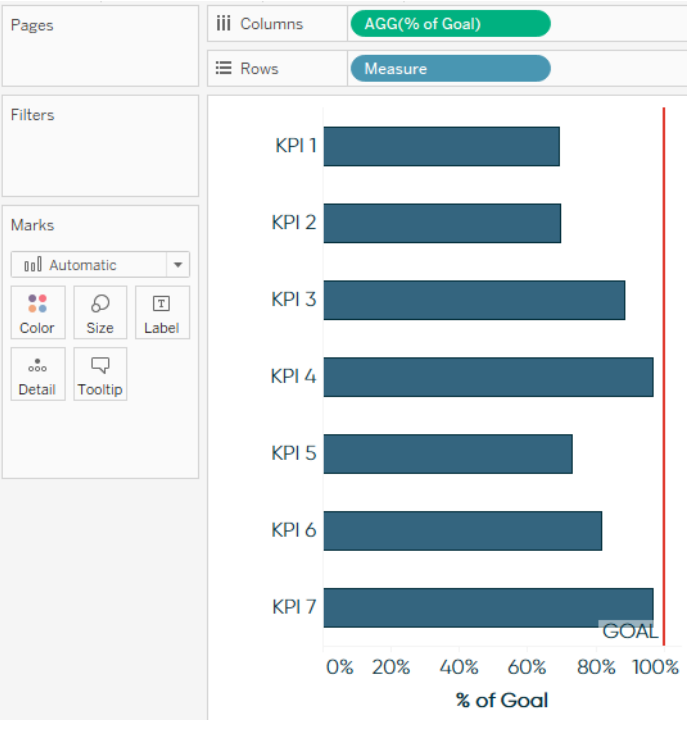
To normalize the bars in a pace chart, create a calculated field that calculates the progress to goal. This is used to represent the bars instead of the current values. The formula for this calculation is SUM([Current Value])/SUM([Goal]):



After you have created the Progress to Goal calculated field, create a horizontal bar chart showing the progress to goal for each KPI:



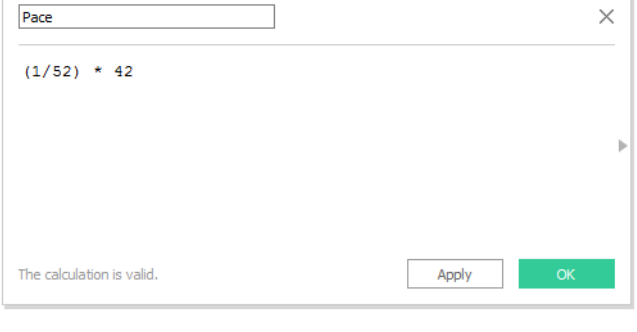
With a pace chart, the goal line is always normalized across every KPI at 100%. To add this reference to the visualization, simply add a reference line with a constant of 1 (which equals 100%):



For this illustration, we will pretend that the pace to goal should be the same across all seven of our KPIs. When using a linear pace, a calculated field can be created to calculate how far to goal each KPI should be at the current point in the year. For example, if we are in week 42, the pace calculation would be:

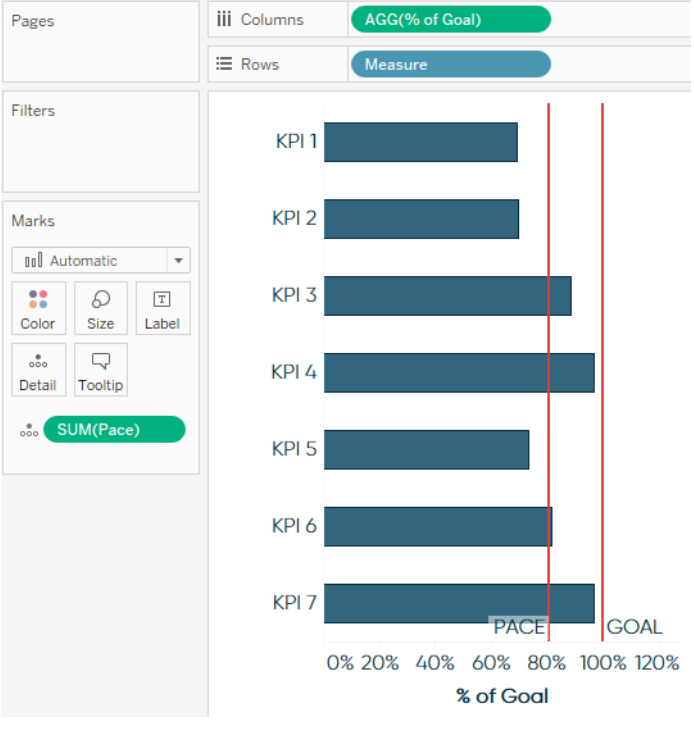
```
(1/52) * 42
```

This calculation is dividing the year into 52 equal parts (i.e., weeks), then multiplying that fraction by the number of weeks that have passed in the year:

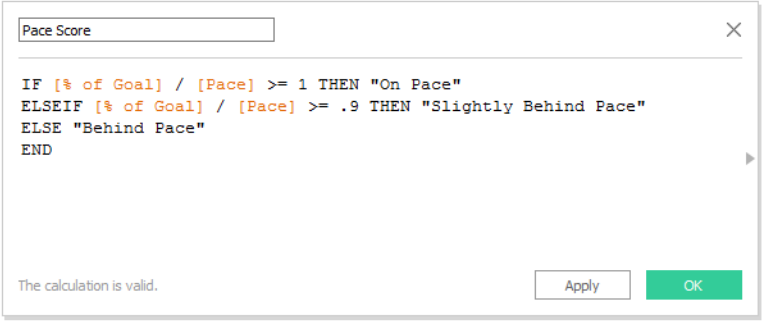


In Tableau, you have the option to replace the 42 with a parameter that allows the end user to change the multiplier.

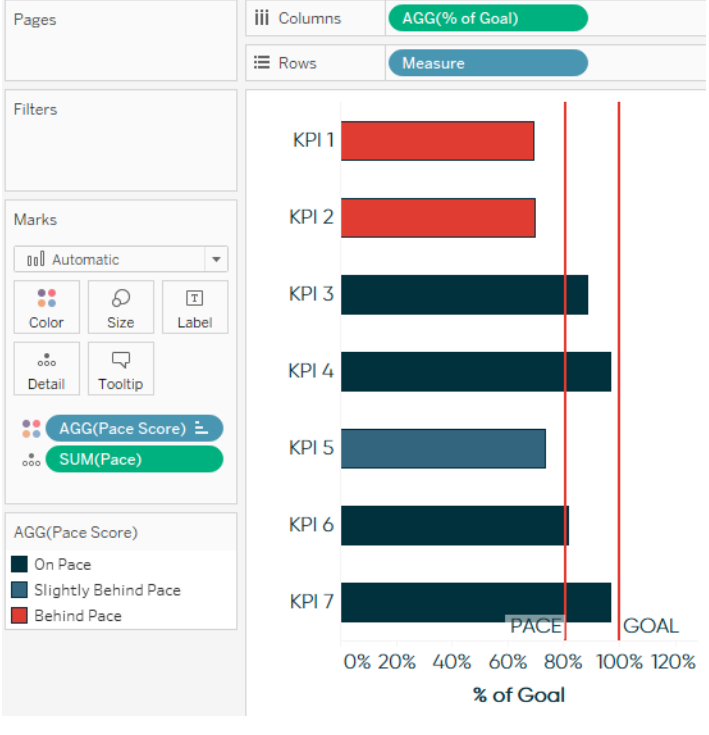
Once you have the Pace calculated field, add it to the Detail Marks Card so it can be used as a reference line. Then add a reference line that shows where the pace should be at this point in the year:



Lastly, to color the bars to illustrate whether each KPI is on pace, slightly behind pace, or behind pace, create a calculated field with the scoring logic. This will vary based on your own requirements, but as one example, I'll pretend that 100% or above is on pace, 90%–99.99% is slightly behind pace, and anything less than 90% is behind pace:



This pace score is then dragged to the Color Marks Card to color each bar by its progress to goal classification:

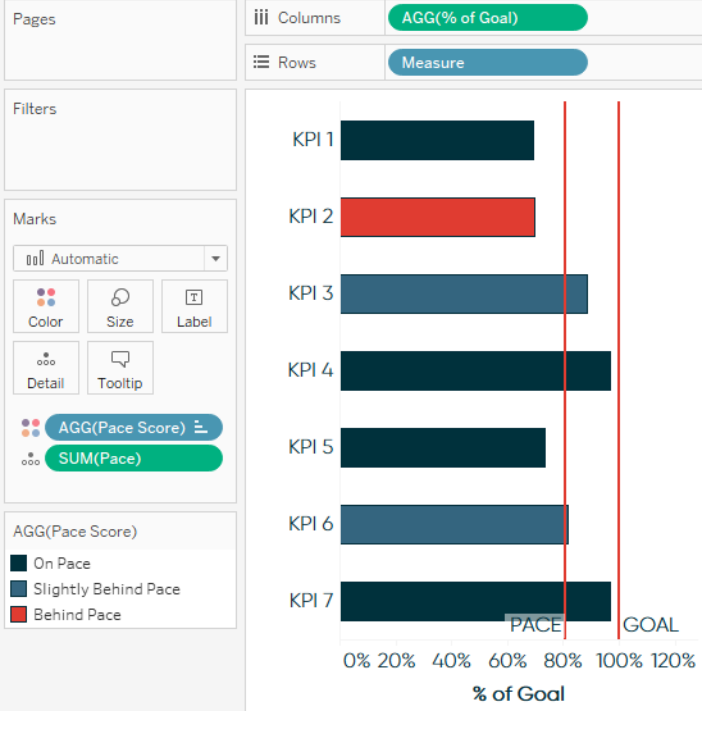


In this tutorial, we used a linear pace that was calculated by taking 1/52 of the year multiplied by the current week of the year. However, this pace can be replaced with a different metric such as the value for each KPI at this point last year, or a goal for each respective KPI at this point in the year. Here is one more example using a differ-

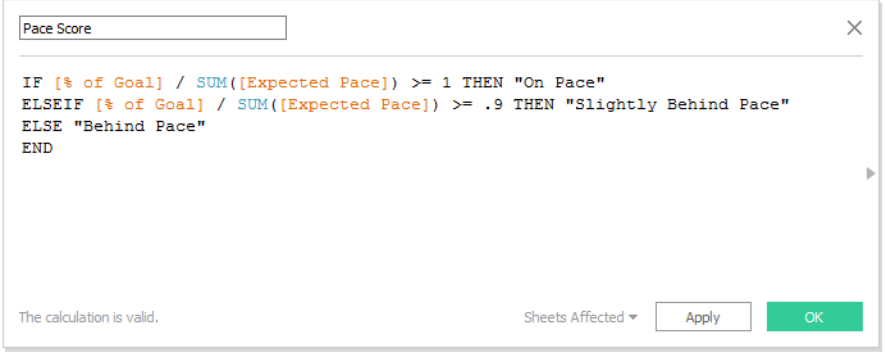
ent expected pace for each respective KPI. In this example, I have the expected pace at this point in the year as an additional field in my underlying data:

	A	B	C	D
1	Measure	Current	Goal	Expected Pace
2	KPI 1	26308469	37900000	0.6
3	KPI 2	12235114	17550000	0.8
4	KPI 3	7959	9000	0.9
5	KPI 4	28005548	29000000	0.95
6	KPI 5	6000000	8200000	0.7
7	KPI 6	7000000	8588000	0.85
8	KPI 7	3100000	3200000	0.95

The seasonal pace chart using this data looks like this in Tableau:



To create this version, I replaced the calculated linear pace reference line with a reference line for expected pace, which shows the expected pace for each respective KPI. I also replaced the linear pace calculation in the Pace Score calculated field with the Expected Pace measure from the underlying data:



Notice how this seasonal pace chart tells a different story regarding the progress to goal for each KPI than the pace chart with the linear pace. But in both cases, pace charts were used to normalize progress to goal calculations across KPIs to get a more effective visualization about the business.

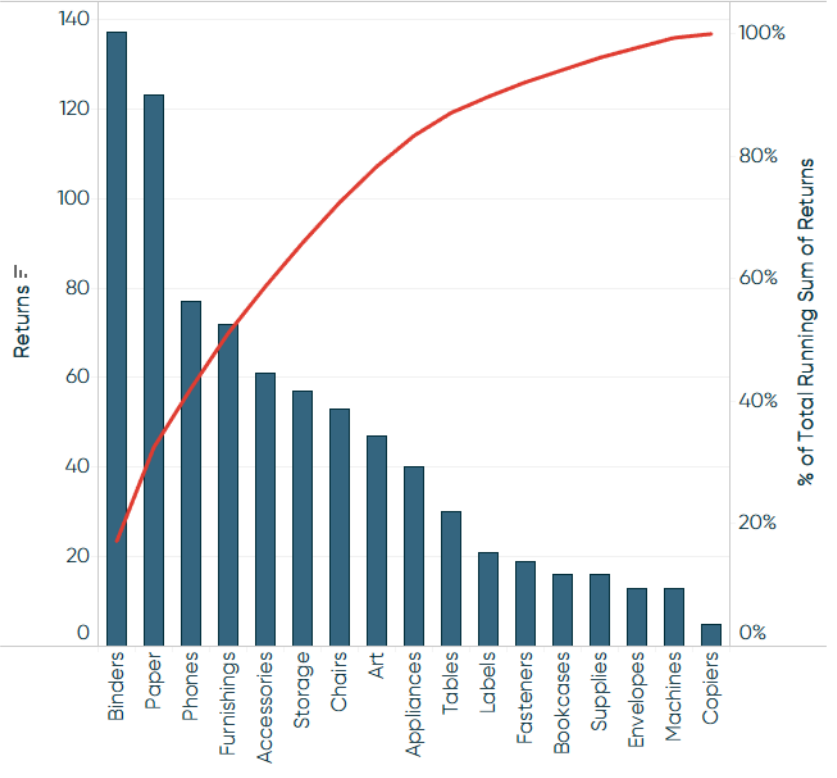
CHAPTER 42

How to Make a Pareto Chart

A Pareto chart, named for Vilfredo Pareto (the 80-20 guy!), is a dual-axis combination chart used to highlight dimension members that are having the biggest impact to the measure in question. On the primary axis, bars are used to show the raw quantities for each dimension member, sorted in descending order. On the secondary axis, a line graph is used to show the cumulative total in percent format. While this chart type can serve a variety of purposes, it is part of the seven basic tools of quality control, and is traditionally used to identify the biggest contributors to a cumulative total and opportunities for improvement.

How to Make a Pareto Chart in Tableau

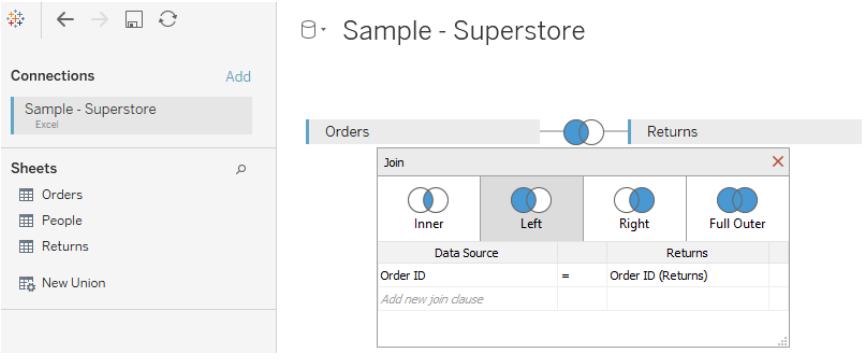
In sticking with a quality control scenario, this tutorial will use the Sample – Superstore dataset to look at which product sub-categories contribute the most returned items. The final chart will look like this:



As with the rest of *Practical Tableau*, you can follow along using the Sample – Superstore data that comes with every download of Tableau. However, this tutorial required me to take two additional steps to prepare the data:

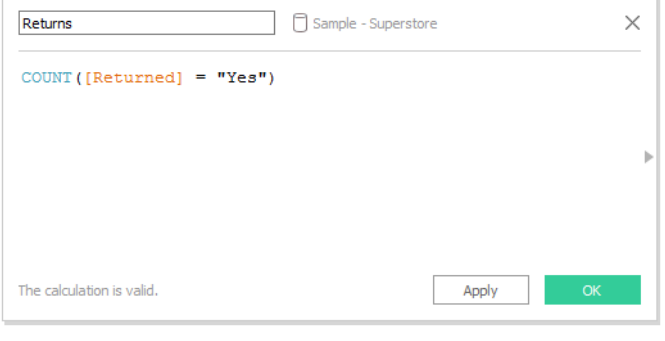
1. Left join the Returns table to the Orders table.

   To do this, right-click the Sample Superstore data connection and choose "Edit data source". A new interface will appear where you can drag the Returns table next to the Orders table and set up a left join on Order ID:

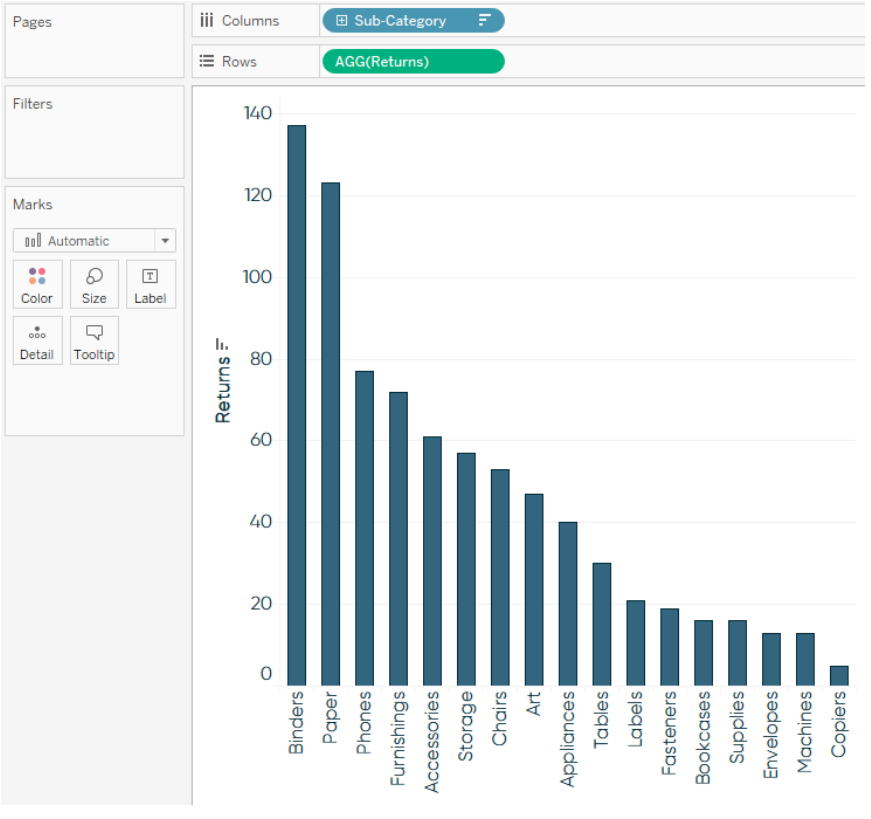


2. Create a calculated field to count the number of returns.

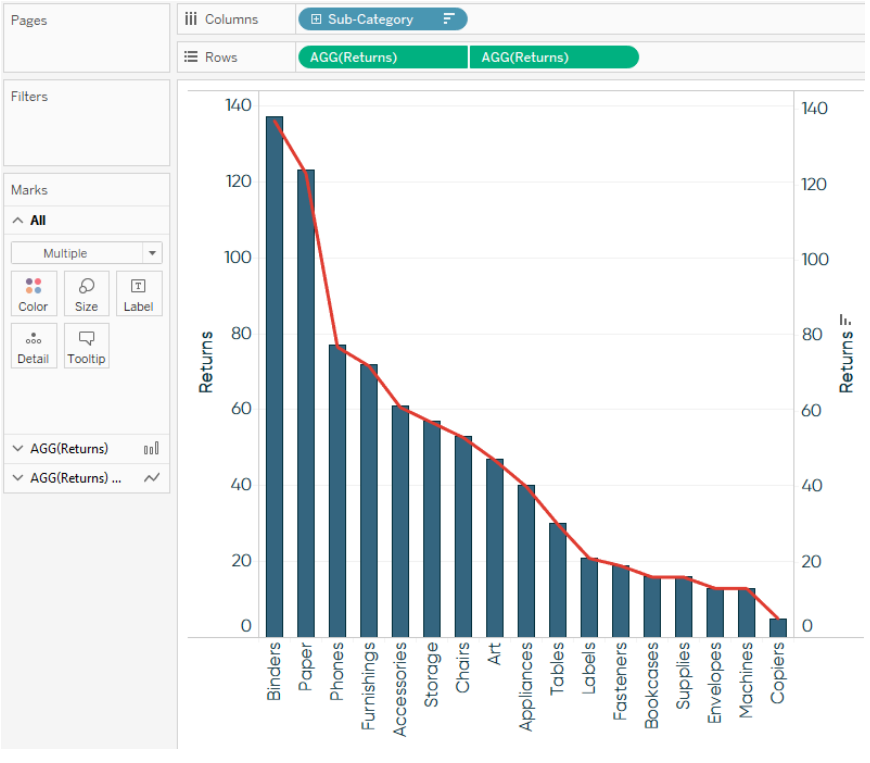
   The formula for this calculation is COUNT([Returned] = "Yes"):



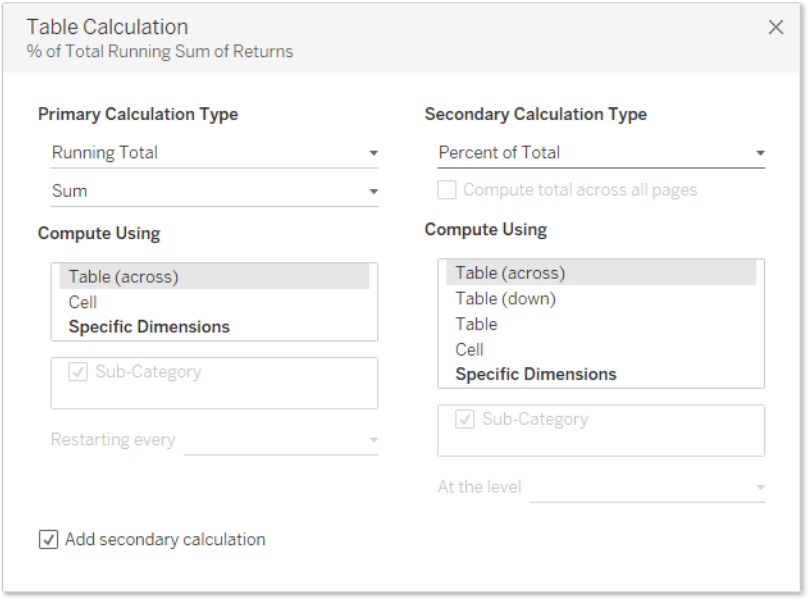
   We are now ready to build out the Pareto chart. First, create a bar chart that looks at the number of returns per product sub-category and sort it in descending order:



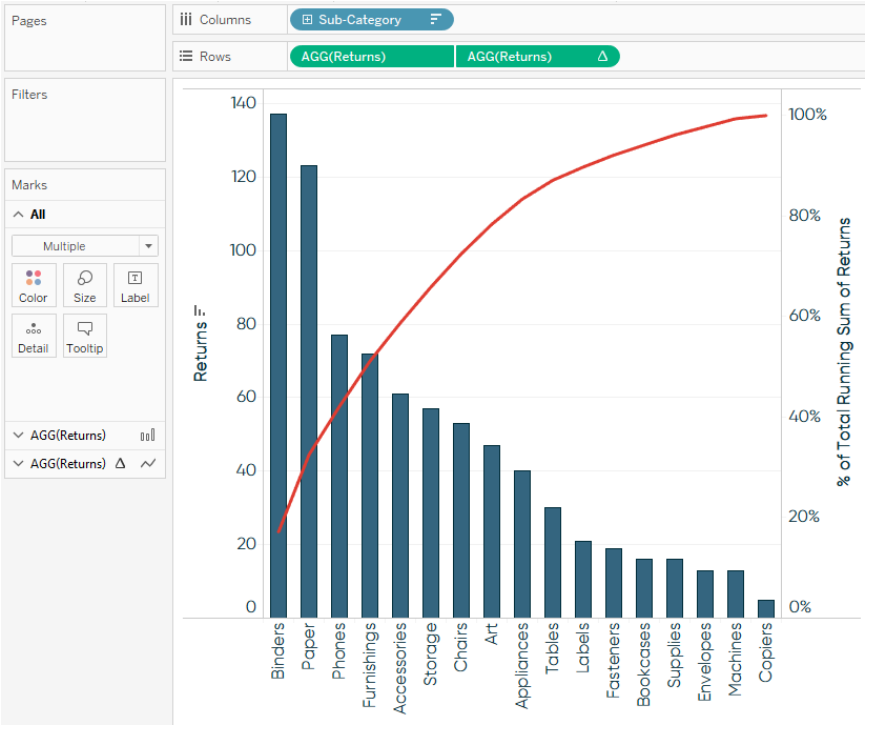
3. Next, create a dual-axis combination chart by dragging the Returns measure from the Measure area of the Data pane to the right axis, changing the mark type on the primary axis back to Bar, and changing the mark type for the secondary axis to Line. For a refresher on creating this type of chart, see Chapter 21. At this point, the visualization will look similar to this:



4. Now the step that makes this a Pareto chart. We will add a table calculation and a secondary table calculation to the second Returns pill to display the cumulative percent of returns across product sub-categories. To add the first table calculation, click the second Returns pill, hover over "Quick table calculation," and choose "Running total." With the table calculation for running total, you can add a second table calculation on the result. This is how we can calculate the raw running total number at each product sub-category, then add a secondary calculation to determine the cumulative percent of total. To add a secondary table calculation, click the second Returns pill again, choose "Edit table calculation," and check the box at the bottom of the interface that says Add Secondary Calculation. Changing the Secondary Calculation Type to Percent of Total will display the cumulative percent of total:



After changing the axis tick marks for a cleaner look, I am left with this Pareto chart:



This chart can now be used for insights such as, "The business' three most returned product sub-categories are causing about 40% of the total returns." Pareto charts are an effective way to quickly highlight opportunities for improvement and provide a scale for how urgently a quality control problem should be treated.

CHAPTER 43

How to Make a Control Chart

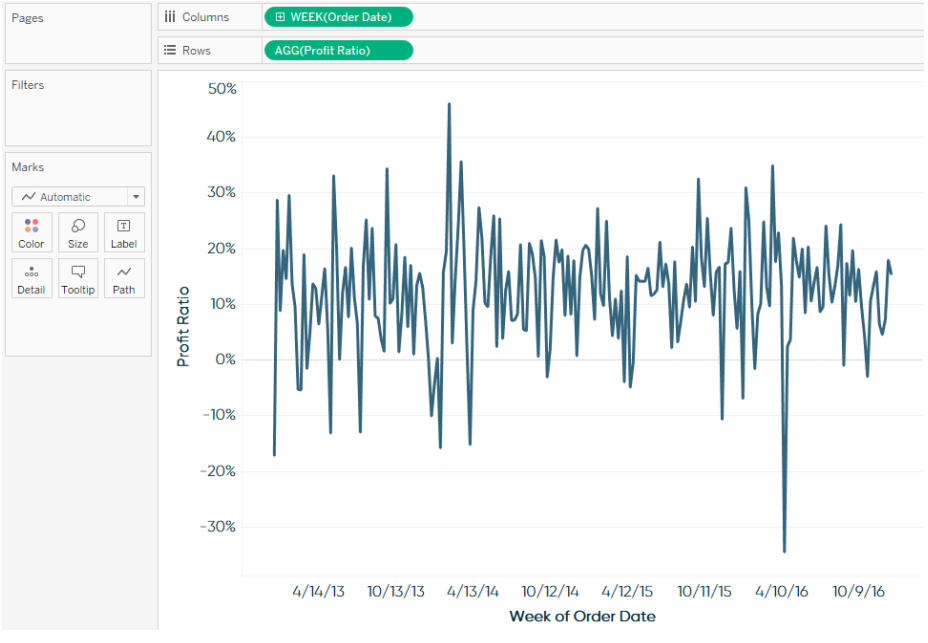
Control charts, or Shewhart charts, were designed to evaluate whether production is in a state of statistical control. Along with Pareto charts, histograms, and scatter plots, they are one of the seven basic tools for quality control. Basic control charts show a time-series analysis with reference lines that show the average performance along with an upper control limit, typically three standard deviations above the mean, and lower control limit, typically three standard deviations below the mean.

Control charts were originally called Shewhart charts, named for Walter Shewhart, who invented the visualization type while working on a way to improve the reliability of the telephony transmission systems at Bell Labs in the 1920s. Control charts were born in, and are still primarily used for, evaluating manufacturing processes. For example, a factory may use this type of analysis to predict production levels and have a statistical indicator for when corrective action needs to be taken.

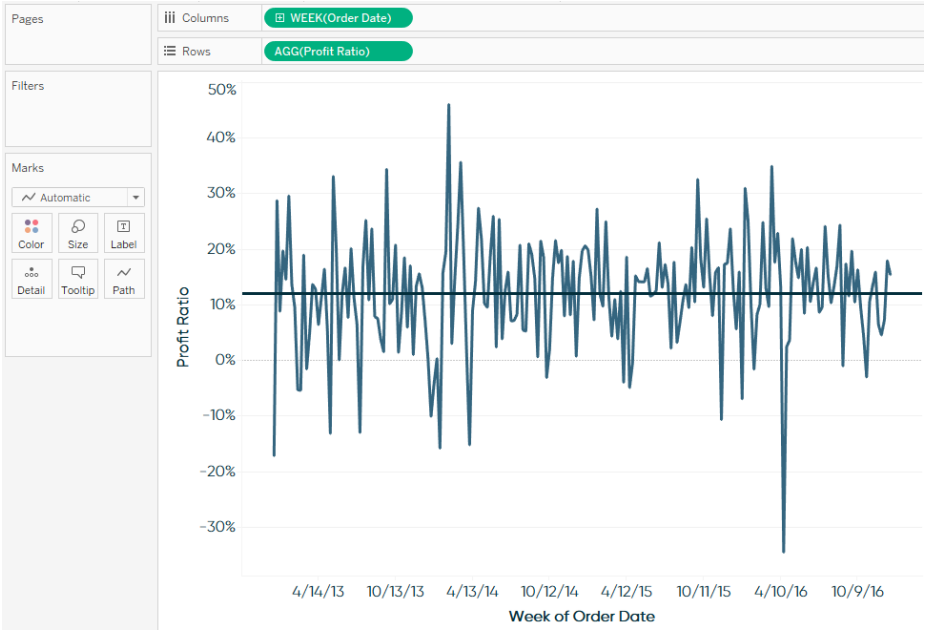
How to Make Control Charts in Tableau

While designed for manufacturing, I find control charts to be useful for any measure in which historical data can be used to predict performance and provide visual cues for when a statistically significant change is occurring in the business. In that vein, and so we can all follow along using the same data, this tutorial will create a basic control chart that evaluates the Profit Ratio measure in the Sample - Superstore dataset.

To create a control chart, start by creating a continuous line graph that looks at Profit Ratio by week:

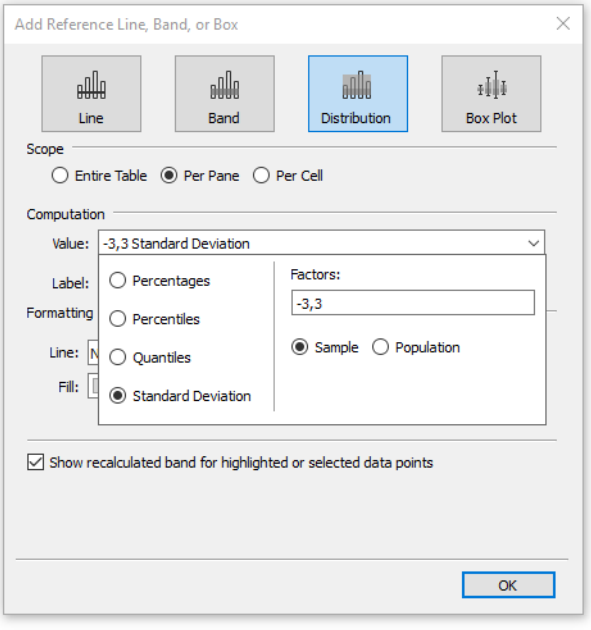


It may be concerning that we see several weeks with a negative profit ratio, but based on our historical data, is it a *statistically significant* cause for concern? The answer comes through control charts. Control charts start with a line that shows the mean performance for all of the data points. This is easily added in Tableau as a reference line. To add the reference line, right-click the axis for Profit Ratio, choose "Add reference line," and click OK. Here's how the view looks at this point after removing the label on the reference line and formatting the line for color and weight:



To finalize the control chart, we need to add the upper and lower control limits, which are typically three standard deviations from the mean. This step will also be accomplished through reference lines in Tableau. More specifically, a reference *distribution* will be used that draws two lines on the view, one for the upper control limit and one for the lower control limit. To do this, follow these steps:

1. Right-click the axis for Profit Ratio and choose "Add reference line."
2. In the reference line options listed at the top of the dialog box, choose Distribution.
3. In the drop-down for the distribution value, select the Standard Deviation option.
4. Change the factors to –3 and 3:



Here's how my final view looks after removing the line labels and doing some formatting to the reference distribution:

This final control chart shows not only the average profit ratio during the weeks in our analysis, but that the business can expect profit ratio to fall between –19% and 42%. In the current example, there was a week that fell outside of the lower control limit and deserves further investigation.

For more on this chart type, see Ben Jones' Tableau Public post, How to Make Control Charts in Tableau.

CHAPTER 44

How to Make Dynamic Dual-Axis Bump Charts

Bump charts are an effective way to show how dimension members are ranking across different measures over time. For example, you may want to see how specific product categories have ranked in sales for your company from year to year. Or maybe you want to use discrete months as your element of time to see if the rankings for product categories change based on seasonality. Or maybe you want to do something outside of work and see how your fantasy football players are ranking across different statistics from week to week!

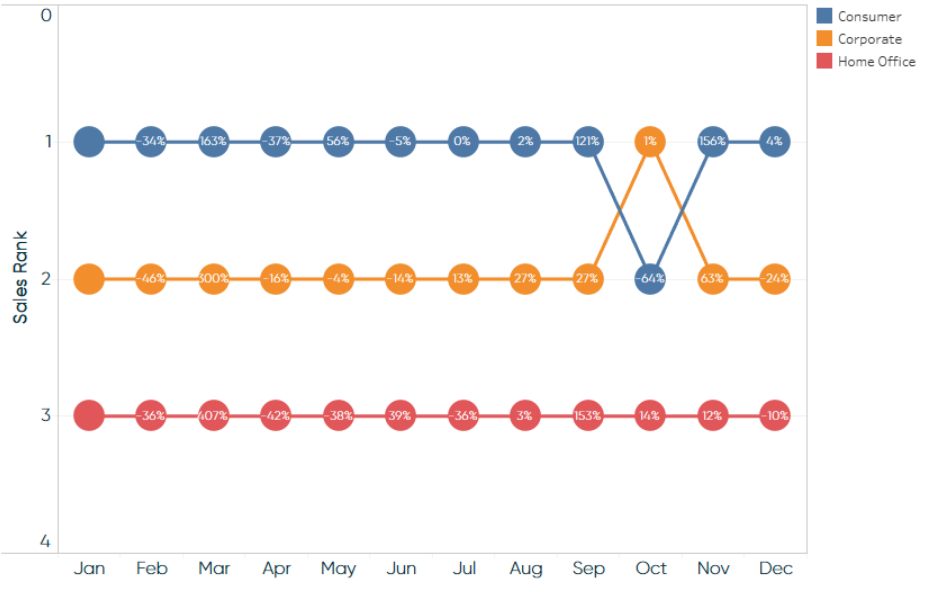
Whatever the case may be, I've found bump charts to be an engaging and easy-to-interpret means of visualizing ranks over time. Fellow Tableau Zen Master, Matt Chambers, created one of the most popular Tableau Public visualizations of the year with a bump chart and provided his technique at his site, Sir Viz-a-Lot.

In addition to being engaging and easy to understand, this chart type is very easy to make in Tableau.

This tutorial will show you how to make bump charts in Tableau, but also how to leverage a dual-axis to provide additional context on the view, and how to allow the end user to choose the measure being ranked and the dimension members being compared.

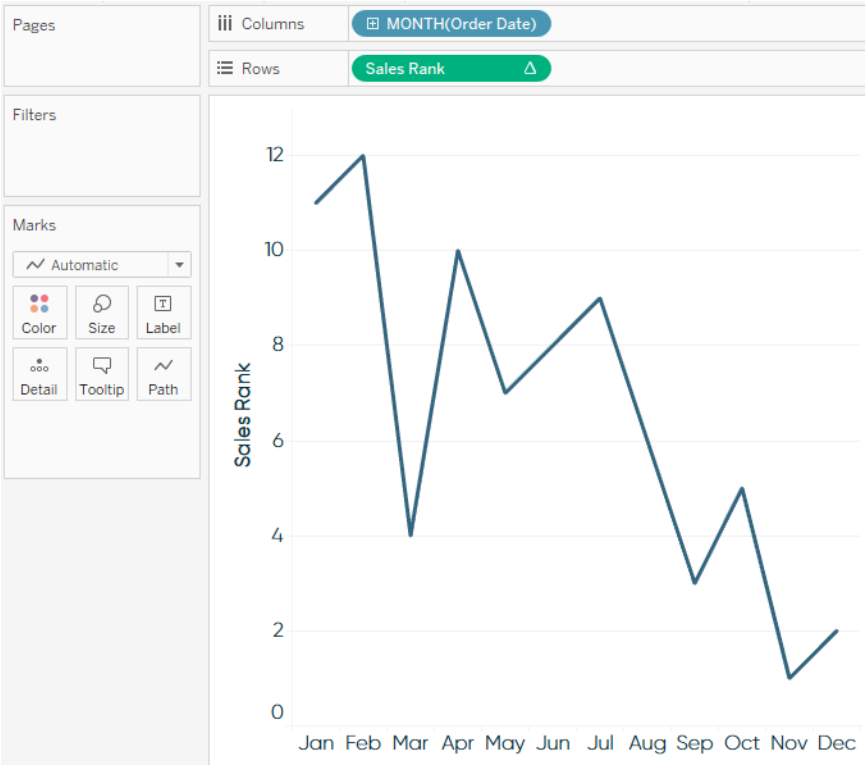
How to Make Bump Charts in Tableau

For the first part of the tutorial, we will re-create this bump chart that shows us how the segments from the Sample – Superstore dataset rank between January and December for the SUM(Sales) measure:

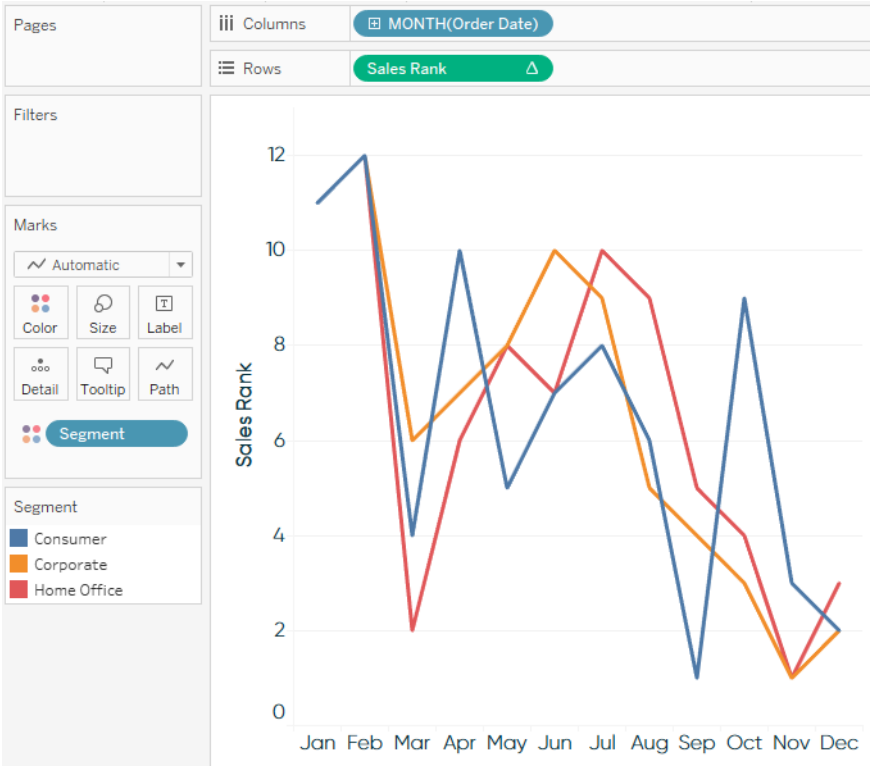


To create a bump chart, start by placing the element of time that you want to evaluate onto the Columns Shelf. To re-create the bump chart pictured, I placed discrete MONTH(Order Date) onto the Columns Shelf. I want to point out that by choosing the month date part, and because the dataset I'm working with here has four years of data, that each of the twelve months will have four years of data aggregated together. I'm doing this for illustrative purposes, but if you are using this chart type to show changes over time, make sure you choose the most appropriate date for your analysis.

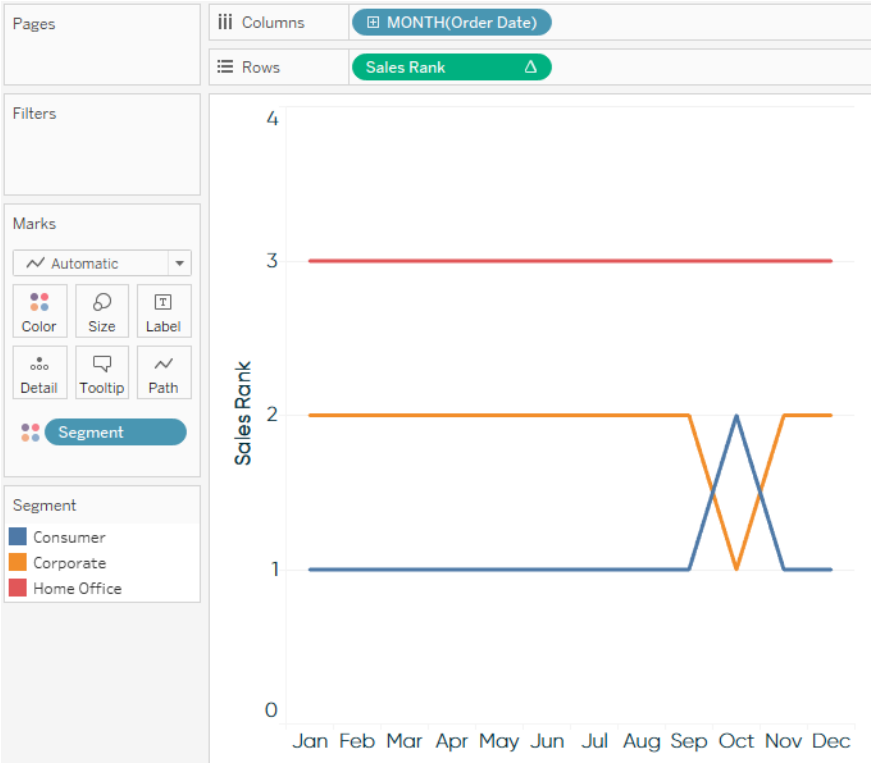
On the Measures Shelf, I will place a calculated field for the measure that we want to rank; in this case Sales. The formula for my calculated field is RANK(SUM(Sales)). At this point, the view looks like this:



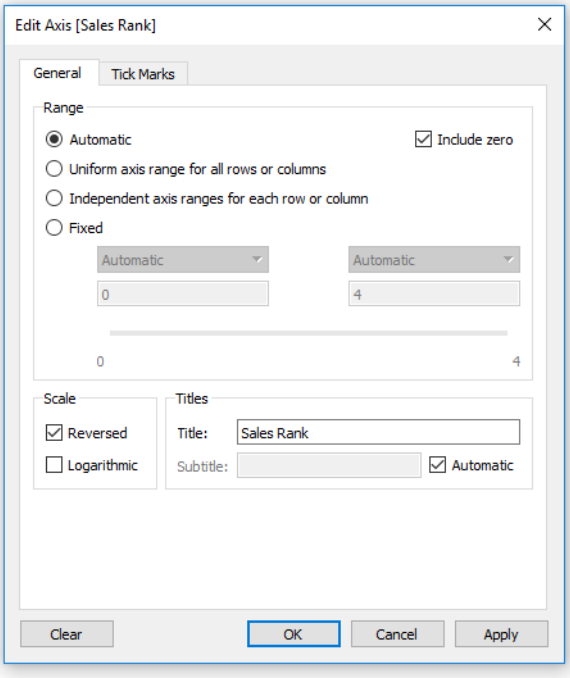
The RANK function acts as a table calculation in Tableau, so when it was added to the view, it was computed across the table from left to right. To rank the sales amounts by the Segment dimension, let's make the level of detail in the view more granular by dragging the Segment dimension to the Color Marks Card:



That got us closer, but we still need to change the RANK table calculation to compute for each segment. The delta symbol on the Sales Rank measure tells us that a table calculation is happening. To edit how the table calculation is being computed, right-click the Sales Rank measure, hover over Compute Using, and choose Segment:



At this point, we've technically got a bump chart in Tableau. It may be more intuitive for the best rank (i.e., 1) to be on top, so I will reverse the axis by right-clicking the Sales Rank axis, clicking Edit Axis, and checking the box to reverse the scale:



We have a bump chart, but wouldn't it be great to provide some additional value to the analysis? It's good to know that the consumer segment led the way for the first nine months of the year, but we don't know if the sales for that segment were going up or down throughout the year. To provide that context I will create a dual-axis combination chart. The first axis will be the bump chart that we just set up, and the second axis will be a dot plot that displays the percent change in month-over-month sales.

To make this bump chart dual-axis, duplicate the Sales Rank measure, place it on the opposite axis, and right-click the right axis to click Synchronize Axis.

At this point, we have the same line graph twice, with one laying on top of the other. Now that we duplicated the Sales Rank measure, we have two separate sets of Marks Cards for Sales Rank, and they can be edited independently. I will change the mark type on the second Marks Shelf for Sales Rank to Circle to make my bump chart look a little bit nicer: