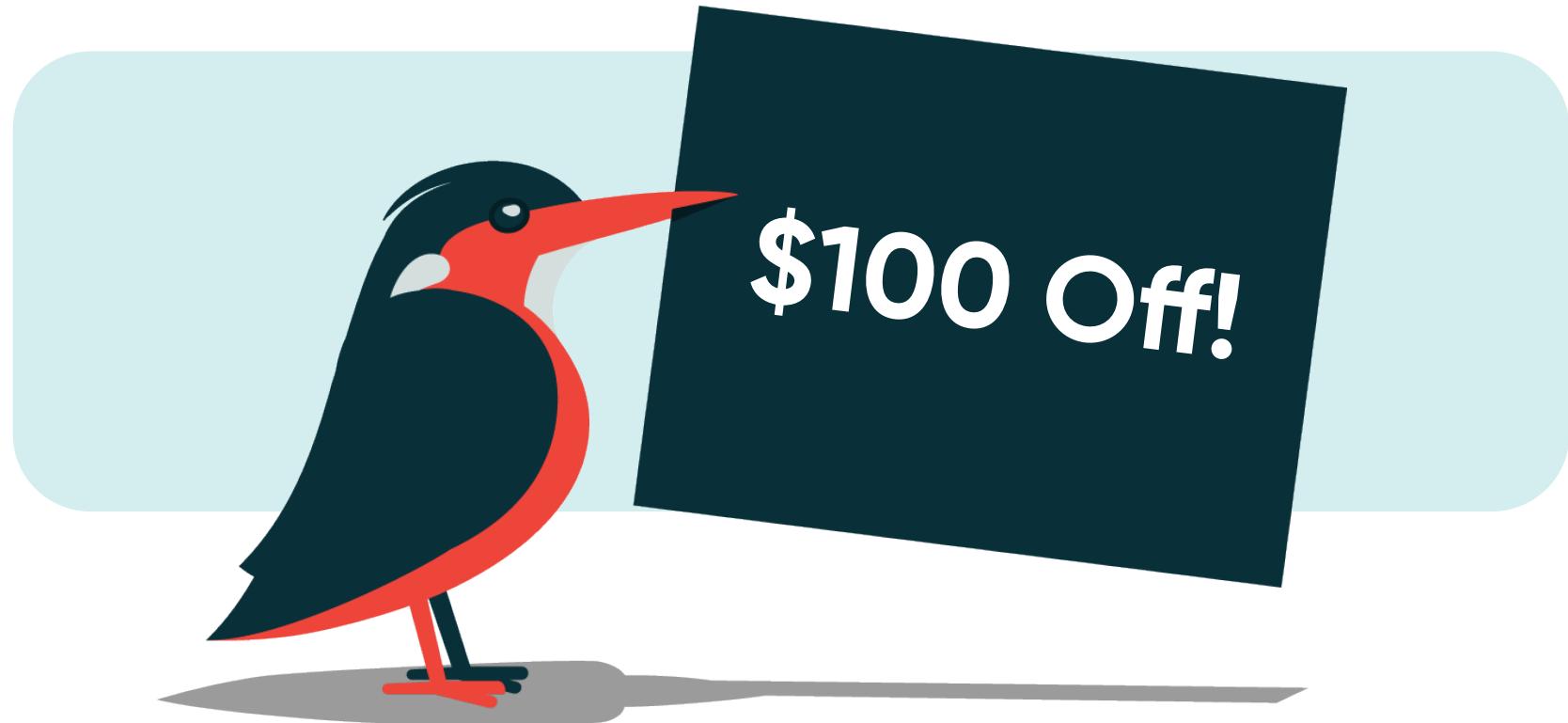
The background features a stylized landscape with rolling green hills under a blue sky with yellow clouds. In the foreground, there's a dark green forest at the bottom, a small tree with red and black leaves on the left, and a large grey mountain peak in the center. A red banner with white text is overlaid on the bottom.

# Tableau Tips

33 Ways to Tableau by Ryan Sleeper

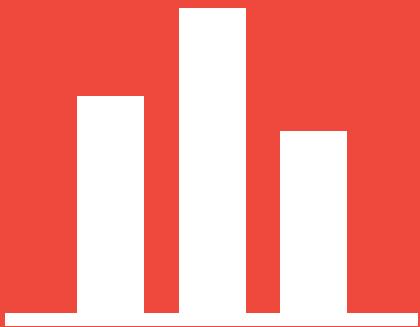


Thank you for subscribing and downloading this e-book. As a thank you, use code PLAYFAIR for an extra \$100 discount at a Tableau Training with Ryan Sleeper event. Learn more at [ryansleeper.com](http://ryansleeper.com).

# Table of Contents

<b>Chapter 1:</b> 3 Ways to Make Beautiful Bar Charts in Tableau	1
<b>Chapter 2:</b> 3 Ways to Make Lovely Line Graphs in Tableau	18
<b>Chapter 3:</b> 3 Ways to Use Dual-Axis Combination Charts in Tableau	33
<b>Chapter 4:</b> 3 Ways to Use Tableau Dashboard Actions	47
<b>Chapter 5:</b> 3 Ways to Use Google with Tableau Dashboard Actions	63
<b>Chapter 6:</b> 3 Ways the Describe Feature Can Help You Use Tableau	74
<b>Chapter 7:</b> 3 Ways to Add Alerts to Your Dashboards	83
<b>Chapter 8:</b> 3 Ways to Use Alt-Code Characters to Enhance Your Tableau Dashboards	90
<b>Chapter 9:</b> 3 Ways Psychological Schemas Can Improve Your Data Visualization	101
<b>Chapter 10:</b> 3 Ways to Make Charts More Useful Than Pies	108
<b>Chapter 11:</b> 3 Ways I Make Data Visualization Decisions	116

# Chapter 1



## 3 Ways to Make Beautiful Bar Charts in Tableau

When it comes to data visualization, bar charts are still king. With all due respect to my other favorite fundamental chart types such as line graphs and scatter plots, nothing has the flexibility, ease of use, and ease of understanding, as the classic bar chart. Used to compare values of categorical data, bar charts work well because they take advantage of a basic preattentive attribute: length. Our ability to process the length of bars with extreme efficiency and accuracy makes the bar chart arguably the most powerful data visualization choice available to us.

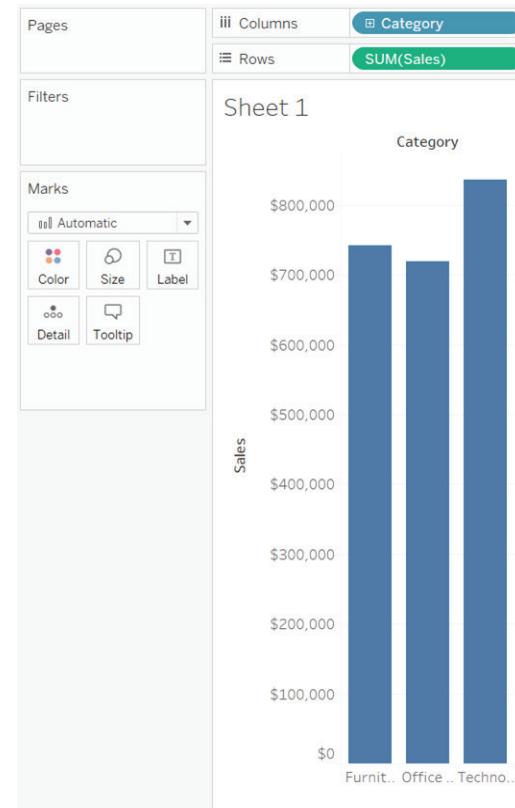
The invention of the bar chart is credited to William Playfair, with his *Exports and Imports of Scotland to and from different parts for one Year from Christmas 1780 to Christmas 1781* being the first appearance. Extraordinarily long and descriptive titles aside, bar charts have been making an impact for a long time. In fact, I hypothesize that the fact bar charts have been around for so long is one of the reasons some attempt to find a “more engaging” chart type to tell their data story.

This chapter attempts to add some love for bar charts by sharing three ways to make them more engaging in Tableau.

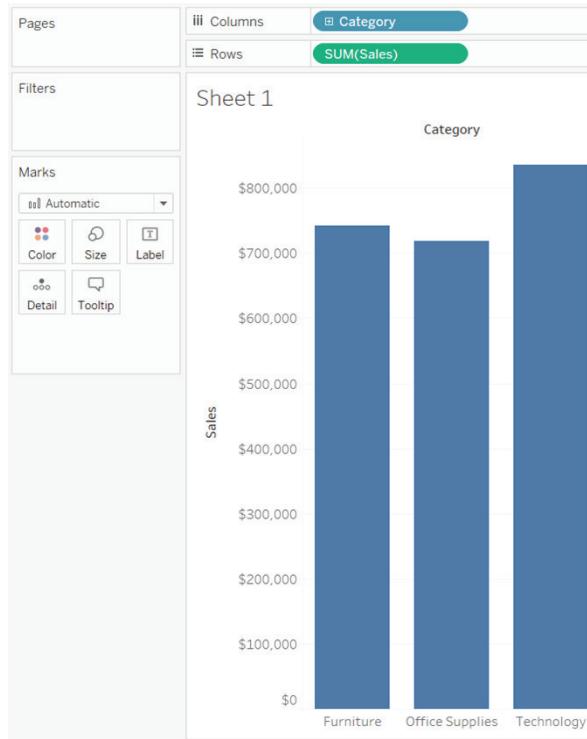
# 1. Use Formatting Available in Tableau

My first tip for making beautiful bar charts in Tableau is to use the formatting options you already have available in Tableau. Consider the following Sales by Category bar chart pictured to the right that shows all of the default Tableau settings.

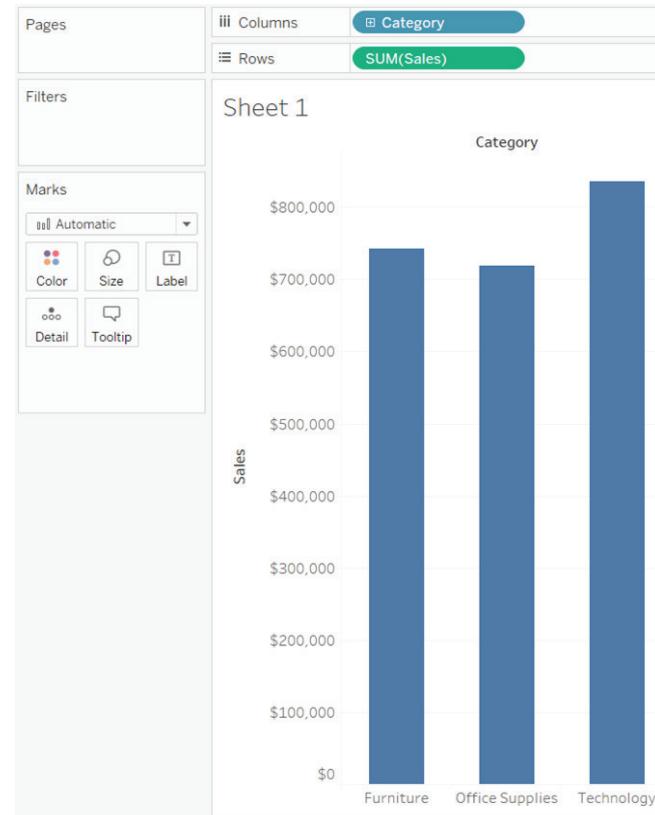
This bar chart gets the job done, as you can immediately decipher that Technology leads the way with over \$800,000 in sales, Furniture contributes the second most, and Office Supplies contribute the least. However, there are several opportunities to make this bar chart more engaging and effective. The most obvious of which is to widen the columns so the categories can be read.



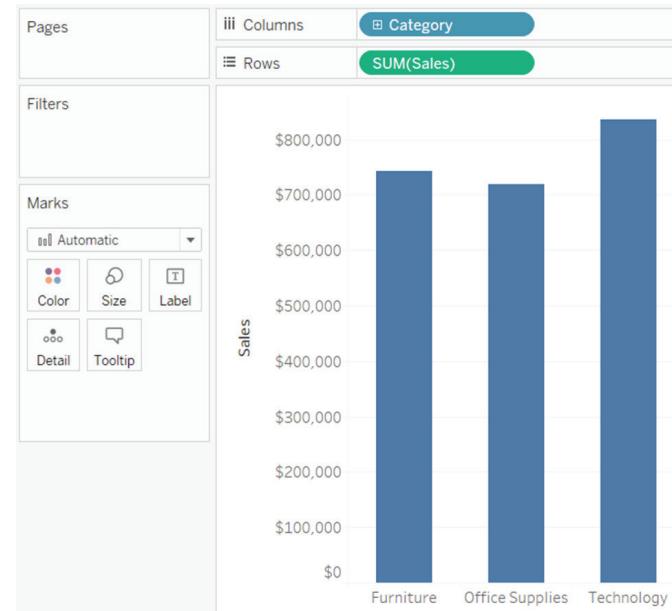
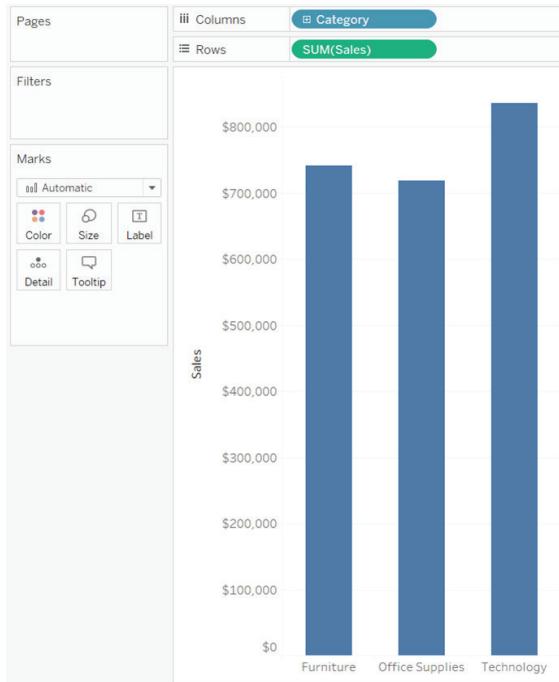
Making the columns wider makes the bars themselves wider. In my opinion, these bars are now too heavy relative to the rest of the visual. The next step I'll take is to reduce the size of the bars by clicking on the Size Marks Card and dragging the slide to the notch in the middle.



The next tip is arguable, but I'm not as descriptive as William Playfair was above with his 110-character chart name. In my experience, the context of the chart is provided in surrounding text and/or dashboard titles, so I am going to hide the sheet name by right-clicking on the title and choosing "Hide Title". I am also going to right-click on the bar chart header, "Category", and click "Hide Field Labels for Columns". If this is a standalone visualization, I recommend keeping the title.

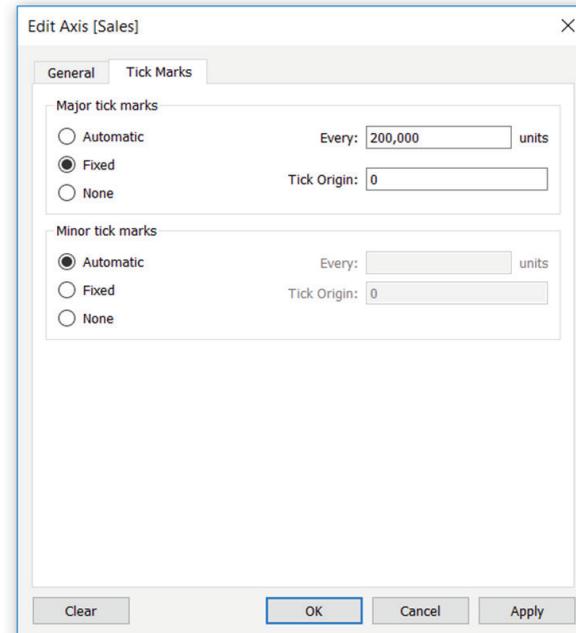


The bars in this chart are unnecessarily tall because there is not much variance between the categories in this analysis.

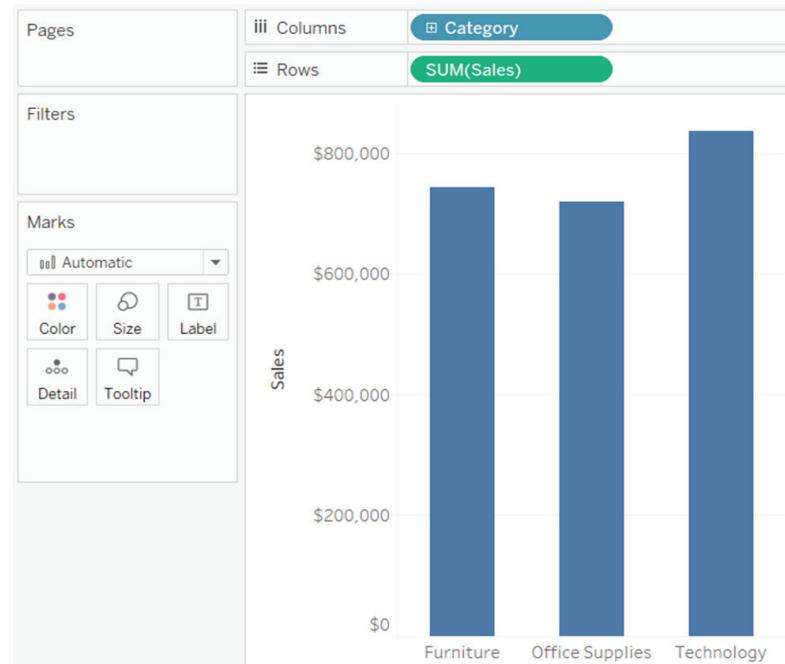


Here's how the bar chart looks after I reduce the height by about 40%.

Take this next step on a case by case basis, but another side effect of having limited variance between the three bars is that there are too many gridlines and axis marks. This is negatively impacting the data-ink ratio and can be cleaned up. To reduce the number of axis ticks, right-click on the axis, click “Edit Axis...”, and navigate to the Tick Marks tab.



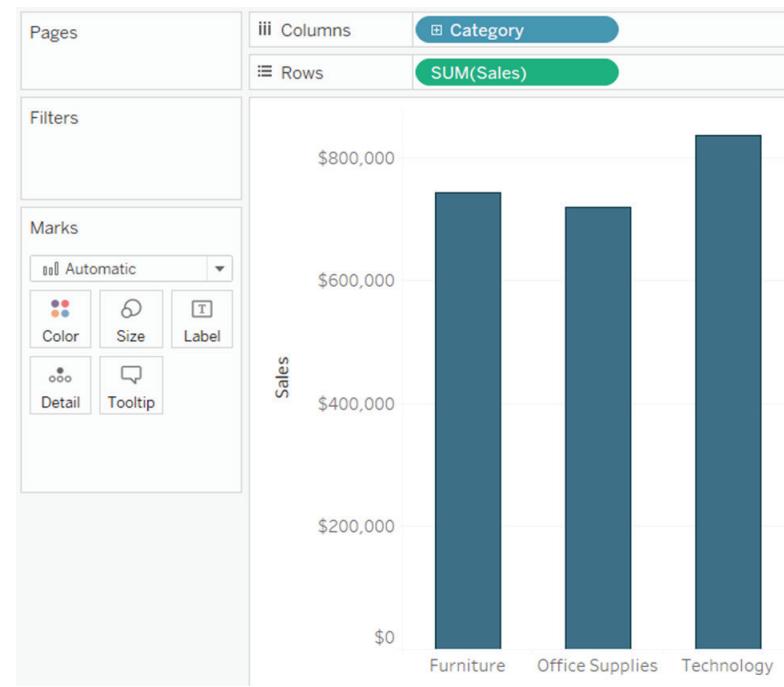
Here's how the bar chart looks after fixing the tick marks at 200,000 as pictured in the dialog box above.



Last, but not least: color. I have written quite a bit about color including [Leveraging Color to Improve Your Data Visualization](#) and [3 Ways Psychological Schemas Can Improve Your Data Visualization](#). There is so much to be said about color, but for the purposes of this chapter, I will offer just three thoughts on coloring your bar charts:

1. Avoid double encoding. The bars in our chart are already separated by category. Adding category again to the Color Marks Card to color each bar with a unique color is unnecessary and potentially confusing. The one caveat to this is if the colors are being used to provide a link between multiple visuals on a dashboard.
2. Reduce the opacity from 100% to 80% – 90% by clicking on the Color Marks Card and moving the opacity slider to the left. This is a very subtle technique for reducing the saturation of the color and making the visual a little easier on the eyes.
3. Use this as an easy opportunity to brand your data visualization. If you can't decide on a color, consider using a primary or secondary color from you or your end users' brand.

Here's how my final bar chart looks after choosing a secondary color from my personal brand and reducing the opacity of the bars to 90%. Note that I also added a very subtle border to the bars, which can be found in the options on the Color Marks Card.



## 2. Use Axis Rulers to Add a Baseline

In the last tip, I mentioned the data-ink ratio. This a concept introduced by Edward Tufte that essentially says you should dedicate as much “ink” on a data visualization to the data as possible. One way I sometimes achieve this is by hiding the axis altogether and adding labels to the bars. (see Fig. 1)

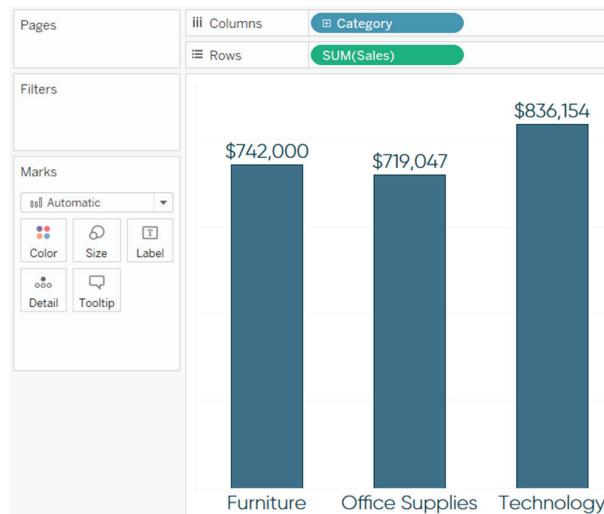


Fig. 1

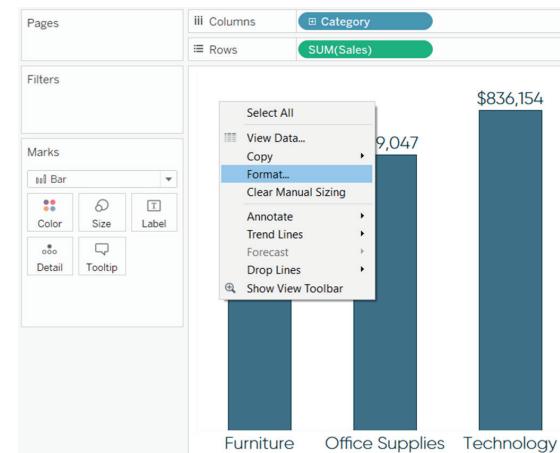
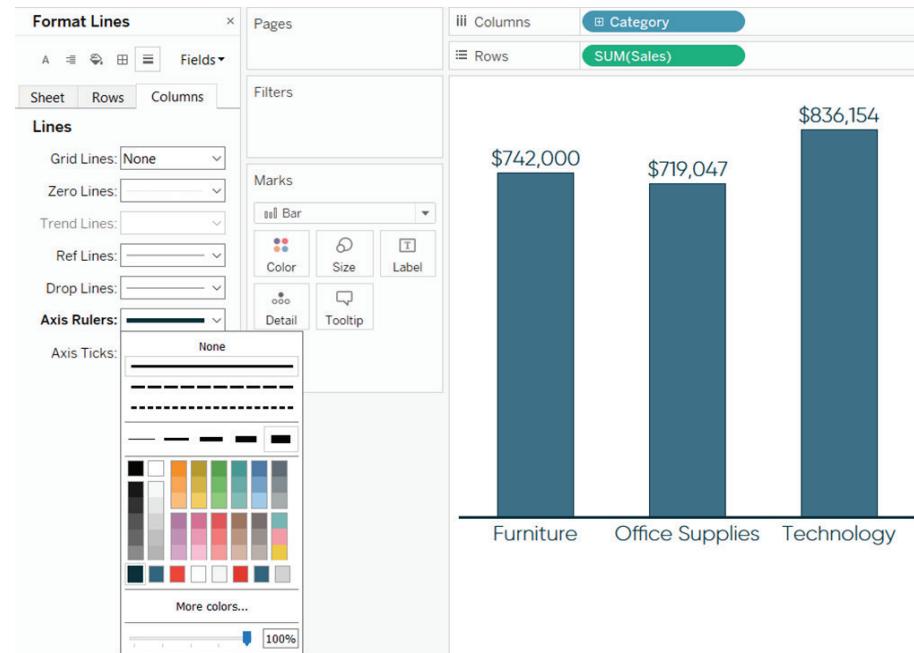


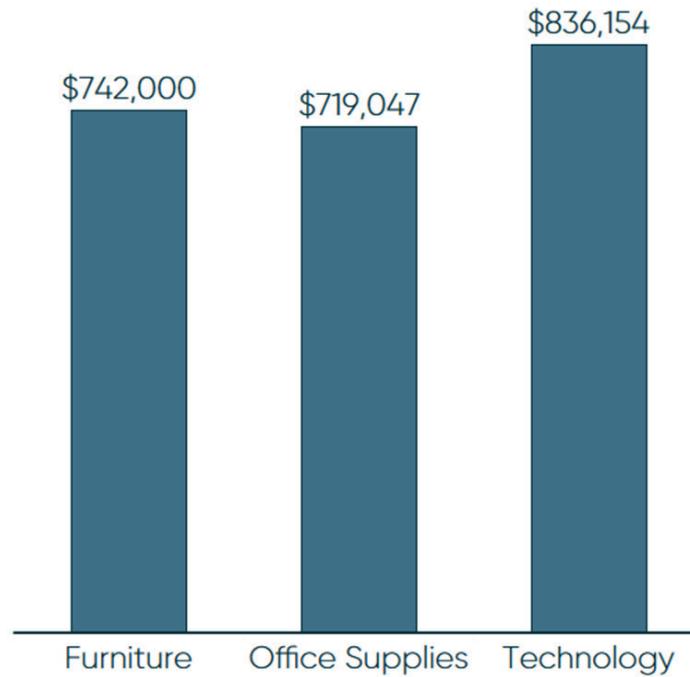
Fig. 2

To add to the first tip, formatting, I also made the font larger and in brand. I like this look, but don't like how the bars appear to be floating. What I would like to do is add a solid foundation for the bars to sit on; this provides a practical purpose and also enhances the design. The easiest way to add a baseline is to modify the formatting of the view's axis rulers. By default, axis rulers are set to be a very light, thin gray line. To make the line heavier and match the color of the bar chart you are creating, right-click on the view and click “Format...” (see Fig. 2)

This will open the formatting pane on the left. Navigate to the “Format Lines” tab and modify the formatting for the Axis Rulers for Rows and Columns. To clean up the view, I’m going to set the Axis Rulers for Rows at None, which removes the thin gray vertical line on the left side of the bar chart. I’m then going to format the Axis Rulers for Columns to be a solid, thick line that matches the color of the bars’ borders.



Here's how my final bar chart looks after removing all of the other lines except for the baseline.

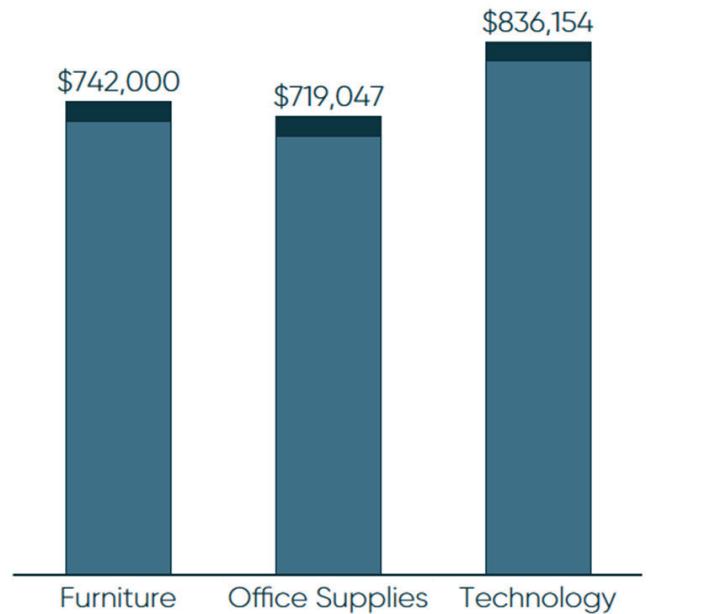


Compare this to the default Tableau bar chart in the first image above!

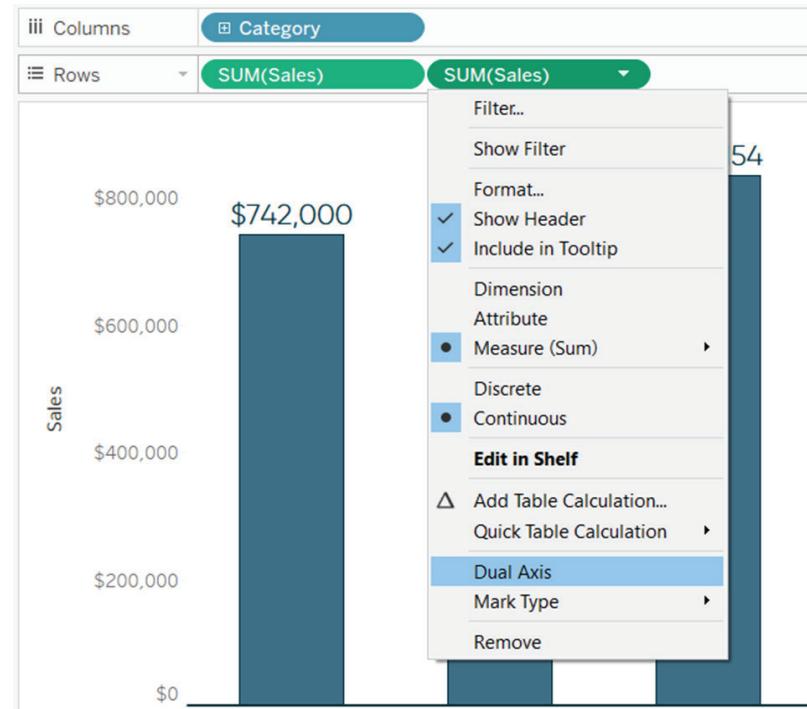
### 3. Add Caps to Bars

As I explained in [Tableau Blueprint 2: My Tableau Public Viz Views](#), capped bar charts have two benefits: (1) they enhance the design of a bar chart and (2) add value by potentially doubling as a secondary comparison point. For example, the size of the cap can represent the value of a goal or be colored by a different measure (i.e. length of the bars represent sales; color of caps represent profitability). While I like the style of capped bars and the subtle way they can provide additional information, beware they can easily be confused with stacked bar charts, and should be used with caution.

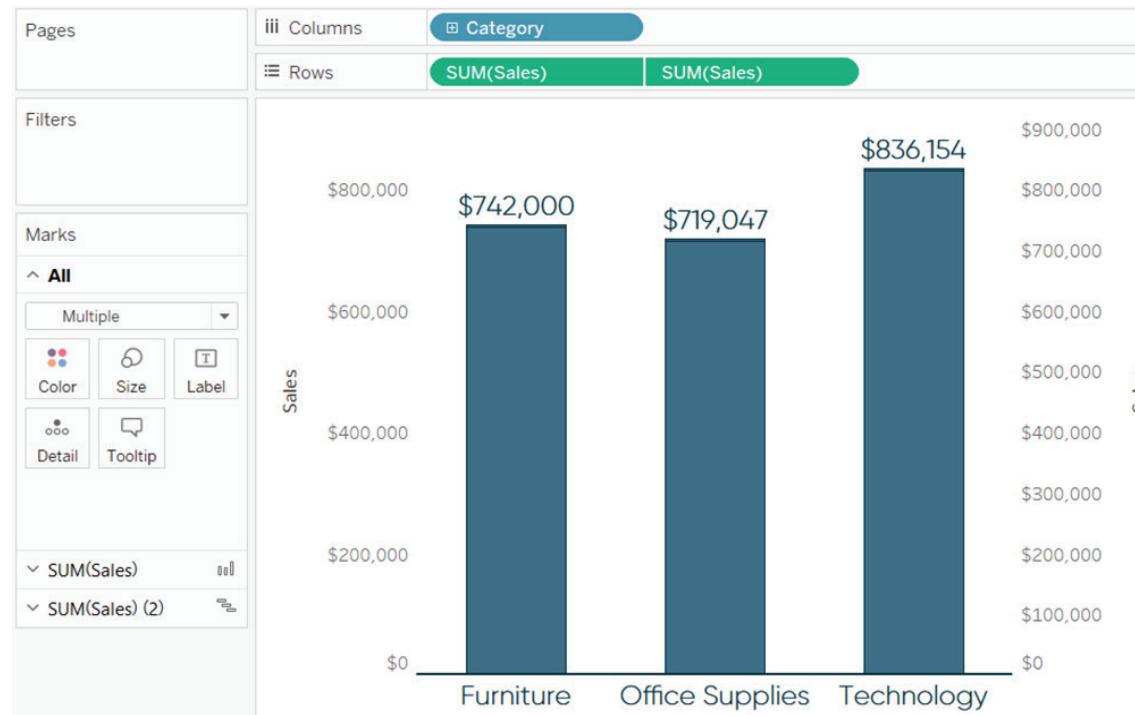
Here's what a capped bar chart looks like when added to our previous example.



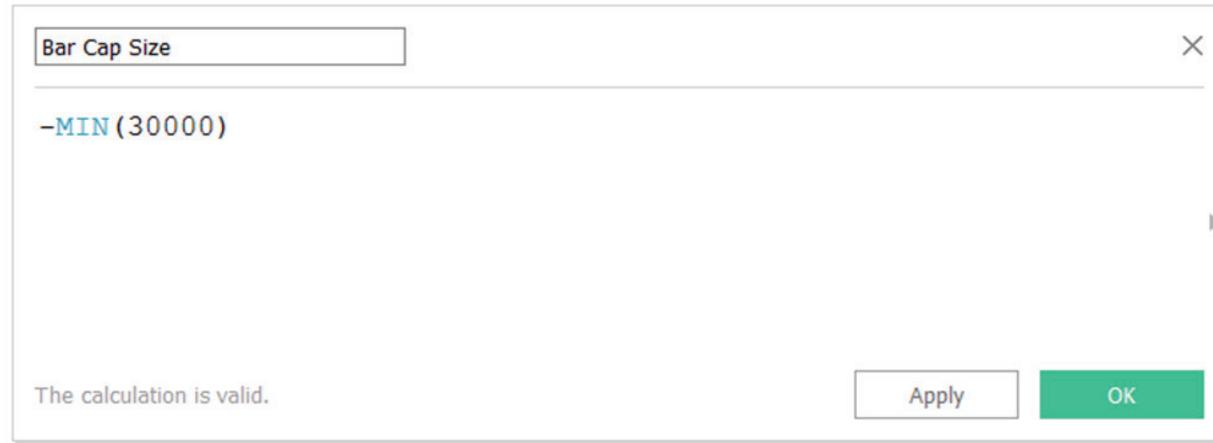
To create a capped bar chart in Tableau, start by adding a dual axis to the original bar chart. This can be accomplished by dragging the measure onto the Rows Shelf a second time, clicking on the second occurrence of the pill, and choosing “Dual Axis”.



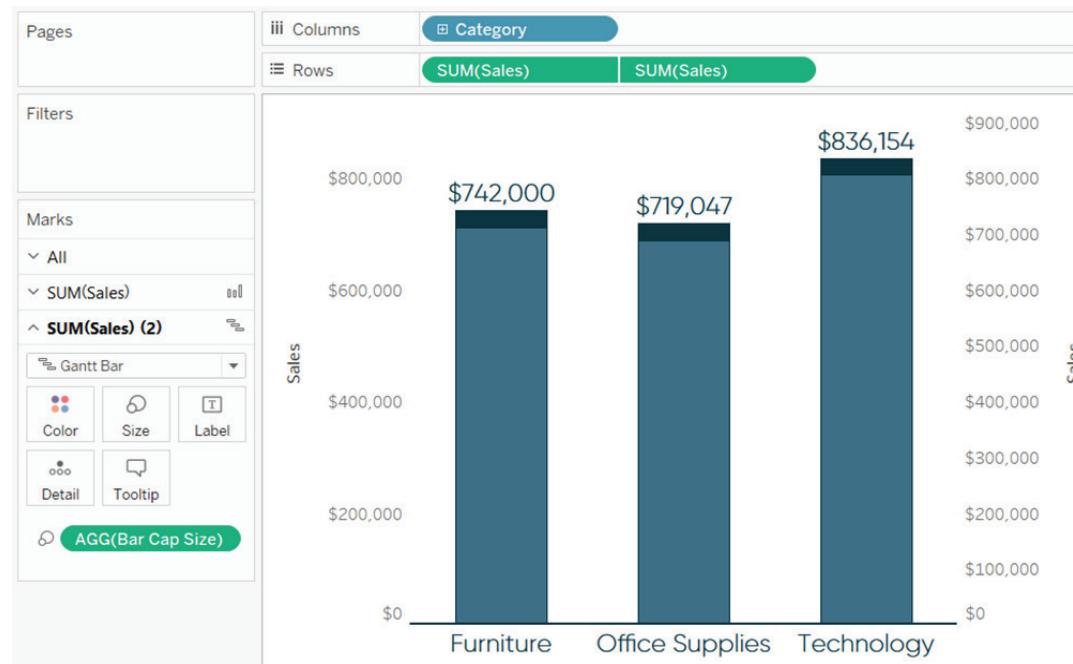
Once you've created the dual-axis chart, change the mark type of the first measure back to bar and the mark type for the second measure to Gantt Bar. Also ensure the axes are synchronized by right-clicking on the right axis and choosing "Synchronize Axis". At this point, my view looks like this:



The next step to creating capped bar charts in Tableau is to create a calculated field for the size of the caps. The calculation is simply  $-\text{MIN}([\text{insert size}])$ . It is critical to add the negative sign before the bar size to ensure the caps go the correct direction. Here's how my calculated field looks when setting the cap size at 30,000:

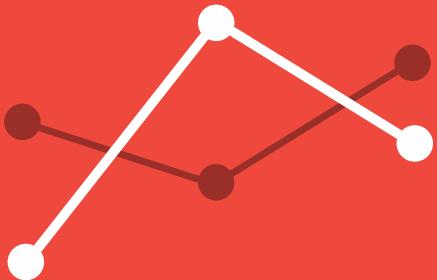


Lastly, drag this newly created Bar Cap Size calculated field to the Size Marks Card on the Marks Shelf for the measure you set as a Gantt Bar.



To finalize my capped bar chart, I colored the caps to match the bar borders and hid both axes.

# Chapter 2



## 3 Ways to Make Lovely Line Graphs in Tableau

Due to the popularity of [3 Ways to Make Beautiful Bar Charts in Tableau](#), I decided to follow it up with some ideas for making your line graphs more engaging in Tableau. Line graphs are a close second to bar charts as my favorite fundamental visualization type and are the obvious choice for evaluating trends over time.

Like bar charts, the invention of line graphs is generally credited to William Playfair at the end of the 18th century. Also like bar charts, I blame their age and people's familiarity with line graphs as the reason some data visualization enthusiasts look for "more engaging" choices. Line graphs have stood the test of time and their effectiveness cannot be denied. I'm hoping that these three approaches help cement line graphs as a top choice for your Tableau data visualizations.

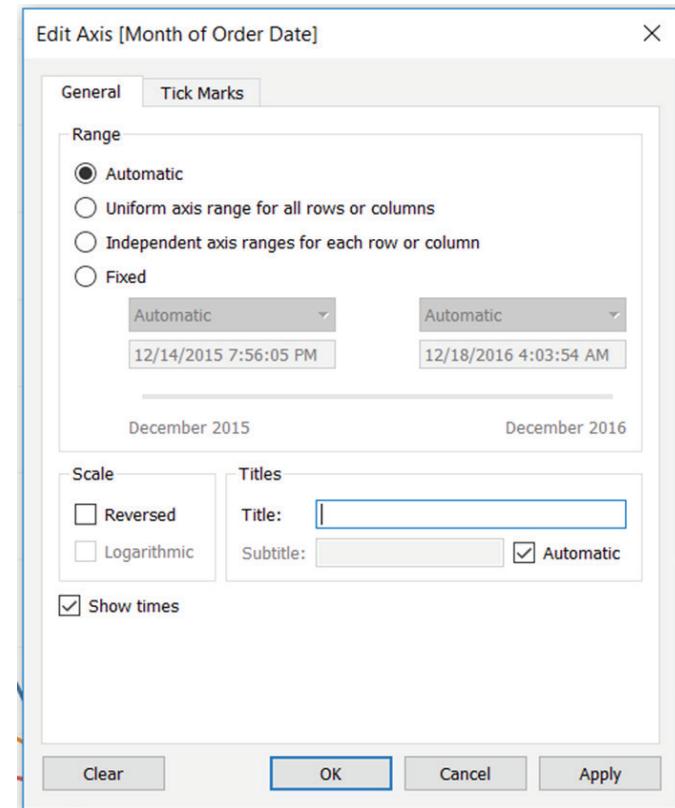
# 1. Use Formatting Available in Tableau

My first tip for making line graphs more engaging is to use the formatting options available to you in Tableau. Consider the following sales by segment line graph with all of the default Tableau format settings. Note that I've used the Month([Order Date]) field as continuous and filtered the data to the year 2016.



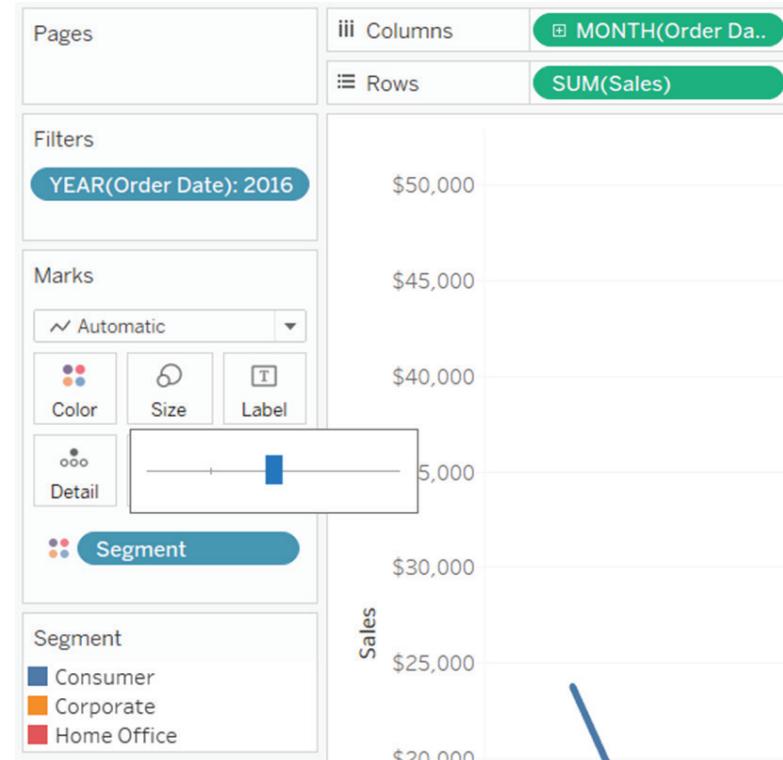
A great deal of thought went into Tableau's default formatting including the fonts, colors, and mark sizing. So as can be expected from my data visualization tool of choice, this graph is already getting the job done and providing good insights. That being said, if one of your objectives is to make the graph as engaging as possible, there are several opportunities to add value to this view.

First, I've never personally liked the axis title that is used to designate the date granularity when you are using a date field continuously. While I appreciate the clarity the axis title provides, this information is typically already implied in the graph itself and usually stated in surrounding context (i.e. "This is a graph about Sales by Month last year"). I almost always get rid of this axis title. Unfortunately, if you right-click on the axis and deselect "Show Header", both the axis title and month names disappear. To get rid of only the axis title but keep the month names, right-click on the axis, and choose "Edit Axis...". This will open a dialog box where you can change the axis title. The trick is to simply delete all of the text on this line; then click the OK button.



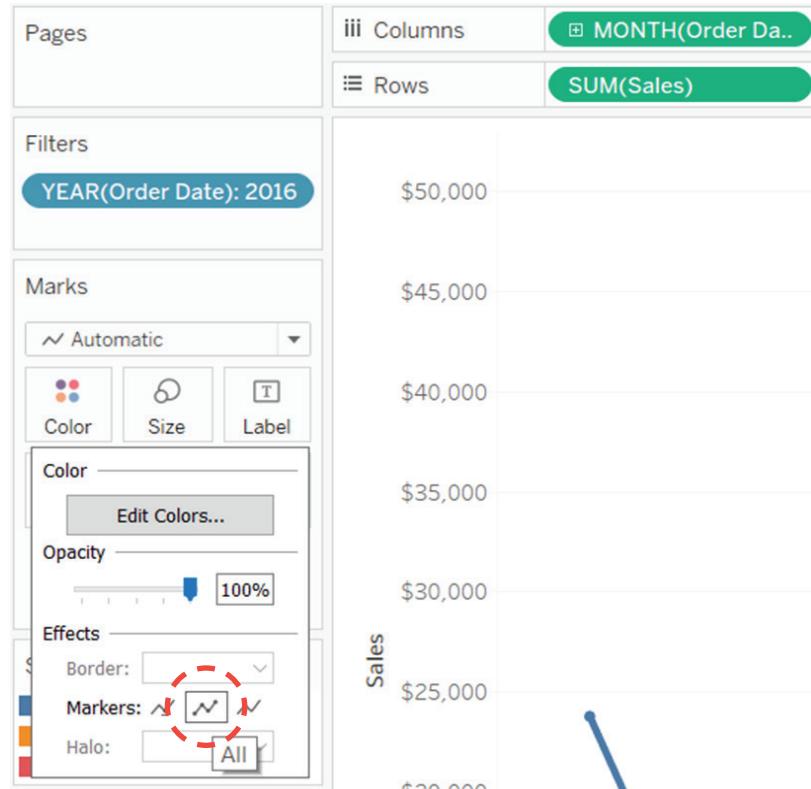
Next, if this is a standalone visualization, the weight of the lines is too light. To change the weight of the lines, click on the Size Marks Card and drag the slider left or right until the lines are weighted as desired. I normally find that the second ‘notch’ on the Size Marks Card is a good option for lines.

Take this on a case by case basis because sometimes heavier lines won't look good when used in a tighter space (such as one of several components on a dashboard).

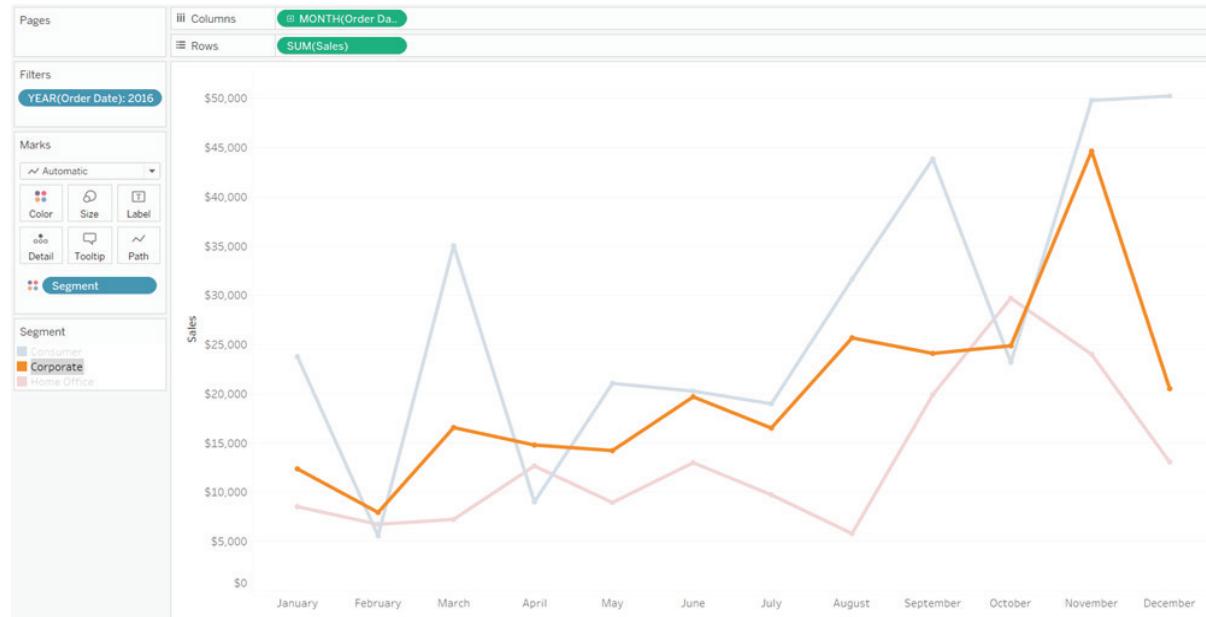


The next formatting tip addresses one of the most common questions I am asked while [speaking about Tableau](#): Markers. Markers are the small circles that are sometimes seen on data points of a line graph. Markers serve a practical purpose of telling the end user where there is data in a subtle way, and they also are an easy way to enhance a line graph.

To add markers, click on the Color Marks Card and select the second Markers option.

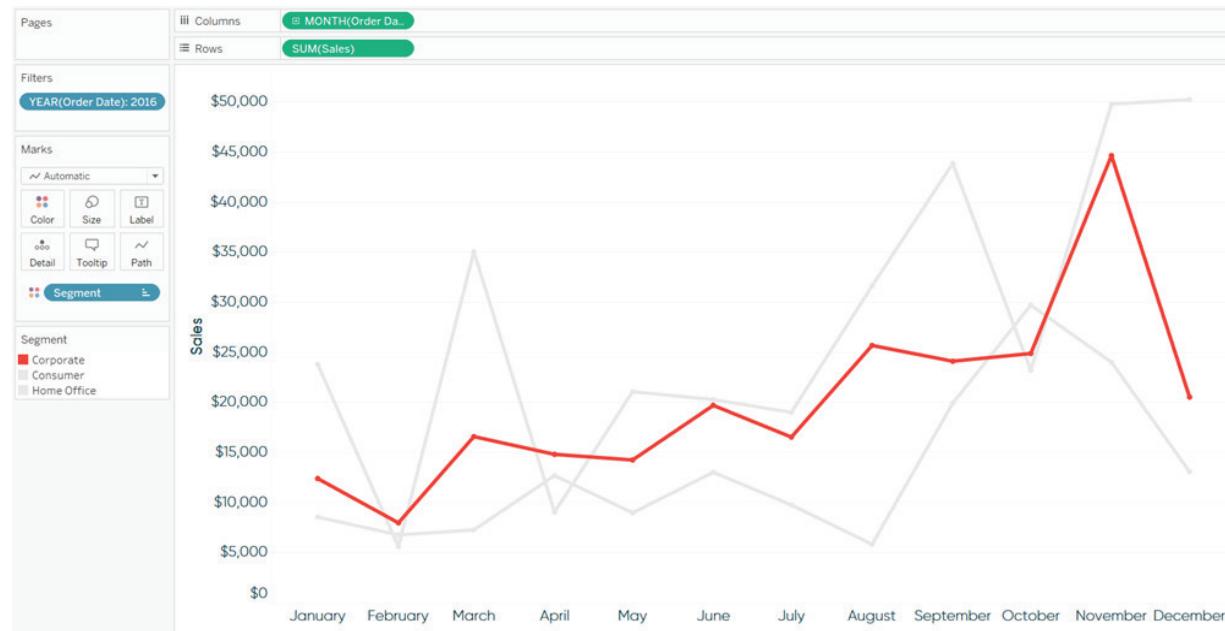


The next tip is optional, but many times I like to make the line or insight that I care about “pop” on the view. This can be accomplished in several ways; the easiest of which is to click a dimension member on the color legend.



This approach also allows the end user to make the selection that they care about. Another way to highlight a dimension member would be to show a highlighter for the Segment dimension. This can be accomplished by right-clicking on the Segment dimension (currently on the Color Marks Card) and choosing “Show Highlighter”.

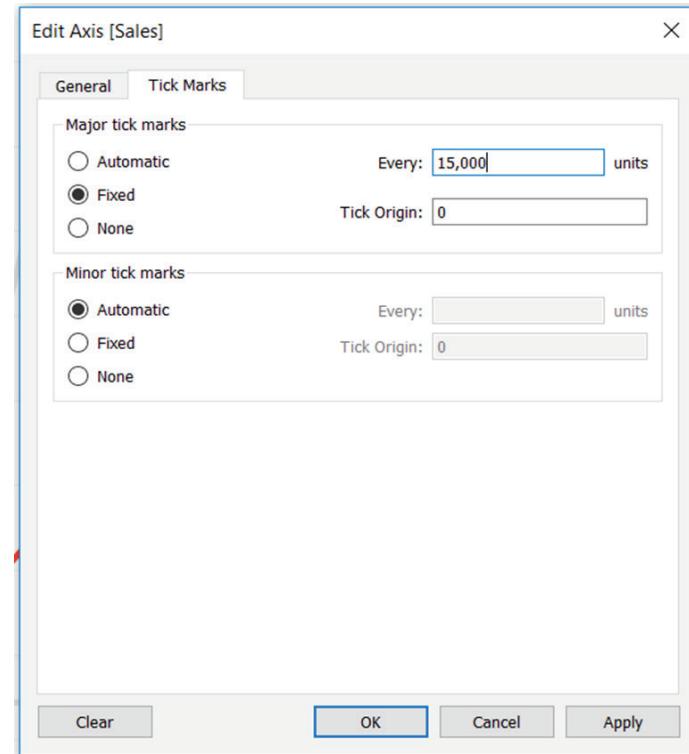
There are other times that you want to make the highlight more permanent. Maybe I am the manager of the corporate segment, so I always want it highlighted throughout all views. Or maybe I'm not allowed to see the performance of the other dimension members, but my boss wanted me to see my segment's performance in context of the others. There is a way to do this with parameters which is beyond the scope of this chapter, but it can also be accomplished easily by changing the colors of the dimension members. Here's how my line graph looks after "graying out" two out of three dimension members by clicking on the color legend and remapping the colors. To finalize the view, I've also put the colors and fonts in brand.



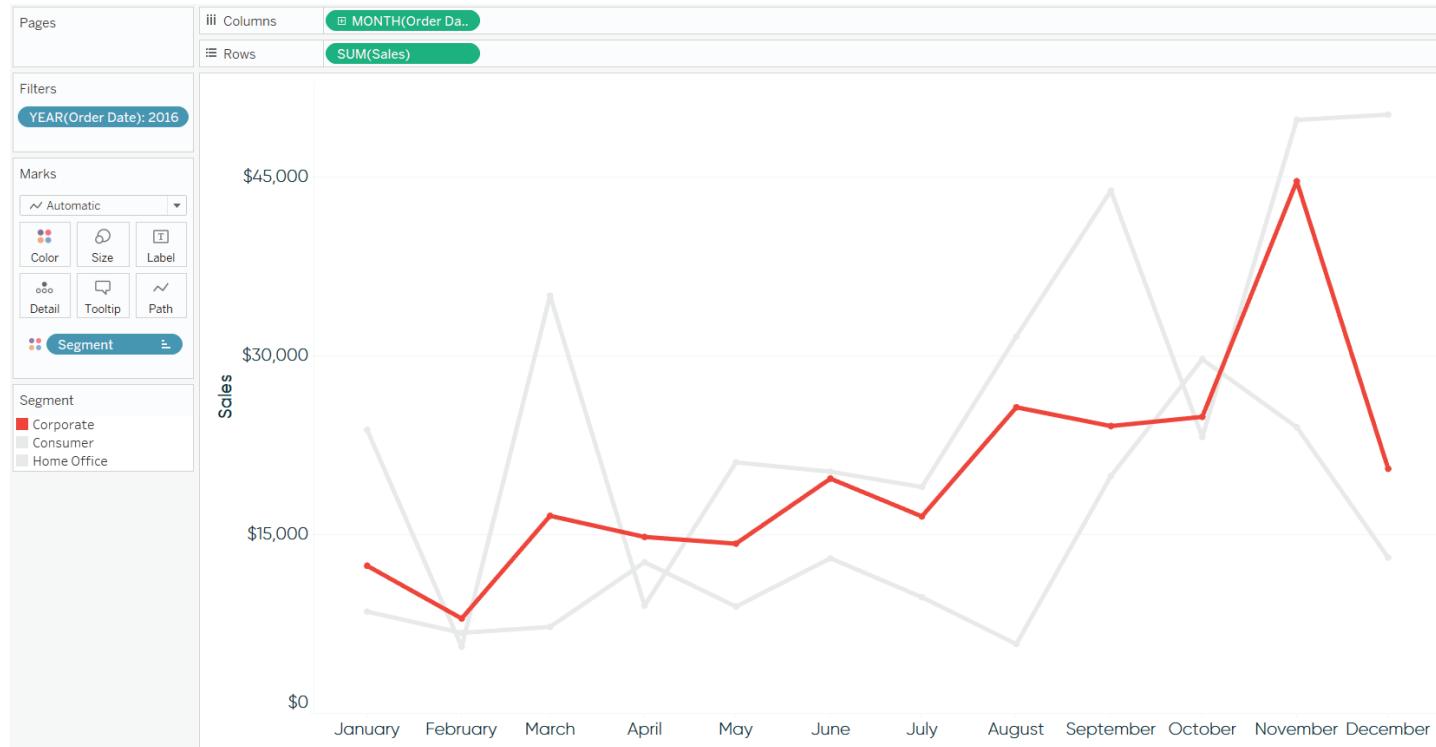
## 2. Maximize the Data-Ink Ratio

The next tip should be considered for any visualization you create: Maximize the ‘Data-Ink Ratio’. The Data-Ink Ratio is a concept introduced by Edward Tufte, who says you should dedicate as much ‘ink’ on a view to the data as possible. This means getting rid of unnecessary lines, effects, and anything else that detracts from the data itself.

There are a couple of opportunities to make our line graph even more minimalistic than it already is. First, the axis ticks on the Sales axis can be reduced by at least half. The axis ticks for Sales on this line graph are the multiples of \$5,000 going up the Y-axis. By default, these are often too granular for my taste, and cause too many extra lines and too much ink. To fix the axis ticks, right-click on the Y-axis, click “Edit Axis...”, and navigate to the “Tick Marks” tab. You should beware that this option truly does fix the axis tick marks, so if you filter the view later, this setting will stick.



Here's how my line graph looks after fixing the tick marks at 15,000 units.



This same approach can be taken on the X-axis with months if you are using a continuous axis. This is not always an appropriate choice, but often the range of the axis is enough to communicate what the visualization is about. I often show only the starting point and end point of the X-axis of a line graph, especially when using sparklines. Here's how my line graph looks after fixing the axis ticks at eleven months with an origin of January 1st.



To finalize the view, I will hide the axis rulers and change the gridlines to dotted. Both of these options are found in the formatting pane which you can reach by right-clicking on the graph and choosing “Format...”.

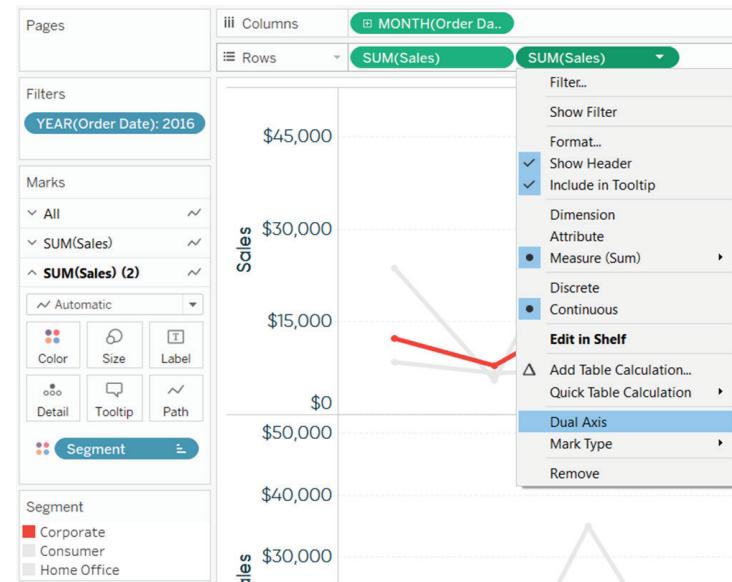


Compare this to the default line graph at the beginning of the chapter!

## 3. Leverage the Dual-Axis

There are at least two ways the second axis on a line graph can be used to enhance the visualization. Let's say that we really like the markers from the formatting tip above, but would like to make them more pronounced. To do this, start by putting the Sales measure on the Rows Shelf a second time, right-click on it, and choose "Dual Axis".

Ensure the axes line up by right-clicking on the right axis and choosing "Synchronize Axis".



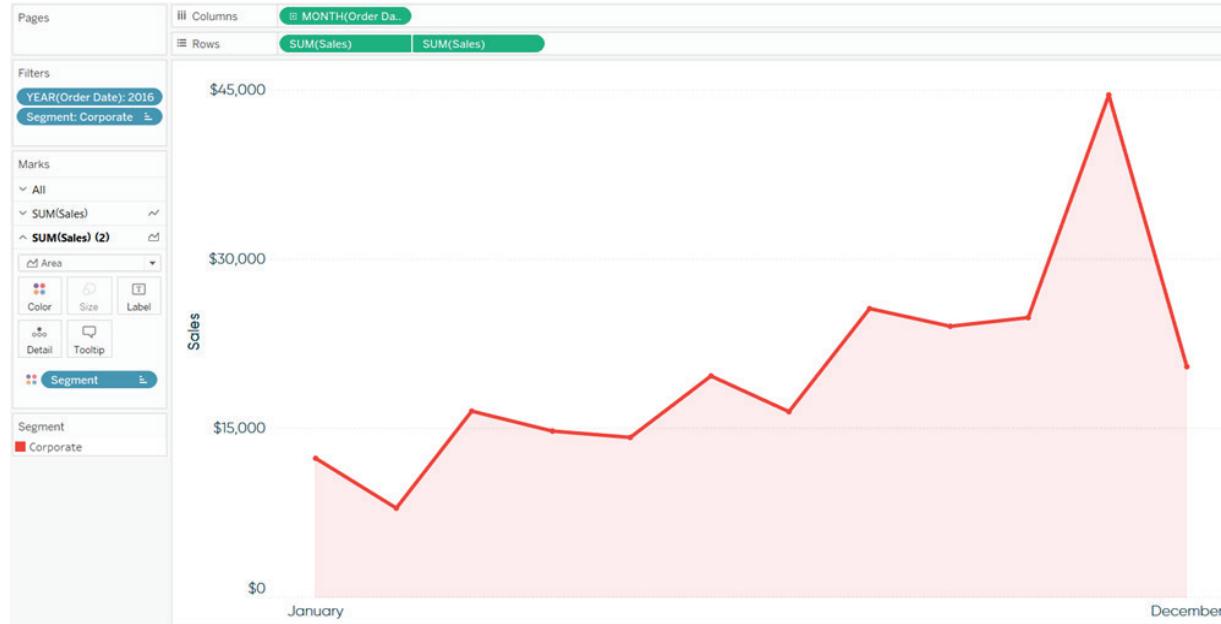
Now that you've got two measures on the Rows Shelf, you've also got two sets of Marks Cards on the Marks Shelf. These can be edited independently, which means you can keep the mark type for the first occurrence of Sales as Line, but change the mark type for the second occurrence of Sales to Circle. After sizing the circles, which are your new markers, hide the right-axis by clicking on it and deselecting "Show Header".



The last tip for using the dual-axis to enhance a line graph is a design inspired by Google Analytics. When showing only one measure and dimension member at a time, Google Analytics shows a dual-axis line graph / area graph combination. To show you this approach, I am going to filter the view to only the segment that I care about: Corporate.



Now that we've isolated a single dimension member, change the Mark Type on the second occurrence of Sales to Area. Lastly, reduce the opacity of the area to something very light, such as 10%. Here is how my final view looks.



# Chapter 3



## 3 Ways to Use Dual-Axis Combination Charts in Tableau

Dual-axis combination charts, or combo charts, are named that because they have two axes and they display a combination of different mark types. For example, you can create a visualization that displays a measure with bars on one axis and another measure as lines on the second axis. This is one of my favorite chart types to use in Tableau because the ability to add a second axis, and control the axes independently of each other, unlocks some additional flexibility. This new-found flexibility creates several practical applications that can be used to improve your analysis, user experience, and design.

This chapter will show you how to make a dual-axis combo chart in Tableau as well as three different ways to use them: (1) their traditional use (2) a method for making your end user part of the story and (3) an option for improving the aesthetics of your dashboard.

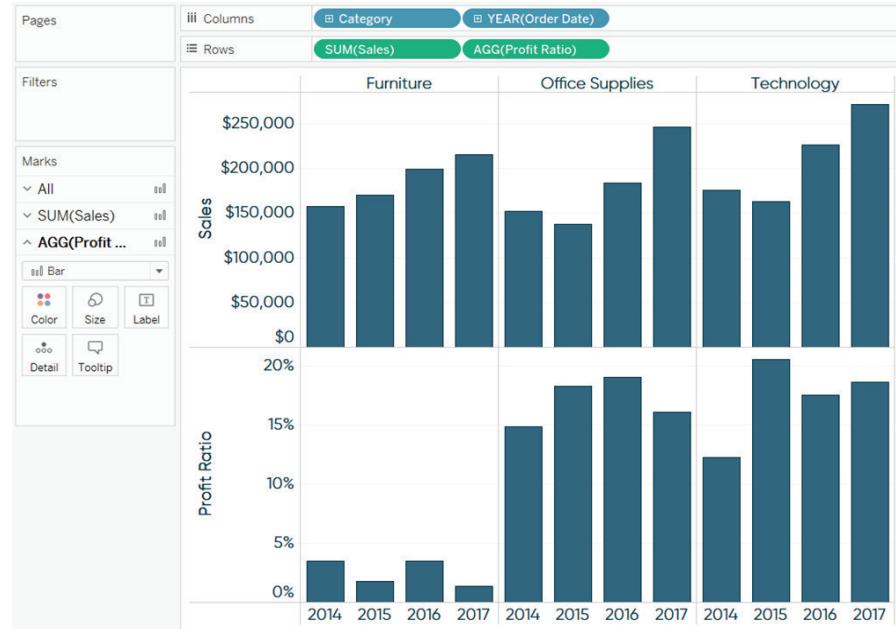
# 1. How to Make Tableau Dual-Axis Combination Chart

Let's start by making a traditional dual-axis combination chart using the Sample – Superstore dataset. Even if this is familiar to you, I'll be sharing a second approach that you may not know that will save you a click. This first chart will show sales by year as bars on one axis, and profit ratio by year as lines on the other axis; both measures will also be broken down by the Category dimension.

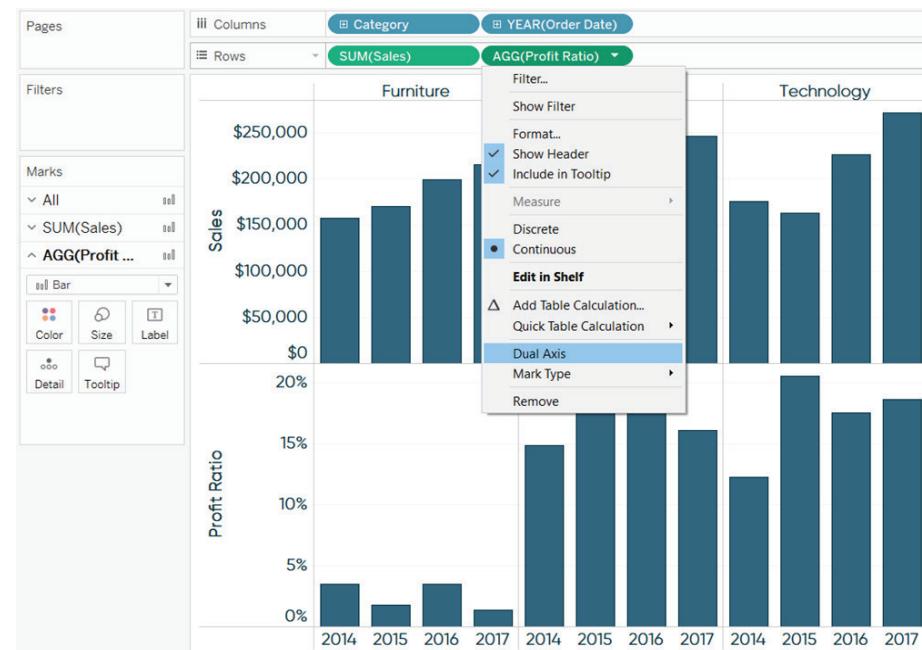


To begin, I like to create one of the charts, so I'll start with Sales by Year by Category.

Next, place the second measure (in this case, Profit Ratio) on the Rows Shelf.

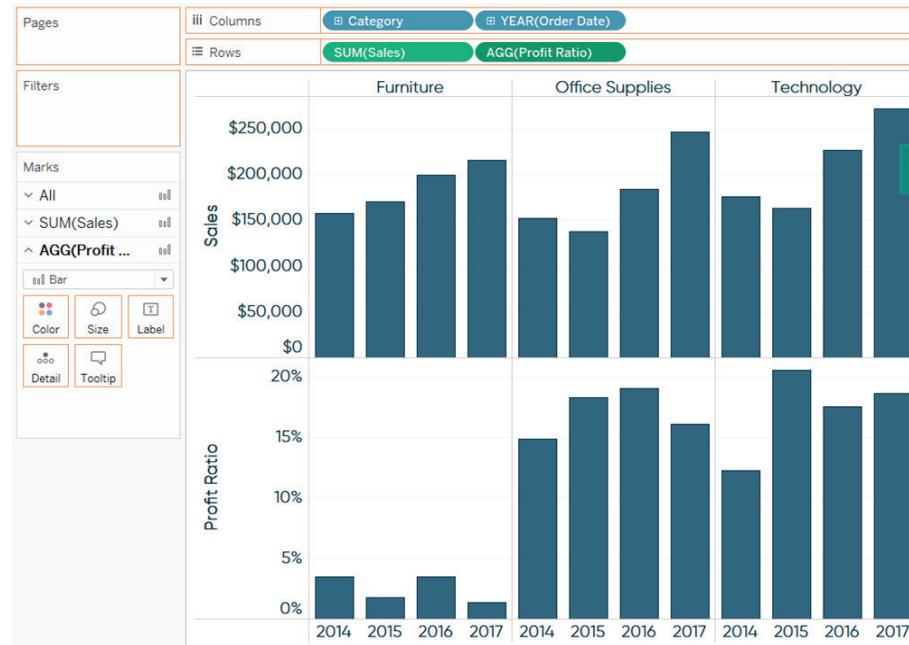


At this point, we have two individual bar charts on two rows. There are two ways we can convert these two separate bar charts into a dual-axis bar chart. First, and the way most people learn, is to click the second measure pill on the Rows Shelf, and choose “Dual Axis”.

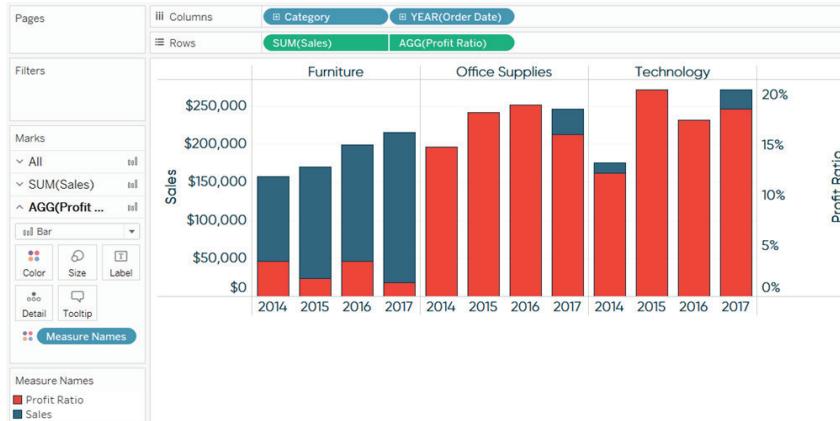


The second, and slightly more efficient method, is to hover over the axis of the second row. While hovering, a green triangle will appear in the upper left-hand corner of the axis. You can left-click on the green triangle and drag it to the opposite axis of the left axis on the first row. When you hover over the right side of the chart, Tableau will show you a dashed line; this is where the axis will be drawn after releasing the left mouse key.

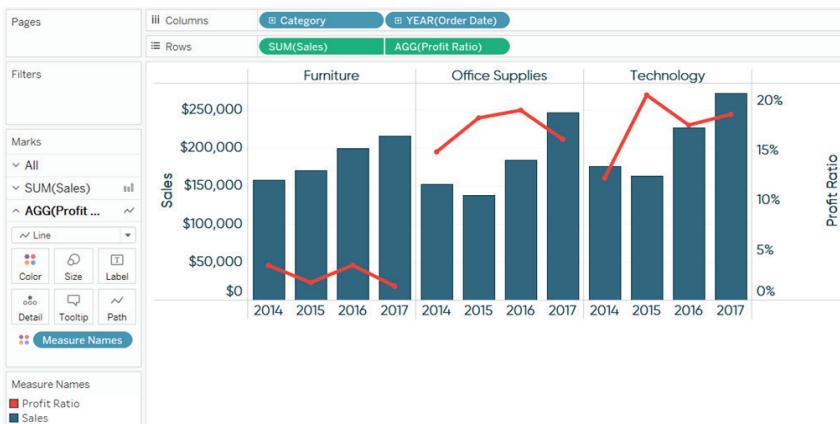
In both cases, we end up with a dual-axis bar chart.



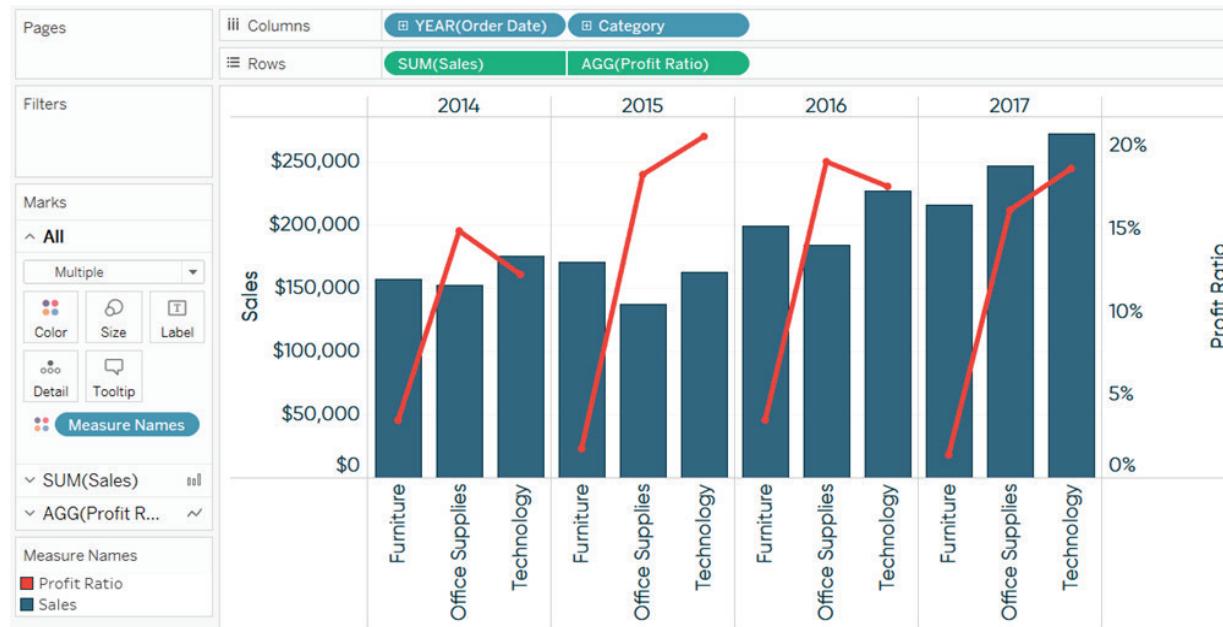
This isn't quite a dual-axis combination chart because the mark type for both charts is Bar, with the bars for profit ratio lying on top of the bars for sales. For a dual-axis combination chart, we need a combination of mark types. When you have more than one measure on the Rows Shelf and/or Columns Shelf, each measure gets its own Marks Shelf. This means you can edit the mark types for the measures independently of each other.



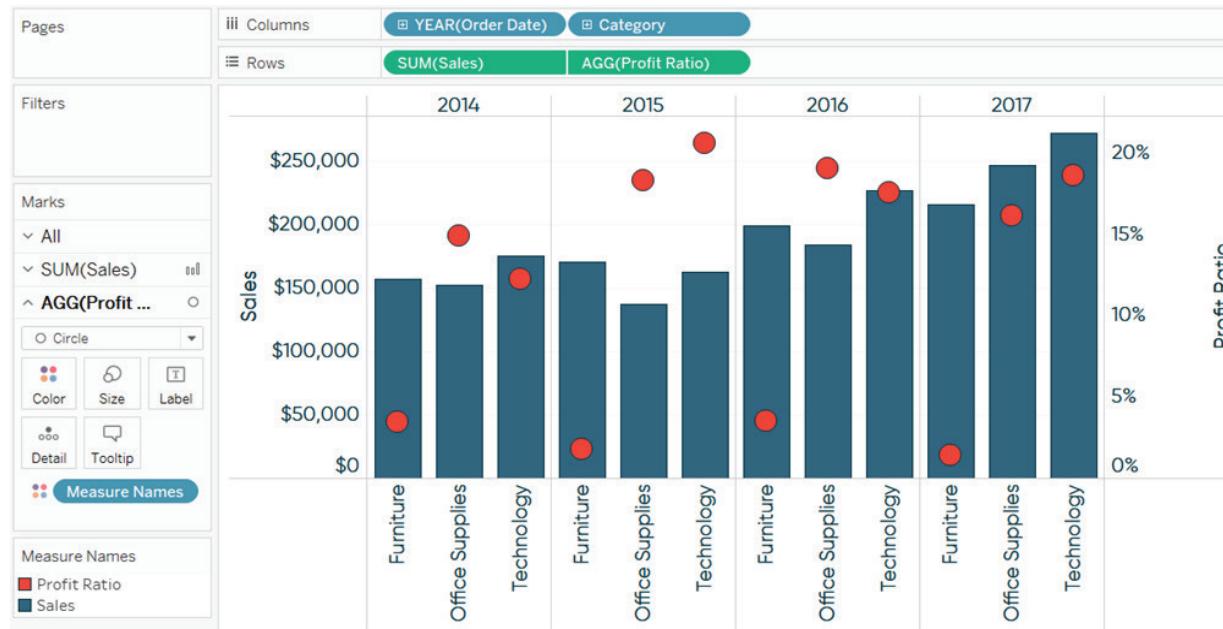
Here's how the final view looks after I change the mark type of the Profit Ratio measure from bar to line.



One word of caution when using the Line mark type. Lines should only be used when there is a relationship between the data points over time or if they're being connected by a continuous dimension / measure. Discrete fields are processed in their order on the Rows Shelf or Columns Shelf in Tableau. So for my example, sales and profit ratio are broken down by the Category dimension first, then the Year(Order Date) dimension second. That works, but what if we wanted to break the two measures down by Year(Order Date) first, then Category? Here's how it looks:



Note that we're now connecting dots with lines for data points that are not related by time. If you're ever in this situation, I suggest changing the mark type from Line to Circle to create a dot plot on the second axis.



## 2. Using a dual-axis combination chart to make your user part of the story.

For my second way to use a dual-axis combination chart, we will be recreating a portion of my Tableau dashboard, [How Does Your Household Income Rank?](#)

The visualization features a curve that shows how each household income ranks by percentile. This is a common “descriptive” view in that it describes high-level statistics. The real value comes by using the second axis to display where the person using the dashboard ranks on that curve. This makes the end user part of the story and is a much more engaging user experience.

To achieve this effect, I used a [parameter built in Tableau](#) that would allow the end user to choose any of the household income options between \$2,000 and \$450,000. After making the selection, in addition to the descriptive curve, the circle moves to the appropriate place based on the user’s selection. The label also updates with a caption telling the end user what percentile rank their household income is in.

**How does your household income rank?**  
Percentile rankings of US household incomes between \$2K and \$450K.

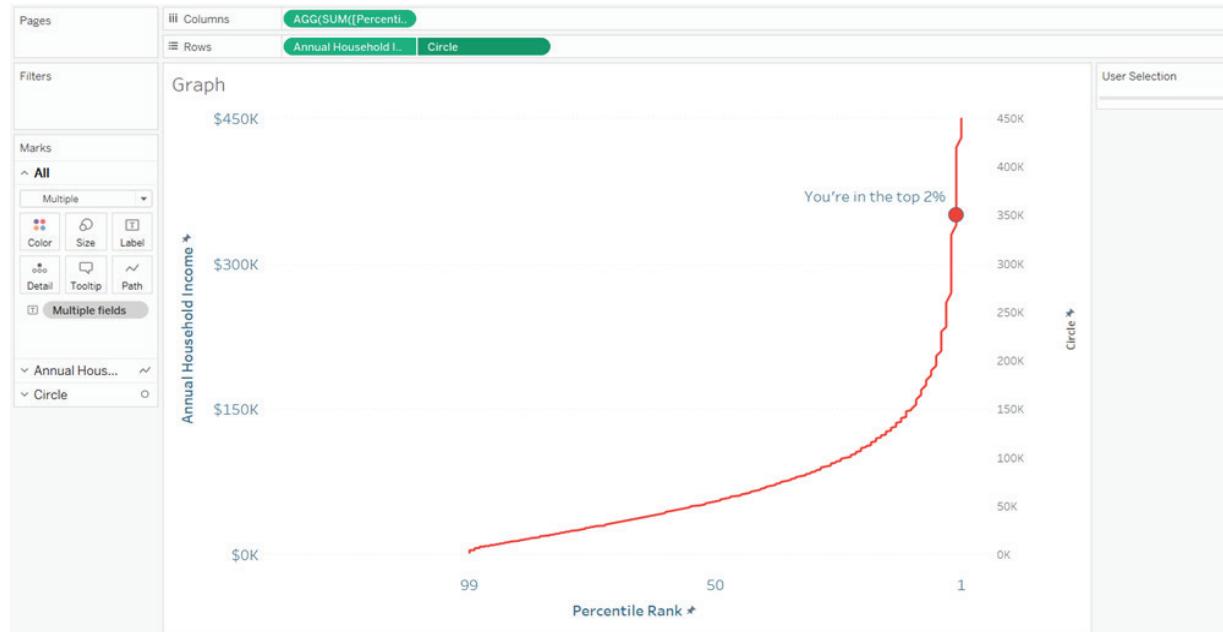


This is a remake of a CNN Money calculator. Inspiration and data sourced from money.cnn.com. It was created by Ryan Sleeper of ryansleeper.com.

This is a dual-axis combination chart. The curve for the left axis uses a mark type of Line, and the circle is a second measure that only displays a circle for the end user's selection on the right axis. The trick to getting just one circle to show up is a simple formula that computes whether the user's parameter selection matches the household income value on the Y-axis. Here's what it looks like:



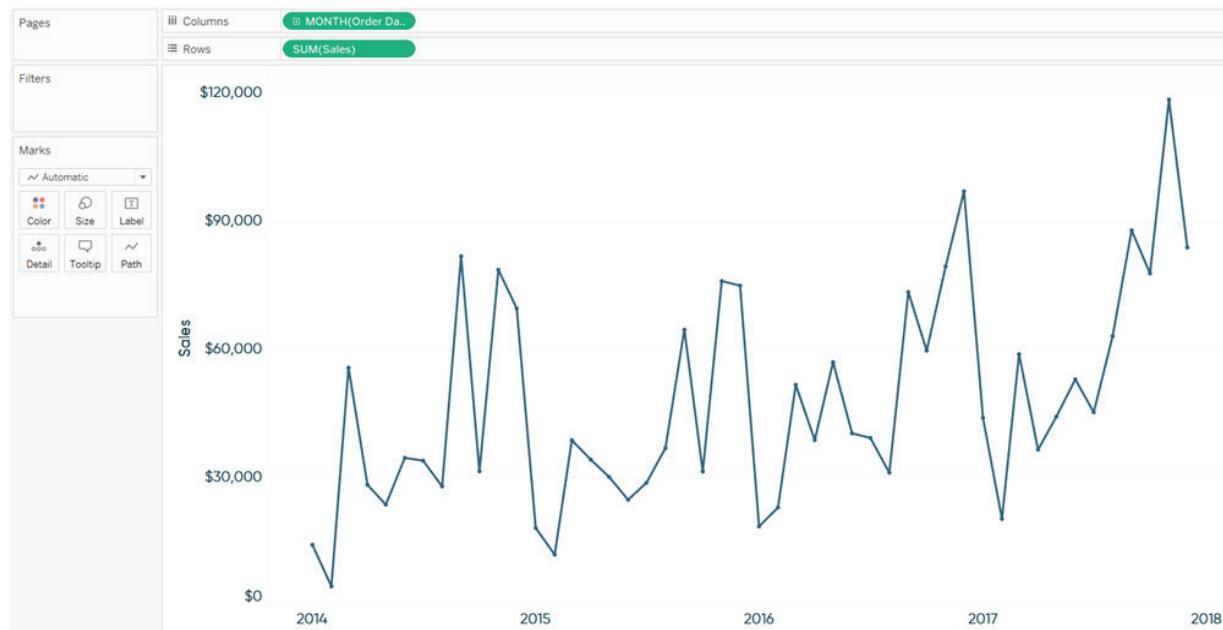
Once I had this calculation, I built the curve on the left axis and put the measure for the circle on the right axis.



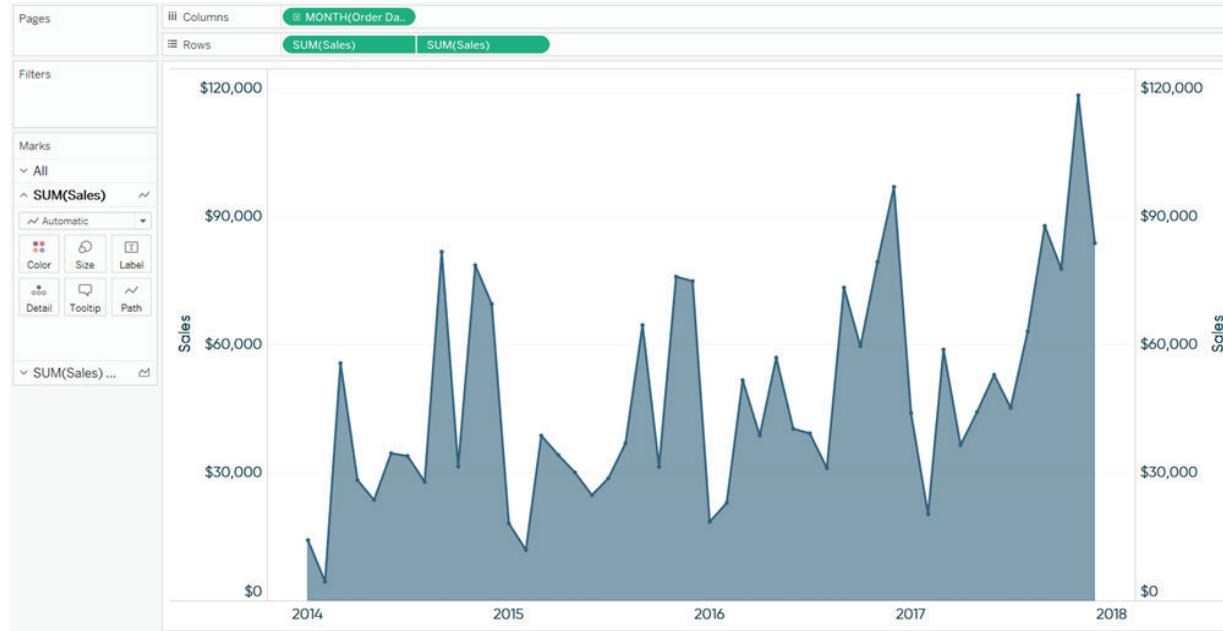
To finalize the view, I synchronized the axes by right-clicking the axis for "Circle" and chose "Synchronize Axis". This ensures the circle will always line up perfectly with the line on the opposite axis. Lastly, I hid the axis on the right by right-clicking on it and unchecking "Show Header".

## 3. Using a dual-axis to improve the design of a line graph

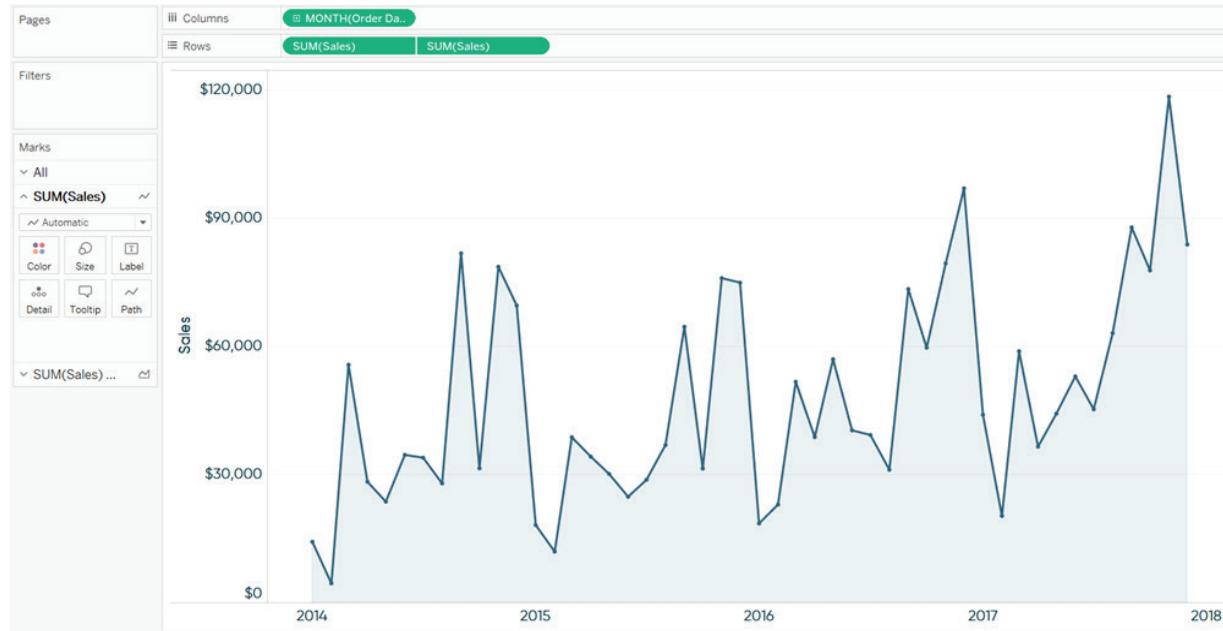
In the previous chapter, 3 Ways to Make Lovely Line Graphs, I provide a few recommendations for making traditional line graphs more engaging in Tableau. One of those approaches involves a dual-axis combination chart; here's how it's done. Say you have a line graph showing a sales by month trend:



Now I'll put the Sales measure on the right axis, synchronize the axes, and change the mark type for the second axis to Area.

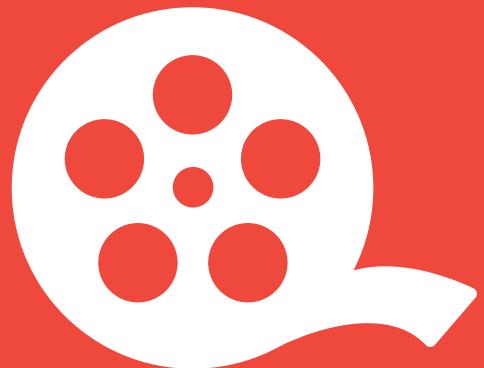


At this point we have a dual-axis combination chart with sales by month as a line graph and sales by month as an area chart. To finalize the view, I will hide the right axis and reduce the opacity of the area to 10%.



You can only use the second axis for one purpose at a time, but if you are not using it for one of the first two reasons above, this third application can be an easy way to enhance your traditional line graphs.

# Chapter 4



## 3 Ways to Use Tableau Dashboard Actions

Dashboard actions, filters, and parameters are three of the best tactics to use in Tableau because they provide a way to transfer control of the analysis from you to your end users. This is critical in data visualization because it helps you allow discovery for your users and makes the insights they find easier to retain. In this chapter, I provide an introduction to dashboard actions and we will walk through three different ways to leverage dashboard actions to improve your user experience.

# An Introduction to Tableau Dashboard Actions

Before I share three ideas on how to use dashboard actions, here's a quick primer in case you are not familiar with this functionality. If you're comfortable with the concept of dashboard actions, feel free to jump down to the first tip, Tableau Dashboard Action 1: Use Every Sheet as a Filter.

Dashboard actions in Tableau allow you to add logic to dashboard components that create actions somewhere else. For example, you can add logic that says, "If a user clicks on Dashboard Sheet 1, I want something to happen on Dashboard Sheet 2." If you think about dashboard actions this way, their setup is very intuitive in Tableau. To set up a dashboard action, navigate to Dashboard > Actions in the top navigation from any dashboard view. A dialog box will appear, and when you click the button "Add Action >" in the bottom left corner, you will be presented with three options for the type of dashboard action that you want to add:

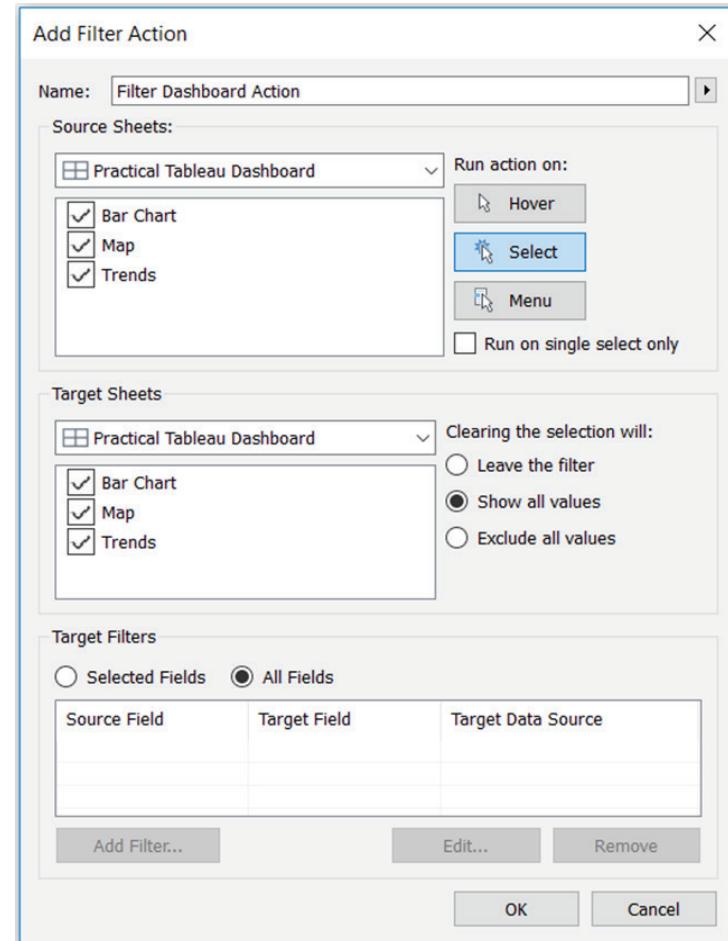
1. Filter – Example: If you click on sheet one, sheet two will be filtered to whatever you clicked on sheet one.
2. Highlight – Example: If you click on sheet one, sheet two will be highlighted by whatever you clicked on sheet one.
3. URL – Example: If you click on sheet one, open a URL (this can either be opened on a webpage dashboard component or in a new browser window).

Once you choose which type of dashboard action to add, you are taken to a new screen where the logic for your dashboard action is coded:

The “Source Sheets” list shows you all of the sheets you have on the dashboard where you are adding the action. Any sheets selected in this list will cause the dashboard action to execute.

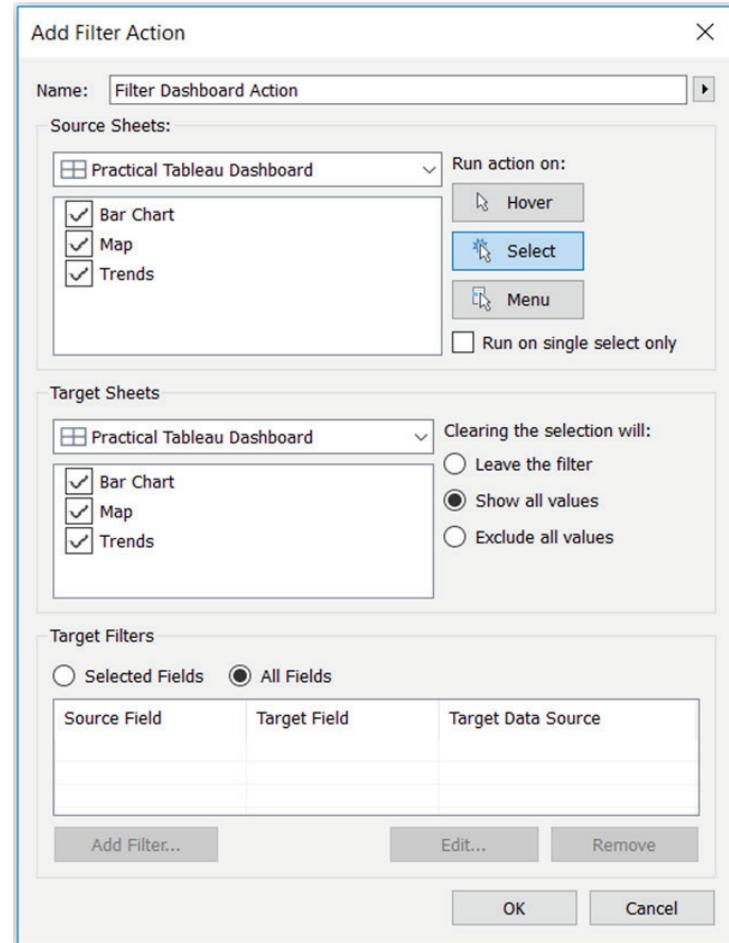
The “Target Sheets” list also shows all of the sheets in the dashboard, but these are the sheets where you want the action to take place.

Pretty intuitive so far, right?



From here, there are a few options for your dashboard actions. First, you can have the action execute on three different interactions (pictured in the top right corner of the dialog box):

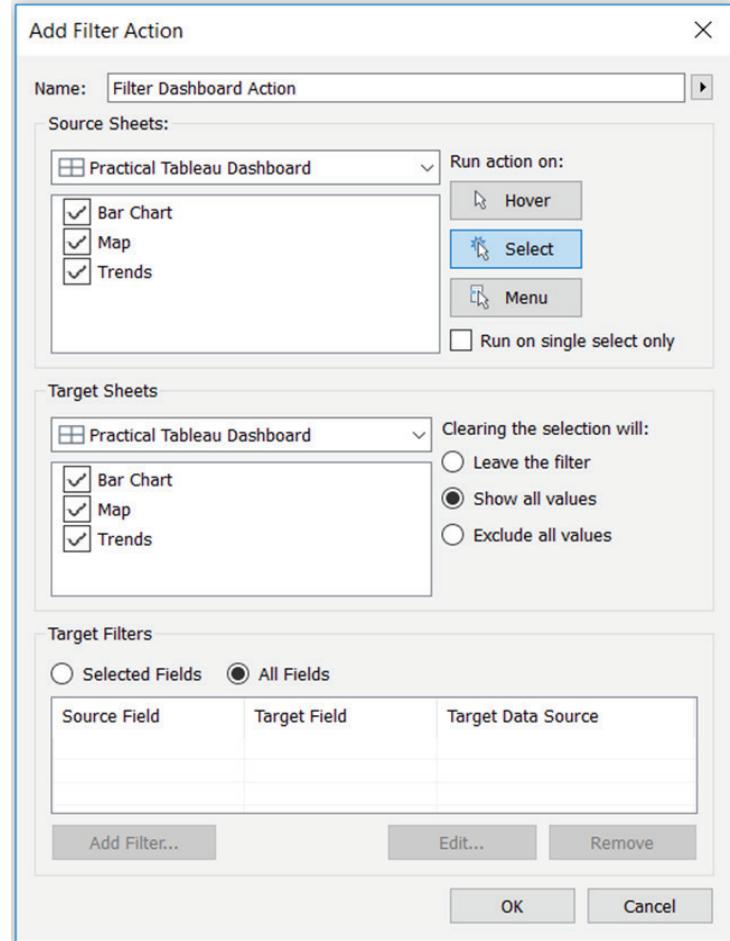
1. Hover – Example: If you hover over the source sheet, the action will take place on the target sheet.
2. Select – Example: If you click on the source sheet, the action will take place on the target sheet.
3. Menu – Example: If you hover over the source sheet, a menu of dashboard actions will appear in the tooltip. Clicking on one of the menu items will execute the action on the target sheet.



You can also tell Tableau what you want to happen if the dashboard action is cleared (which can be done by clicking escape):

1. Leave the filter – leaves the last dashboard action that happened in place. If you filtered sheet two by something clicked on sheet one, the filter on sheet two will stick.
2. Show all values – reverts back to the original view as if no dashboard action took place.
3. Exclude all values – clears everything off of the target sheet, meaning the target sheet will not show unless a dashboard action is executed.

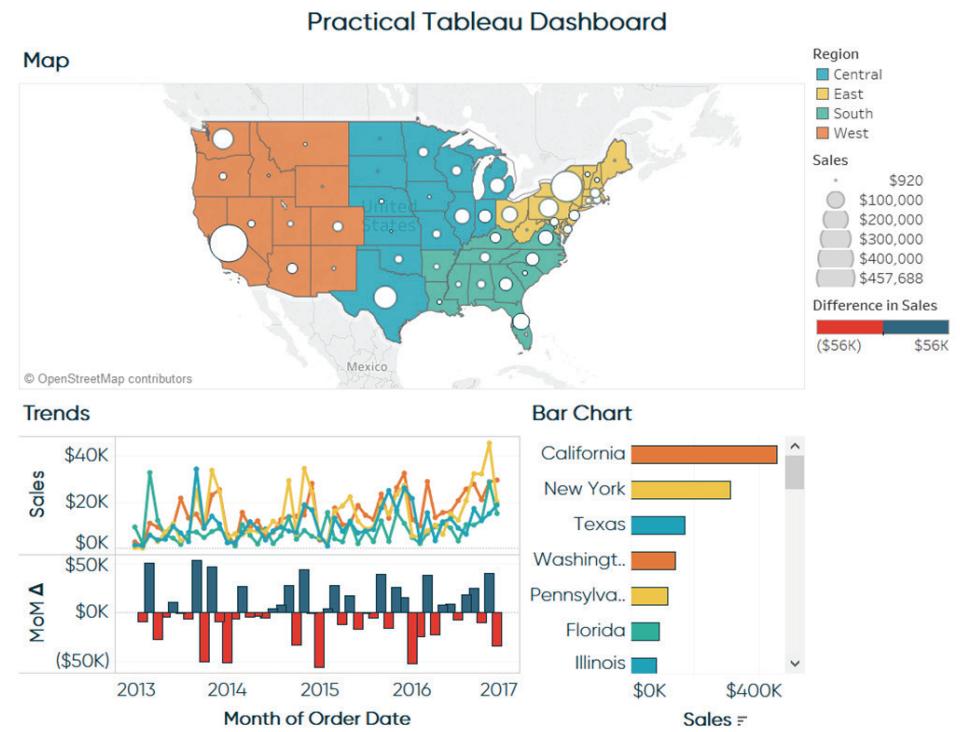
Lastly, you can refine the dashboard action to take place against certain fields using the “Target Filters” options at the bottom of the dashboard actions dialog box. By default, the dashboard action will run on every shared field between the source sheets and the target sheets. If you want to change the level of detail that a dashboard action runs on a target sheet, you would need to add a target filter to specify the fields – essentially telling Tableau how granular you want to be.



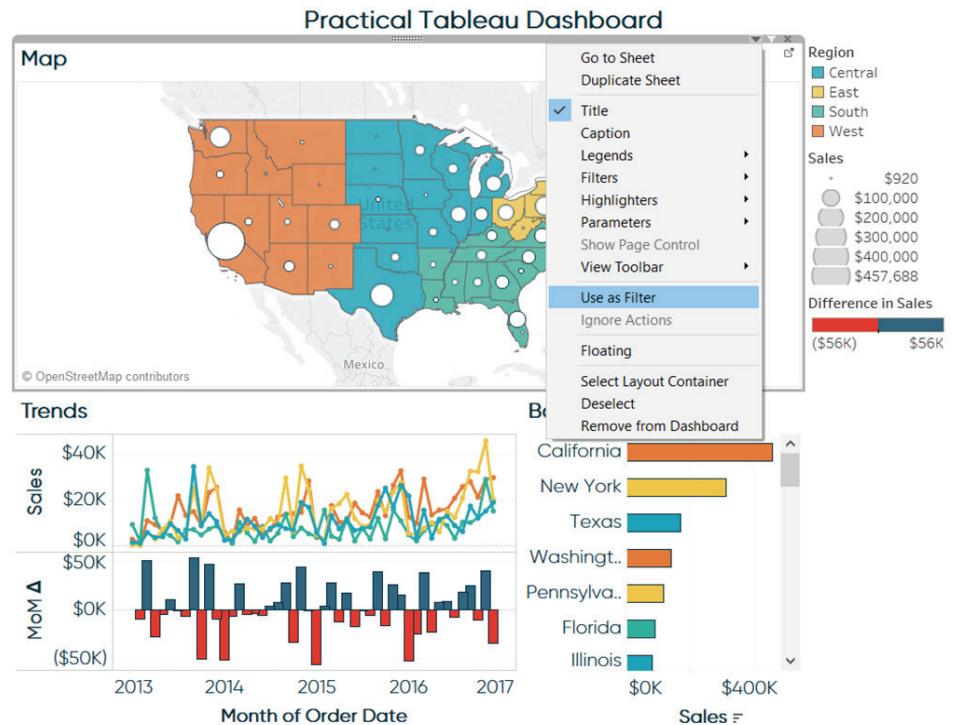
# Tableau Dashboard Action 1: Use Every Sheet as a Filter

Any sheet in a dashboard can be used as a filter by simply hovering over the sheet, clicking the down arrow that appears in the top right corner, and choosing “Use as Filter”. This is basic Tableau functionality, but it wasn’t until I saw a presentation from Kevin Krizek of Tableau, that I realized it would be good practice to allow users to use every sheet in a dashboard as a filter. Adding this functionality to your own dashboards gives your users flexibility in choosing how they want to look at the data.

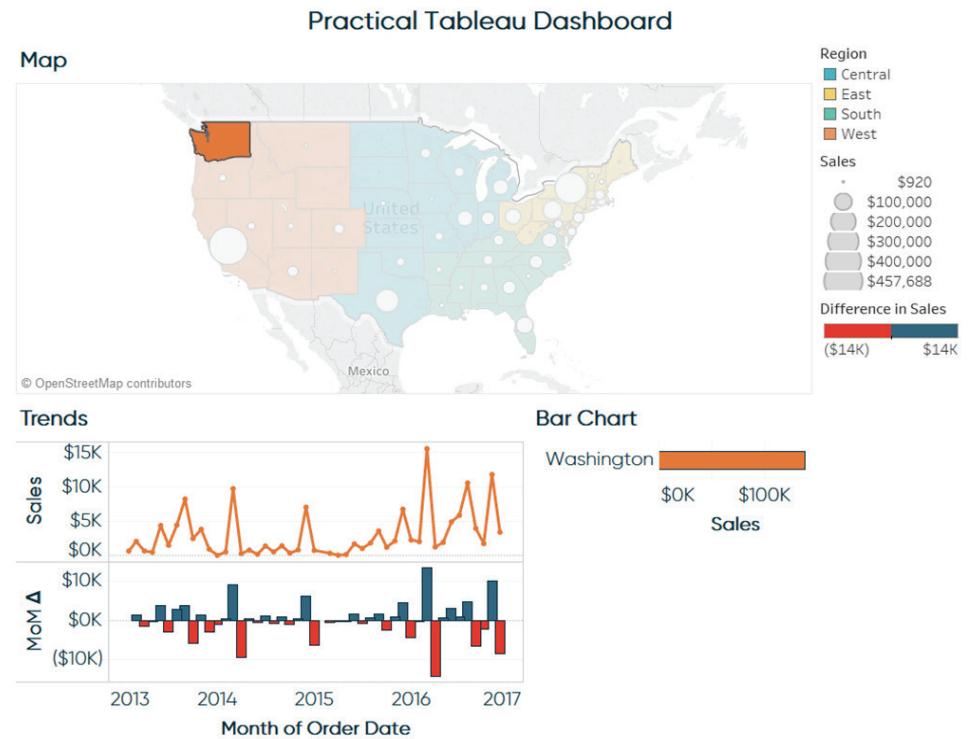
Let’s take a look at how this would work with a sample dashboard that I often create while [speaking about Tableau](#). Here is the original view:



I will now choose to use every individual dashboard sheet as a filter for the entire dashboard by hovering over the sheet, clicking the down arrow that appears in the upper right, and selecting “Use as Filter” (on all three sheets):



Now when I click on any sheet, the other sheets are filtered to whatever I clicked on. For example, if I click on Washington in my map view, the trend line and bar chart sheets will be filtered to just that state:



What is actually happening behind the scenes when I choose to use a dashboard sheet as a filter is that Tableau is automatically creating a dashboard action like shown in the dashboard actions introduction above. You can confirm Tableau created dashboard actions by navigating to Dashboard > Actions.

The good thing about setting up each dashboard as a filter instead of sharing one dashboard action between all of the source sheets is that you can refine the actions independently of each other. For example, the bar chart doesn't add much value when it is filtered to just Washington as pictured above. Maybe it would be better if I added a target filter that filtered the bar chart to Region instead of State when I click on the map. On the other hand, I would like the map to show just one state on the map if I click an individual state on the bar chart. In order to have these two different levels of filter granularity, I would need two separate dashboard actions.

**Actions**

Connect sheets to external web resources using URL actions, or to other sheets in the same workbook using Filter actions and Highlight actions.

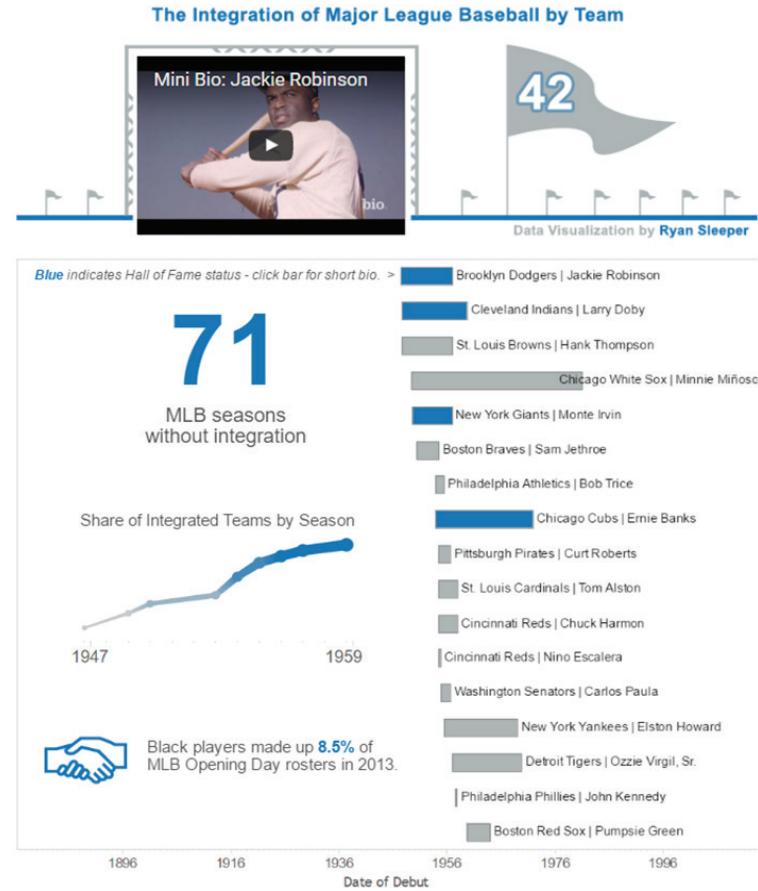
Name	Run On	Source	Fields
Filter 1 (generated)	Select	Practical Tableau Da...	All
Filter 2 (generated)	Select	Practical Tableau Da...	All
Filter 3 (generated)	Select	Practical Tableau Da...	All

**Add Action >** **Edit...** **Remove**

Show actions for all sheets in this workbook OK Cancel

## Tableau Dashboard Action 2: Embed YouTube Videos in a Dashboard.

This is a tip I first picked up from Ben Jones at his excellent blog, [DataRemixed.com](http://DataRemixed.com). The first time I embedded a video in a Tableau dashboard was my 2013 viz, [MLB Integration by Team](#).

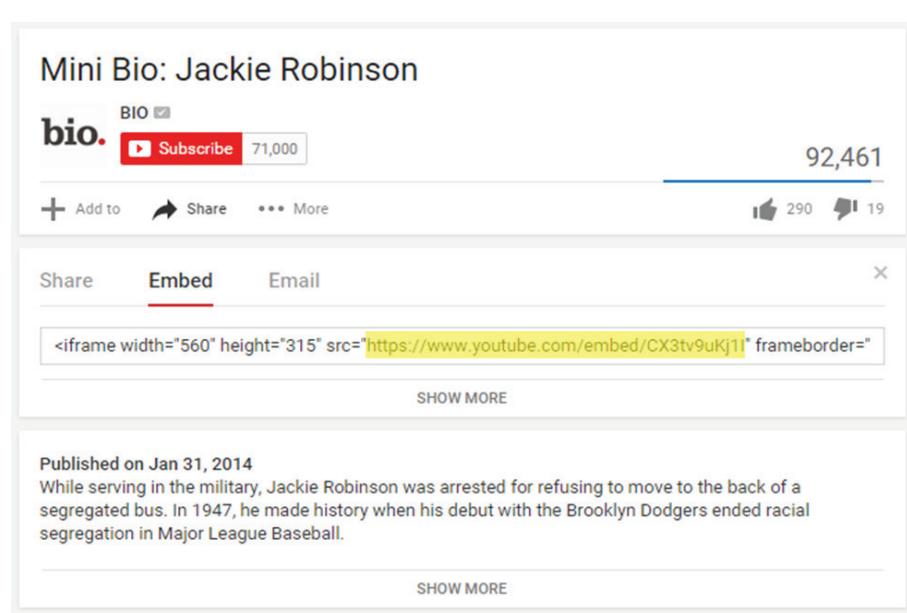


This novel effect was actually quite easy to create in Tableau. The only tricky part is finding the correct YouTube links and having the YouTube links as a field in your underlying data source. Here's a look at the underlying data for this viz:

	A	B	C	D	E	F	G	H	I	J
1	Player	Team	League	Date of First Game	Date of Last Game	Span of Career	Seasons	Debut	Integration	YouTube
2	Jackie Robinson	Brooklyn Dodgers	National	4/15/1947	9/30/1956	3456 ten	1	6.00%	<a href="https://www.youtube.com/embed/CX3tv9uKj1I">https://www.youtube.com/embed/CX3tv9uKj1I</a>	
3	Larry Doby	Cleveland Indians	American	7/5/1947	7/26/1959	4404 thirteen	2	13.00%	<a href="https://www.youtube.com/embed/m6yq0BR8o0Y">https://www.youtube.com/embed/m6yq0BR8o0Y</a>	
4	Hank Thompson	St. Louis Browns	American	7/17/1947	9/30/1956	3363 ten	3	19.00%		
5	Minnie Miñoso	Chicago White Sox	American	4/19/1949	10/5/1980	11492 thirty-two	4	25.00%		
6	Monte Irvin	New York Giants	National	7/8/1949	9/30/1956	2641 eight	5	31.00%	<a href="https://www.youtube.com/embed/K6ZHOKRRi84">https://www.youtube.com/embed/K6ZHOKRRi84</a>	
7	Sam Jethroe	Boston Braves	National	4/18/1950	4/15/1954	1458 five	6	38.00%		
8	Bob Trice	Philadelphia Athletics	American	9/13/1953	5/2/1955	596 three	7	44.00%		
9	Ernie Banks	Chicago Cubs	National	9/17/1953	9/26/1971	6583 nineteen	8	50.00%	<a href="https://www.youtube.com/embed/HPL-Z6cst1Q">https://www.youtube.com/embed/HPL-Z6cst1Q</a>	
10	Curt Roberts	Pittsburgh Pirates	National	4/13/1954	6/8/1956	787 three	9	56.00%		
11	Tom Alston	St. Louis Cardinals	National	4/13/1954	9/29/1957	1265 four	10	63.00%		
12	Nino Escalera	Cincinnati Reds	National	4/17/1954	9/25/1954	161 one	11	69.00%		
13	Chuck Harmon	Cincinnati Reds	National	4/17/1954	9/15/1957	1247 four	12	69.00%		
14	Carlos Paula	Washington Senators	American	9/6/1954	6/23/1956	656 three	13	75.00%		
15	Elston Howard	New York Yankees	American	4/14/1955	9/29/1968	4917 fourteen	14	81.00%		
16	Ozzie Virgil, Sr.	Detroit Tigers	American	9/23/1956	6/27/1969	4660 fourteen	15	88.00%		
17	John Kennedy	Philadelphia Phillies	National	4/22/1957	5/3/1957	11 one	16	94.00%		
18	Pumpsie Green	Boston Red Sox	American	7/21/1959	9/26/1963	1528 five	17	100.00%		

To find the correct link to use in your data source, follow these steps:

1. Find the video you want to embed on YouTube.
2. Click the Share icon below the video.
3. Click the Embed tab.
4. Copy the URL that appears between the quotation marks.



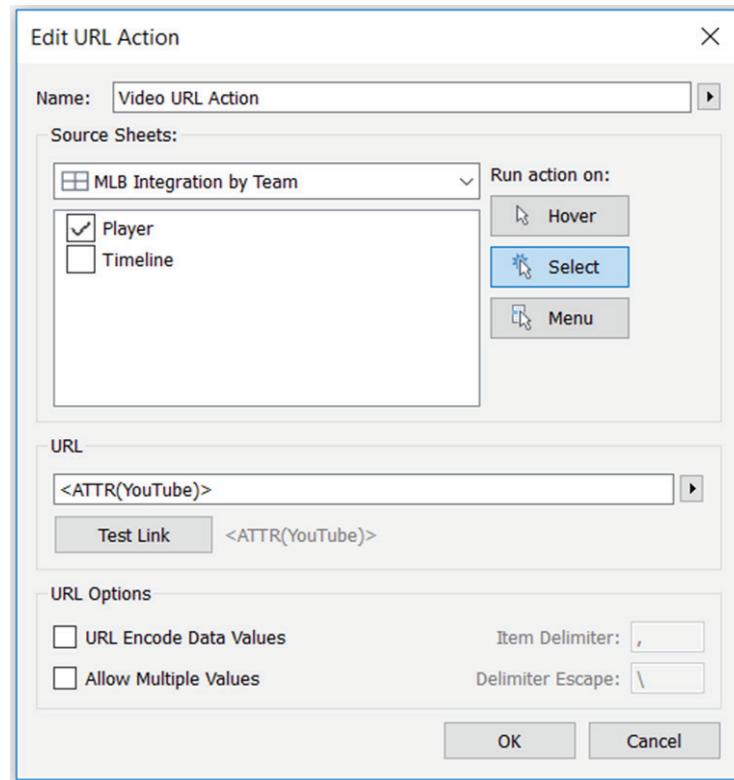
The screenshot shows a YouTube video page for 'Mini Bio: Jackie Robinson'. The video has 92,461 views. Below the video player, there are sharing options: 'Add to', 'Share' (which is highlighted), and 'More'. The 'Embed' tab is selected, showing the embed code: <iframe width="560" height="315" src="https://www.youtube.com/embed/CX3tv9uKj1" frameborder="". Below the embed code, there are 'SHOW MORE' buttons. At the bottom, it says 'Published on Jan 31, 2014' and provides a brief description of Jackie Robinson's historical significance.

Now that you have the data, follow these steps to use dashboard actions to embed a YouTube video in your Tableau dashboard:

1. Add a “Web Page” dashboard object by dragging and dropping it on your view from the left navigation of a dashboard view. In my case, I used a floating Web Page object so that I could make it the exact same dimensions as the scoreboard; providing the illusion that the video is being displayed on the jumbotron. When you add the Web Page object, Tableau will ask you to enter the URL for the page you want to display; leave this blank for now because we will be using a dashboard action to provide these instructions.
2. Add a dashboard action by navigating to Dashboard > Actions > Add Action.
3. The dashboard action should be a URL action. You can refine the settings to have the action run on Hover, Select, or Menu; I have chosen for the action to run on Select (which is the same as click).
4. Lastly, for the URL, click the arrow that appears next to the empty URL box. You should be shown a list of options including the URL field in your underlying data. Click the URL field so that the video associated with a particular record will start when the action is run.

You can see on the right how my final dashboard action looks.

Now if I click a name on the sheet called “Player” that has a corresponding video link, the YouTube video will load in the empty Web Page object that I added. The videos can be consumed right there in line with the rest of the Tableau viz!



## Tableau Dashboard Action 3: Do a Google Search or Google Image Search from a Dashboard.

One of the most effective implementations of my third tip was in the [winning entry of 2015 Iron Viz Championship by Shine Pulikathara](#). In the viz, users can explore news stories or related images by following links provided within the Tableau dashboard for thousands of different data points. The first time I integrated functionality that would do a Google Image Search from a Tableau dashboard was in a viz I created for my wife to document our travels, [\*We're Not in Kansas Anymore\*](#). Clicking on any location will open a new browser with a Google Image search for that location.



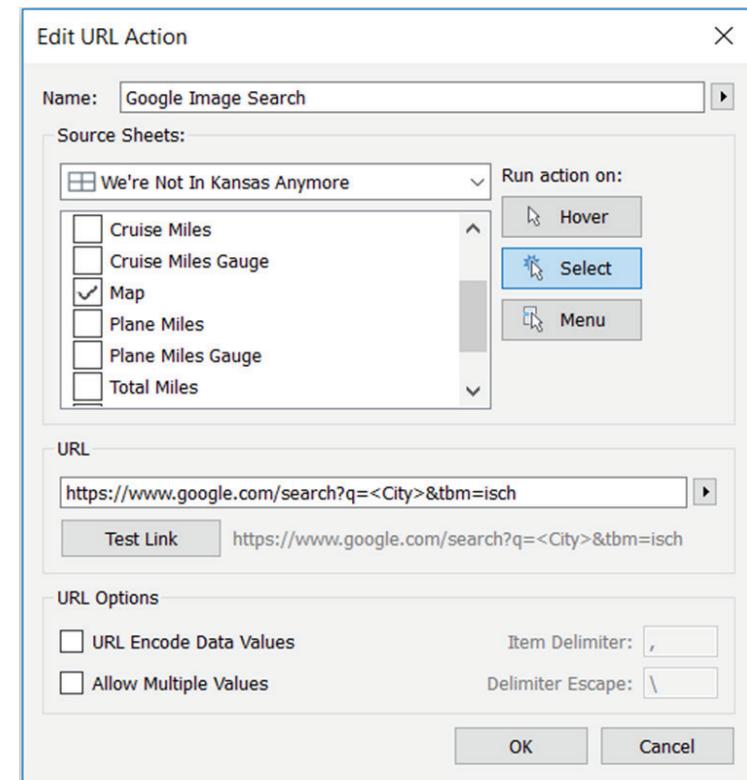
## Here are the steps required to create this user experience:

1. Go to Google Images and search for one of the locations.
2. Record the URL that appears in the browser. A search for Kansas City resulted in this URL: [https://www.google.com/search?q=kansas+city&source=lnms&tbo=isch&sa=X&ved=0ahUKEwiH6eqWhoDKAh-VWz2MKHctYBJsQ\\_AUICSGD&biw=1366&bih=643](https://www.google.com/search?q=kansas+city&source=lnms&tbo=isch&sa=X&ved=0ahUKEwiH6eqWhoDKAh-VWz2MKHctYBJsQ_AUICSGD&biw=1366&bih=643)
3. You can keep the entire link, but all that you actually need is: <https://www.google.com/search?q=kansas+city&tbo=isch> – this URL includes the query (Kansas City) and tells Google to show results on the image tab (isch = image search).
4. Add a dashboard action by navigating to Dashboard > Actions > Add Action
5. The dashboard action should be a URL action. You can refine the settings to have the action run on Hover, Select, or Menu; I have chosen for the action to run on Select (which is the same as click).
6. Lastly, for the URL, paste the URL that you recorded from the search above. Replace the portion of the URL for the query, which is the text immediately following the "?q=", with a field from your data. You can add a field from your data source by clicking the arrow that appears next to the empty URL box. I wanted to run a search for each city, so I used the field <City> from my data source.

Here's how my final dashboard action looks:

Now if I click a place on my map, a Google Image search is executed with the name of that city (from my underlying data) as the search query!

This chapter has provided an introduction to dashboard actions in Tableau, showed you how to use every sheet as a filter, how to embed a video in a Tableau dashboard, and how to do a Google search from a Tableau dashboard. All three of these dashboard action examples share a common thread of allowing discovery, one tactic for improving the retention of insights for you and your audience.



# Chapter 5



## 3 Ways to Use Google with Tableau Dashboard Actions

Google has a number of powerful tools that can help with your business analyses. In addition to the native Google Analytics Tableau connection which allows you to create customized Tableau dashboards from your web analytics data, other Google services can be integrated with Tableau through the use of dashboard actions.

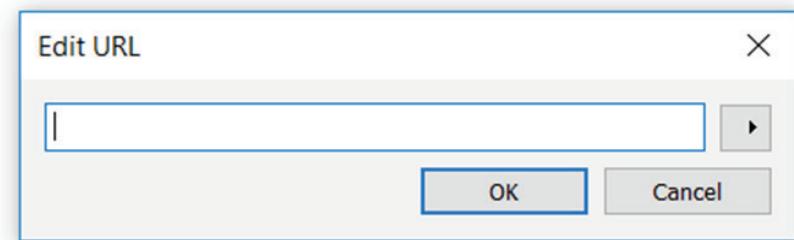
This chapter will show you how to (1) do an image search, (2) view your paid search ads, and (3) lookup a Google Trend, all from within a Tableau dashboard.

The following tips have several benefits including: efficiently adding a visual impact to your dashboards, providing context to you and your end users, and improving your analyses with very little manual effort.

If you need a crash course on dashboard actions, check out the previous chapter, 3 Creative Ways to Use Tableau Dashboard Actions. Then come back and learn the new ways to improve your Tableau dashboards by integrating Google!

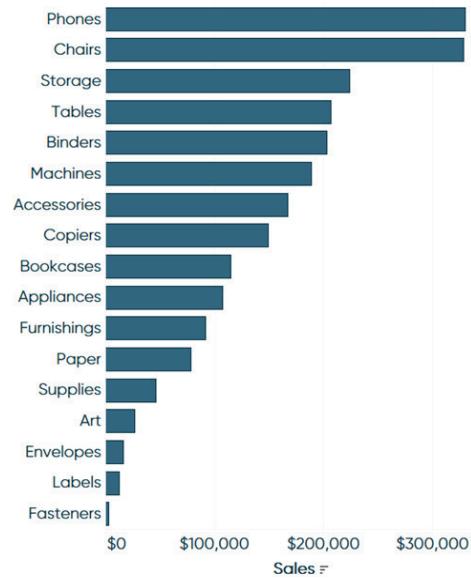
# How to add a URL dashboard action to a Tableau dashboard.

To do a Google image search, view your Google paid search ads, or lookup a Google Trend, you will need to add a URL action to your Tableau dashboard. To lay the foundation for this tutorial, I am going to make a simple dashboard. First, I'll place a horizontal layout container on a new dashboard. Within that layout container, I'll place a bar chart showing Sales by Sub-Category and a Web Page object. Note that when you add the Web Page object, Tableau will ask you to enter a URL, but you need to leave this blank for now.



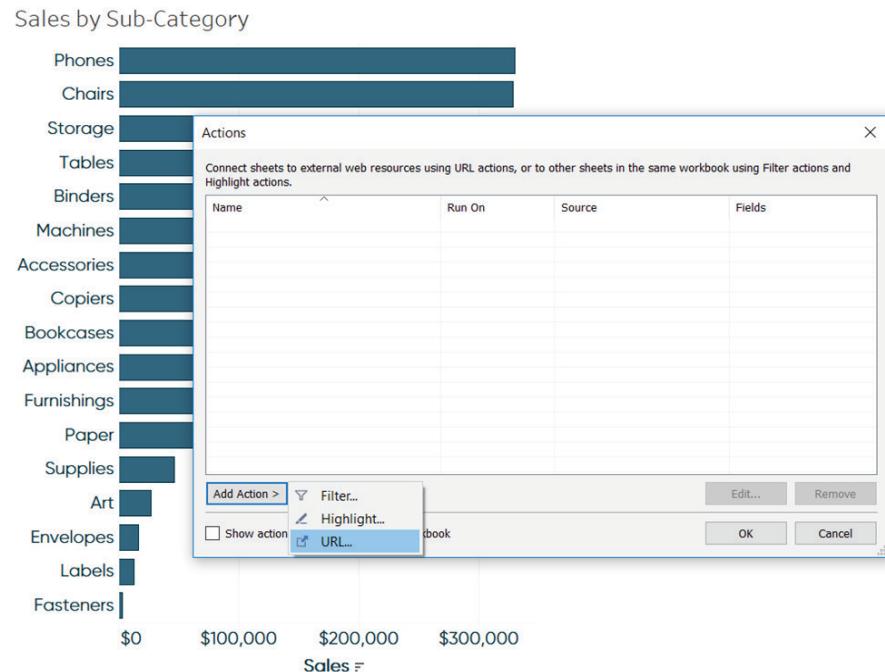
At this point, my dashboard looks like this. Note that the blank space on the right side of the horizontal layout container is the Web Page object. This will soon be filled with a Google image search, Google paid search ads, or Google Trends.

Sales by Sub-Category





To add a URL dashboard action, navigate to “Dashboard > Actions…”, click “Add Action”, and choose “URL…”.



## 3 ways to integrate Google with Tableau URL actions

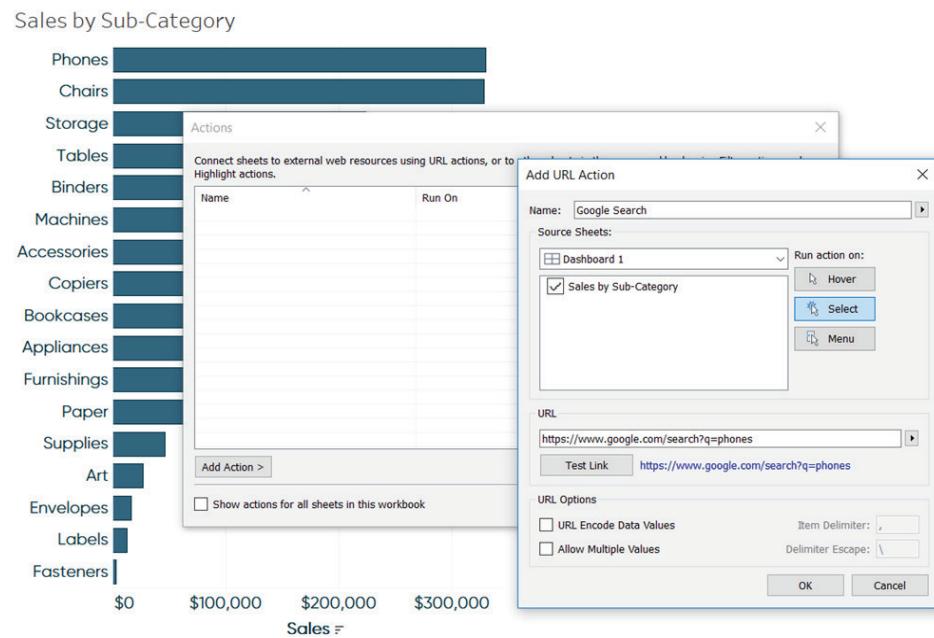
When doing a Google image search, a Google search (which contain paid search ads), or a search for a Google Trend, a URL is generated with a query string. Here is how each respective URL looks when searching for the top Sub-Category (Phones) from our dashboard:

Google Image Search: <https://www.google.com/search?q=phones&tbo=isch>

Google Search: <https://www.google.com/search?q=phones>

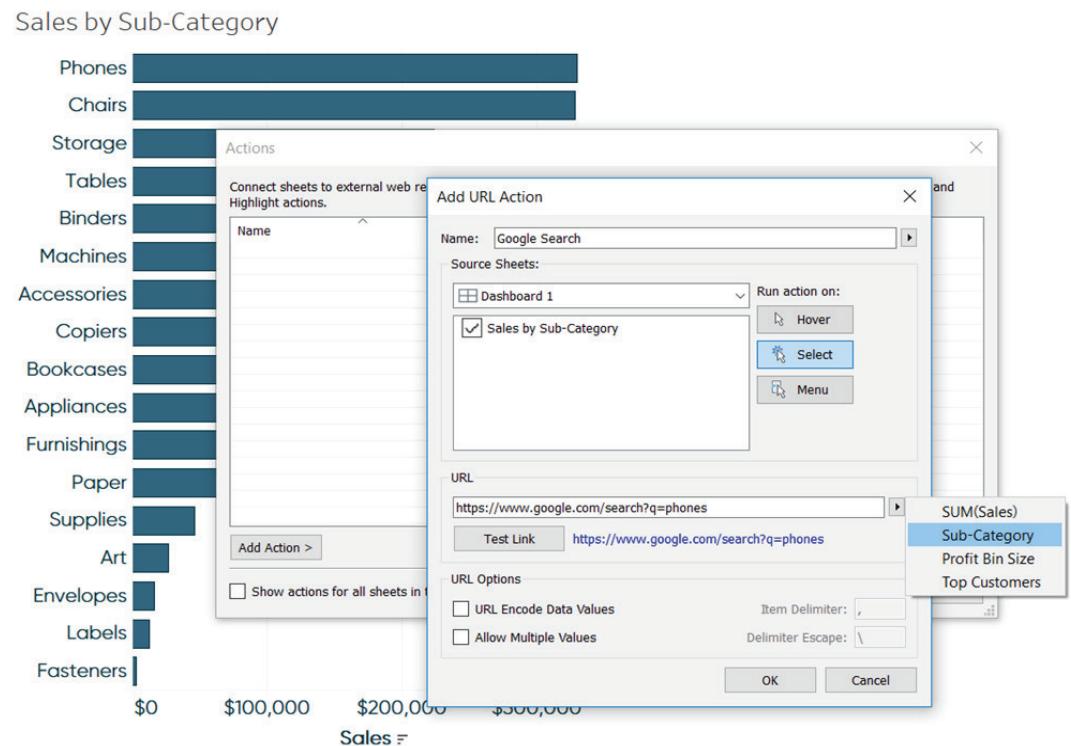
Google Trends: <https://trends.google.com/trends/explore?q=phones>

I'm consolidating all three examples because they all work the same way. Depending on which of the three tactics you want to execute, you will use the respective URL from above. Here's how my URL dashboard action looks if I wanted to set up a traditional Google search to see if my paid search ads are showing up with my competitors'.

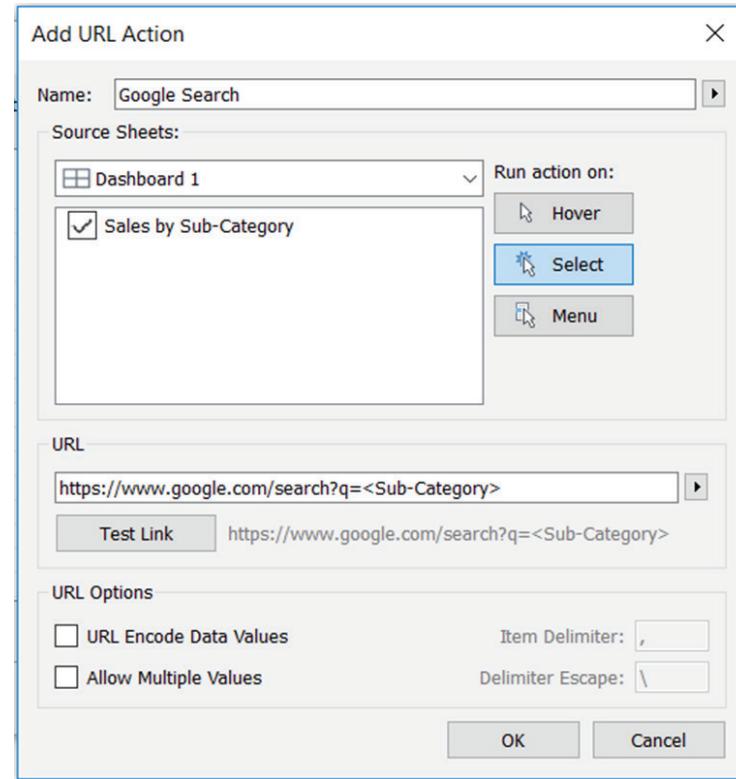


## Here's where the magic happens.

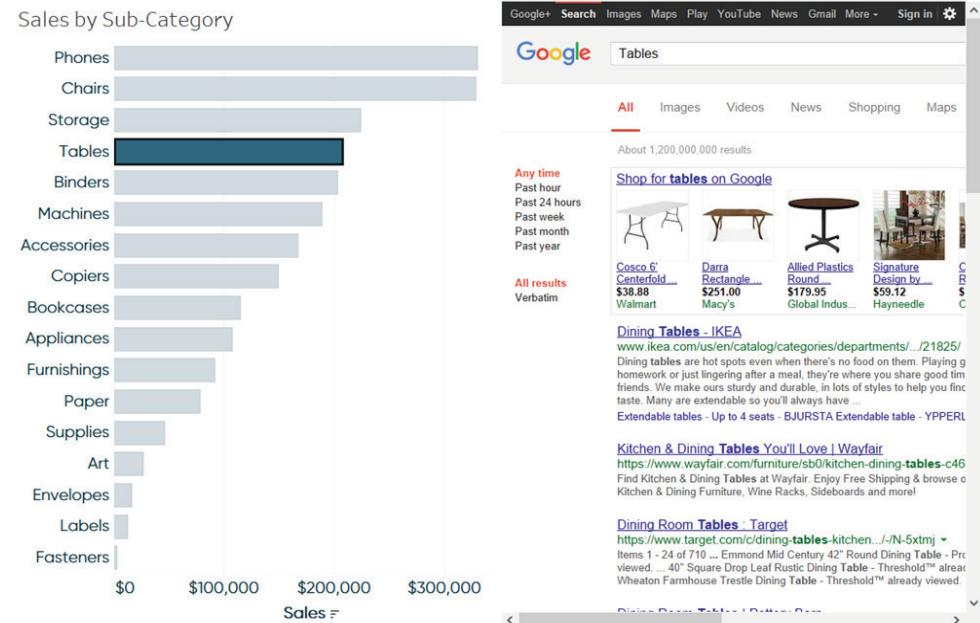
Instead of keeping phones as a static search, we can replace the text after "?q=" with a field from our dashboard using the right-arrow seen next to the URL.



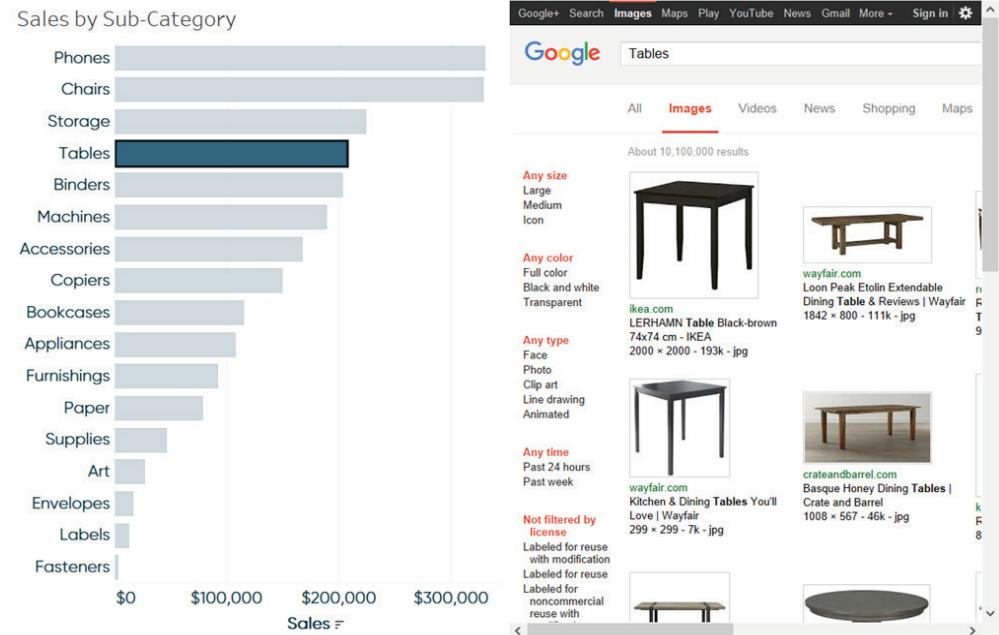
Here's how my URL dashboard action looks after replacing "phones" from my query with the Sub-Category dimension from my dashboard.



The way this is set up, now when I click on any of the bars on my dashboard, the Web Page object on the right will be filled with a Google search! Here's how the dashboard looks after clicking on the Tables Sub-Category.

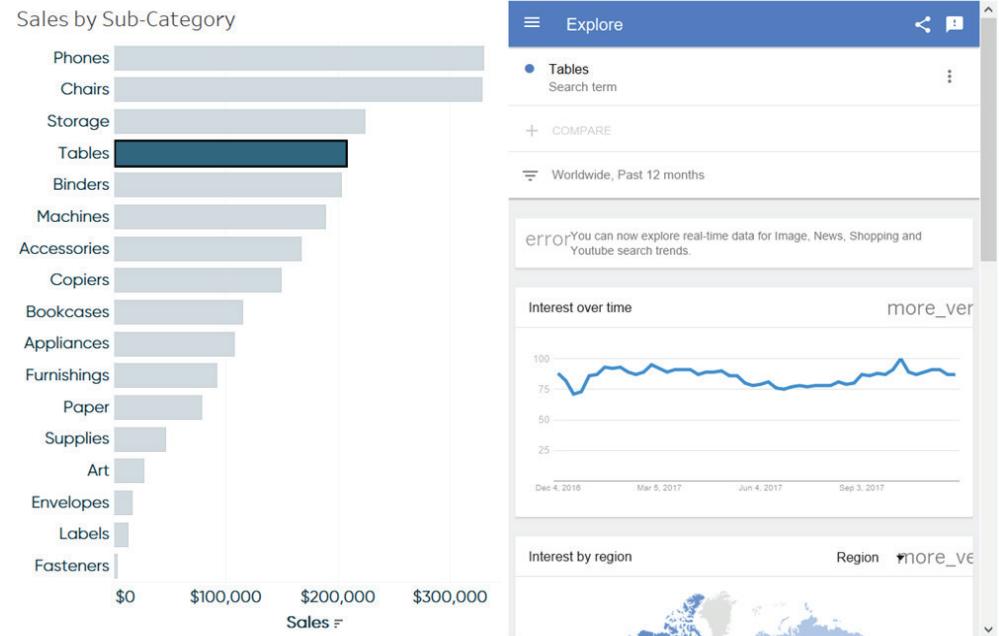


Here's how the same view would look if I used the Google Image Search URL instead.

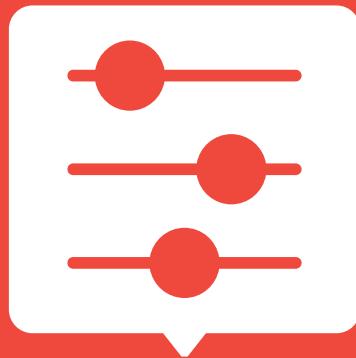


Lastly, here's how the same view would look if I used the Google Trends URL.

If you don't like the way the web page looks within the Tableau dashboard, follow the same steps from this tutorial but do not add a Web Page object. If there is no Web Page object to fill with the URL from the URL dashboard action, a new browser window will open outside of Tableau instead.



# Chapter 6



## 3 Ways the Describe Feature Can Help You Use Tableau

The “describe” feature in Tableau is one that somehow took me years to stumble upon, but is now one of my favorites. If you’ve heard of this feature, you’ve most likely used it to describe an individual dimension, measure, set, or parameter. Right-clicking on a field and choosing “Describe...” gives you information such as the data type and which table it comes from, but there are several other ways to use the describe feature in Tableau to help you speed up your learning and analyses.

This chapter will show you (1) how to get the most out of the aforementioned usage, (2) a method for reverse engineering somebody else’s (or your own) Tableau workbooks, and (3) a way to view underlying formulas and parameter allowable values.

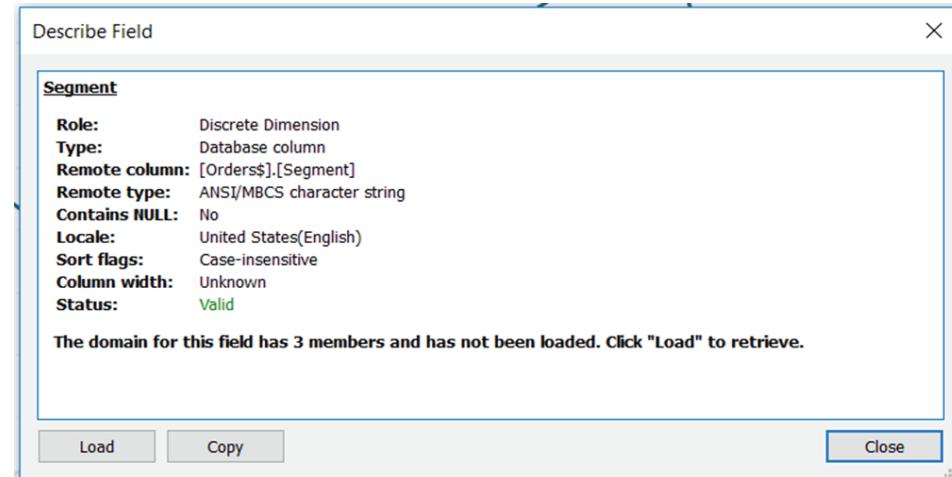
These tips have several benefits including saving processing time of your VizQL queries, speeding up your learning or reminding you of your previous approaches, and making your calculated field writing more efficient.

This chapter will use the [Super Sample Superstore](#) dashboard to show you three different applications of the describe feature in Tableau.

## Using “Describe...” to save processing time

As mentioned in the introduction, if you’re familiar with this feature, you’ve likely right-clicked on an individual field and chose “Describe...” to learn more about the field’s properties.

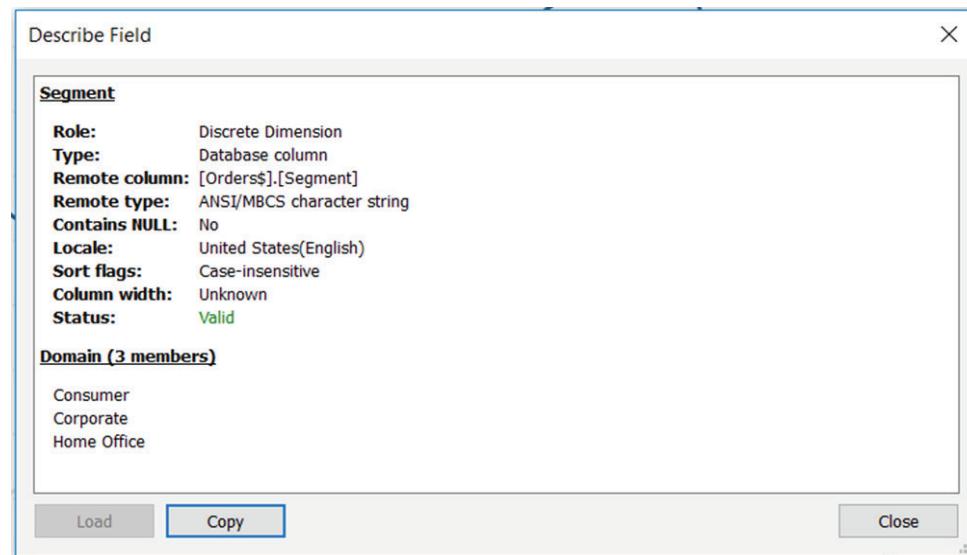
Here’s what is shown when doing this for the Segment dimension in the Sample – Superstore dataset.



My favorite part of the description when used with a dimension is that if you click the “Load” button pictured in the bottom left corner of the window, Tableau will provide a preview of the first 20 dimension members. Before I knew about this feature, I would drag and drop the field onto the view to get an idea of what dimension members were in the field.

Unfortunately, this meant that Tableau had to query the data source to visualize the answer, which could take quite some time depending on the size of my dataset. The describe feature is much more efficient and gives me an idea of what dimension members are in the field without requiring me to create a new view/wait for processing.

Here's how the Describe Field information looks when loading the dimension members for the Segment dimension.



# A method of reverse-engineering

Sometimes I feel like the younger me is smarter than the current me. I don't know whether to laugh or cry when I do a Google search, only to answer my own question with an old blog post I wrote. Other times, I'll go back to old workbooks to remind myself how I executed a certain tactic. I'm also a big proponent of Tableau Public and am always encouraging people to download dashboards they like to "look under the hood" and see how they were created.

Whether you are reverse-engineering your own or somebody else's work, this next tip will help. Even if you've heard of describing fields, you may not know that you can describe entire sheets.

Here's how a sheet description looks, which can be accessed by clicking "Worksheet" in the top navigation, and choosing "Describe Sheet...".

**Sheet Description**

**Description of "Sales Trend"**

The trends of CP Sales and PP Sales for Date Equalizer with Granularity. Color shows details about CP Sales and PP Sales. The data is filtered on Action (Region = Region Parameter,Country,State), Action (Region = Region Parameter,Region Abbreviation,Region), Region Filter, Action (US Map Color,State) and Action (Region = Region Parameter,State). The Action (Region = Region Parameter,Country,State) filter keeps 49 members. The Action (Region = Region Parameter,Region Abbreviation,Region) filter keeps 4 members. The Region Filter filter keeps True. The Action (US Map Color,State) filter keeps 49 members. The Action (Region = Region Parameter,State) filter keeps 49 members.

**Marks**  
The mark type is Line (Automatic).  
Stacked marks is off.

**Shelves**

**Rows:** Measure Values  
**Columns:** Date Equalizer with Granularity  
**Filters:** Measure Names, Action (Region = Region Parameter,Country,State), Action (Region = Region Parameter,Region Abbreviation,Region), Region Filter, Action (US Map Color,State), Action (Region = Region Parameter,State)  
**Color:** Measure Names

**Dimensions**

**Measure Names** has 2 members on this sheet  
Members: CP Sales; PP Sales  
Measure Names is sorted manually.  
**Region Filter** has 1 members on this sheet  
Members: True  
The formula is [Region Parameter] = "US" OR [Region Parameter] = [Region]  
**Date Equalizer with Granularity** ranges from 9/1/2016 to 3/1/2017 on this sheet.  
The formula is DATE(CASE [Date Granularity]

```

WHEN "Week" THEN DATETRUNC('week',[Date Equalizer])
WHEN "Month" THEN DATETRUNC('month',[Date Equalizer])
WHEN "Quarter" THEN DATETRUNC('quarter',[Date Equalizer])
WHEN "Year" THEN DATETRUNC('year',[Date Equalizer])
END)

```

**Measures**

**Measure Values** ranges from 4,969 to 27,645 on this sheet.

**Sets**

**Action (Region = Region Parameter,Country,State)** has 49 members on this sheet  
Members: False, United States, Alabama; False, United States, Arizona; False, United States, Arkansas; False, United States, California; False, United States, Colorado; ...  
**Action (Region = Region Parameter,Region Abbreviation,Region)** has 4 members on this sheet  
Members: False, C, Central; False, S, South; False, W, West; True, E, East  
**Action (Region = Region Parameter,State)** has 49 members on this sheet  
Members: False, Alabama; False, Arizona; False, Arkansas; False, California; False, Colorado; ...  
**Action (US Map Color,State)** has 49 members on this sheet  
Members: NULL, Alabama; NULL, Arizona; NULL, Arkansas; NULL, California; NULL, Colorado; ...

**Copy** **Close**



This provides so much information that I actually ran out of room in the screenshot. In addition to the information pictured, you can also see all of the worksheet's parameters as well as their current values, and even information about the data source being used to generate the worksheet.

Most of this information can be seen by looking at the location of pills on the Rows Shelf, Columns Shelf, Filters Shelf, and Marks Cards – but not all of it. For example, you are now able to see whether or not the marks are stacked and, in this example, view the underlying calculation for the “Date Equalizer with Granularity” field.

Not only does ‘Describe Sheet’ provide a concise total picture of what’s happening in the worksheet, but you can use the Copy button pictured in the bottom-left corner of the previous image to copy and paste the text into a notepad or Word document. You can then use the code to recreate formulas you may not otherwise be familiar with as calculated fields.

**Sheet Description**

**Description of "Sales Trend"**

The trends of CP Sales and PP Sales for Date Equalizer with Granularity. Color shows details about CP Sales and PP Sales. The data is filtered on Action (Region = Region Parameter,Country,State), Action (Region = Region Parameter,Region Abbreviation,Region), Region Filter, Action (US Map Color,State) and Action (Region = Region Parameter,State). The Action (Region = Region Parameter,Country,State) filter keeps 49 members. The Action (Region = Region Parameter,Region Abbreviation,Region) filter keeps 4 members. The Region Filter filter keeps True. The Action (US Map Color,State) filter keeps 49 members. The Action (Region = Region Parameter,State) filter keeps 49 members.

**Marks**

The mark type is Line (Automatic).  
Stacked marks is off.

**Shelves**

**Rows:** Measure Values  
**Columns:** Date Equalizer with Granularity  
**Filters:** Measure Names, Action (Region = Region Parameter,Country,State), Action (Region = Region Parameter,Region Abbreviation,Region), Region Filter, Action (US Map Color,State), Action (Region = Region Parameter,State)  
**Color:** Measure Names

**Dimensions**

**Measure Names** has 2 members on this sheet  
Members: CP Sales; PP Sales  
Measure Names is sorted manually.  
**Region Filter** has 1 members on this sheet  
Members: True  
The formula is [Region Parameter] = "US" OR [Region Parameter] = [Region]  
**Date Equalizer with Granularity** ranges from 9/1/2016 to 3/1/2017 on this sheet.  
The formula is DATE(CASE [Date Granularity])  
  
WHEN "Week" THEN DATETRUNC('week',[Date Equalizer])  
WHEN "Month" THEN DATETRUNC('month',[Date Equalizer])  
WHEN "Quarter" THEN DATETRUNC('quarter',[Date Equalizer])  
WHEN "Year" THEN DATETRUNC('year',[Date Equalizer])  
END)

**Measures**

**Measure Values** ranges from 4,969 to 27,645 on this sheet.

**Sets**

Action (Region = Region Parameter,Country,State) has 49 members on this sheet  
Members: False, United States, Alabama; False, United States, Arizona; False, United States, Arkansas; False, United States, California; False, United States, Colorado; ...  
Action (Region = Region Parameter,Region Abbreviation,Region) has 4 members on this sheet  
Members: False, C, Central; False, S, South; False, W, West; True, E, East  
Action (Region = Region Parameter,State) has 49 members on this sheet  
Members: False, Alabama; False, Arizona; False, Arkansas; False, California; False, Colorado; ...  
Action (US Map Color,State) has 49 members on this sheet  
Members: NULL, Alabama; NULL, Arizona; NULL, Arkansas; NULL, California; NULL, Colorado; ...

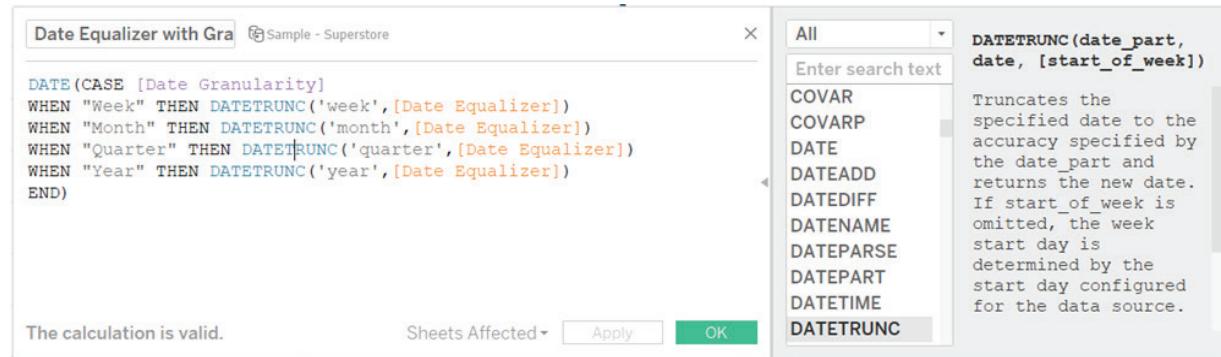
**Copy** **Close**

# Calculated Field Writing

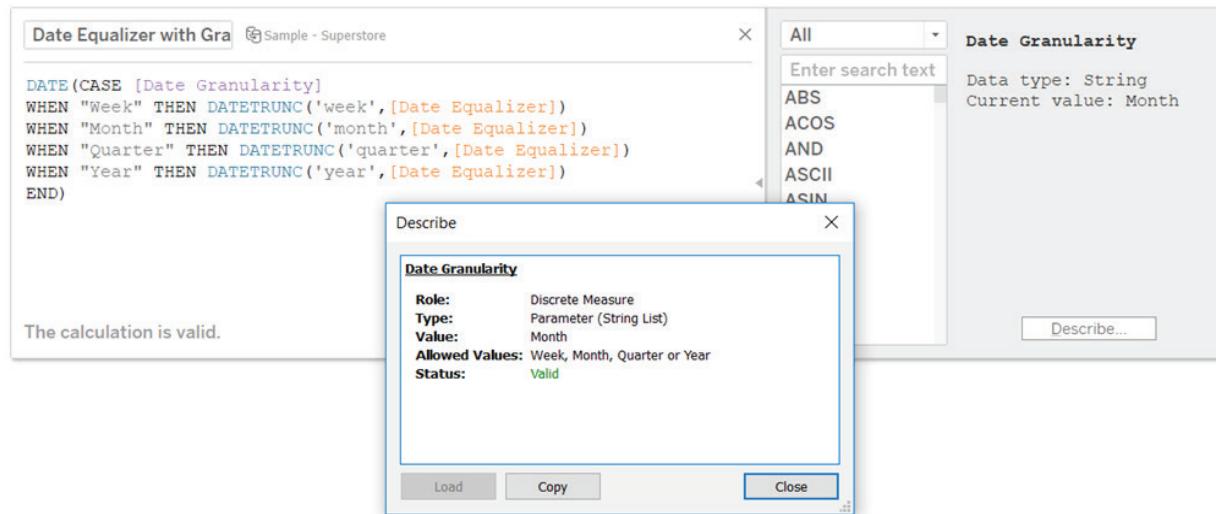
Speaking of writing calculated fields, the describe feature also has ‘an app(lication) for that’ – see what I did there? Let’s say that instead of getting the formula for the “Date Equalizer with Granularity” calculated field, or just to take a closer look, we right-clicked on the field and chose “Edit...”. This would open a calculated field dialog box that looks like this:



Clicking on anything colored blue, purple, or orange will give you additional information in the ‘fly-out’ box on the right-side. Here’s what happens when I click “DATETRUNC”:



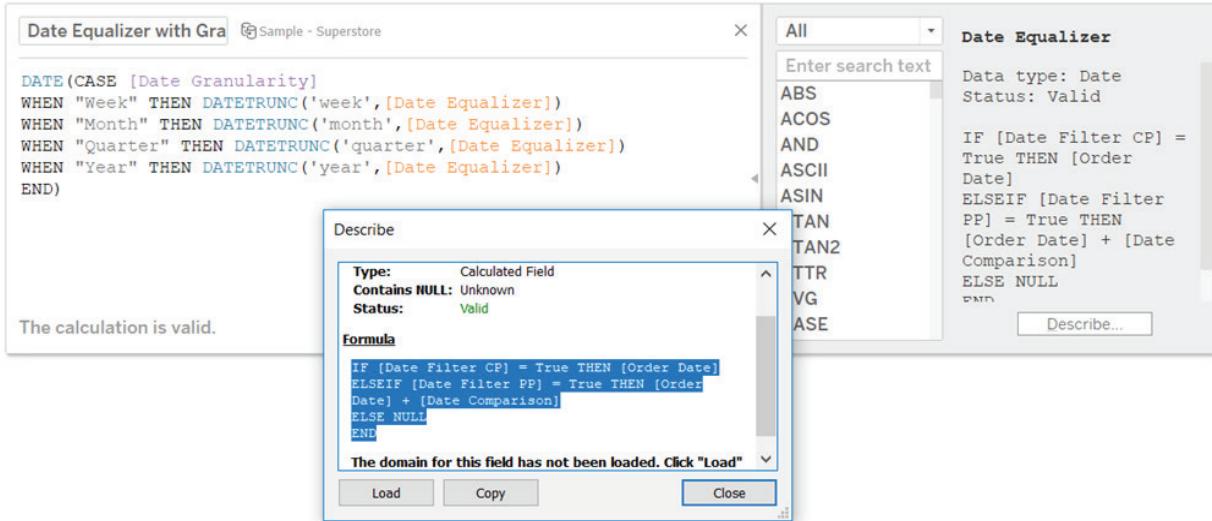
When clicking anything that's purple (parameters) or anything that's orange (fields), you will see the option to describe them. For parameters, I like to use this describe button to see all of the allowable values, which in this case, helped me write the CASE logic.



```
DATE(CASE [Date Granularity]
WHEN "Week" THEN DATETRUNC('week', [Date Equalizer])
WHEN "Month" THEN DATETRUNC('month', [Date Equalizer])
WHEN "Quarter" THEN DATETRUNC('quarter', [Date Equalizer])
WHEN "Year" THEN DATETRUNC('year', [Date Equalizer])
END)
```

The calculation is valid.

For calculated fields, I like to use the describe button to copy and paste underlying code. While you can view the underlying code on the right by just clicking on an orange calculated field, using the describe button opens more information including text which can be copied.



The screenshot shows the Tableau Calculated Field Editor. A modal window titled "Describe" is open, displaying the following information:

- Type:** Calculated Field
- Contains NULL:** Unknown
- Status:** Valid
- Formula:**

```
DATE(CASE [Date Granularity]
WHEN "Week" THEN DATETRUNC('week',[Date Equalizer])
WHEN "Month" THEN DATETRUNC('month',[Date Equalizer])
WHEN "Quarter" THEN DATETRUNC('quarter',[Date Equalizer])
WHEN "Year" THEN DATETRUNC('year',[Date Equalizer])
END)
```
- Note:** "The domain for this field has not been loaded. Click "Load""
- Buttons:** Load, Copy, Close

In the background, the main editor window shows the calculated field name "Date Equalizer" and its definition:

```
IF [Date Filter CP] = True THEN [Order Date]
ELSEIF [Date Filter PP] = True THEN [Order Date] + [Date Comparison]
ELSE NULL
END
```

A tooltip-like sidebar on the right provides additional details about the field:

- Data type: Date
- Status: Valid
- IF [Date Filter CP] = True THEN [Order Date]
- ELSEIF [Date Filter PP] = True THEN [Order Date] + [Date Comparison]
- ELSE NULL
- END

In both of these cases, I'm able to write new calculated fields much faster because I don't need to close the calculated field window I am working in to re-open a parameter to see its allowable values or re-open a different calculated field to take advantage of code that was already created!

# Chapter 7



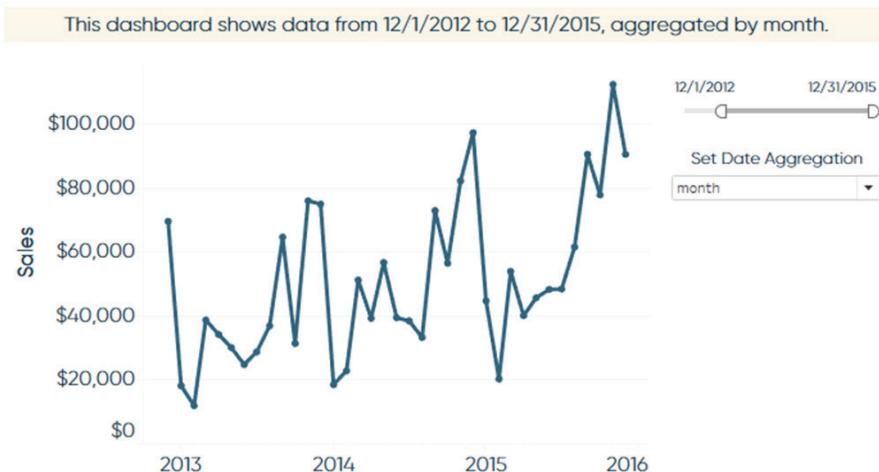
## 3 Ways to Add Alerts to Your Dashboards

One of the biggest advantages to using a tool like Tableau is that the software can do much of the ‘heavy lifting’ for you. For example, you can build in logic to improve the usability of your dashboards once, and have it work for you or your end users from that point on. One application of this is to add alerts to your dashboards that help communicate notifications or insights.

You may want to use alerts to notify users or remind yourself that a certain filter is on. Perhaps you have specific performance thresholds that are tolerable to your business, but want to be alerted somehow if performance is ever higher or lower than expected. Adding alerts to your dashboards helps reduce the time to insight and elicit action that helps your business – the primary goal of analytics. This chapter will cover three different examples of dashboard alerts.

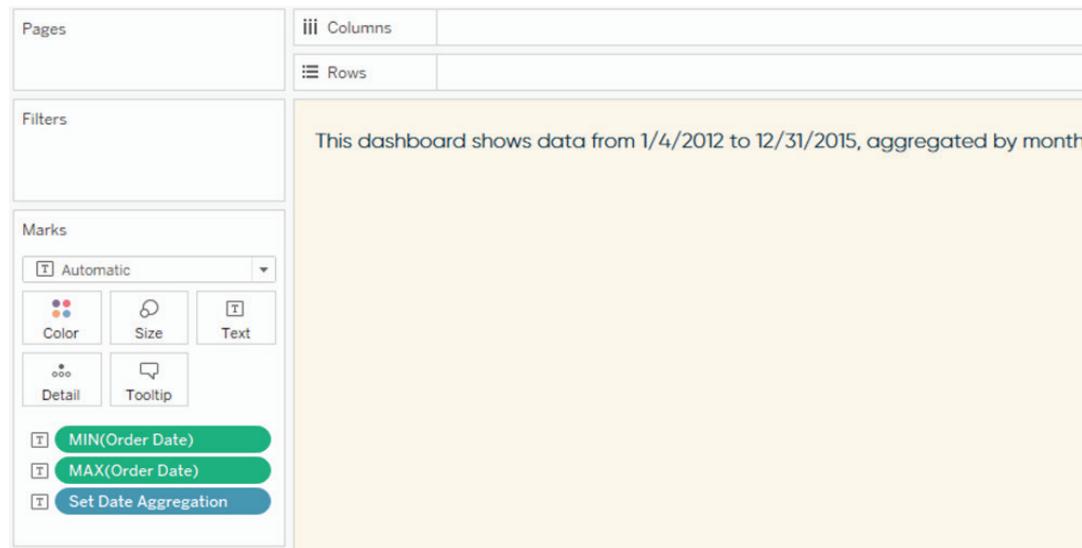
## Alert 1 – Date settings

The first type of alert that I frequently use in my own dashboards communicates which dates are being displayed and, if I'm using a line graph, how finely the dates are aggregated (daily, weekly, monthly, etc.). This alert serves two key purposes: (1) it clearly communicates how current the data is, and (2) helps avoid confusion in the case that the data has not updated or a view doesn't look as expected due to the aggregation. Here is an example using the Sample – Superstore dataset:



To create this alert, simply start a new worksheet and make a text-based view by adding the information you want included in the alert to the Text Marks Card. In this case, I added MAX(Order Date), MIN(Order Date), and the 'Date Aggregation' parameter. To add the two min and max dates, I right-clicked and dragged the 'Order Date' dimension onto the Text Marks Card, which allowed me to select the date field I wanted to display.

Once you have the information you want on the view, you can click into the Text Marks Card to format the text and written logic as you wish. At this point, your text sheet should look like this:



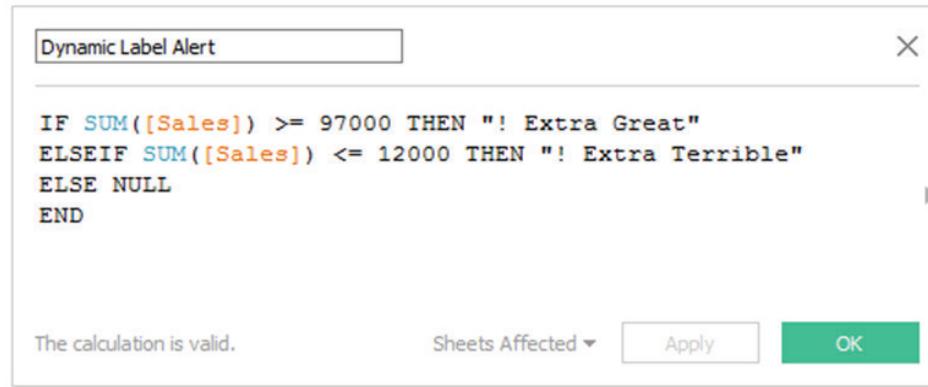
Finally, ensure the Order Date filter on your line graph is applied to this text sheet. This way, when an end user changes the date range, the MIN and MAX dates will change to match the filter.

## Alert 2 – Dynamic Labels

Wouldn't it be great if mark labels only showed up when something extra important happened? This is actually straightforward in Tableau. The trick is to build your logic for what's "extra important" in a calculated field, then use that calculated field as your label. Using the same sales over time example from above, here's a look at dynamic labels in action:



Notice that I have placed a calculated field called 'Dynamic Label Alert' on the Label Marks Card. Here is the underlying logic for that calculated field:



This is a simple example, but you can code the logic in the calculated field to alert you when anything important to you occurs. Perhaps you want to know when a goal is met or if any marks are outside of one or two standard deviations from the mean.

## Alert 3 – Heat map dashboard with optional Tableau Server email

In the third dashboard alert example, we will again leverage calculated fields. Only this time, instead of adding our logic of interest to the Label Marks Card, we will add it to the Color Marks Card – creating a heat map that immediately draws attention to our best and worst performing segments.

Let's imagine a C-Level executive has requested a dashboard showing the sales by region and product sub-category. They normally receive this information in a spreadsheet, but Tableau allows us to add value by immediately surfacing the most pertinent information. We decide to add value to the existing crosstab by alerting the executive in-line with the raw numbers whether the sales for a particular segment are at or below 40% of last year's sales or at or better than 160% of last year's sales.

This logic looks something like this under the hood:

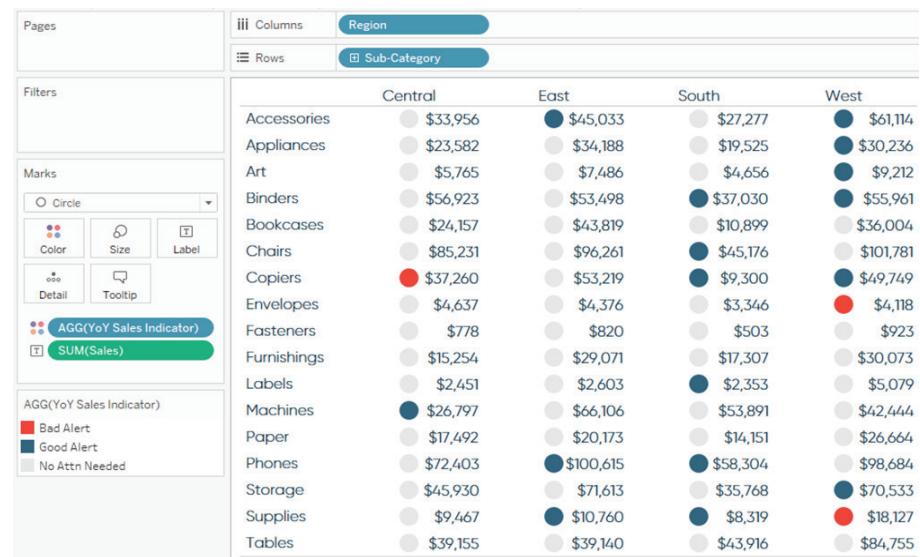
```
IF [YoY Sales Index] <= 40 THEN "Bad Alert"  
ELSEIF [YoY Sales Index] >= 160 THEN "Good Alert"  
ELSE "No Attn Needed"  
END
```

Once this calculated field is created, you can add it to the Color Marks Card and change the mark type to Circle, creating a view like pictured on the right.

A high-level dashboard like this can be [subscribed to in Tableau Server or Tableau Online](#), dropping the most current overview in your or your executive's inbox every morning or week. From here, they can decide for themselves based on the alerts if they need to click through to a more robust, interactive dashboard on Tableau Server or Tableau Online for further analysis.

Note that this type of alert dashboard can be coded to any comparison you wish, such as performance compared to a goal, competitor, or date range. If you are comparing relative dates year over year (such as "last week"), use Week Number or an equalized date to ensure there is data in both years.

This chapter shared just three ways to add alerts to your dashboards, but the possibilities are almost limitless. Experiment with elements of these three approaches to design the alerts that make the most sense for your business and stakeholders.



# Chapter 8



## 3 Ways to Use Alt-Code Characters to Enhance Your Tableau Dashboards

ASCII, or American Standard Code for Information Interchange, is the standard when it comes to computers converting numbers into readable text. In addition to all of the ASCII text characters, Alt codes were introduced to cover all of the characters not available on a computer's hardware. While the primary purpose of these alternative symbols was to provide characters that required accent marks and foreign-language characters with different styles, there are many other symbols available on your machine when you hold the "Alt" key while entering a numeric code.

Since these codes exist, there are several characters you can use to enhance your data visualizations.

This chapter will show you how to dynamically show increases and decreases with the ▲ and ▼ characters, how to format with the ★ and • symbols, and how to create a viz in tooltip with the Alt code.

# Alt Code 1: Dynamically formatting numbers with triangles to show changes in performance

In my post, [How to Dynamically Format Numbers in Tableau](#), I illustrate how to provide a dynamic number format whether the number is positive or negative, regardless of the data type, and even when the number has a prefix and/or suffix. While that is my favorite approach when dealing with more than two scenarios, if I am only trying to illustrate a positive or negative change, I like to format the numbers with an up ( $\blacktriangle$ ) or down ( $\blacktriangledown$ ) triangle.

To use any of the Alt codes in this chapter:

1. Ensure the Num Lock key is turned on
2. Hold the 'Alt' key while entering the numerical code for the symbol
3. Let go

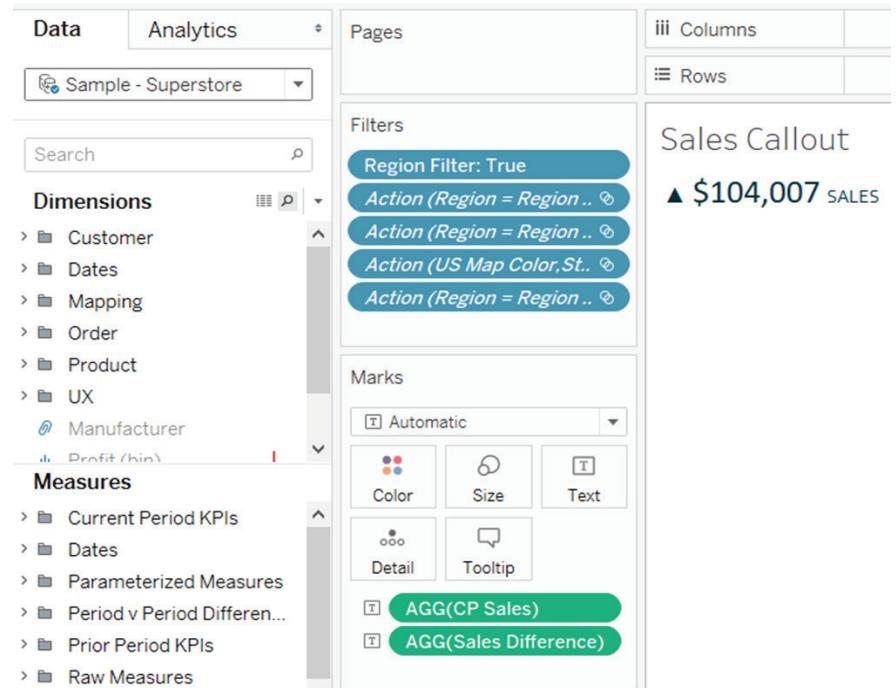
The Alt code for a triangle pointing up is 30 and the Alt code for a triangle pointing down is 31. If you are on a laptop and don't have access to the Num Lock key, you can also just copy and paste the characters within this chapter.

To show you how to put these into practice, we will be recreating one of the triangle indicators from the [Super Sample Superstore dashboard](#).



In the dashboard, positive changes are illustrated with the up triangle and negative changes are illustrated with a down triangle. This is a simple way to use [psychological schemas](#) (discussed in the next chapter) to communicate the insight efficiently.

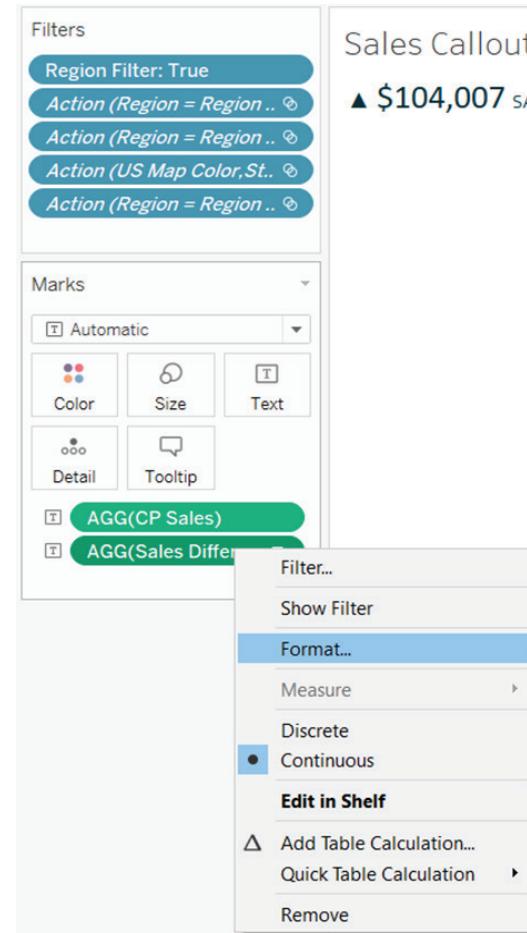
This is achieved by custom formatting a measure in Tableau Desktop. Here's how my Sales Callout from the previous screenshot looks 'under the hood'.



The screenshot shows the Tableau Desktop interface with the following components:

- Data Pane:** Shows the connection "Sample - Superstore".
- Analytics Pane:** Shows the "Region Filter: True" and several actions related to the Region filter.
- Marks Shelf:** Shows the following marks settings:
  - Automatic
  - Color
  - Size
  - Text
  - Detail
  - Tooltip
 Two measures are highlighted in green:
  - AGG(CP Sales)
  - AGG(Sales Difference)
- Sales Callout:** A callout box on the right side of the interface displays the text "Sales Callout" and "▲ \$104,007 SALES".

Note that this entire view is generated by just two measures: CP Sales and Sales Difference. The Sales Difference measure is what is being formatted into the up and down triangles. To do this, right-click on the measure you want to format and choose “Format...”.

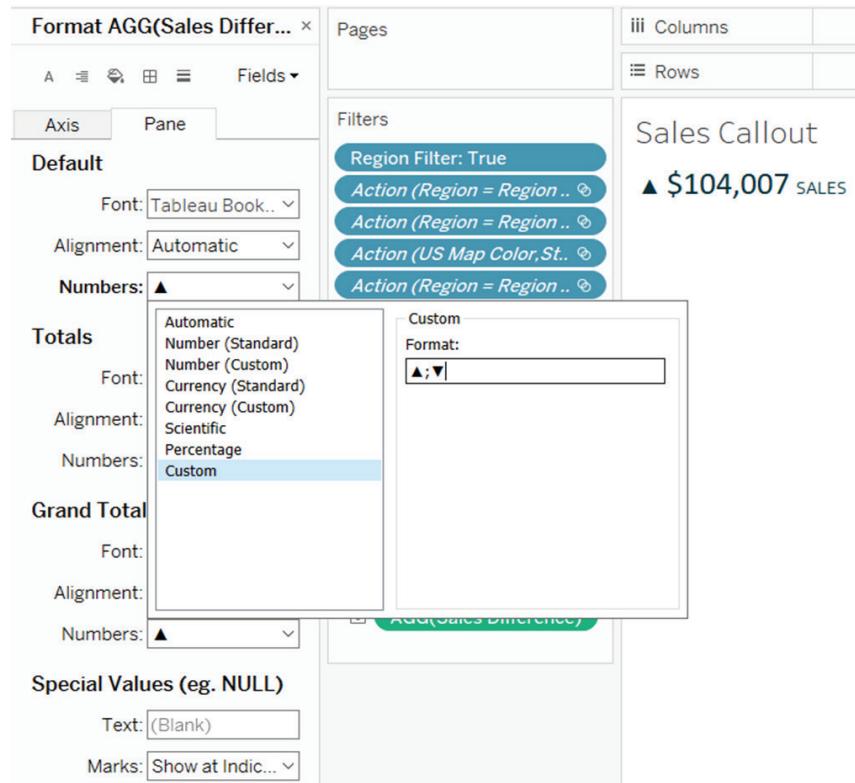


The screenshot shows the Tableau interface with the following components:

- Filters:** Region Filter: True, Action (Region = Region ...), Action (Region = Region ...), Action (US Map Color, St...), Action (Region = Region ...)
- Marks:** Automatic, Color, Size, Text, Detail, Tooltip.
- Selected Measure:** AGG(Sales Difference)
- Context Menu (open over AGG(Sales Difference)):**
  - Filter...
  - Show Filter
  - Format...** (selected)
  - Measure
  - Discrete
  - Continuous** (selected)
  - Edit in Shelf
  - Add Table Calculation...
  - Quick Table Calculation
  - Remove

A sales callout in the top right corner displays: ▲ \$104,007 SALES.

After clicking, a new formatting pane will appear on the left. Choose “Numbers”, then “Custom”. This is where you can copy and paste the up and down triangles, separating them with a semicolon “;”. Whatever symbol you place before the semicolon will show up when the value is positive; whatever symbol you place after the semicolon will show up when the value is negative.

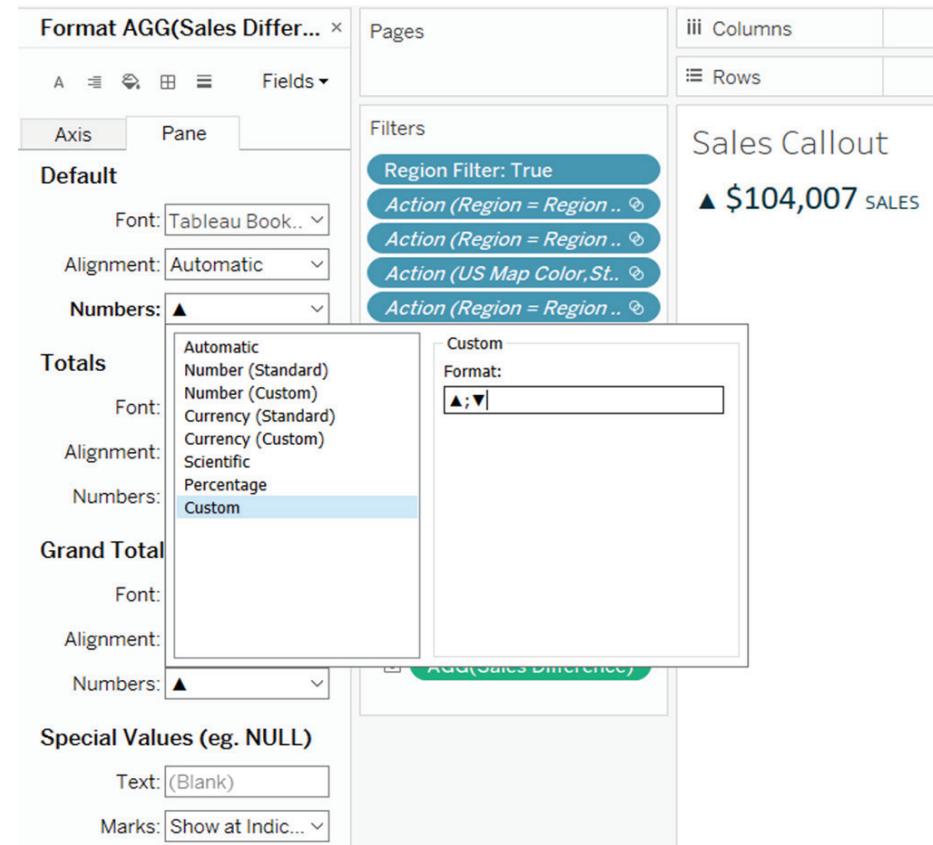


The screenshot shows the Tableau Format pane for the field "AGG(Sales Differ...)". The "Custom" format is selected under "Numbers". The "Format" field contains the custom separator ";". The "Font" is set to "Tableau Book..", "Alignment" to "Automatic", and "Numbers" to "▲;▼". The "Totals" section also shows the same custom format. The "Grand Total" and "Special Values (eg. NULL)" sections have their "Numbers" dropdowns set to "Custom".

For ease of this illustration, and to provide myself flexibility, I often like to create individual elements that can be combined in larger calculations. That being said, in this case, I could have also simply formatted the CP Sales measure with this “ \$#,##; \$#,##” to get the same effect with just one field. One disadvantage to the latter approach is that the sizing will be the same for the entire field (i.e. I can't have smaller triangles with larger numbers).

Lastly, you can add a third format by entering a second semicolon followed by the format. For example, sometimes I use a third format to communicate when something is not applicable or available to display as a number.

The custom format would look something like this “ \$#,##; \$#,##; -”. This custom format would display an up triangle with a dollar sign when there was a positive dollar change; a down triangle with a dollar sign when there was a negative dollar change, and a dash when there was no change or the calculation couldn't be computed.

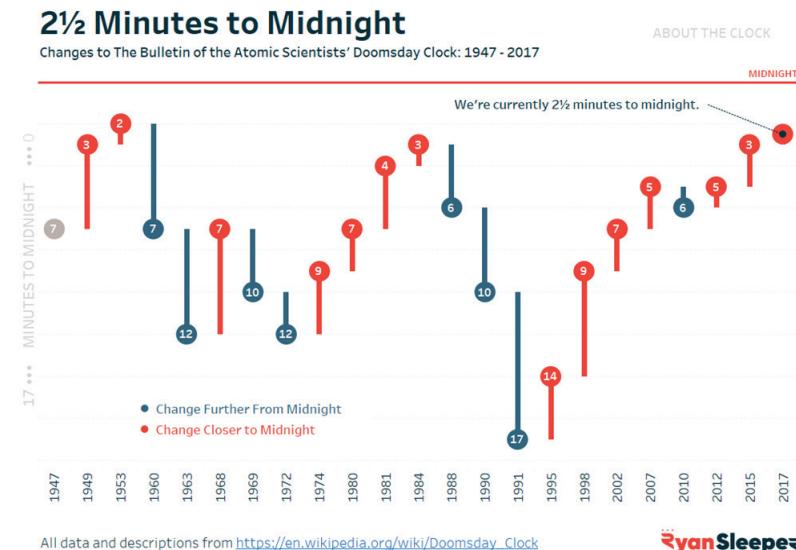


The screenshot shows the 'Format' dialog box for the 'AGG(Sales Difference)' field. The 'Default' section is selected, showing 'Font: Tableau Book...', 'Alignment: Automatic', and 'Numbers: ▲'. A dropdown menu for 'Numbers' is open, showing options like 'Automatic', 'Number (Standard)', 'Number (Custom)', etc., with 'Custom' selected. The 'Custom' section shows a 'Format:' input field containing the custom format code: ▲;▼. Other sections visible include 'Pages' (with a 'Sales Callout' card showing '▲ \$104,007 SALES'), 'Filters' (with several filter actions listed), 'Totals' (with a dropdown menu showing various numeric formats), 'Grand Total' (with a dropdown menu showing various numeric formats), and 'Special Values (eg. NULL)' (with 'Text: (Blank)' and 'Marks: Show at Indic...').

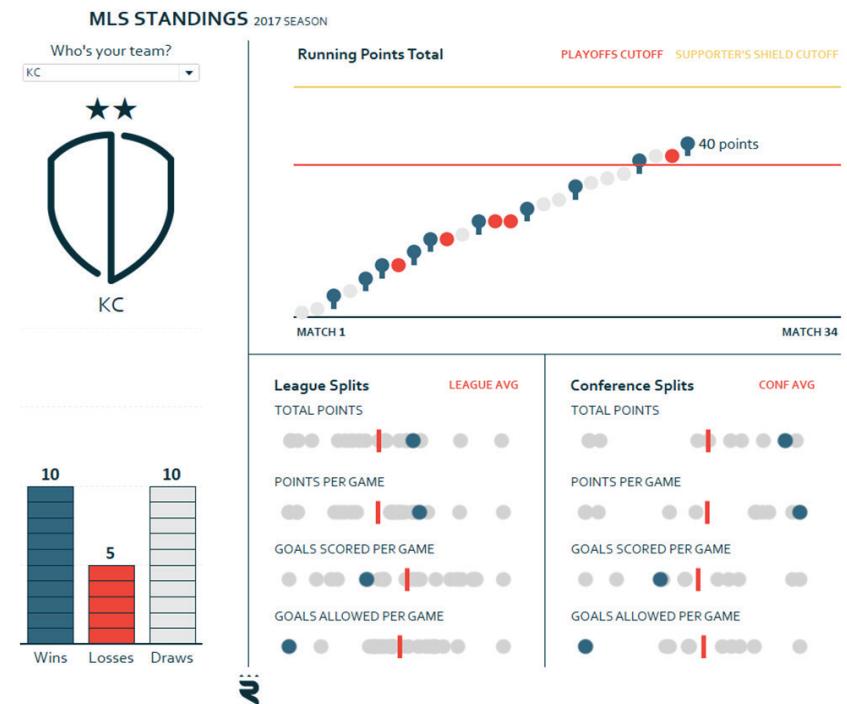
## Alt Code 2: Formatting your visualizations with the star and bullet symbols

Alt code symbols can also be used as a way to improve the formatting and add a unique style to your dashboards. One of the most common characters I use is the bullet (●), or Alt code 7. This symbol can be used in ordered lists, but also doubles as an alternative approach to color legends. I prefer the circle shape to the default color legend squares in Tableau, particularly when using a circle mark type as you would with a scatter plot.

Here is an example from my visualization, [2½ Minutes to Midnight](#).



In another example, I took this a step further by integrating the star symbol (★), or Alt code 9733. In soccer (football), it is tradition to display a star over the team's crest for each championship that they've won. So in my viz, [2017 Major League Soccer Standings](#), there is a field in the underlying data for championships won. Instead of displaying the number of championships, I put one star symbol for one championship, two star symbols for two championships, and so on. That way when a team is selected that has won at least one championship, the stars representing their championship(s) appear in the dashboard.



This same effect could have been achieved with a calculated field that tells Tableau how many stars to display depending on the number of championships won. The formula would look something like this:

```
IF [Championships Won] = 1 THEN "★"  
ELSEIF [Championships Won] = 2 THEN "★★"  
ELSEIF [Championships Won] = 3 THEN "★★★"  
ELSEIF [Championships Won] = 4 THEN "★★★★"  
ELSEIF [Championships Won] = 5 THEN "★★★★★"  
END
```

"A picture tells a thousand words" and these Alt code characters are a way to pack a lot of information into a single symbol. For the Kansas City fans in attendance for the coldest game in MLS history, a championship that went to a 10-round penalty kick shootout (the longest in MLS history), that star reminds them of quite a bit.

These are just two ideas for using Alt codes to format a visualization, but there is [an entire site](#) dedicated to different options.

## Alt code 3: How to make a bar chart viz in tooltip with squares or rectangles

<update> Tableau 10.5 with ‘Viz in Tooltip’ was released in January 2018, so this tip is finally antiquated! To learn more about this new functionality and see one application, read [How to Add an Image to a Tableau Tooltip.](#) </update>

While we have been excitedly awaiting the ‘viz in tooltip’ functionality that Tableau is currently working on, I have been trying to share alternative ways to add value to tooltips. Viz in tooltip allows you to show a secondary data visualization when you hover over a data point on a primary data visualization on a view.

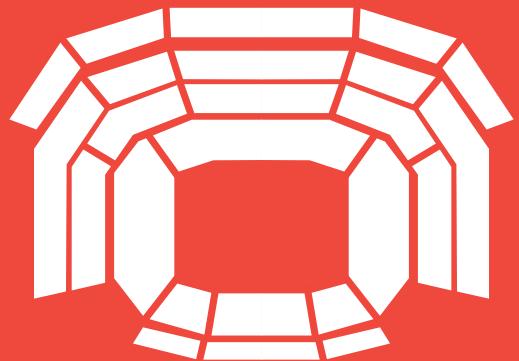
Believe it or not, Andy Cotgreave (Tableau evangelist and [data viz hero](#)), came up with a creative way to do this nearly seven years ago. Seriously... download the workbook in his post and it was made with version 6!

The tip involves combining a parameter with a string of Alt code squares (Alt+9632) or rectangles (Alt+219) with calculated fields that compute how many symbols to display. As Andy has had seven years to optimize his approach, and I do not have anything to add at this time, I’m going to point you his way for this last [tutorial](#).

For two more recent posts about adding value to tooltips, see:

- [How to Make Dynamic Tooltips in Tableau](#)
- [How to Add a Dimension to a Tooltip in Tableau](#)

# Chapter 9



## 3 Ways Psychological Schemas Can Improve Your Data Visualization

Whether you are aware of it or not, you are constantly recognizing and processing patterns in your everyday life. Think about when you go to dinner at a sit-down restaurant that you've never been to before. Even though you're trying the restaurant for the first time, you will have some expectation about the order of events, which usually looks like this:

- You walk in and are greeted by a host or hostess who shows you to a table
- The waiter comes by to introduce themselves and takes your drink order
- After returning with your drinks, the waiter takes your dinner order
- Throughout the meal, the waiter may stop by every once in a while to make sure you're doing okay and refill your drinks
- After the meal, the waiter will make a bad joke about 'saving room for dessert'
- After declining, they will either produce your bill from their pocket or run over to the register to get it for you
- You pay and go on your way

This is one of many examples of a psychological schema (not to be confused with database schema) in your everyday life. These patterns help society align diverse audiences and help us process varying situations very efficiently. These schemas are so powerful and ingrained that a disruption to the pattern can be confusing and challenging to overcome. As one extreme example, imagine showing up to the restaurant from above and having the waiter bring you a check before you've sat down.

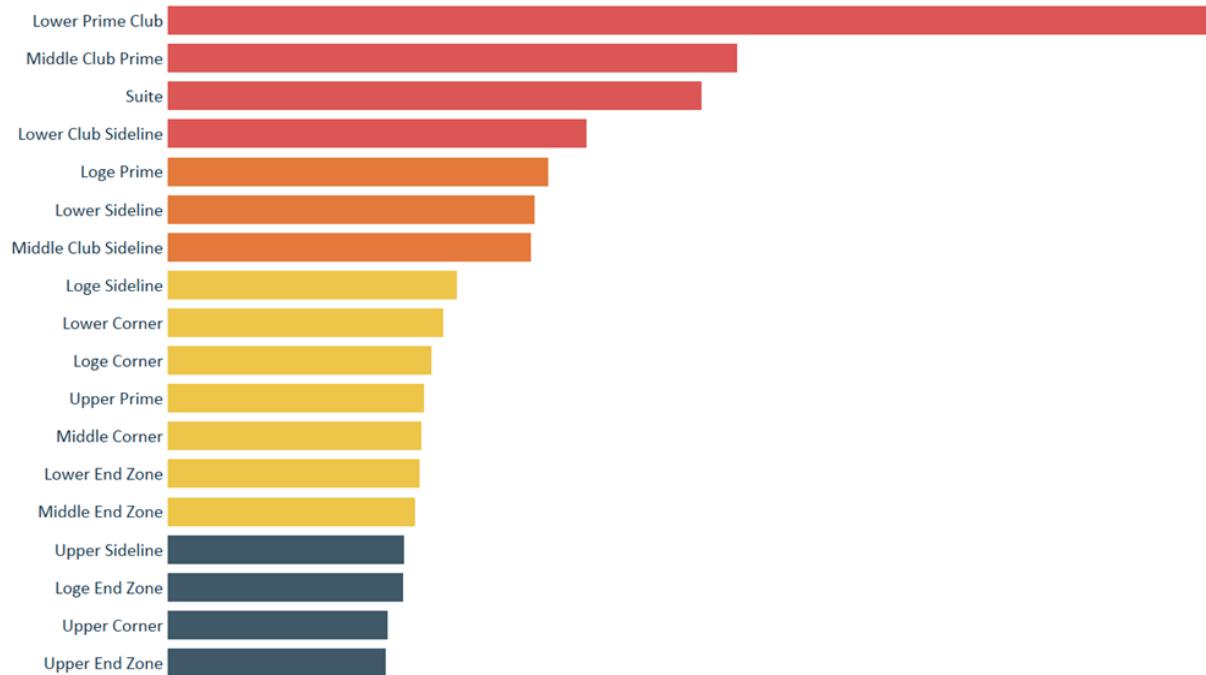
The restaurant pattern goes in an intuitive order so it is unlikely to vary much, but you also create schemas that are personal to you based on your own life experiences and worldview. These expectations help you avoid ‘reinventing the wheel’ because you’ve experienced the same or similar situation before and know how to handle it.

Schemas play an important role in data visualization because they have the ability to make or break the two biggest benefits of visualizing data: reducing time to insight and improving the accuracy of insights. Tap into your audience’s schemas and you improve their experience; disrupt their schemas and you run the risk of leading your audience in the wrong direction.

This chapter shares three ways to leverage schemas to improve your data visualization.

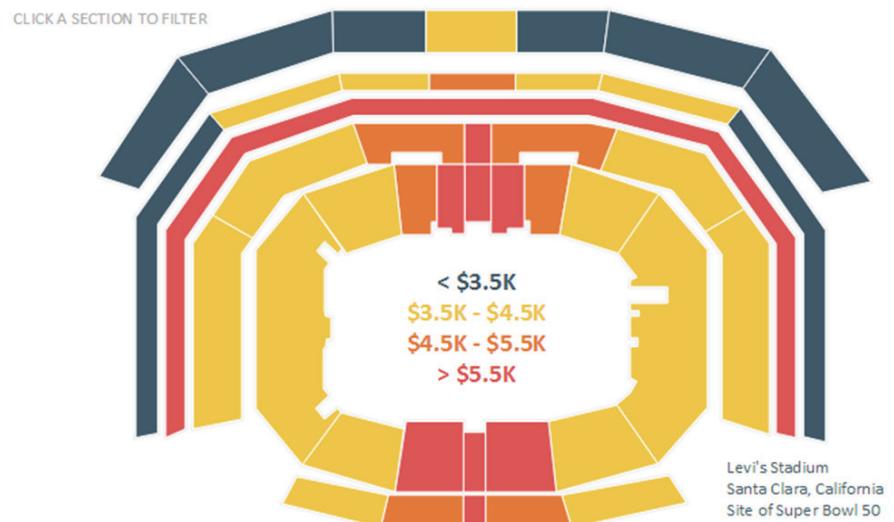
## Schema 1: Spatial Context

Maps help us process data because in addition to the data point, they provide spatial context that help our analyses. Consider the following bar chart showing the lowest cost per section to attend Super Bowl 50:



This is a good data visualization within best practices, and there are definitely insights to be found in this chart. However, adding spatial context immediately helps the analysis make sense, even if you are not familiar with the stadium where this game was played:

I can use the schema I've constructed in my mind from my experiences buying tickets to many sporting events over my lifetime to know that the lower and closer to midfield you are, the more expensive the ticket will be. This will help reduce my time to insight because it's much faster for me to determine if the numbers on the bar chart make sense to me intuitively or if there is a disruption to my schema, which in this case would also lead to insight (i.e. if lower bowl tickets are going for less money than the upper deck). Depending on what story you are trying to tell, it may make sense to display a map like this in addition to, or even instead of, the bar chart. For my [Cost of Attending Super Bowl 50 visualization](#), I chose to show the stadium map and a line graph.



## Schema 2: Icons / Shapes / Symbols

A picture tells a thousand words. The second schema that can improve your data visualization is the use of icons, shapes, and/or symbols. There's a thin line between graphics that enhance a data visualization and chartjunk, but when done tastefully, graphics have the ability to provide much more information than words alone.

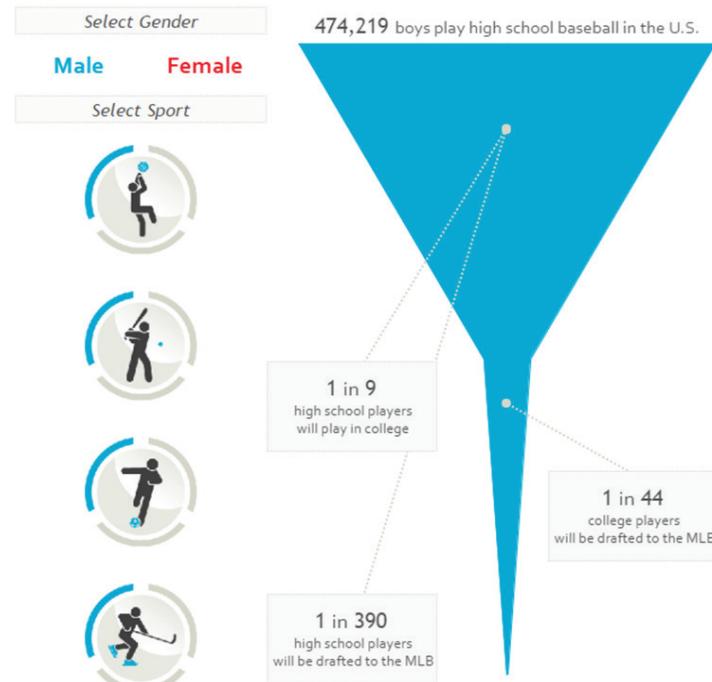
At a basic level, think about how much value a "+" or "-" sign adds when it precedes a number on a dashboard. When used to show if there was a positive or negative change in a KPI, just one character reduces the time to insight and increases the accuracy of insights. Arrows or triangles pointing up or down work the same way.

That's a simple example, but many graphics work in this way because we have preconceived notions about what they mean. Take a look at the navigation I used on my visualization, The Odds of Going Pro in Sports (pictured to the right).

I used icons instead of words to display the different sport options. In my opinion, this makes the view more engaging, but icons also provide the advantage of reducing, and often eliminating, language barriers.

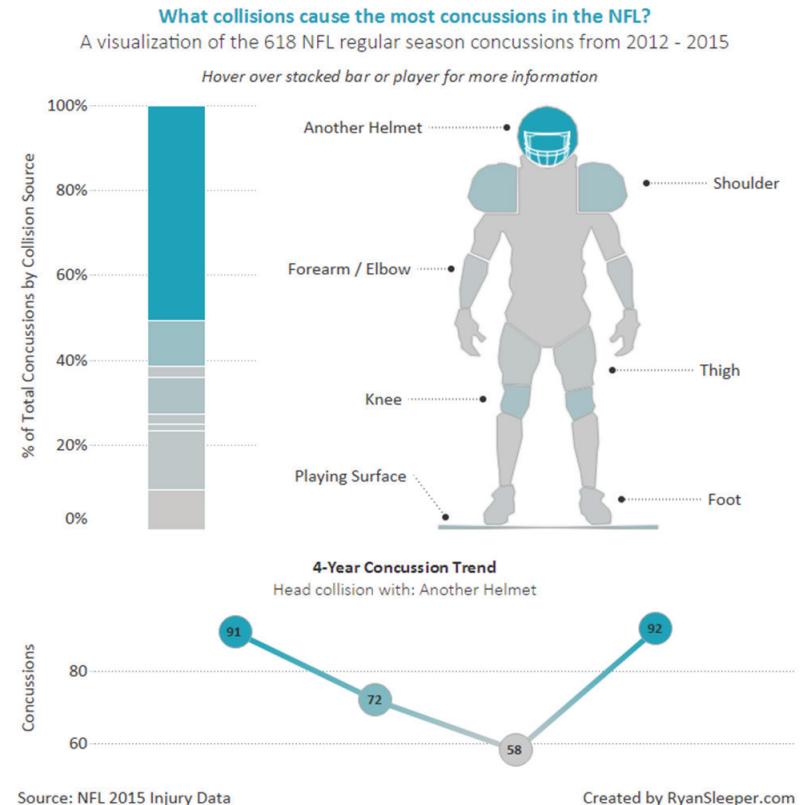
### WHAT ARE THE ODDS OF GOING PRO IN SPORTS?

An analysis of high school, college, and pro sports in the United States by gender.



In another example, I used a polygon map of a football player to help illustrate what body parts cause the most collisions in the NFL:

I could have simply made a bar chart by collision source, which would be a fine approach, but by overlaying the heat map onto the player's shape, the viewer can tap into their existing schemas to process the view and discover insights.



## Schema 3: Color

For better or worse, I bet you associate green with good and red with bad. I'm not exactly sure where that preconception was born, but it seems to be with us to stay. It is a schema. You don't have to teach your audience what red and green mean; they know (or think they know, so be careful!).

I don't recommend this color combination for both scientific (color blindness) and personal reasons (I believe it's ugly), but this is an example of a schema you can leverage to help your audience make sense of your data visualizations.

Being aware of your audience's existing associations can help you help them decrease their time to insight and improve the accuracy of their insights. It works both ways though, so be careful not to completely disrupt their schemas. If you're making a visualization about fruit, don't color oranges purple and grapes orange.

If you're using color outside of the common green/red or blue/orange color palettes, be consistent so your audience becomes conditioned to understand your use of color.

Leveraging these three schemas in a thoughtful way can go a long way toward maximizing the two biggest benefits of data visualization: decreasing the time to insight and improving the accuracy of insights. At the very least, be aware that your audience has their own preconceptions, and disrupting them can make it more challenging for your audience to find value in your data visualization.



# Chapter 10



## 3 Ways to Make Charts More Useful Than Pies

Despite being one of the least effective means of communicating data, I often see Tableau pie charts in corporate dashboards and Tableau Public visualizations. New users likely see pie charts as an easy way to spruce up their dashboards, but they are doing themselves a disservice because pie charts increase time to insight and reduce accuracy of insights – the opposite of what we are trying to achieve with data visualization. I've talked before about the science behind why you shouldn't use pie charts, so this chapter will be different.

When I share the shortcomings of pie charts, I am usually asked, "... but if I can't use pie charts, then how do I show a parts of a whole relationship?" Despite the limitations of pie charts – especially the fact that we can process bars more efficiently than the areas of a pie – this is still one of the most common questions I receive during [my Tableau trainings](#). This inspired me to document some better alternatives to using pie charts in Tableau.

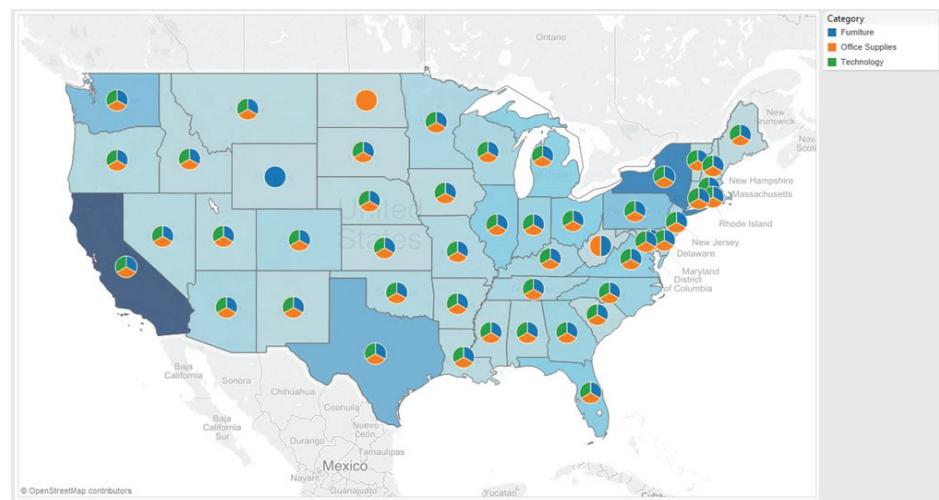
*Why does Tableau allow pie charts in the first place?*

You may be wondering, if pie charts are so bad, why does Tableau even allow you to build them with Show Me or by changing the mark type to 'Pie'?

When pie charts were first introduced to Tableau, they were meant to be used for the specific purpose of being a secondary mark type on a filled map. You may have seen this example from Tableau where they show the sales by category by US state using a pie chart on each state.

In this example, the colors on the filled map represent the total sales per state, and the pie is used to show the make-up of total sales in each state. In this scenario, using pie as the secondary mark type is the only way to accomplish this view.

Though their intentions were good, Tableau let the genie out of the bottle by introducing this mark type. I have actually never seen this intended use in a real dashboard, but users have instead adopted pie charts in several ways that are outside of best practices. Even in this example, you basically see 48 or so little peace signs because it is almost impossible to differentiate between the three categories. In this chapter, I will share a couple of alternatives to pie charts while building up to my recommended approach to visualizing a parts of a whole relationship.

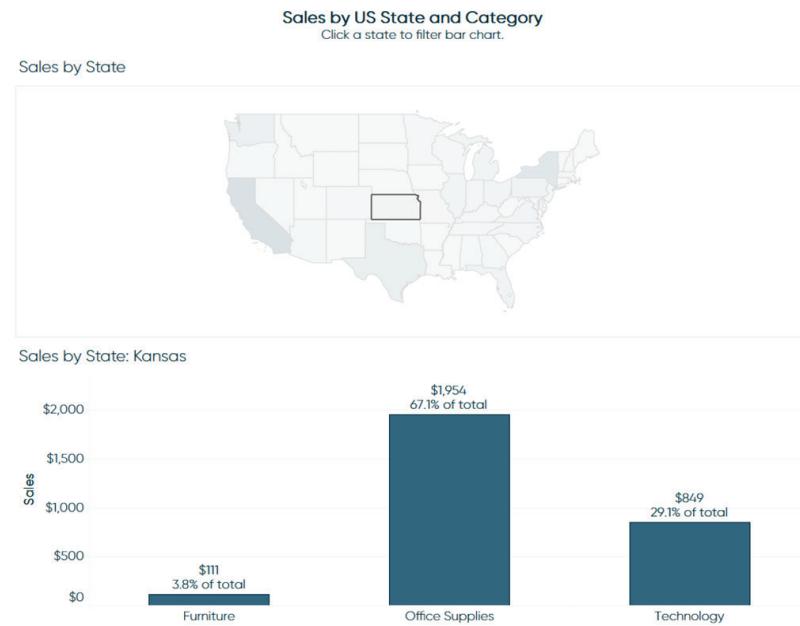


## Tableau pie chart alternative one: Bar Chart

I have two rules if you absolutely have to use pie charts in your Tableau workbooks. The first is to use five slices or fewer including “Other”. The “Other” slice acts as a catch-all for the remaining slices outside of the top four. More importantly, pie charts should never be used in a time-series analysis. So my first pie chart alternative is to simply use a bar chart, which is a great choice for comparing categorical values at one point in time.

One thing you lose in this approach compared to the map above is the quick comparisons between states. However, you can use [Tableau dashboard actions](#) to achieve something similar by having the bar chart update when a state is clicked on or hovered over.

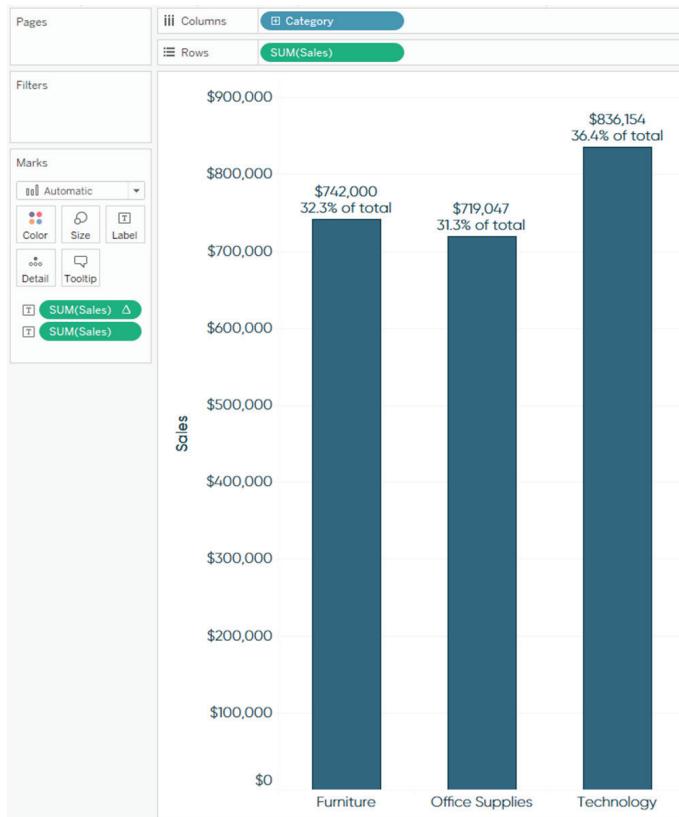
Another reason I like this approach is you gain real estate to include the actual sales numbers and/or the percent of the whole that each category is contributing.



## Tableau pie chart alternative two: Stacked Bars or Areas

Another Tableau pie chart substitute would be to use a stacked bar chart. While this option is slightly better than a pie chart, I do not like this alternative as much because it is challenging to compare trends across the view unless the stacked bar is on the bottom. On the positive side, a stacked bar adds some value because it is easier to process the category on the bottom, easier to process the value of the whole, and you have additional real estate to add context when compared to a pie.

It is easy to convert a bar chart to a stacked bar in Tableau by simply removing the dimension that is creating each bar from the Rows or Columns Shelf. With a stacked bar, the top of the highest bar represents the total, and each color below represents a contribution to that total.

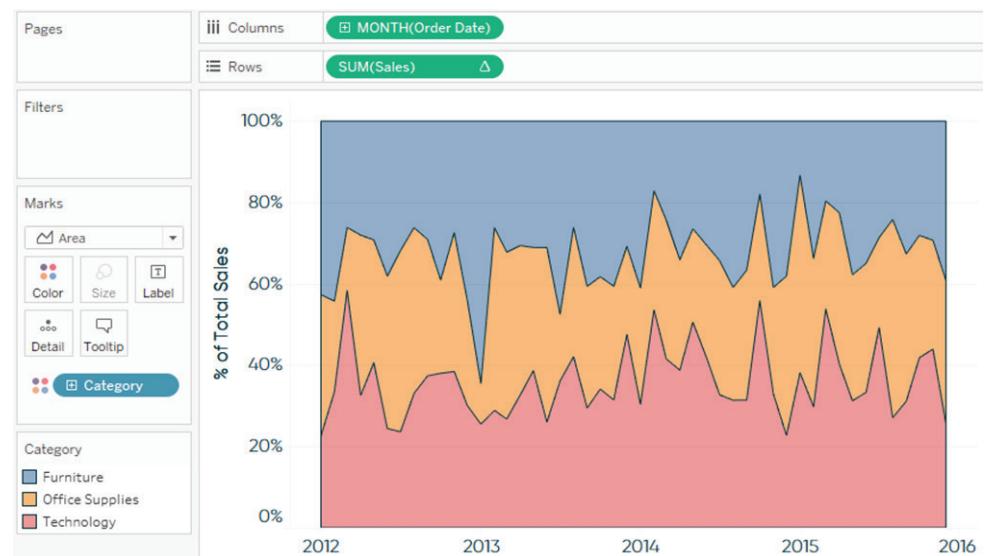


Whether you use a bar chart or a stacked bar chart, the values are for one point in time. For me, these chart types do not provide much value because they get stale very quickly in a corporate dashboard. In other words, the bars likely will not fluctuate a significant amount from week to week or month to month. Even if they do, because you are always looking at one point in time, you lose the comparison to prior time frames.

If you are going to use stacked bars, I suggest you use them to show how the distribution is changing over time (i.e. have one stacked bar for each time period). To take this a step further, you can add a quick table calculation to your measure called “Percent of Total” and change “Compute using” to “Table (Down)” to see what percent of the whole each category represented during each time frame.

To accomplish this, start by adding a table calculation for ‘Percent of Total’ to the measure by right-clicking the measure on the view (or clicking the down arrow on the measure’s pill), hovering over “Quick Table Calculation”, and choosing “Percent of Total”. Then change the direction of the table calculation to Table (down) instead of the default Table (across) by clicking into the measure’s pill again, hovering over “Compute Using”, and choosing “Table (down)”.

This table calculation will make every bar’s height 100%, then each color represents a share of that 100%. For a more cohesive view, change the mark type from Bar to Area to get a result like pictured to the right.



## Tableau pie chart alternative three: My Recommended Approach

I mentioned that we would be building up to my recommended approach, and while the alternatives provided to this point are all more effective than pie charts, they have their own limitations. Stacked areas like the one shown in the previous example can be challenging to decipher because unless the slice is on the bottom, it is difficult to precisely read the trend of each individual slice.

The first thing I recommend for the optimal parts of a whole visualization, is to change the mark type from Area to Line:

It is now easier to see the trend of each individual category. In this example from the Sample – Superstore dataset, the lines follow a similar trend so there is quite a bit of overlap. Clicking on the color legend highlights each category to help illustrate each individual trend.

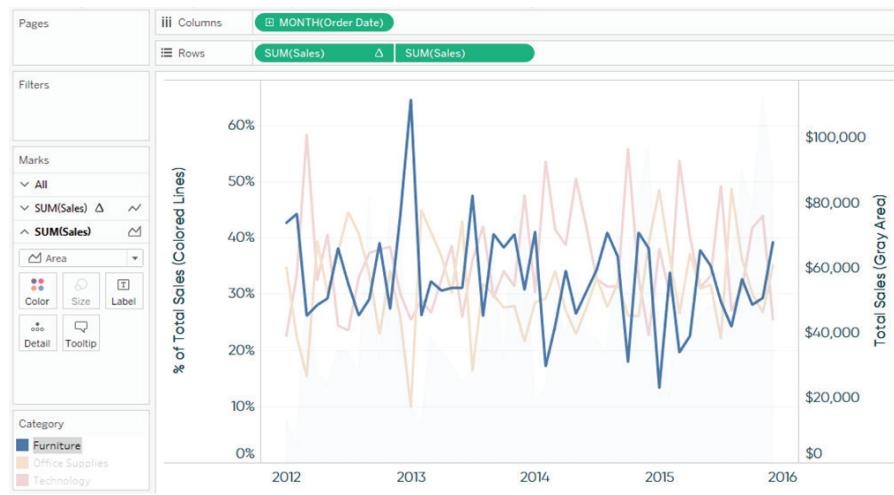
We can now see the sales contribution in percentages of each category to the total sales over time. One piece of context we lose with this approach is whether the total sales amount is trending up or down. The last thing I recommend is placing the Sales measure on the opposite axis, which creates a dual axis line graph. By default, Tableau colors the total sales by category. In this case, we only care about how the total sales amount is trending over time, so remove Category from the Color Marks Card on the SUM(Sales) Marks Shelf.



You are left with four lines instead of three, which is causing even more overlap. Total sales is a secondary insight, so I suggest changing its mark type to area and washing it out. To do this, navigate to the Marks Shelf for SUM(Sales) (without the Category dimension on the Colors Marks Card), change the mark type to 'Area', and change the opacity by clicking on the Color Marks Card. Finally, you can add total sales to the Tooltip Marks Card on the Percent of Total Sales Marks Shelf so that both the percent of total and total sales show up when you hover over each data point. Your finished product will look like the image on the right.

In review, in this chapter we have evolved our visualization of a parts to whole relationship from a static pie chart during one point in time, to a dual-axis combination chart showing how the distribution changes over time. We did this all without losing the context of how our total is trending.

This alternative to pie charts in Tableau will help reduce your time to insight, while also making your analysis more accurate, precise, and actionable.



# Chapter 11



## 3 Ways I Make Data Visualization Decisions

According to Google's dictionary, a mission statement is "a formal summary of the aims and values of a company, organization, or individual." Some of these stated missions are quite broad, leaving a lot of runway to innovate while keeping track of the business' original purpose. Consider Microsoft's, "Our mission is to empower every person and every organization on the planet to achieve more." Others are laser-focused on a specific product, such as Honest Tea's, "to create and promote great-tasting, healthy, organic beverages."

For me, a great mission statement is short, incorporates primary values, and provides some direction. The most important aspect of mission statements is that they help in decision making because they remind stakeholders of the purpose at hand. The idea is that these short statements can scale to keep everybody pulling in the same direction.

You've likely at least heard of mission statements, if not a few famous examples, but have you considered them in the context of data visualization? This chapter shares my data visualization mission statement and how it helps me solidify some of my opinions including why I believe data visualization is superior to spreadsheets and why I don't use pie charts.

# My Data Visualization Mission Statement

When I started my company, I put a lot of thought into my mission statement and ended up on, “Be the best-possible partner resource for translating data into valuable information.” Whether that’s through a training, dashboard development, or discovery analytics, these twelve words help me decide what projects to partner on. If I don’t have the capacity or skill alignment to be the best choice, I don’t try to get the project. If the project isn’t related to turning raw data into value, I don’t try to get the project. I have no problem telling a potential partner this because I don’t want them to waste money, and for me to waste time.

This mission statement has helped guide me where to partner, but I also have a set of principles I follow once I start working on a data visualization project. While somewhat unorthodox from a traditional mission statement, I believe it fits the definition from above. My data visualization mission statement is:

1. Reduce the time to insight
2. Increase the accuracy of insights
3. Improve engagement

If these three principles look familiar, it’s because I also call these the benefits of data visualization. If I ever have a decision to make while I’m creating a data visualization, I ask if the change will accomplish all three of these things. If it fails at any one of them, I simply don’t do it.



My favorite example, and the way I often get an audience to evolve from their spreadsheet mentality, is to show the difference between a raw table of data and a highlight table. When adding preattentive attributes to improve processing time, you always (1) reduce the time to insight and (2) increase the accuracy of insights. I also argue that a highlight table is much more (3) engaging than the raw table, so all three criteria are met.

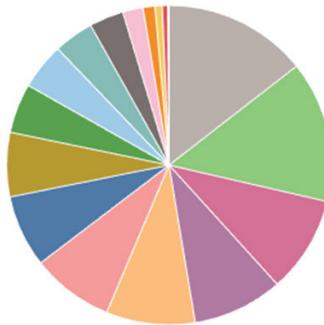
I wouldn't be doing what I'm doing if I didn't believe data visualization was the key to understanding data, but just illustrating that these principles affirm my reasoning. Let's look at another example: pie charts. When I argue against the use of pie charts, it's almost become a humorous debate. I'm often confronted with people who are just positive they have the one perfect use case where they are forced to choose a pie.

I want to go on the record to say that when I recommend avoiding pie charts, it's not because I'm a data visualization elitist living up on the hill in my castle. It's because of my data visualization mission statement. While it can be argued that pie charts are more engaging than a bar chart, they fail relatively dramatically at reducing the time to insight and improving the accuracy of insights.

Let's try to answer a basic business question using Tableau's Sample – Superstore dataset: What are the three worst-performing product sub-categories in sum of sales?

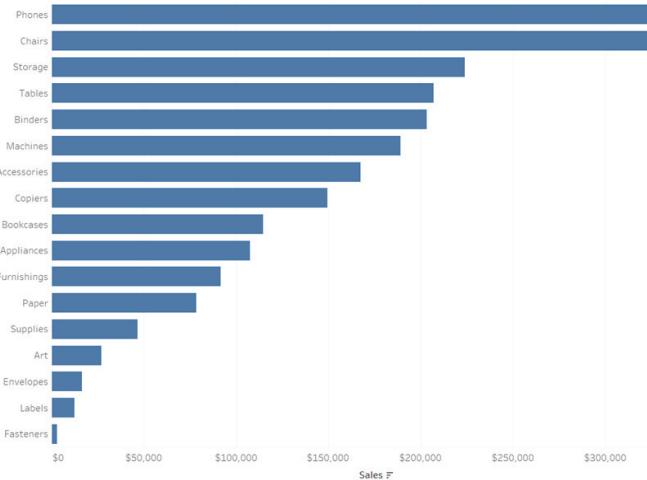


Sales by Sub-Category



*Pie Chart*

Sales by Sub-Category



*Sorted Bar Chart*

To keep it a somewhat fair fight, I sorted the dimension members in both charts. With the pie chart, the size of the slices prevents us from showing all of the labels. You can provide this information in a tooltip if using Tableau, but it doesn't help non-interactive versions of the view. This means you are forced to provide the sub-category names through a color legend, increasing the cognitive load on your end user (i.e. they have to look back and forth at the chart and the color legend). This increases the time to insight.

Further, we are much better at comparing the lengths of bars compared to areas of a pie. If I wasn't such a nice guy and sorted the charts, it would be much harder to determine the bottom three sub-categories when looking at the pie chart compared to the bar chart. Using a bar chart would provide more confidence that you are correctly picking the bottom three. In other words, it would increase the accuracy of your insight.

I know I'm not going to convince everyone, but sharing this because the reasoning behind my opinion on pie charts is objective. If you think you have the one case where a pie chart is easier to process and improves the accuracy of insights – go for it! I just have never come across one.

Pie charts are engaging, which is one of the reasons they're still so pervasive in reporting. I'm a huge believer in engagement, but it has to be done tastefully with a balance between the first two principles of the mission statement. To help make some more effective chart types more engaging, I documented [3 Ways to Make Beautiful Bar Charts](#) and [3 Ways to Make Lovely Line Graphs](#).

For some general ideas on making your data visualization more engaging, check out the [Design section of the Triple Crown Framework](#).



Thanks for  
reading,  
- Ryan

Ryan Sleeper

Founder and Principal, [Playfair Data](#)

Author of [Practical Tableau](#)