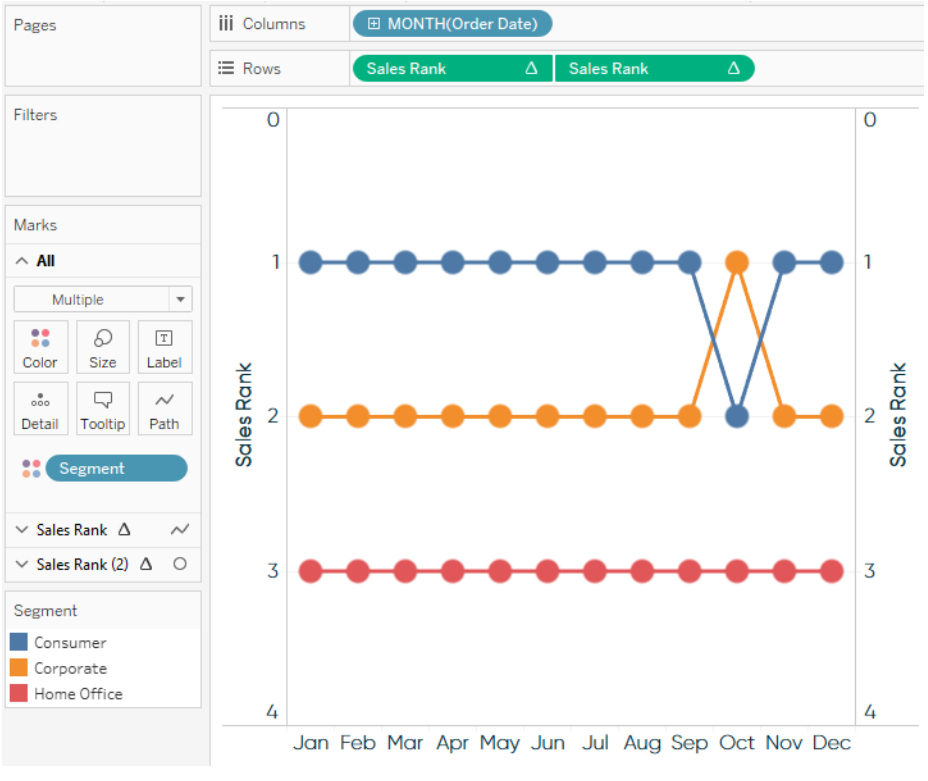
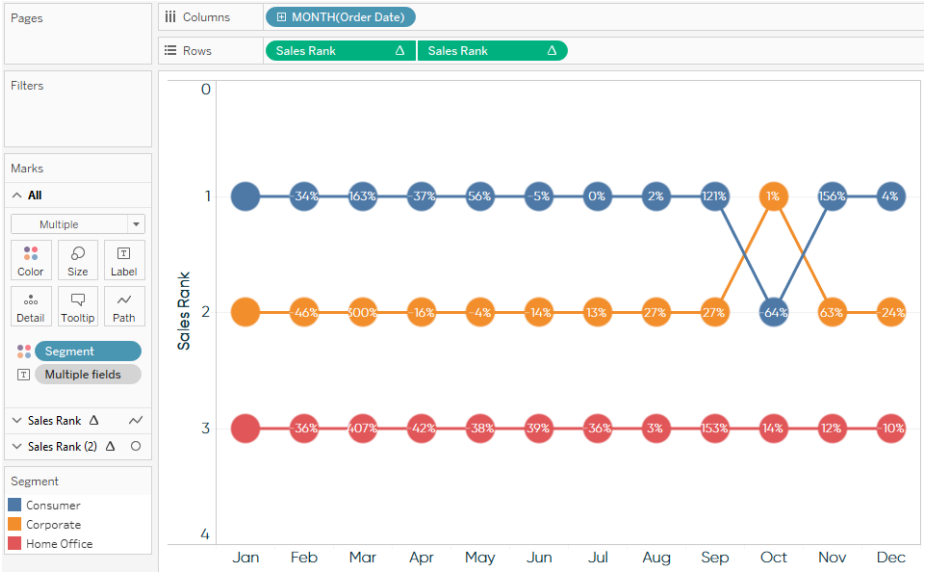
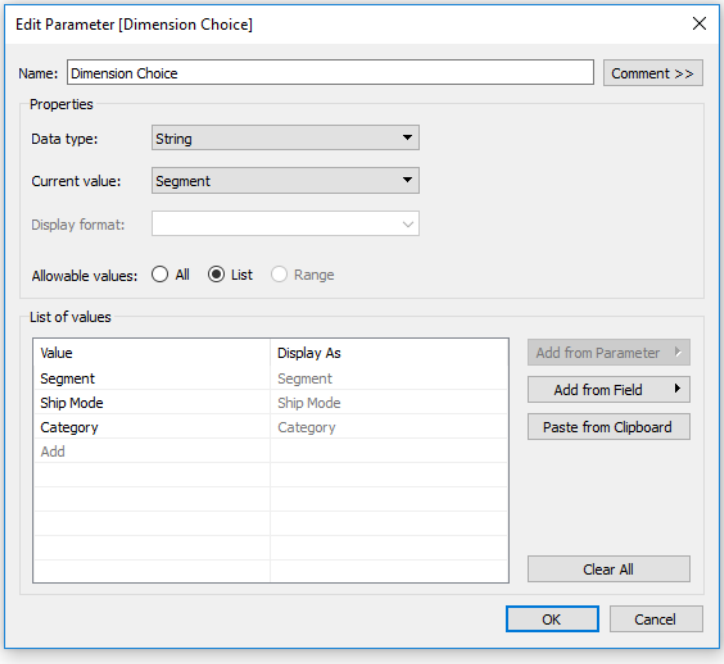


I now have some real estate to add some additional context to the marks. One possibility is to display the rank (1 through 3) for the mark on each circle. That may be a good use if you're dealing with many separate rows of marks, but with only three, that seems like a waste of valuable real estate. Instead, I will add the Sales measure to the Label Marks Card (for the circles only), then add a table calculation that computes the month-over-month percent change in sales for each segment. I also hid the right axis to finalize my bump chart:

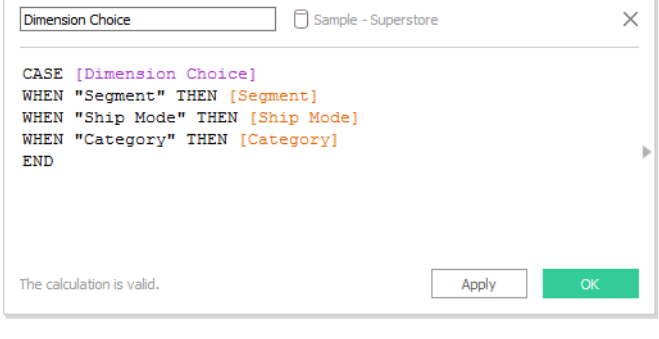


Our Tableau bump chart is now showing the month-over-month rank per segment, but also the month-over-month percent change in sales for each segment. This is turning out to be a solid static analysis, but why don't we add even more value in Tableau by allowing our end user to choose what dimension is used in the sales ranking.

To accomplish this, I will use this trick outlined in Chapter 64 to allow end users to choose measures and dimensions. For this example, I will allow my end users to choose from the Segment, Ship Mode, or Category dimensions. The first step is to create a string parameter with the choices of Segment, Ship Mode, and Category:



Then I will create a calculated field to give Tableau instructions on what to display for each parameter choice:

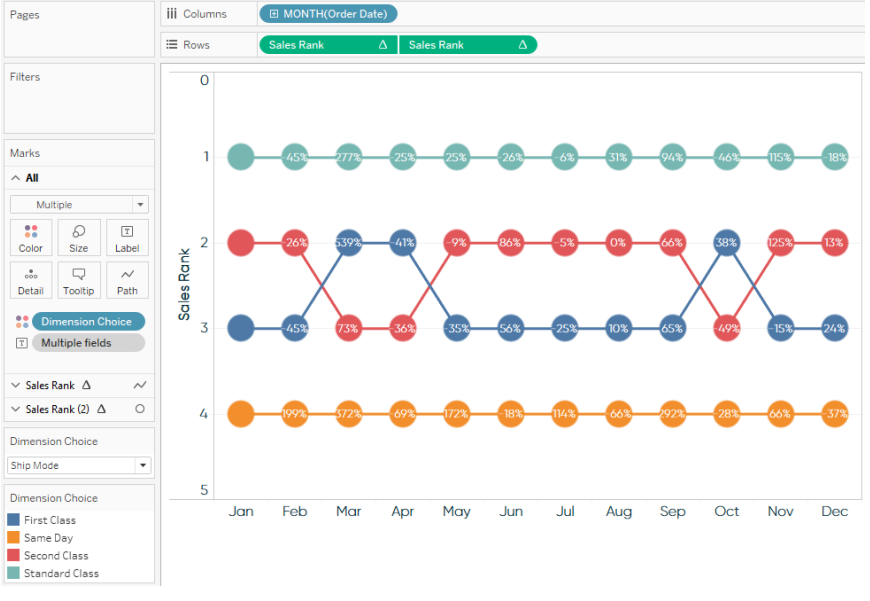


Now if I replace the Segment dimension on the Color Marks Cards of my bump chart with my newly created Dimension Choice field, the lines and circles will be colored based on what my end user selects in the parameter.

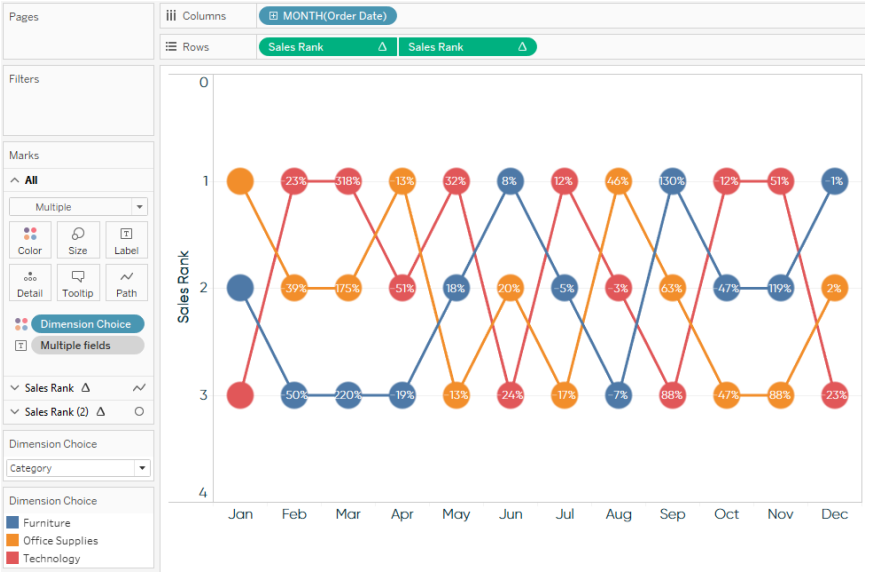
In addition to placing the Dimension Choice dimension on the Color Marks Card for both axes, there are two more small steps to get this working:

1. Ensure the table calculations on the Rows Shelf are now computing on Dimension Choice (they were previously computing on Segment).
2. Show the parameter control for Dimension Choice so the end user can change the selection (to do this, right-click the parameter and choose Show Parameter Control).

   Now if I select Ship Mode in the parameter control, my bump chart rankings are based on Ship Mode:



   If I choose Category, my bump chart rankings are based on Category:



We now have a dynamic, dual-axis bump chart that is providing multiple layers of insight for our end users, and allows them to control the visualization. Your wheels may already be turning at the possibilities for your own analyses, and yes, you can also use the parameter selection approach to change the measure for the ranking!

CHAPTER 45

How to Make Dumbbell Charts

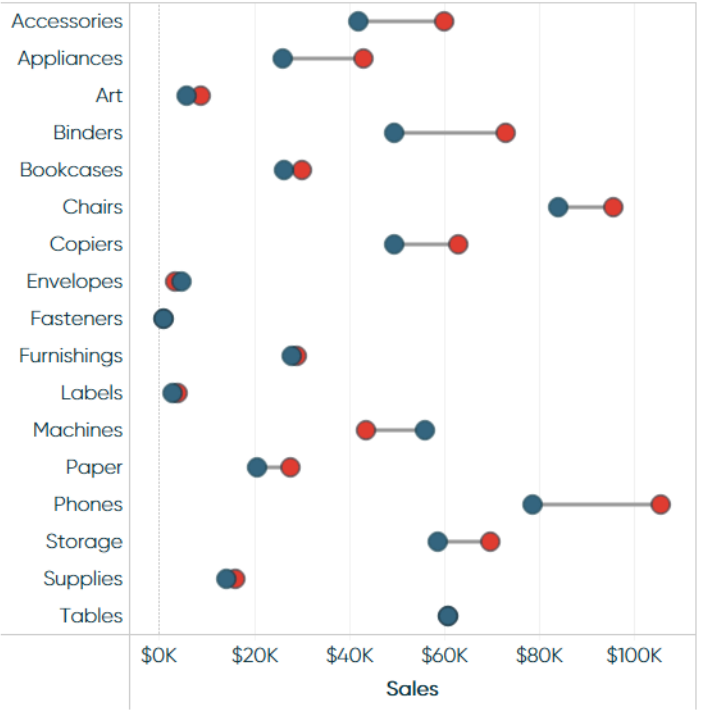
Tableau dumbbell charts, also known as DNA charts, are an alternative visualization choice for illustrating the change between two data points. Dumbbell charts get their slang name from their appearance, which look similar to weights, and sometimes strands of DNA, when they are in a horizontal orientation.

I personally love that there is an outside-of-the-box chart type that isn't named after a delicious food such as donuts or waffles. You may even say that dumbbell charts are a healthy alternative…

Names aside, this is another chart type that I find to be engaging, effective, and relatively easy to create in Tableau. This tutorial will illustrate how to make a dumbbell chart in Tableau in just a few steps.

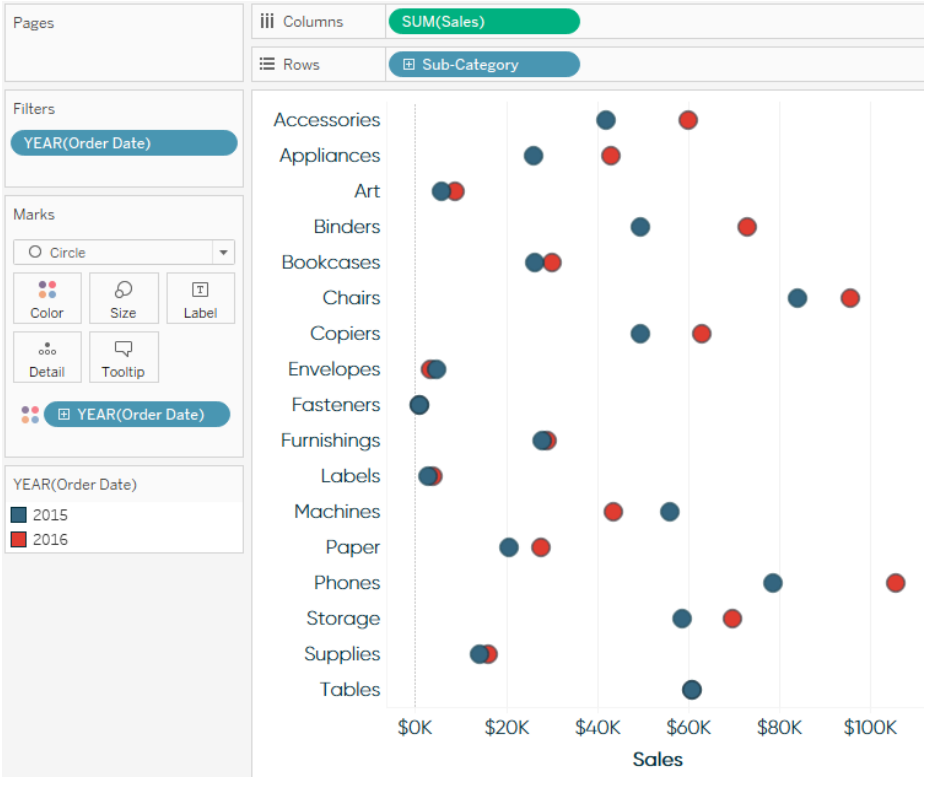
How to Make Tableau Dumbbell Charts

Tableau dumbbell charts are actually dual-axis combination charts, where one of the axes has a mark type of Circle and the other has a mark type of Line. For this tutorial, we will re-create this visualization, which compares the year-over-year sales per subcategory in the Sample - Superstore dataset:



The first step to creating a dumbbell chart in Tableau is to create a dot plot with the measure and dimension you want to visualize. In this example, I will place the Sales measure from the Sample – Superstore dataset onto the Columns Shelf and the Sub-Category dimension onto the Rows Shelf. This creates a sales by sub-category bar chart that can easily be converted to a dot plot by changing the mark type from Automatic to Circle.

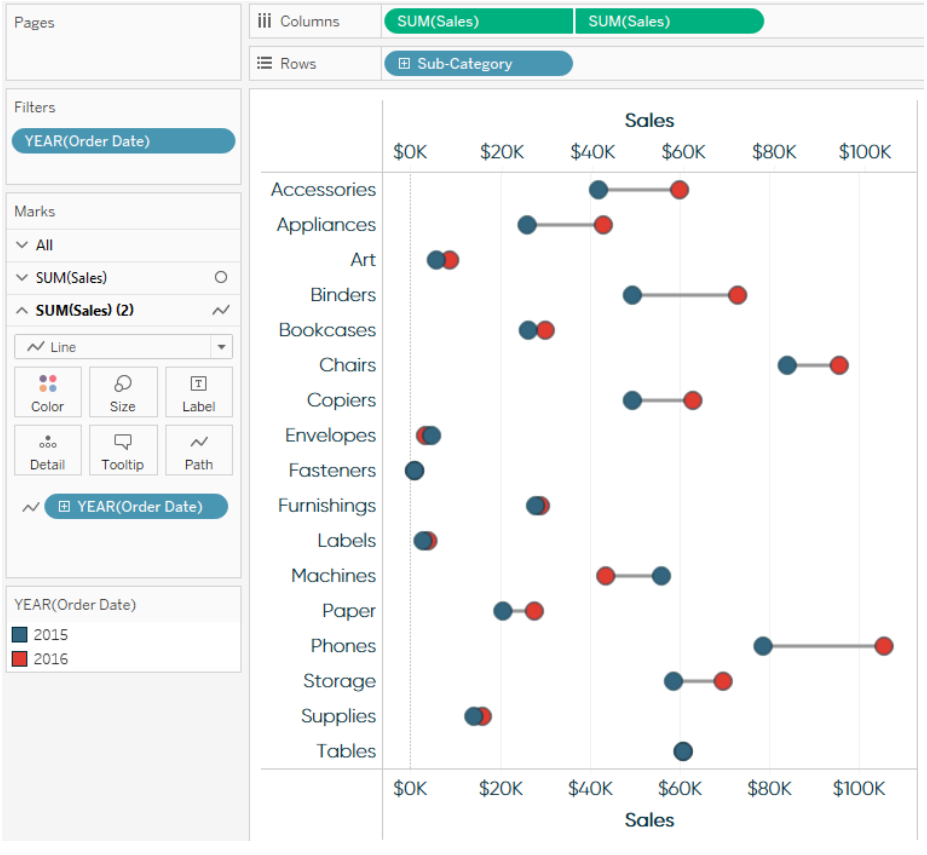
I will also filter the visualization to the last two years in the dataset so that I have only two comparison points per sub-category, and also color the marks by year to distinguish which year is which. At this point, my view looks like this:



Note that this chart type can also be created with a vertical orientation by swapping the location of the fields on the Rows and Columns Shelves, but we will stick with the horizontal orientation for this tutorial.

The second, and actually final, step is to create a second axis with the Sales measure and change its mark type to Line. To turn this dot plot into a dual-axis combination chart, drag the Sales measure near the top of the chart, directly across from the sales axis on the bottom of the chart; when a dashed line appears, drop the measure on the view. Alternatively, you can place a second occurrence of the Sales measure onto the Columns Shelf, right-click the second pill, and choose Dual Axis. Ensure the axes always line up by right-clicking the top axis and choosing Synchronize Axis.

At this point, you have a set of Marks Cards for each occurrence of the Sales measure, and they can be edited independently. Navigate to the second set of Marks Cards for SUM(Sales), change the mark type to Line, then drag the YEAR(Order Date) dimension on the Marks Shelf to the Path Marks Card:



That's all it takes to get to a nice-looking dumbbell chart in Tableau. From here, you can format the size and color of the marks and hide the top axis by right-clicking it and deselecting Show Header.

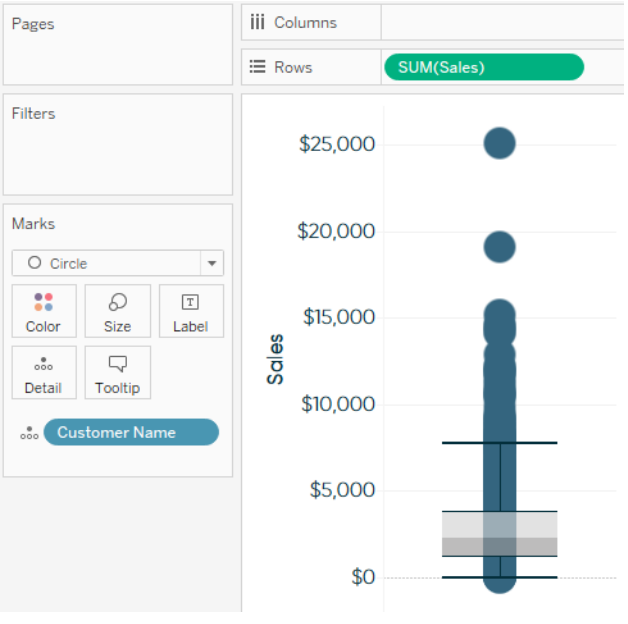
CHAPTER 46

How and Why to Make Customizable Jitter Plots

"Jittering" is a technique for separating overlapping marks on a view. By giving marks some extra room by separating them into different columns, hidden data is often revealed and it is easier to visualize how the data points are distributed. The bad news is that Tableau does not provide an out-of-the-box option to jitter data points. The good news is that Tableau has an amazing community of very smart people who are willing to share their ideas. At least three Tableau Zen Masters that I know of—Steve Wexler, Mark Jackson, and Jeffrey Shaffer—have written about jitter plots.

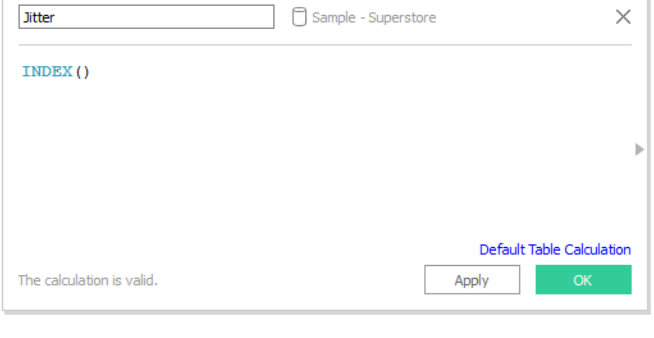
If it's good enough for them, it's good enough for me. This chapter shares my favorite technique for creating jitter plots in Tableau, and also shows you how to put the intensity of the jitter into the hands of your end users.

Consider the following box and whiskers plot, which shows the sales distribution of all of the customers in the Sample – Superstore data source:

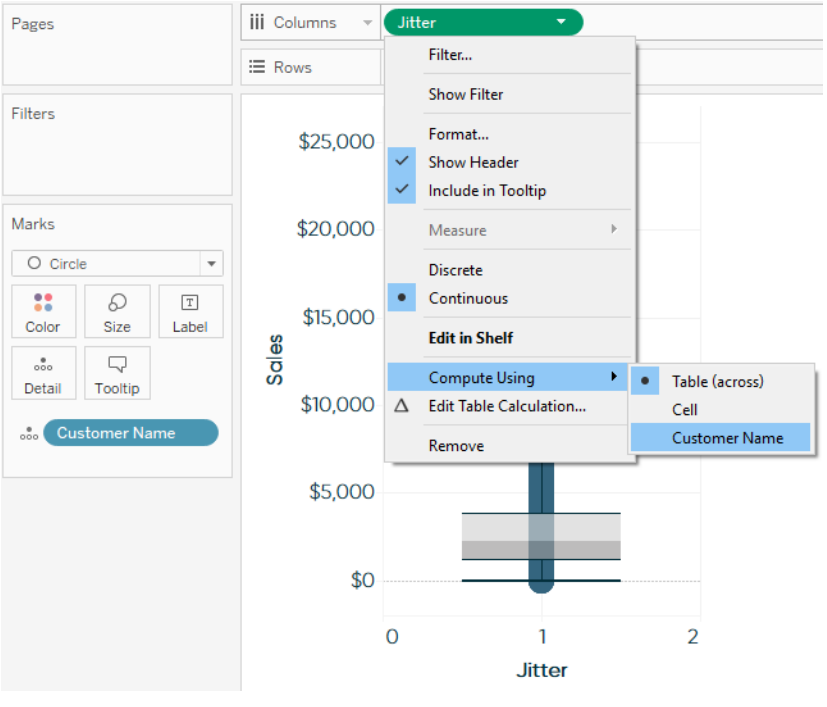


While the box and whiskers themselves are providing some valuable information about the distribution of the customers, it is impossible to get a good sense of how many customers are at each sales value on the y-axis. We are able to see a couple of outliers at the very top, but the view summary in the bottom-left corner of the authoring interface is telling me that there are 793 marks (i.e., customers) on the view. I cannot tell what is happening with the other 791 customers because they are all overlapping on a single column in the chart.

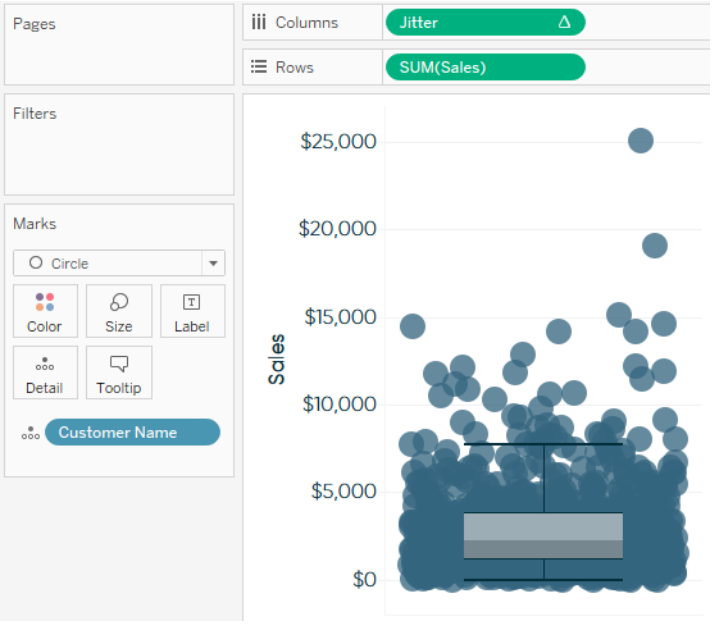
Jitter plots are created using the INDEX() function and changing the addressing to the most granular level of detail—which in our case is Customer Name. To get started with jitter plots, create a calculated field named Jitter that includes only the INDEX() function:



When this newly created calculated field is placed on the Columns Shelf, a new axis will appear, but all of the marks will still be lined up in one column. To apply the jitter, click the calculated field, hover over Compute Using and choose the most granular level of detail—Customer Name:

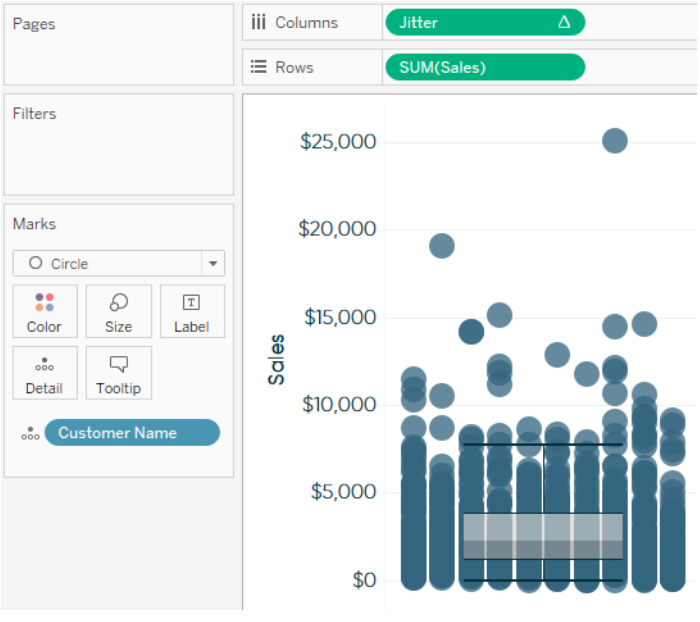


After changing the transparency and size of the marks, hiding the Jitter axis, and bringing the right side of the view in to make the chart skinnier, I am left with this jitter plot:



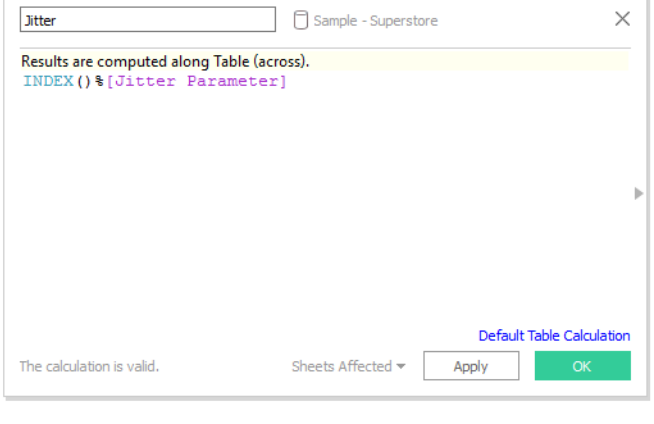
As you can see, it is now much easier to see how our customers are distributed on the y-axis. The horizontal position of each mark type does not mean anything to the analysis, but it separates the marks so they are not all lying on top of each other.

This is a perfectly usable jitter plot, but there may be times when you want to control the intensity of the jitter, or how many columns there are on the x-axis. To hardcode the number of columns used to jitter the marks, simply edit the Jitter calculated field and add %X, where X represents the number of desired columns, immediately after the INDEX() function. Here's how the same analysis from earlier looks when I edit the underlying formula to be "INDEX()%10":

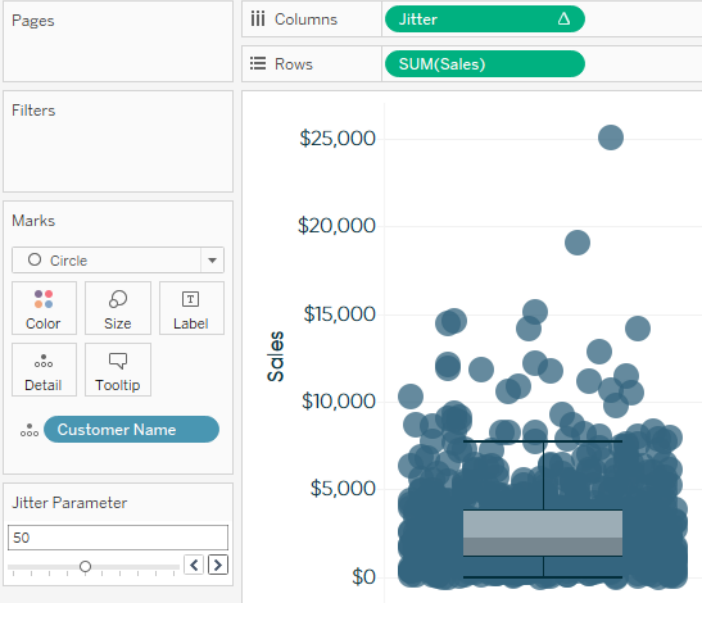


This technique works particularly well if additional dimensions are going to be added to the Columns Shelf for the analysis because all of the columns will have the same jitter intensity specified in the previous step.

Instead of picking just one number to specify the jitter intensity, you can even replace the number in the Jitter calculated field with an integer parameter to allow the end user to quickly change it. Here's how my Jitter calculated field looks after building a parameter and replacing the "10" with the new parameter:



For this example, I set up my parameter to include integers from 10 to 100 with a step size of 10. After showing the parameter control, my end users can change the intensity of the jitter plot, picking any multiple of 10 between 10 and 100. Here's how my final view looks after some formatting to remove the vertical zero line and changing the jitter intensity to 50:



For more on parameters, see Chapter 14.

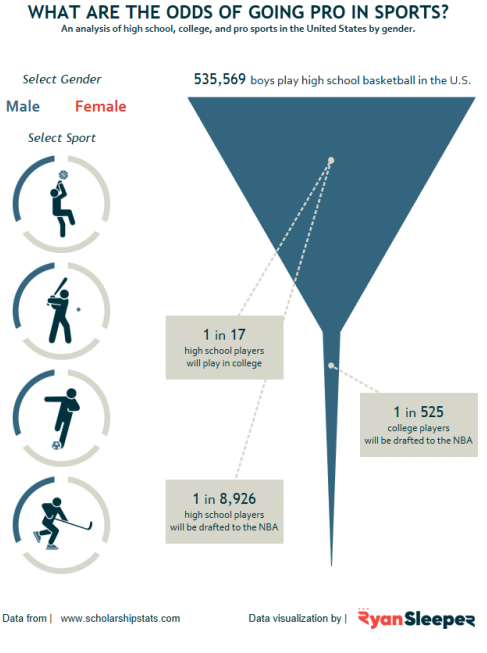
PART III

Tips and Tricks

CHAPTER 47

How to Create Icon-Based Navigation or Filters

Let's take another look at the *Odds of Going Pro* visualization. We discussed how to make funnel charts in Tableau using this as an example back in Chapter 40. Now I will show you how to install custom shapes and use the images to filter and navigate dashboards in Tableau. Here's the Tableau Public dashboard:

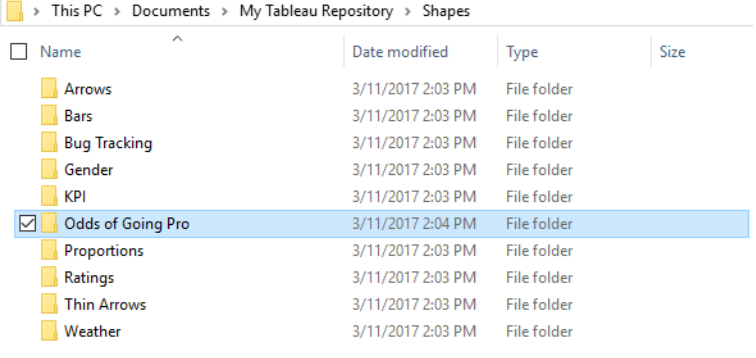


How to Make Icon-Based Navigation/Filters in Tableau

As I mentioned in Chapter 40, I attribute the success of this viz to its simplicity. Not only does it use just one chart to tell the story about the odds of going pro in sports, it uses a simple, intuitive navigation that allows the end user to filter the funnel by gender and sport.

You can easily create a similar navigation by placing the icons you want to use in your Tableau Repository. Every computer with Tableau has a folder called *My Tableau Repository*. On a PC, it's located at *C:Users/[Your User Name]/Documents/My Tableau Repository*. Among other handy Tableau files, the *Shapes* folder in your Tableau Repository holds custom shapes that can be used to create a custom navigation.

I recommend creating a new folder for each unique "shapes palette" that you want to create, and place the corresponding image files there. For example, the preceding viz uses a custom shapes palette I called "Odds of Going Pro" in *My Tableau Repository*. Here is how the folder structure looks behind the scenes on my computer:



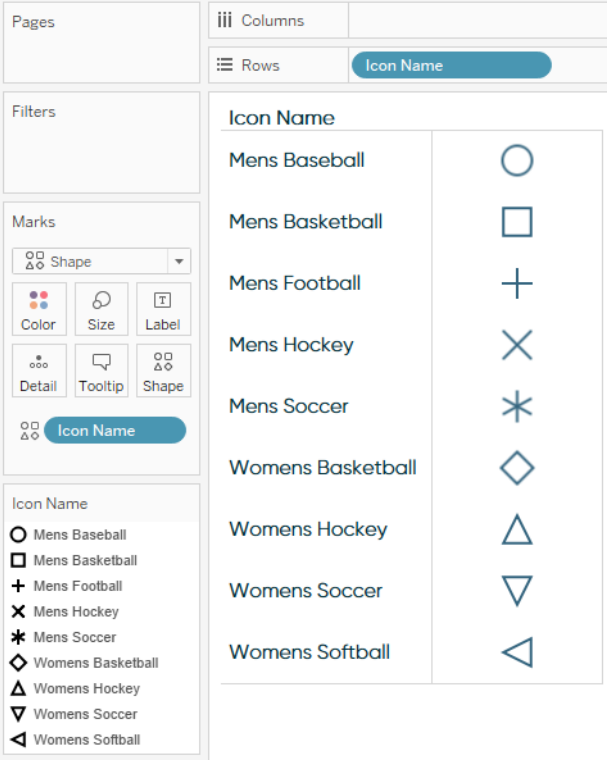
For maximum flexibility with your custom shapes, it is important to use *.png* images with a transparent background. By using shapes with a transparent background, you can use shapes as icons or marks. If you do not have a transparent background, shapes may show up as a colored square instead.

Once the *.png* files are in the *Odds of Going Pro* folder in *My Tableau Repository*, they will be available to use as custom shapes when I reopen Tableau. To create the custom navigation used in the *Odds of Going Pro* viz, follow these steps:

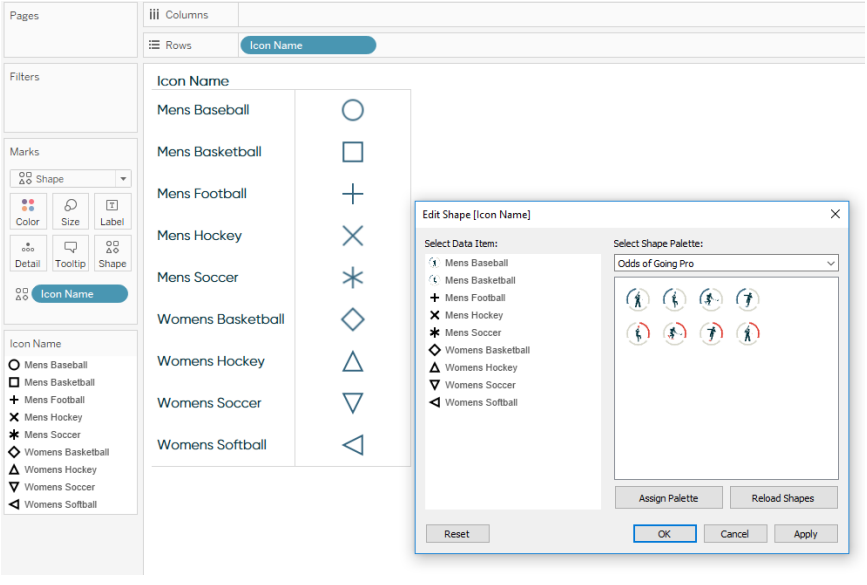
1. On a new sheet, change the mark type from Automatic to Shape.
2. Drag the dimension that contains the names of the buttons you will be using in your navigation to the Shape Marks Card. In my case, the name of the dimension is Icon Name.

3. To adjust the navigation orientation from horizontal to vertical, drag the same dimension on the Shape Marks Card to the Rows Shelf.

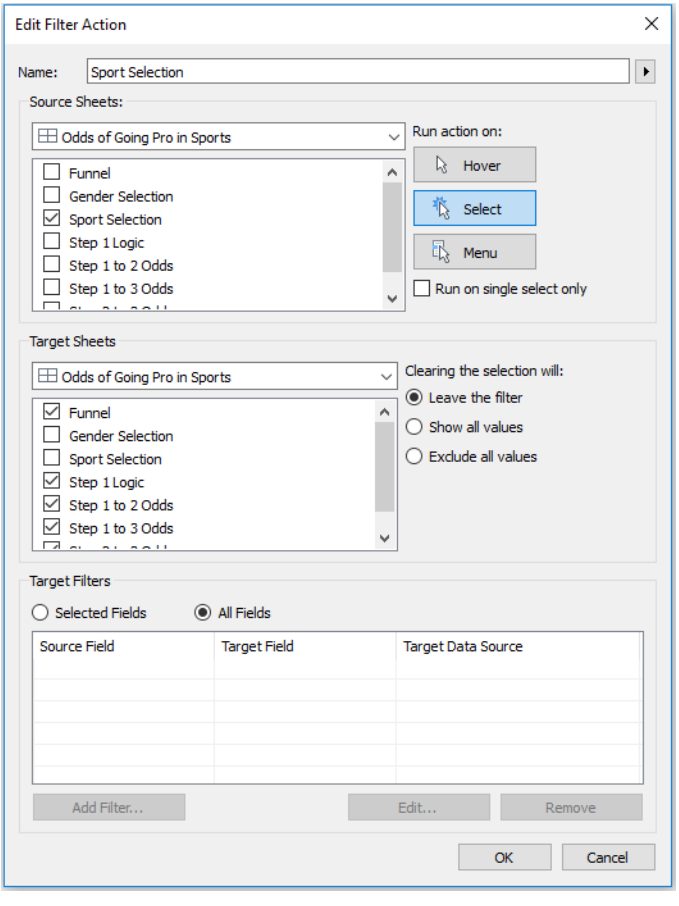
   At this point, your view should look like this:



4. Notice that there is now a shape assigned to each of your button names, but these are the default shapes. To map your custom shapes, hover near the upper-right corner of the shapes legend, click the down arrow, and select Edit Shapes.
5. In the drop-down menu where it says Select Shape Palette, select the custom palette that you added to your Tableau Repository in the preceding steps. Note that you will not see your custom shapes unless you either reopen Tableau after saving your custom shapes in your Tableau Repository or select the option to Reload Shapes in the Edit Shape interface.
6. Map the correct shapes with their corresponding icon names by clicking each item name individually, then clicking the custom shape that is most appropriate:



7. The last step in making your icon-based navigation functional is to add the sheet to your dashboard view, and add a dashboard action that will filter the rest of your dashboard based on the button that is clicked. From within your dashboard view, you can accomplish this by following these four steps:
   a. Navigate to Dashboard → Actions.
   b. Click Add Action.
   c. Choose a "Filter" based action.
   d. Set up logic that basically says "If you click my navigation sheet, I want the dashboard to filter every other sheet." This is how the logic looks on my dashboard:



You now have a fully functional and aesthetically pleasing icon-based filter in your dashboard. Be creative and use the combination of these icon-based filters and dashboard actions to filter the view, highlight insights in the data, navigate between worksheets or workbooks, and even open embedded web pages and videos (discussed in Chapter 56).

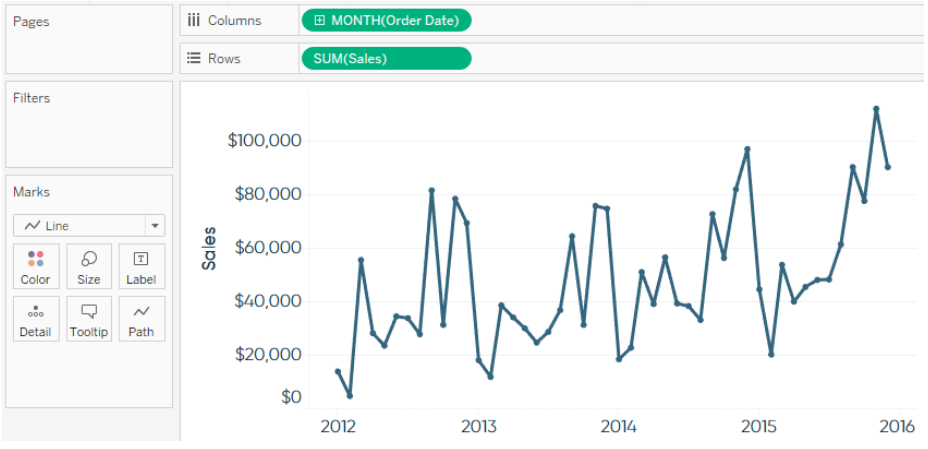
CHAPTER 48

How to Make a What-If Analysis Using Parameters

Parameters are one of the most powerful tools available in Tableau for exploring your data and providing interactivity to your end users. As discussed in Chapter 14, the best way to think of parameters is that they are user-defined values that are not tied to a specific dataset. In the simple formula 2 multiplied by X = 16, the value of X would be the parameter; in this case, equal to 8. The reason parameters are so powerful is that you or your end users can change the value of X from 8 to any number, which will change the results of a view.

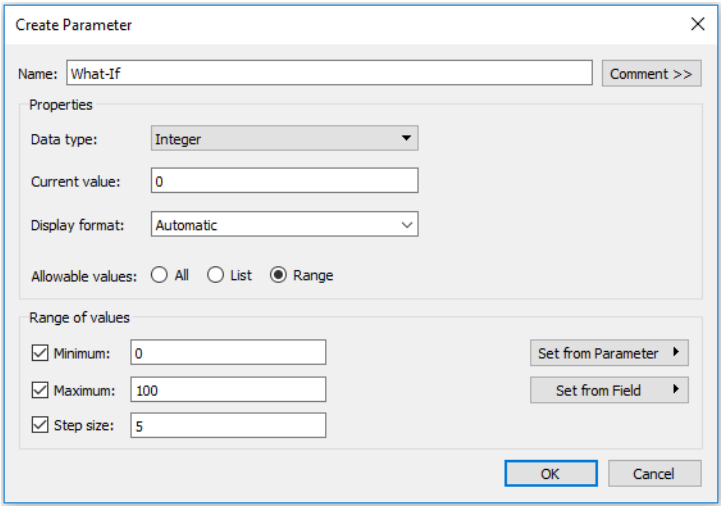
In this chapter, we will use this concept to create a what-if analysis that will show us what would happen if we improved our sales from 0%–100%. Parameters come in many different forms, but for this chapter, we will be creating a parameter from *integers*, or whole numbers.

Let's start by creating a simple line graph showing sales over time:

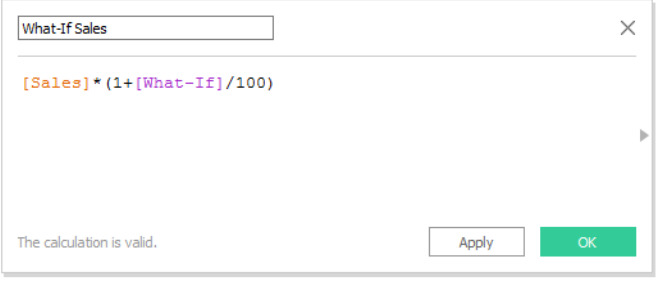


To create a parameter, right-click any white space in the Data pane and select Create Parameter. This is where you can choose from six different data types. We only want the ability to select nonfractional numbers, so choose Integer as the data type. There are some additional options available, including the current value (which will be the default value the first time you use the parameter), allowable values, and within "Range," minimum value, maximum value, and step size.

For the purpose of this what-if analysis, we will set the current value to zero, which will end up not changing our view at all the first time we use the parameter. We will also choose Range and set the range from zero to 100 with a step size of five. This means that the end user can select any number between zero and 100 in increments of five:

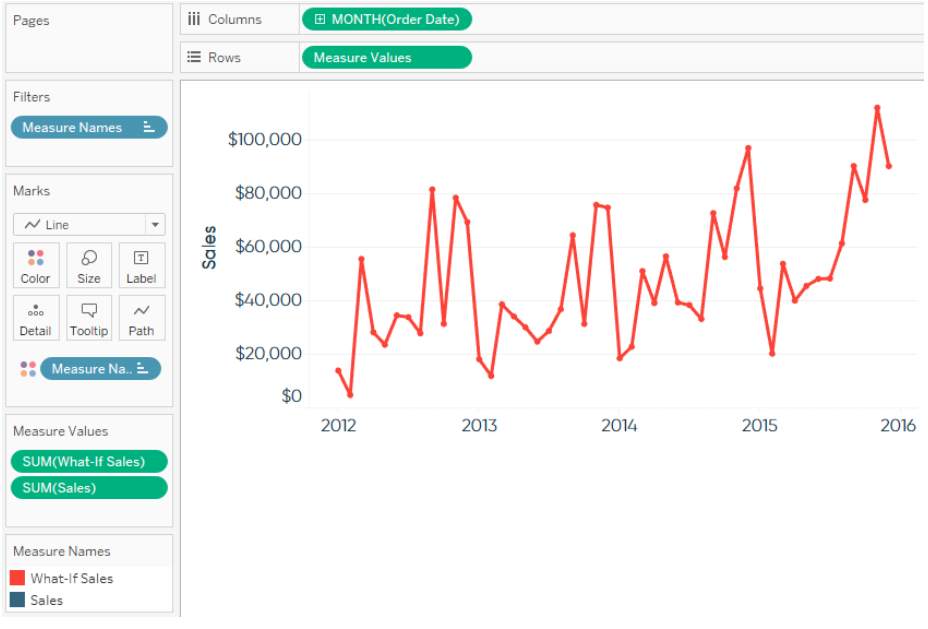


Parameters do little on their own because they are independent values that are not generated from the data, but controlled by the end user. To make the parameter useful, we will integrate the parameter's value in a calculated field that multiplies its value by something else. We are interested in creating a what-if analysis that shows how our sales would change if they improved by 0%–100%. There are a couple of ways to write this calculation; here is one:

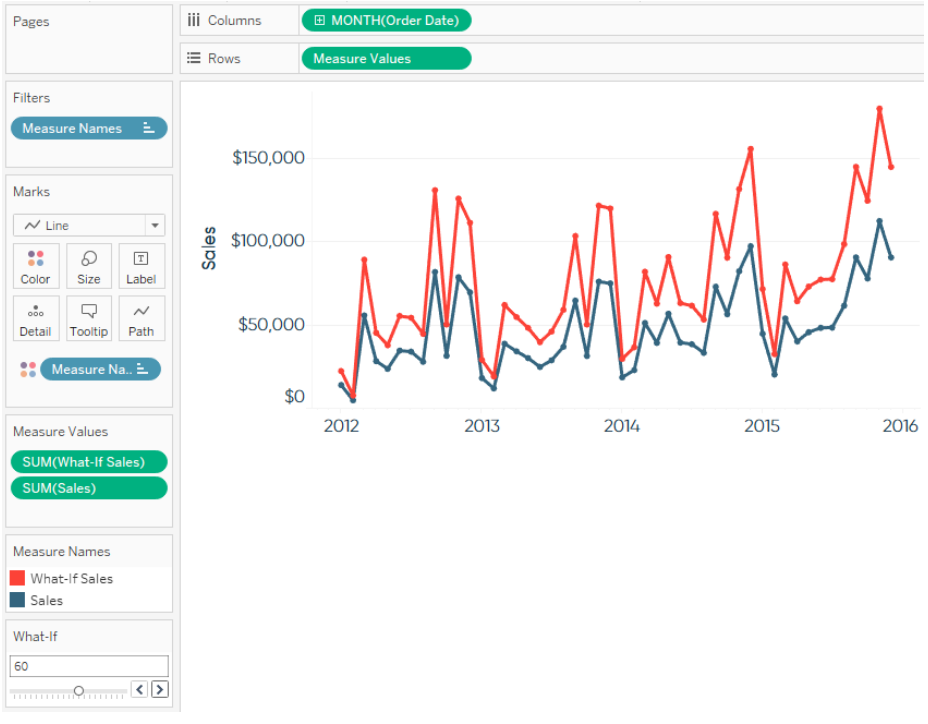


Notice that parameters are colored purple and can be used in a formula just like any other field in your calculation.

Now that we have our parameter in a calculated field, we can use it in a view. To create two lines on the same axis, one for actual sales and one for what-if sales, we will drag our newly created calculated field onto the existing axis for sales. For this view, I have changed the colors of the measure names and placed the line for What-If Sales on top of actual sales by dragging What-If Sales up in the Measure Names color legend. At this point, your view should look like this:



There is one last step needed in order to utilize your parameter. Right-click the What-If parameter and select Show Parameter Control. You now have a way to control the input for the What-If Sales Calculated Field. Notice that the inputs available are the ones that we set up when we first created the parameter; a range of 0–100 in increments of five:



For additional ways to provide interactivity to your end users, see storytelling tip 11 in Chapter 96.

CHAPTER 49

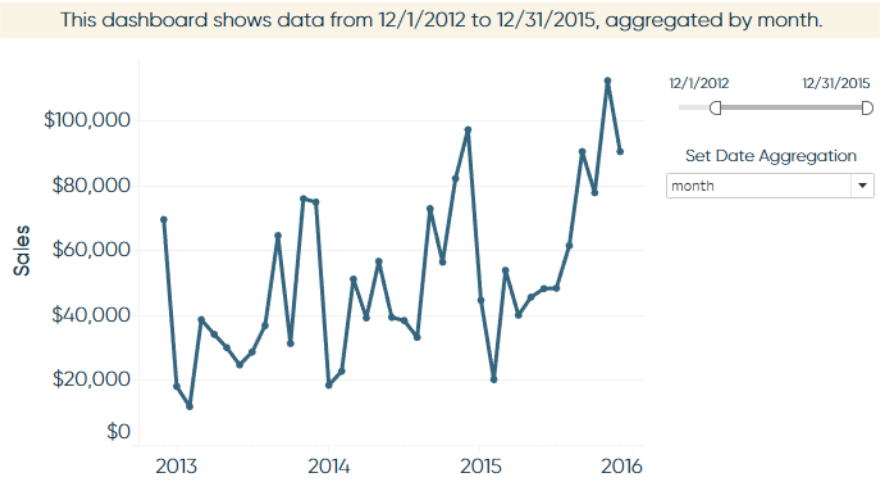
Three Ways to Add Alerts to Your Dashboards

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.

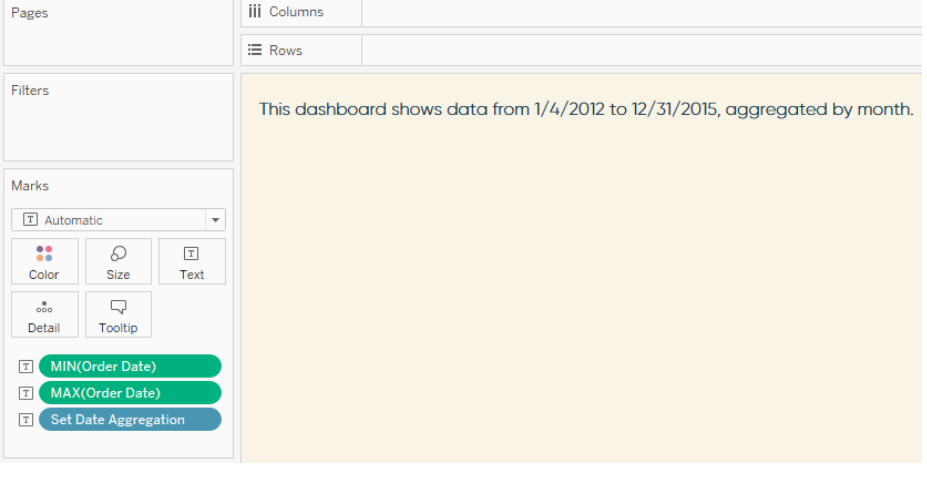
Alert 1: Date Settings

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: (a) it clearly communicates how current the dataset is, and (b) helps avoid confusion in the case that the dataset has not updated or a view doesn't look as expected due to the aggregation. Here is an example using the Sample – Superstore data:



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:

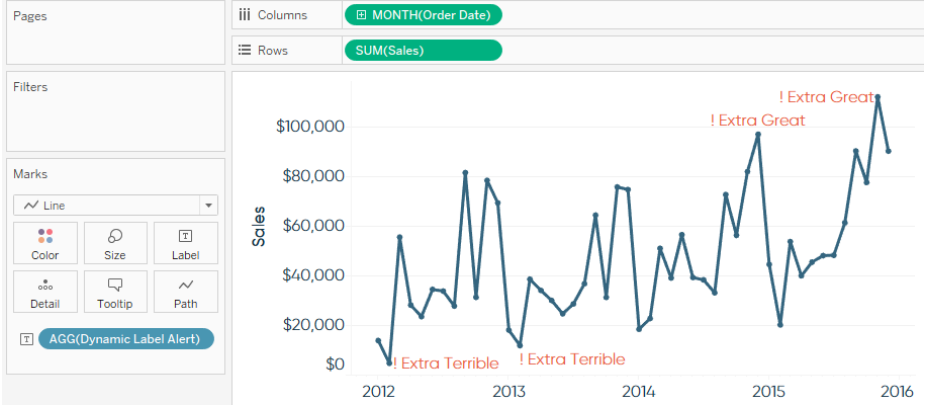


Finally, ensure the Order Date filter on your line graph is applied to this text sheet. This way, when an end user changes the date range, the MIN and MAX dates will change to match the filter.

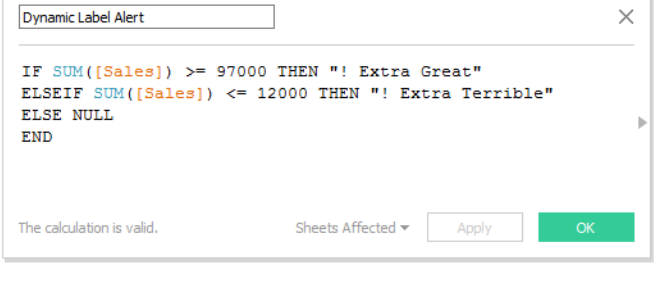
For more, see Chapter 66.

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 before, 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.

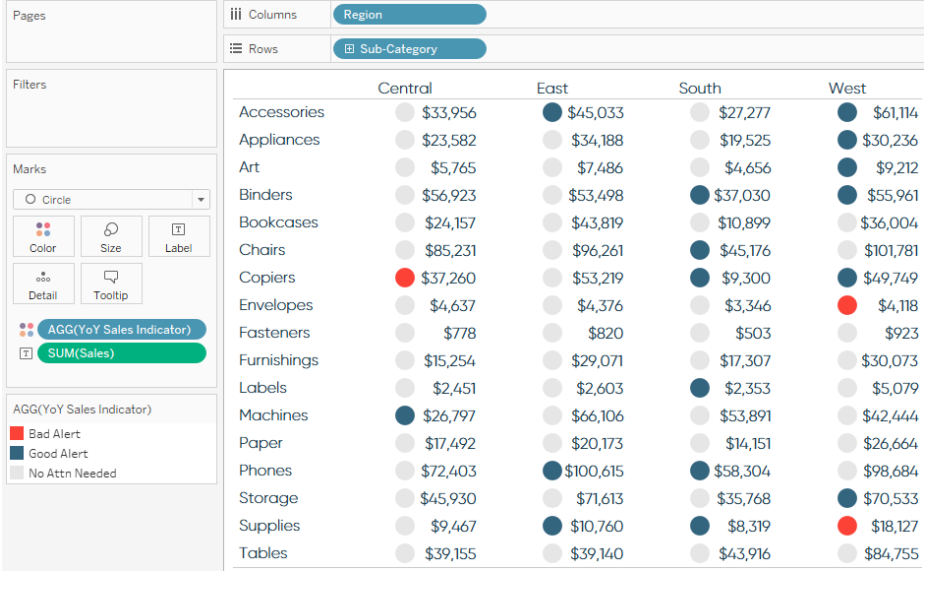
Alert 3: Heat Map Dashboard with Optional Tableau Server Email

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 this:



	Central	East	South	West
Accessories	$33,956	$45,033	$27,277	$61,114
Appliances	$23,582	$34,188	$19,525	$30,236
Art	$5,765	$7,486	$4,656	$9,212
Binders	$56,923	$53,498	$37,030	$55,961
Bookcases	$24,157	$43,819	$10,899	$36,004
Chairs	$85,231	$96,261	$45,176	$101,781
Copiers	$37,260	$53,219	$9,300	$49,749
Envelopes	$4,637	$4,376	$3,346	$4,118
Fasteners	$778	$820	$503	$923
Furnishings	$15,254	$29,071	$17,307	$30,073
Labels	$2,451	$2,603	$2,353	$5,079
Machines	$26,797	$66,106	$53,891	$42,444
Paper	$17,492	$20,173	$14,151	$26,664
Phones	$72,403	$100,615	$58,304	$98,684
Storage	$45,930	$71,613	$35,768	$70,533
Supplies	$9,467	$10,760	$8,319	$18,127
Tables	$39,155	$39,140	$43,916	$84,755

A high-level dashboard like this can be subscribed to in Tableau Server, 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 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 50

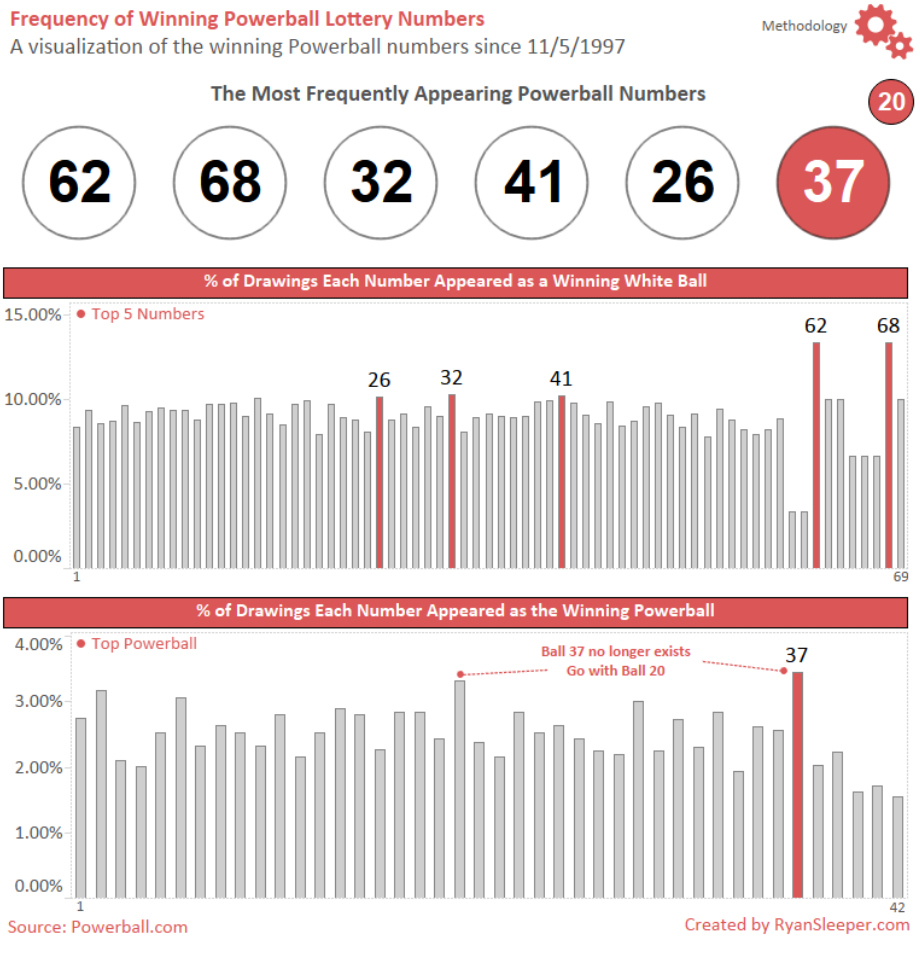
How to Add Instructions or Methodology Using Custom Shape Palettes

Instructions on a Tableau dashboard are vital to ensuring your end users can get the most out of your work whether or not you are there to explain it to them. As with most things in Tableau, there are several different approaches to adding instructions to a dashboard, and there is not necessarily one best method. Just to name a few off the top of my head, you could:

- Add a text box to your dashboard and type out the full instructions.
- Include instructions on a second tab in your workbook.
- Not include written instructions, but display color, size, and mark legends for every dashboard object.

My go-to approach for adding instructions to the corporate dashboards I create in Tableau is to leverage a custom shapes palette to display a familiar icon to my end users. When they see the icon, they know they can hover over it for more information, which is displayed to them as a tooltip.

Here is an example where I used this approach to share the methodology for one of my Tableau Public visualizations:



This is my favorite approach to including instructions because it ensures the end user has access to the instructions without requiring much real estate on my dashboards (I have cooler things to show, after all). My end users have learned that the gear icon is their go-to place for instructions, but they will not necessarily need to be reminded how to use the dashboard after they've used it a few times.

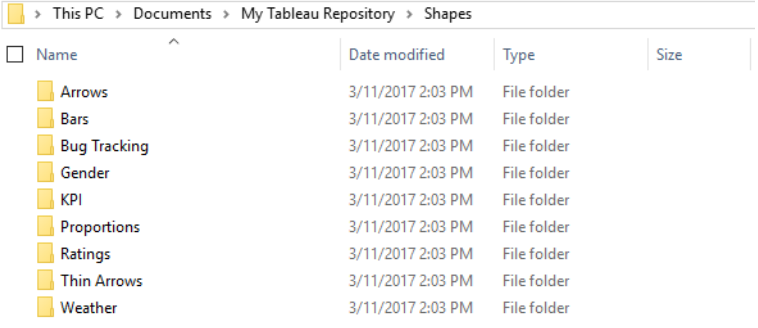
In this chapter, I will walk you through how to add a similar gears icon leveraging a custom shapes palette in Tableau. I will also show you how to add instructions to this custom shape using a tooltip:

1. Create a custom shapes palette with your instructions icon.

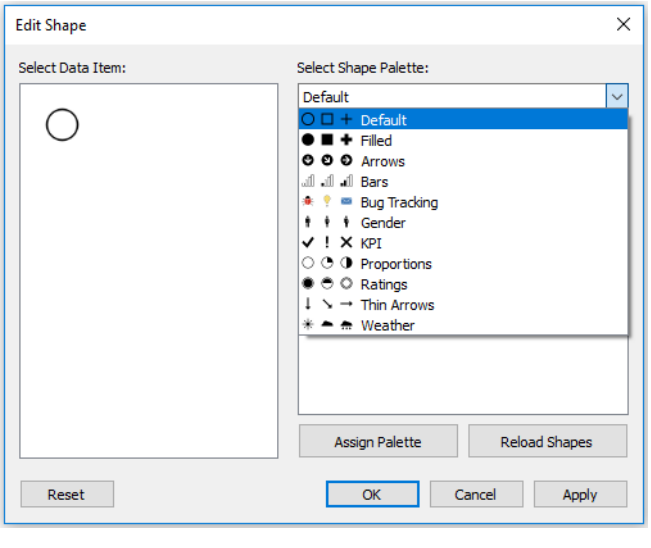
   As mentioned in Chapter 47, every machine with Tableau has a *Shapes* folder in their Tableau Repository. On a PC, it is located at *C:\Users\[User Name]\Docu-*

ments\My Tableau Repository\Shapes. Every folder listed in this location is associated with a shapes palette in Tableau.

Here is how my *Shapes* folder looks on my computer:



Here is how my shape palettes look in Tableau:

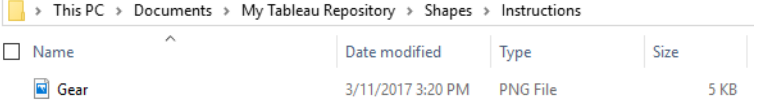


Notice every folder in my shapes directory has a corresponding palette in Tableau. By simply creating new subfolders in your shapes directory and adding images to that folder, you create custom shape palettes that can be used in Tableau.

Let's try to do this by creating a new folder called *Instructions*, and placing an instructions icon in that folder. You can use whatever image you would like, but I used the following *.png* image:

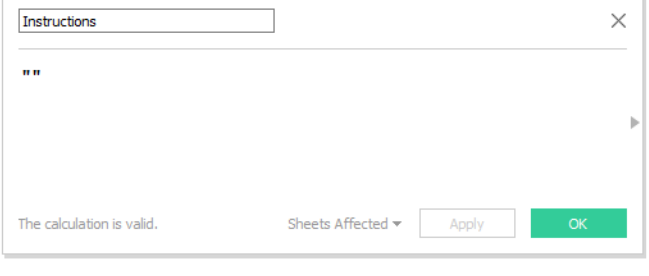
By using *.png* images with transparent backgrounds, you will have the most flexibility possible when using your custom shapes in Tableau. For example, the transparency will allow you to color the image while leaving the background white.

At this point, your custom shapes palette should look like this:



2. Create a "dummy" dimension

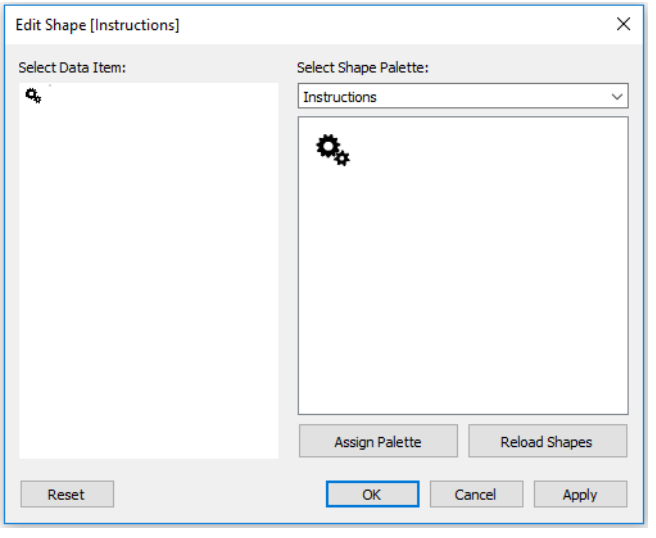
   Creating a "dummy" dimension will provide a field that will act as your instructions. To do this, create a calculated field that looks something like this:



3. Create a sheet with your instructions icon.

   On a new blank worksheet, change the mark type from Automatic to Shape. This will reveal a new Marks Card for Shape. Drag your Instructions dimension to the Shape Marks Card. You can now choose which shape you would like to assign to your instructions by clicking into the shapes palette. If you do not see your new custom shapes palette as an option, choose Reload Shapes in the lower-right corner of the Edit Shape dialog box.

   Map your custom instructions icon to the circle by first clicking the circle shape, then clicking your new icon. At this point, your instructions shape should look like this:



You can also change the color of your instructions icon by clicking into the Color Marks Card. I personally like to drag my dummy dimension to the Color Marks Card, which also allows me to choose from some custom colors that I have set up.

Finally, you can customize what your instructions say by clicking into the Tooltip Marks Card and typing out your instructions. This is what will show up when your end user hovers over the instructions icon. This acts as a full word processor, so you can build in different fonts, colors, and shapes to make your instructions as user-friendly as possible. Get creative and tailor your instructions for your end users.

4. Add the instructions to a dashboard.

   The fourth and final step in adding instructions to your dashboard is to drag the sheet you created in step 3 to a dashboard. I typically make this dashboard object floating and set the dimensions to 50 high by 50 wide. I also clear the borders and remove any sheet legends, such as the shapes legend that will appear when you place the sheet on your dashboard.

There you have it—an elegant way to add customized instructions to your dashboards without using much valuable real estate. As a bonus to this approach, as of Tableau 9.0, tooltips are persistent. This means that end users can hover over your instructions as long as they want without the tooltip disappearing; this alleviates the one drawback to this technique before that version of Tableau was released.

CHAPTER 51

Ten Tableau Data Visualization Tips I Learned from Google Analytics

Eight years ago, I was fortunate to be introduced to Tableau, a tool for data visualization that has led to personal opportunities and contributions to my clients' businesses that I didn't imagine would be possible in such a short amount of time.

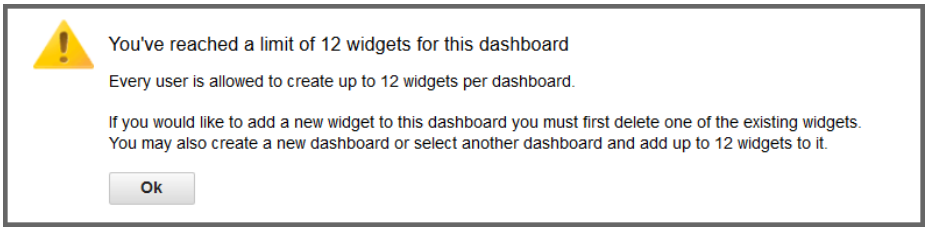
Even before that, though, I was introduced to Google Analytics.

It's no surprise then that some of my inspiration for the design, usability, and analytics of my data visualizations have been drawn from Google Analytics over the years. Sure, the Audience Overview report features an overly large pie chart for new versus returning visitors—a measure that is largely useless in an age of multiple devices, cleared cookies, and anonymous users—but Google tends to get a lot of things right.

What follows is a list of 10 Tableau data visualization best practices I use that I either consciously or subconsciously picked up from Google Analytics. In no particular order…

Use a Maximum of 12 Dashboard Objects

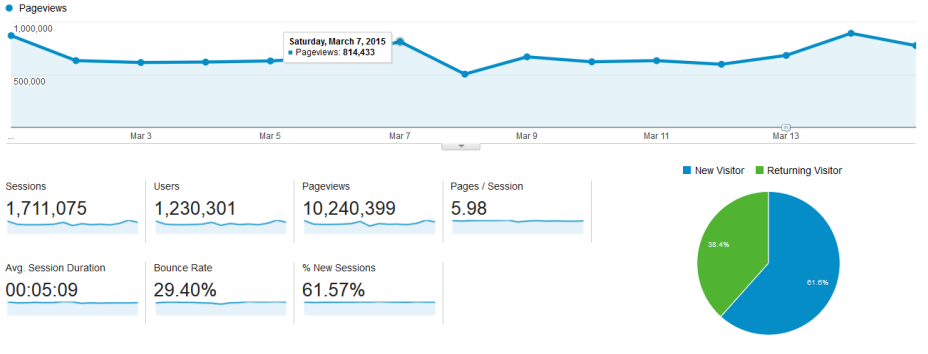
One of the first ideas I remember consciously thinking to myself, "Wow, that's a good idea, Google Analytics; I should use that…," was the 12 object limit Google Analytics has in place for its custom dashboards. Many data visualization specialists often remind us that "less is more," and I have encouraged designers in Chapter 89. In Google Analytics, you have no choice but to keep dashboard objects, or widgets in Google's words, to 12 or fewer. Here is what you see when trying to add a 13th widget in a Google Analytics custom dashboard:



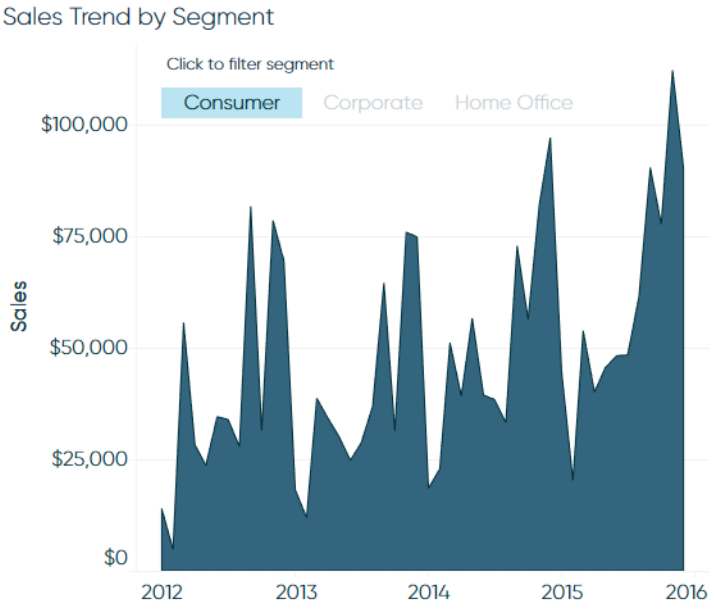
This limit forces designers to prioritize which KPIs and views are truly pertinent to the story.

Improve User Experience by Leveraging Dashboard Actions

On Google's Audience Overview report, which is my favorite report to start my analyses, seven main KPIs are shown along with sparkline trends. By clicking a sparkline, the larger line graph at the top will change to that KPI. This makes it easier to see trends and find extra detail, which is provided if you hover over data points:



Similar usability can be set up in Tableau through the use of dashboard actions. Here is a simple example using a control sheet for Segment and a sales trend over time. By adding both sheets to a dashboard and choosing to use the sheet containing the three different segments as a filter, clicking a segment will filter the trend being shown:



To use a sheet as a filter, click the sheet, click the down arrow that appears in the upper-right corner, and click "Use as Filter."

Allow End Users to Change the Date Aggregation of Line Graphs

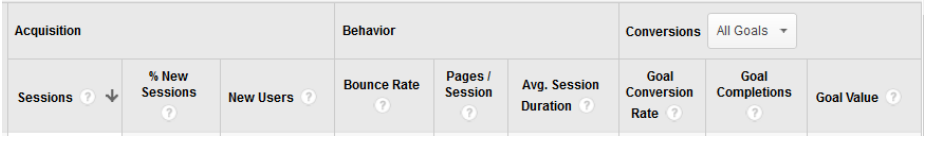
Another feature I like about the Audience Overview report in Google Analytics is the ability to change the date aggregation of the trend being shown. By default, the trends are aggregated by Day, but users also have the option to change the granularity to Hourly, Week, or Month:

This feature can help find different stories in the data based on the date part being displayed. To learn how to replicate this in Tableau, see Chapter 66.

Keep Crosstab Widths to a Maximum of Ten Columns

Tableau allows you to make crosstab views with up to 16 columns. If you ask me, this is about 11 too many. Spreadsheets are not data visualizations and they take far too long for analysts to process. Google Analytics is at least closer, never displaying more

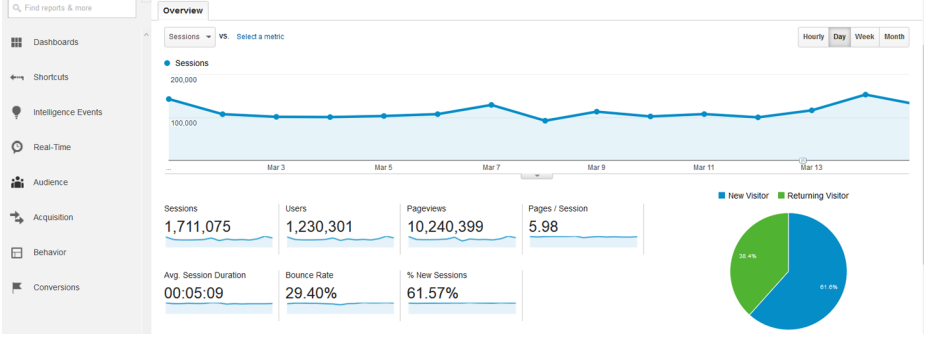
than 10 columns including the dimension in the first column (nine measures pictured here):



Sometimes it makes sense to use crosstabs in small doses, and I am realistic that some audiences will require more detailed data in a text table. If at all possible, I would stick with this implied limit from Google Analytics of ten or fewer columns.

Use a Vertical Navigation in the Left Column

I've always appreciated the intuitive, left navigation of Google Analytics. For reports that have a lot of changeable components, such as filters, the vertical space on the left side of a dashboard is an ideal place for navigation. Placing controls here serves the dual purpose of prioritizing what changes can be made to find stories in the data and having a clear place to remind users what filters have been applied:



This is not a must-use tip for every situation, but makes a lot of sense for dashboards that are meant to provide self-service analytics. If you have dashboards that are not meant to be changed very often, deprioritize filters and buttons by moving them to the right navigation.

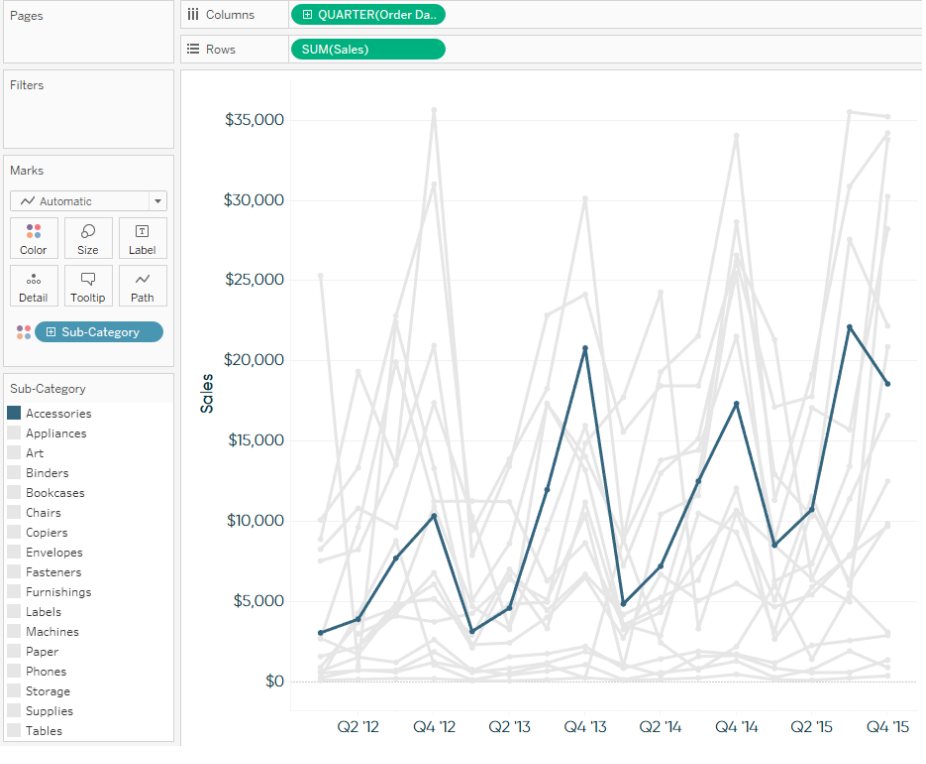
Choose Five or Fewer Colors for Your Dashboards

At times, the number of dimension members within a dimension on a view will dictate how many colors are required in a data visualization. For example, I may have a unique color for each region: North, East, South, and West. When possible, I try to

limit the number of colors to five or fewer—an unspoken rule I picked up from Google Analytics' simple color choices.

Google Analytics does have a set of visuals called "Motion Charts" that contain more than five unique colors, but outside of that, it is challenging to find visuals with more than two different colors. Using five or fewer colors reduces the burden on the end users because they do not have to work as hard to look back and forth to the color legend to determine what the marks on a view represent.

If you are in a situation where more than five categories need distinct colors, consider a two-color palette—making the category of interest a color that stands out, and everything else a more muted color, such as gray. See Chapter 88 for more on getting the most from color in data visualization.



Stick Mostly to Lines and Bars

With only a few exceptions, Google Analytics' powerful analytics platform is built almost entirely on what I consider the most effective data visualizations: lines and bars. Google Analytics has treemaps in its Acquisition reports and highlight tables in

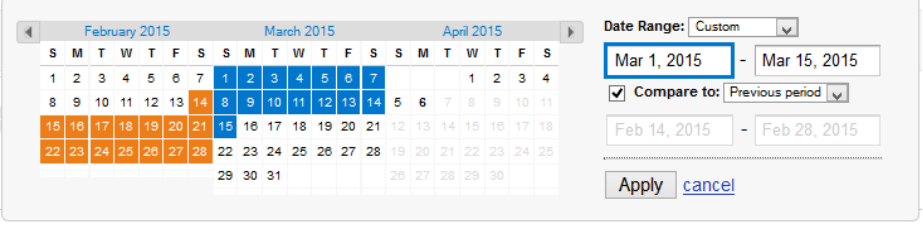
its Cohort Analysis. It also uses scatter plots in the aforementioned Motion Charts and I'm sad to say, pie charts make several appearances.

Even with those other chart types, lines and bars dominate the dozens of available reports in Google Analytics, and for good reason. These simple chart types have been getting the job done for more than 200 years now, and in most cases, continue to be the best starting point for sharing trends and comparisons in data visualization.

Include Comparisons Such as Year Over Year

When I develop data visualizations, I am constantly trying to avoid the dreaded question, "So what?" If I hear this question, it means I did not do a good enough job communicating the stories in the data. One way to avoid this awkward conversation about why you've spent days developing a data visualization that doesn't tell anybody anything… is to use comparisons.

Some of my favorite approaches for showing comparisons are bar charts, small multiples, and period-over-period comparisons. The latter of which can be done very easily in Google Analytics by selecting a date range comparison when you set the date range:

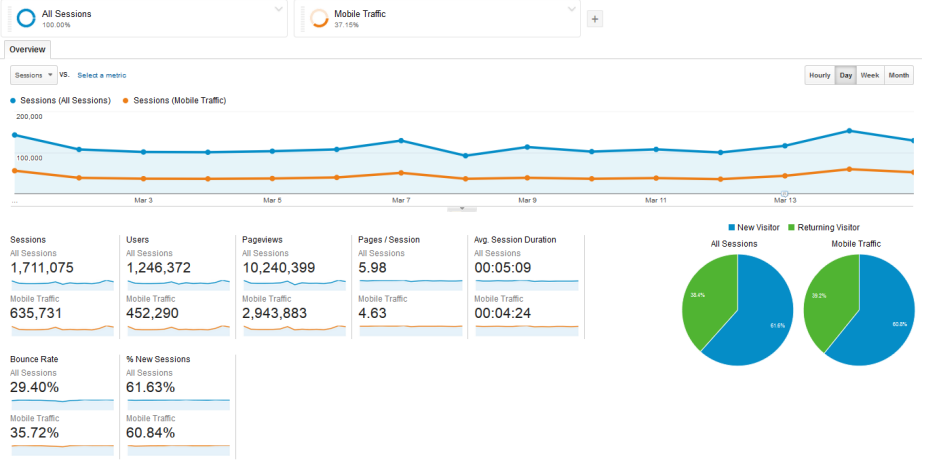


Google Analytics allows you to compare your selected dates to a custom secondary range, the previous period of the same range (i.e., 15 days versus 15 days), or to that same date range versus the prior year. Similar functionality can be set up in Tableau using parameters (see Chapter 69).

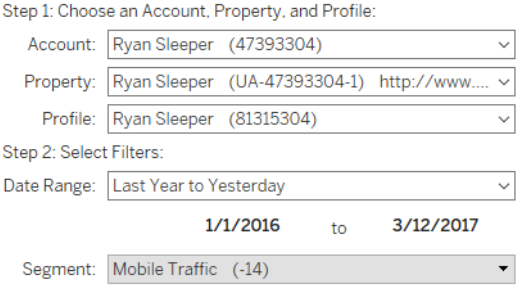
Bring Your Data Visualization to Life Using Segmentation

Another way I like to provide comparisons in my data visualizations is through the use of segmentation. Segmentation is one of the most powerful tactics for adding value to your analyses, and makes it easy to quickly identify your weakest and strongest performers. Segmentation allows you to hone in on your weakest performers to figure out what characteristics to avoid, or your strongest performers to determine what you should be doing more of.

In Google Analytics, several segments including Mobile Traffic, Referral Traffic, and Returning Users (just to name a few), come ready to use "out-of-the-box." You can also set up customized advanced segments to compare users almost any way you can imagine. These standard and custom segments can be used to compare performance to each other, or the site as a whole:



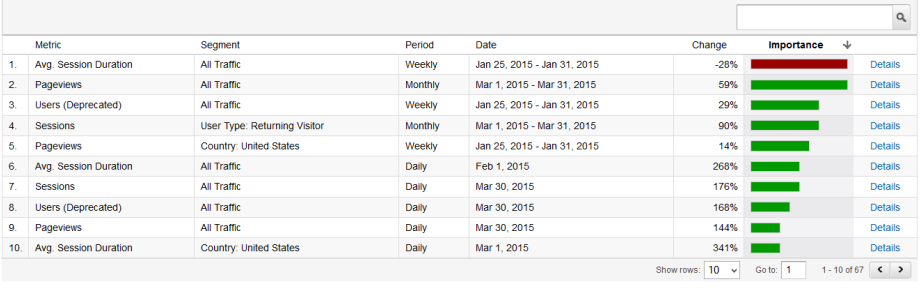
There are a couple of ways to leverage segmentation in Tableau. First, if you are using the Google Analytics connector, you can filter the data coming into Tableau on a specific segment. Note the Segment filter here in the connection setup:



The drawback to this is you can only segment the data to one thing at a time. A much more powerful use of segmentation in Tableau is to create and compare your own segments once the data is already being used in Tableau. For more on customizing your segmentation, see Chapter 53.

Include Alerts of Exceptional or Poor Performance

Last but not least, Google Analytics taught me the power of boiling the most important stories in the data to the top. Google Analytics does this in the form of Intelligence Events, which shows any performance on a site for a particular date range that fell outside of its algorithms' expectations:



	Metric	Segment	Period	Date	Change	Importance	
1.	Avg. Session Duration	All Traffic	Weekly	Jan 25, 2015 - Jan 31, 2015	-28%		Details
2.	Pageviews	All Traffic	Monthly	Mar 1, 2015 - Mar 31, 2015	59%		Details
3.	Users (Deprecated)	All Traffic	Weekly	Jan 25, 2015 - Jan 31, 2015	29%		Details
4.	Sessions	User Type: Returning Visitor	Monthly	Mar 1, 2015 - Mar 31, 2015	90%		Details
5.	Pageviews	Country: United States	Weekly	Jan 25, 2015 - Jan 31, 2015	14%		Details
6.	Avg. Session Duration	All Traffic	Daily	Feb 1, 2015	268%		Details
7.	Sessions	All Traffic	Daily	Mar 30, 2015	176%		Details
8.	Users (Deprecated)	All Traffic	Daily	Mar 30, 2015	168%		Details
9.	Pageviews	All Traffic	Daily	Mar 30, 2015	144%		Details
10.	Avg. Session Duration	Country: United States	Daily	Mar 1, 2015	341%		Details

Show rows: 10 Go to: 1 1 - 10 of 67

Intelligence Events are a great place to start an analysis and I love them for a couple of reasons:

- They do the heavy-lifting for you; making outliers apparent without making you dig to find them.
- They are easily understood by analysts and non-analysts alike.

In Tableau, similar alerts can be built into your dashboards to make them more user-friendly and reduce time to insight. See Chapter 49 to get you started.

Google has taught me a lot over the years and these ten data visualization best practices from Google Analytics have served me well. One of the things I like most about Tableau is the software's flexibility, and I encourage you to take the best ideas from other tools and incorporate—or even build on—them in your own dashboards.

CHAPTER 52

Three Alternative Approaches to Pie Charts in Tableau

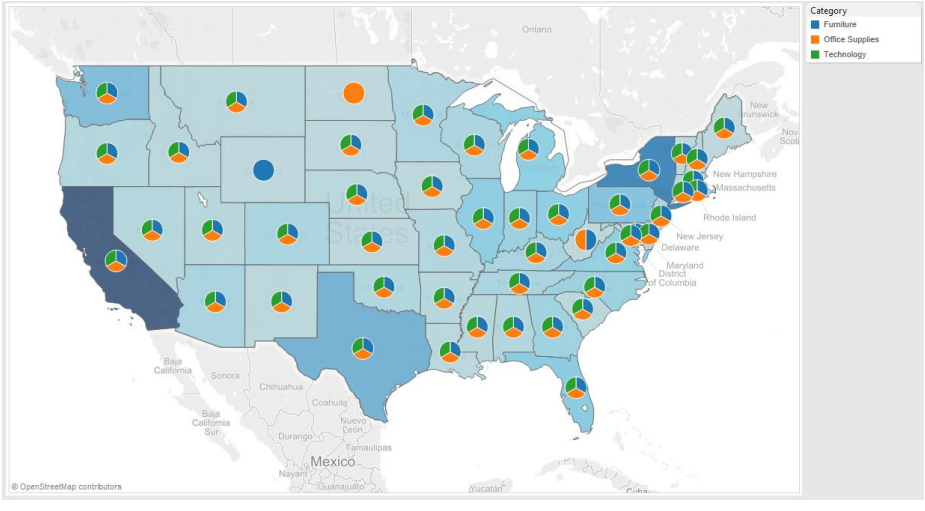
Despite being one of the least effective means of communicating data, I often see pie charts in corporate dashboards and Tableau Public visualizations. New users likely see pie charts as an easy way to spruce up their dashboards, but they are doing themselves a disservice because pie charts *increase time to insight*—the opposite of what we are trying to achieve with data visualization.

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?" For this reason, I want to document a better approach 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 its 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. In this chapter, I will share a couple alternatives to pie charts while building up to my recommended approach to visualizing a parts of a whole relationship.

Tableau Pie Chart Alternative #1: 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. More importantly, *pie charts should never be used in a time-series analysis*. For this reason, my first pie chart alternative is to simply use a bar chart, which is a great choice for comparing values at one point in time.

One thing you lose in this approach compared to the map we just looked at is the quick comparisons between states. However, you can use dashboard actions to achieve something similar by having the bar chart update when a state is clicked or hovered over. I like this approach because you gain real estate to include the actual sales numbers and/or the percent of the whole that each category is contributing:

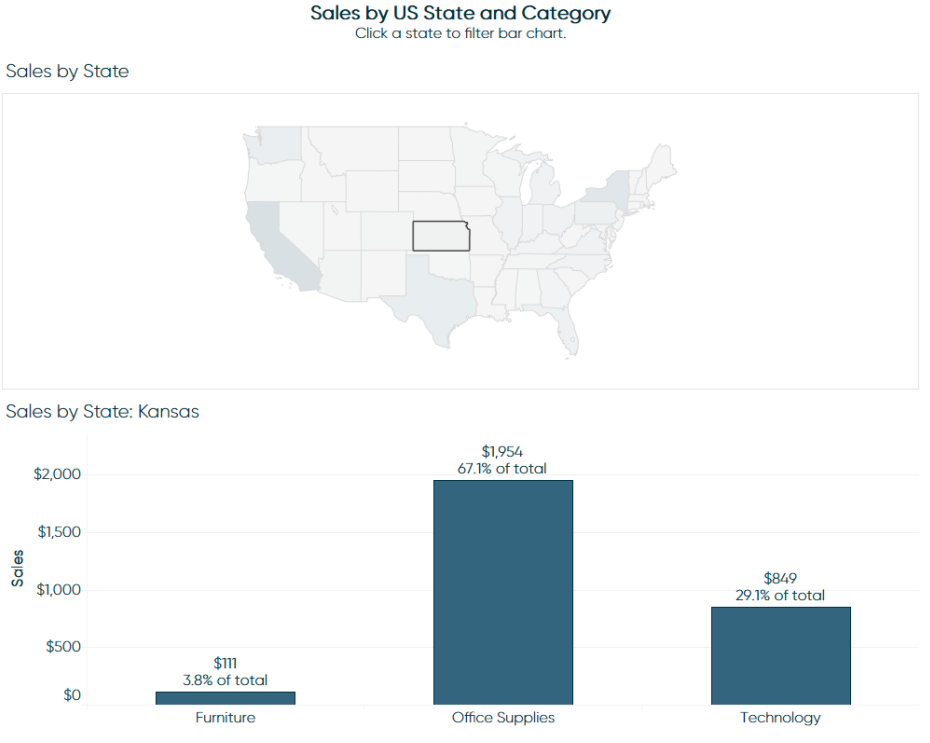
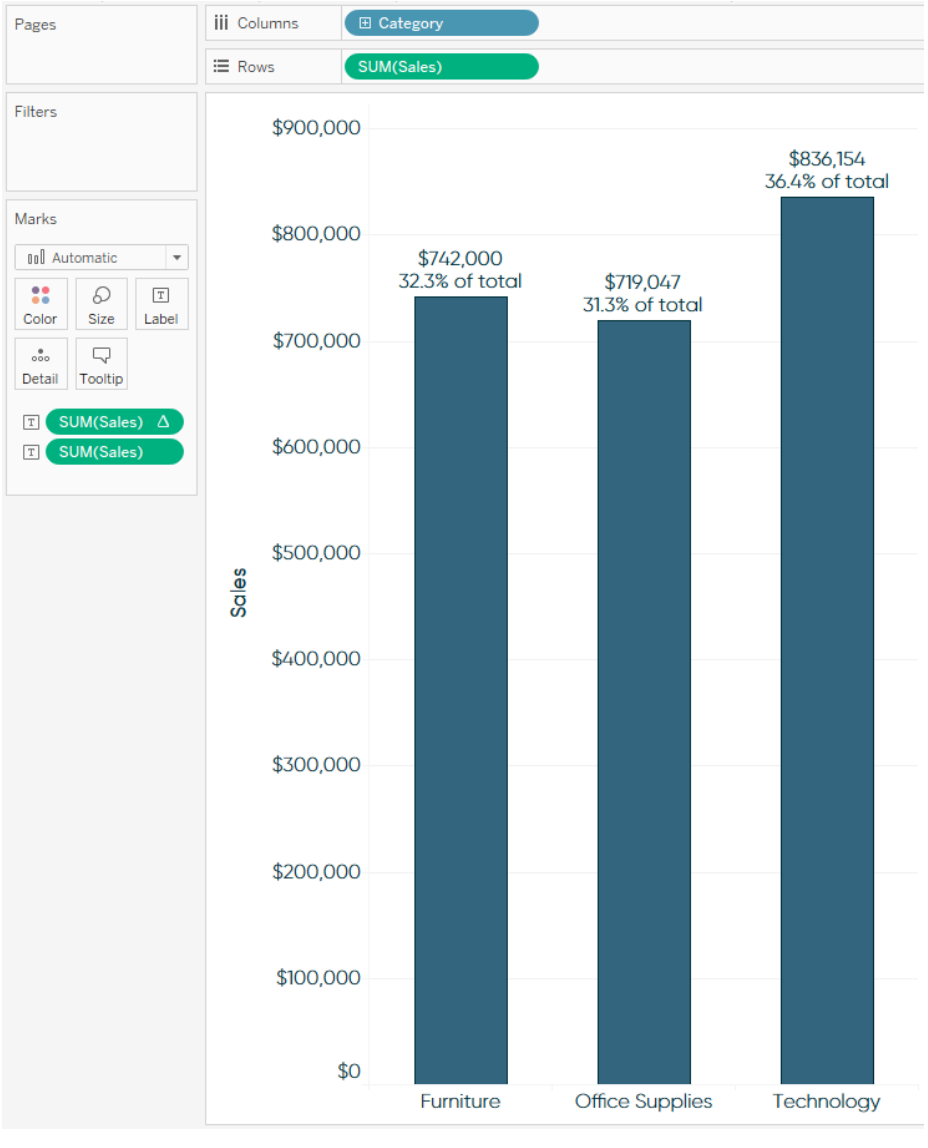
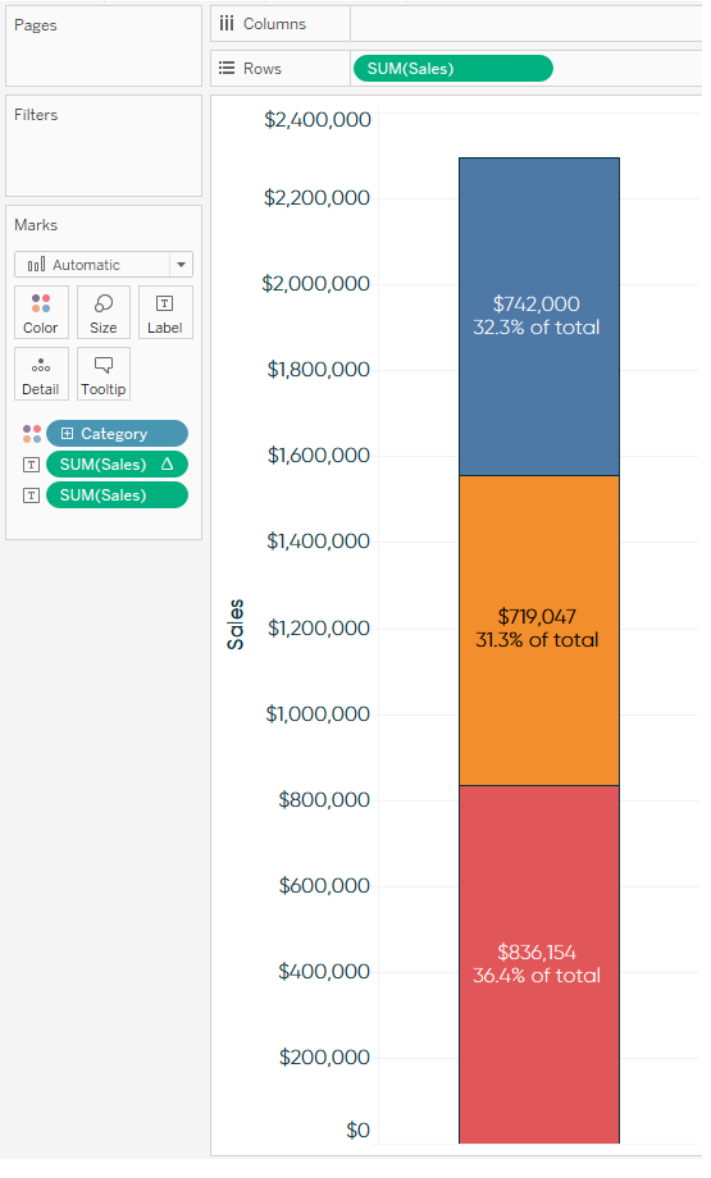


Tableau Pie Chart Alternative #2: Stacked Bars or Areas

Another Tableau pie chart alternative would be to use a stacked bar chart. I do not like this alternative as much because unless the stacked bar is on the bottom, it is very hard to compare trends of individual stacks across the view. This is still better than a pie chart. It is easy to convert a bar chart to a stacked bar in Tableau by simply moving the dimension that is creating each bar from the Rows or Columns Shelf to the Color Marks Card. With a stacked bar, the top of the highest bar represents the total, and each color below represents a contribution to that total. For example, consider the following bar chart:



And now take a look at the same data visualized as a stacked bar chart instead:



Whether you use a bar chart or a stacked bar, 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. For example, the bars likely will not change 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 timeframes.

For this reason, I suggest you at least use stacked bars 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). This quick 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 the following:

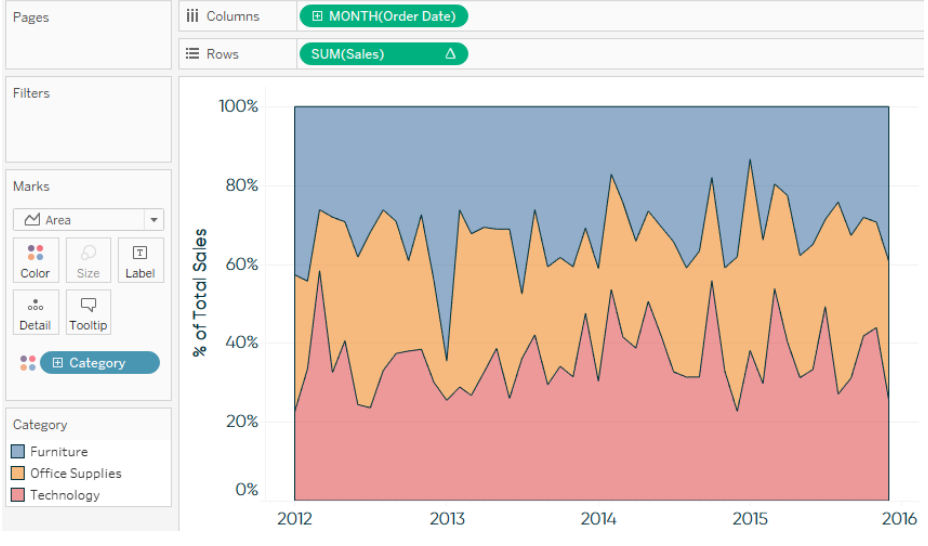
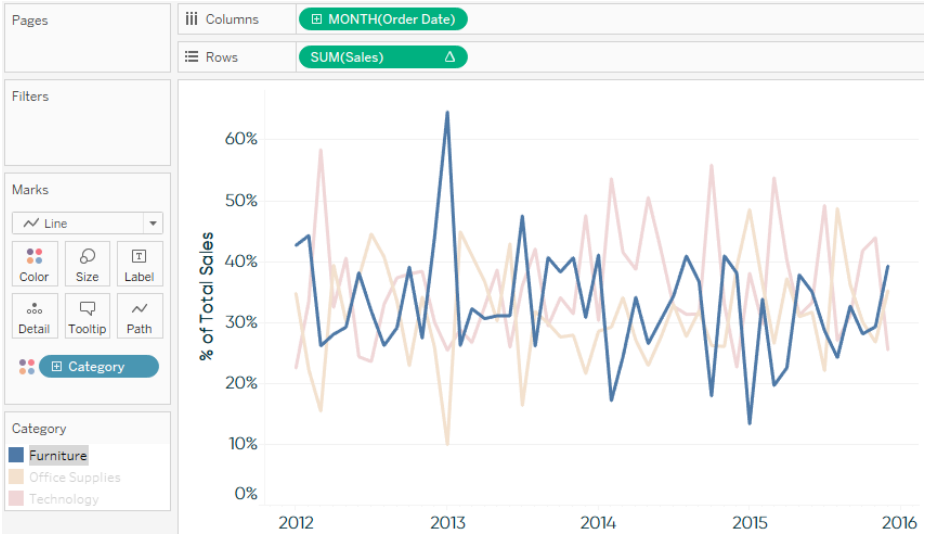


Tableau Pie Chart Alternative #3: My Recommended Approach

I mentioned that we would be building up to my recommended approach, and while the alternatives provided to this point are 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.

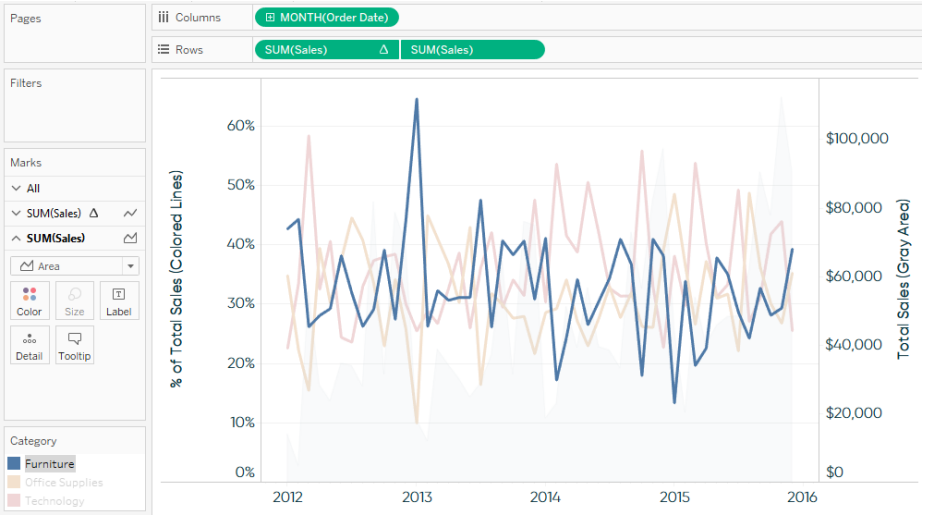
For this reason, the first thing I recommend for the optimal parts of a whole visualization is to change the mark type from Area to Line:



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 the color legend highlights each category to help illustrate each 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. So 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. 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 this:



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 combo chart showing how a distribution changes over time while not losing the context of how our total is changing.

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 53

How to Create and Compare Segments

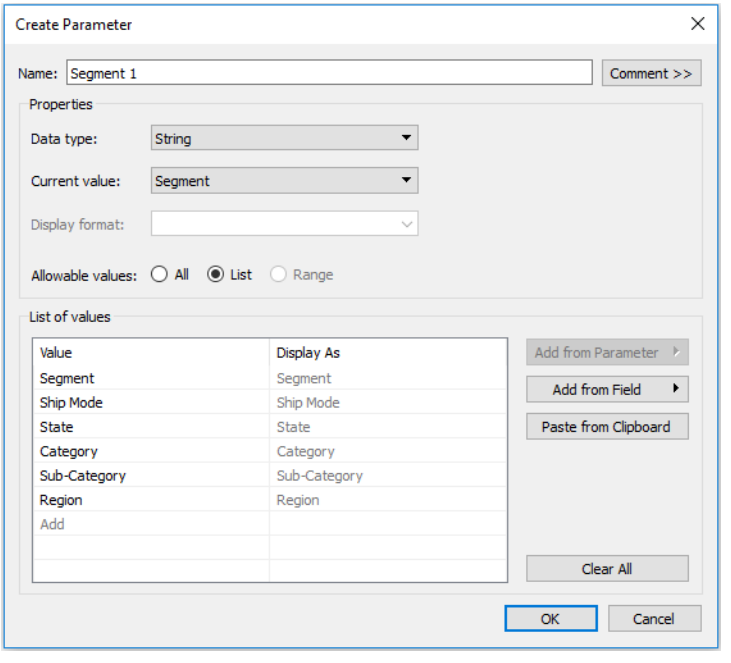
Here's the scenario: The dataset that you're working with in Tableau has several dimensions that represent different customer segments such as married versus single, new versus returning, customer score, or whatever segments are important to your business. It's good that these dimensions are available to slice and dice the data by, but it'd be *great* if you and your end users could change these on the fly, without having to drag and drop the segment of interest onto the view every time you wanted to slice the data by a different segment. It would be even more amazing if you could somehow create a segment hierarchy, allowing you to drill down to different combinations of segments (i.e., Married or Single in the first column drills down to Married or Single, plus a second column for New or Returning).

Solution: Parameters will be our solution to dynamically create and compare segments. In this case, we will create two string-based parameters that will be used to create our combinations of segments:

1. Create a parameter for segment 1.

   For demonstration purposes, I am going to use the dimensions available out of the box with Tableau's Sample - Superstore data. You will likely have segments that are more practical in your data, such as the examples mentioned in the opening sentence of this section (marital status, new versus returning, etc.).

   Create a string parameter with a list of the dimensions you want to be able to use in your segmentation:



2. Create a calculated dimension for segment 1.

   Remember, parameters are dependent, and do nothing on their own. In order to get this parameter to eventually work in our views, we need to create a calculated field that gives Tableau instructions on how to use the parameter. To create our first segmentation, we need to tell Tableau which dimension to display based on the selection made in the Segment 1 parameter. This turns out to be a simple mapping exercise, with the following logic:

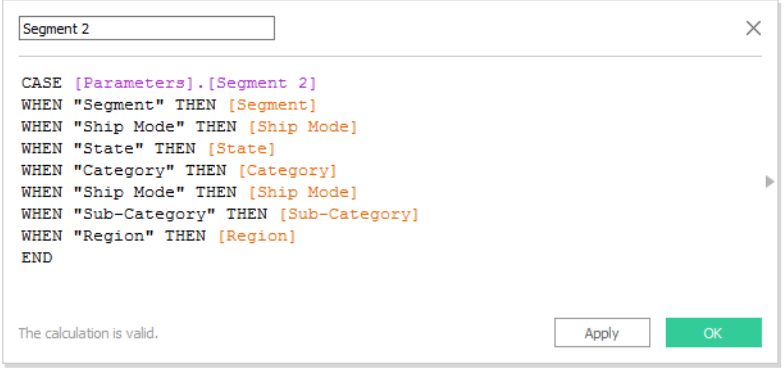
   ```
   CASE [Parameters].[Segment 1]
   WHEN "Segment" THEN [Segment]
   WHEN "Ship Mode" THEN [Ship Mode]
   WHEN "State" THEN [State]
   WHEN "Category" THEN [Category]
   WHEN "Ship Mode" THEN [Ship Mode]
   WHEN "Sub-Category" THEN [Sub-Category]
   WHEN "Region" THEN [Region]
   END
   ```

3. Create a parameter for Segment 2.

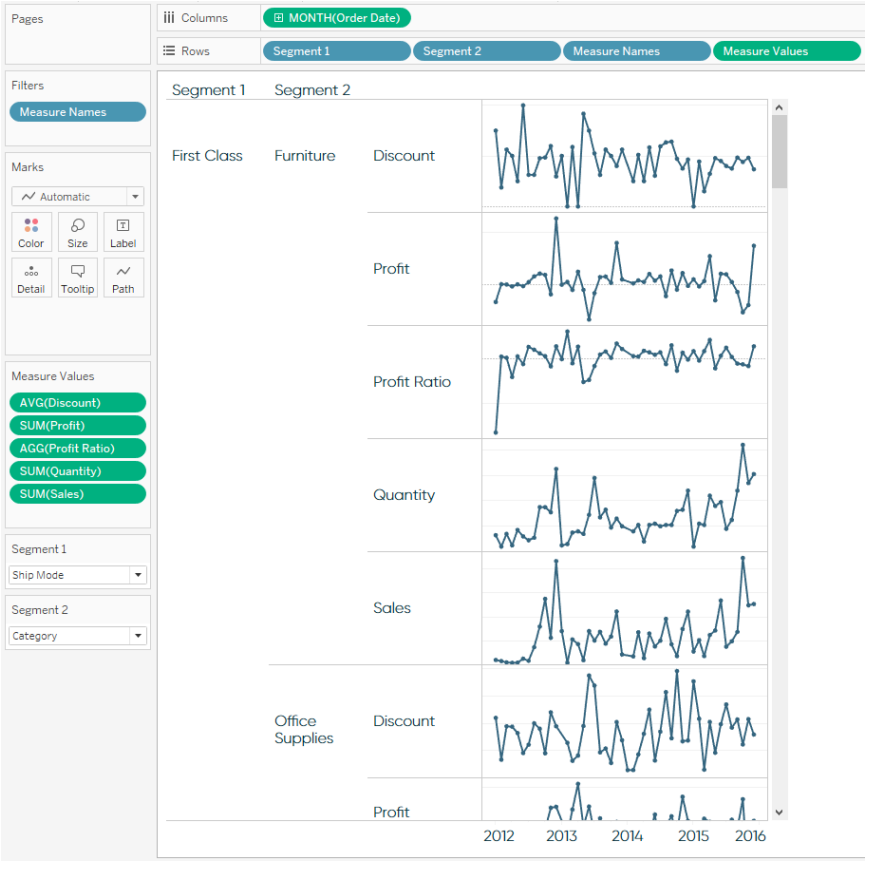
   If you want to have multilevel segmentation, create a second parameter for Segment 2. This can be done by right-clicking the Segment 1 parameter and clicking Duplicate. Edit the copy and rename it Segment 2.

4. Create a calculated dimension for Segment 2.

Instead of retyping all of the logic used to create the Segment 1 calculated field, just right-click the Segment 1 dimension and choose Duplicate. The only trick here is that you not only want to rename the copy to Segment 2, but you also want to change the parameter being used in the logic. To do this, right-click the Segment 1 copy, click Edit, and replace the number "1" with "2" in the first line. At this point, your calculated field for Segment 2 should look like this:



We now have the components needed to create a view that allows you and your end users to choose any combination of two segments. Here is a sparklines view with both segments:



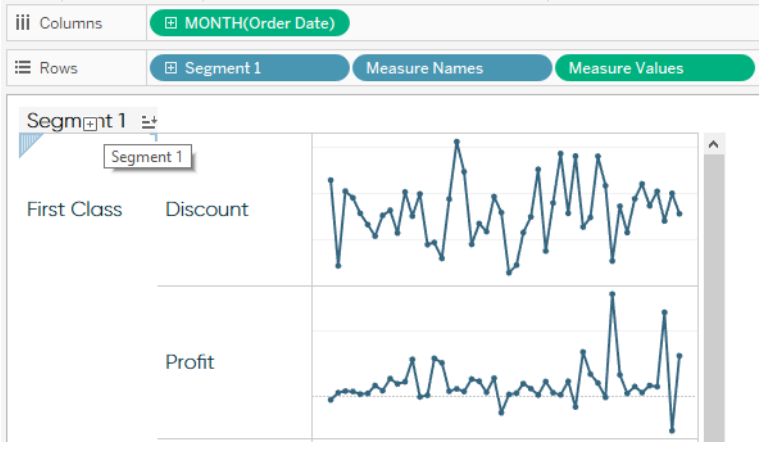
Creating sparklines is beyond the scope of this chapter (see Chapter 24), but the key is to right-click the two parameters we created for segments and choose Show Parameter Control. This is what gives users the ability to change Segment 1 and Segment 2 on the view.

This Is Awesome; Please Tell Me Other Ways This Can Be Used!

Option 1: Create a custom hierarchy with your newly created segments

By creating a custom hierarchy using the Segment 1 and Segment 2 calculated fields, you and your end users will be able to drill down to the secondary level of segmentation and back up to the primary level of segmentation. To create a custom hierarchy, click the dimensions on the Dimensions area of the Data pane

that you want included in the hierarchy (Control-click on a PC to multiselect), right-click, hover over Hierarchy, and choose Create Hierarchy:

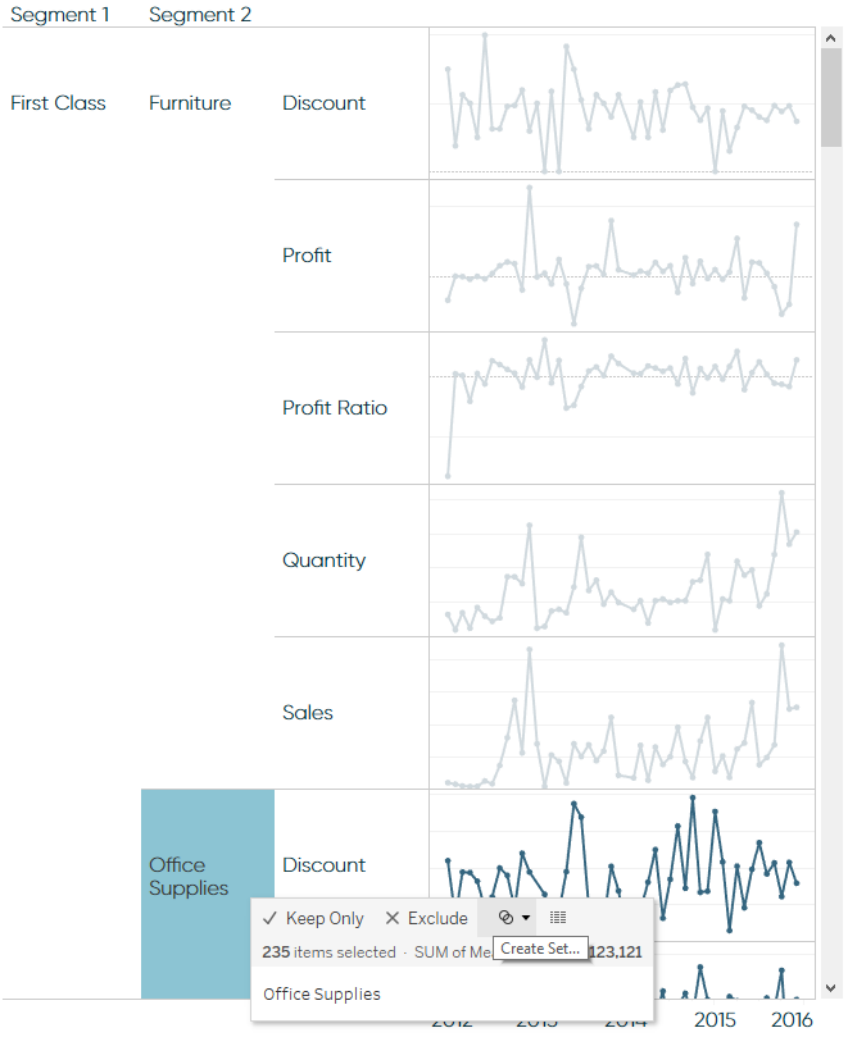


Option 2: Create additional levels of segmentation

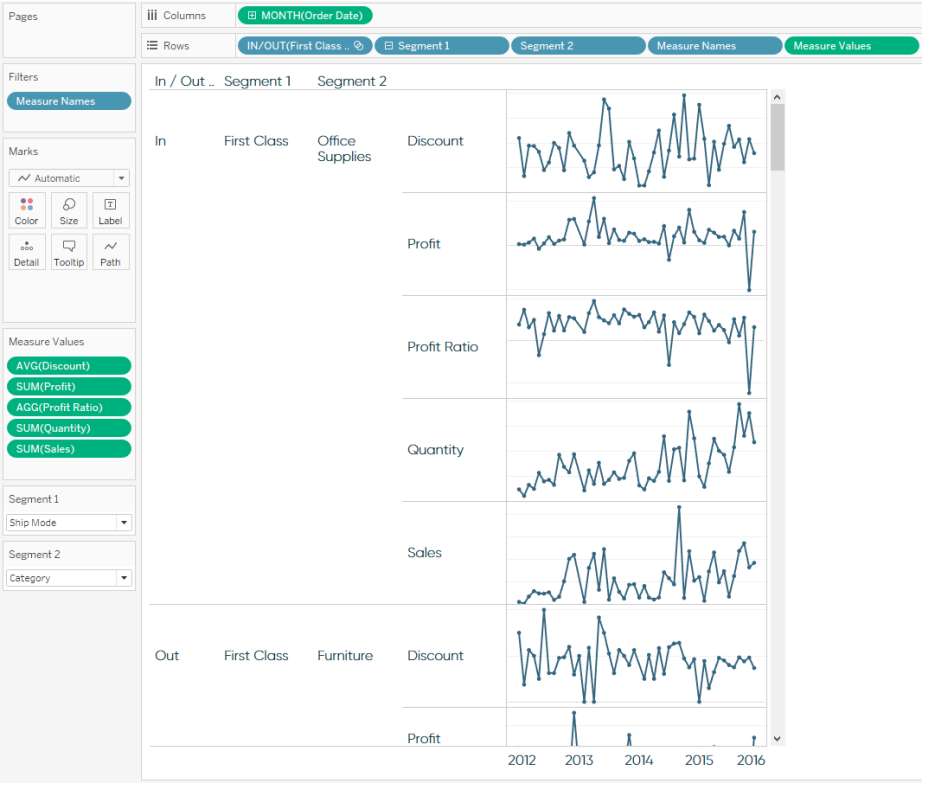
You are not limited to two combinations of segments, though usability may become messy past three or so. If you would like to create more than two levels of segmentation, simply repeat the preceding steps three and four.

Option 3: Save a specific combination of segments for future use

If you find a combination of segments that is particularly interesting, Tableau makes it very easy to create a set straight from the view. These sets can be reused on different worksheets for deeper analysis. Let's say we want to isolate the Ship Mode: First Class → Category: Office Supplies combination for the months on our view. Simply click the axis at the deepest level of segmentation, and click the Venn Diagram icon to create a set:



This set will appear in your left navigation and can now be used as a dimension on views, to filter, or within calculated fields:



Segmentation is one of the best ways to create comparisons that add value to your data visualization. The next time you are mining for stories in your own data, consider leveraging parameters to create custom segmentations on the fly.

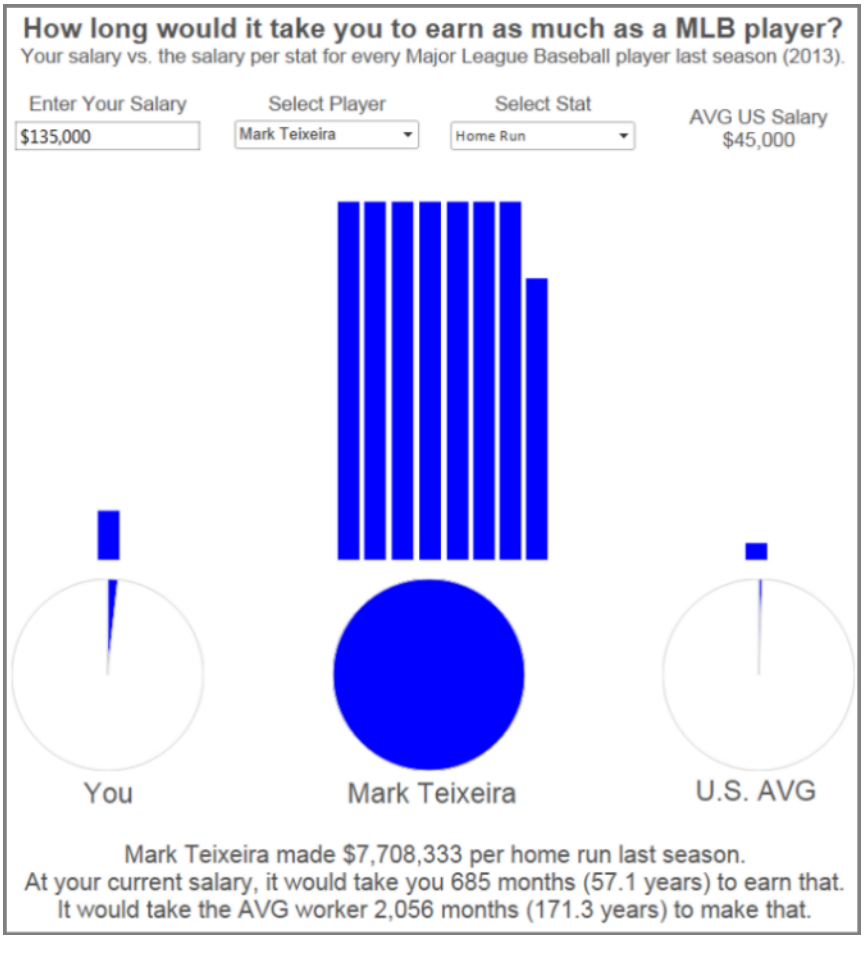
CHAPTER 54

Five Design Tips for Enhancing Your Tableau Visualizations

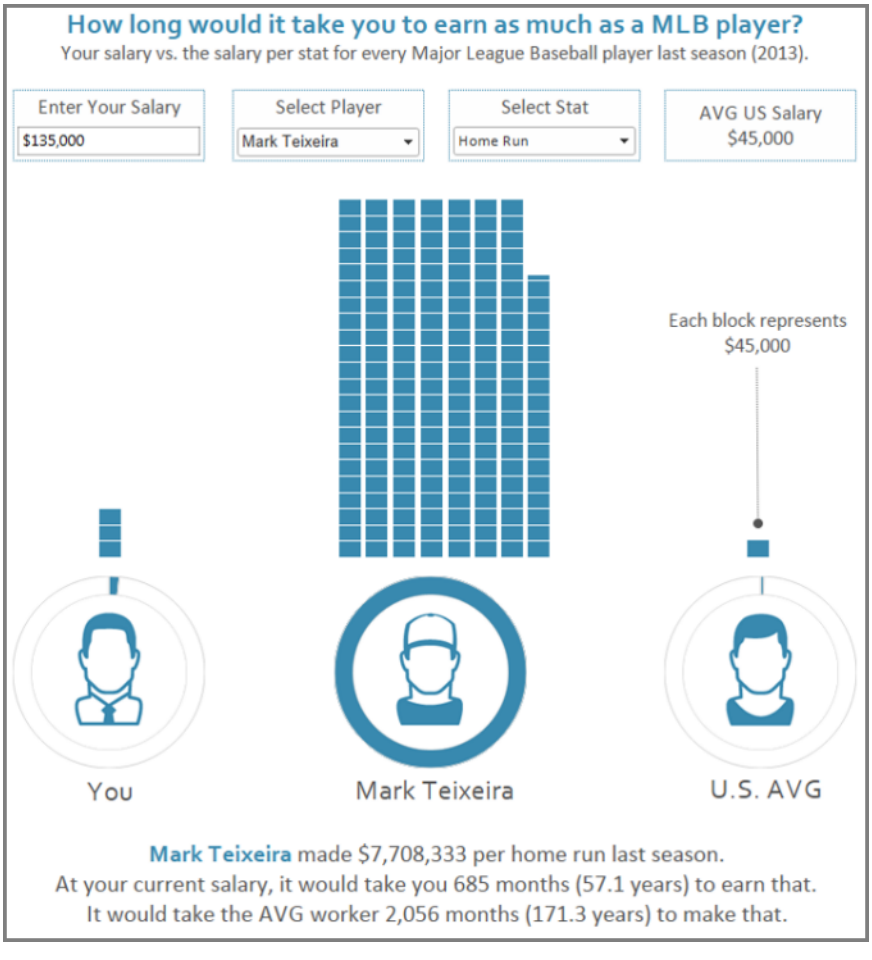
Accurate and honest data is the core of every great data visualization. If you are able to leverage data visualization best practices to make that data easy to understand and act on, you are a huge step ahead of the game. Undoubtedly, you can't have a quality data visualization without quality data and fundamental best practices. That being said, without balancing the quality of your data with a quality design, your data visualization will never reach its full potential. Before dismissing design as an unnecessary—or worse, undoable—component of your data visualization, ask yourself why you visualize data. My answer is to *find and communicate actionable insights*. To make insights actionable, or provoke change, your data visualization needs to be seen by the most relevant and/or largest audience possible. Design is the key to reaching that audience.

I often talk about the concept of being remarkable, my favorite Seth Godin principle. Being remarkable means that your work is so high quality, distinctive, or compelling in some way that it moves your audience to remark about it. In today's world, that may mean sharing your work on social media. What it may also mean to you in the corporate world is that your work gets passed up the ladder—creating change, recognition, promotions … fame … and fortune! The latter may be a longshot, but it's worth a try.

To help illustrate how big of a difference design can make to a data visualization, I have stripped out most of the design elements from my most socially viral work to date, *Your Salary vs. a MLB Player's Salary* (I returned most of the fonts, colors, and formats to their default settings in Tableau, but I want to point out that I could have minimized the design even further by returning the view to its default layout and standardizing the font sizes):



In the viz, the end users can type in their own salary and compare it to the salary of any Major League Baseball player across several different statistics. The concept and data are interesting enough—if not disheartening—to stand on their own, but I credit the "remarkability" of the dashboard to its design. Here is the exact same view with some simple design upgrades:



In addition to the true data purists who do not see the value in complementing data with an aesthetically pleasing design, one reason I hear people give for not trying to balance data and design is that they have no graphic design experience. If you look closely at the "after" image, you will find that the only elements in the entire view that required any Adobe Photoshop or Adobe Illustrator talent are the icons that form the center of the dashboard's donut charts. On top of that, I will let you in on the secret that those icons are from a stock photography file that I purchased from iStock. I simply recolored the icons to my liking and pasted them in the center of a circle to create the donut effect.

My point is: You can do this.

To help you get started, let's take a look at five ways you can balance data and design in Tableau without any graphic design experience.

Color

There are countless articles on color's impact on data visualization, and for good reason. The full possibilities of color are beyond the scope of this chapter, but I will share my two biggest tips for harnessing the power of color:

- Use simple color palettes with five or fewer colors. Of course, sometimes the variety of your dimension members will dictate how many colors your viz includes, but remember that every addition puts more stress on the user to efficiently decipher the story in your data (i.e., think looking back and forth at a color legend).
- Mute your colors. The technical definition of "muting" colors means that you are reducing their hue saturation. The practical definition is that you are making the colors less intense, making them more pleasing to look at. An easy way to do this is to add transparency to the colors.

If you would like to learn more about the theory and psychology of color, Chapter 55.

Typography

Tableau comes out of the box with more than 100 different fonts. If you thought that Tableau's default fonts were your only options, then this tip is for you! Fonts are a powerful way to differentiate your dashboard, and they serve the additional purpose of communicating priorities to your end user.

There is no one-size-fits-all approach to font design, but I recommend picking out one or two fonts that you think look nice, fit the message of your dashboard, and most importantly, are easy to read. Experiment with different point sizes (i.e., 8pt, 10pt, 24pt) to break up your data visualization and prioritize its elements in a subtle way.

Layout

Making thoughtful layout choices can help guide your end user through the story of your data visualization even when you aren't there to explain it. Paying special attention to the spacing of each dashboard component also helps ensure that you end up with a clean design.

One best practice regarding layout is to place the highest-priority content toward the top and left of the view, and relatively lower-priority content toward the bottom and right of the view. As mentioned in Chapter 51, I have also always followed a general rule of thumb to keep dashboard components—including titles, filters, and charts—to twelve or fewer.

For more on layout, see the most-read Tableau guest post of 2013, Kelly Martin's Dashboard Design and Layout Best Practices.

Usability

Similar to layout, implementing good usability is one way you can help end users get the most from your data visualization, even when you don't have the chance to walk them through it.

Make no mistake, usability is an important aspect of design: UX design. Add filters and dashboard actions that allow end users to find their own stories in the data. Inevitably, when users understand how to use your dashboard and find insights on their own, they are more likely to stick and be shared.

Tableau offers so many capabilities, one of the biggest challenges is communicating the interactive features available. Andy Cotgreave offers some great insight and tips on the topic in his article, "How Do You Communicate That People Can Interact with Your Designs?"

Details

In this case, I saved the best for last. You've likely heard the expression, "the devil is in the details." Well, I'm here to tell you: so is the remarkability of a data visualization. Whenever I share a data visualization that doesn't have any graphic design (as in actual graphics), but the audience thinks that it looks great, I credit the meticulous attention to detail in the dashboard. Many times this type of audience can't quite put their finger on why the dashboard looks good, but I know the secret to my success is that the dashboard is more polished than other work they've seen.

Examples include reducing lines, softening gridlines, adding borders to marks, adding transparency to reveal overlapping marks, formatting filters, and potentially dozens more.

This attention to detail, combined with the four other ways mentioned to balance data and design—even without graphic design expertise—will help your data visualization achieve its full potential by reaching and engaging the most relevant and largest audience possible.

CHAPTER 55

Leveraging Color to Improve Your Data Visualization

The use of color can help a story in your data pop off the page. Color can add a level of professionalism to any project and can even elicit specific emotions in viewers. The use of color is one of the easiest ways to take your data visualization from good to great—so why do we see so many blue, orange, and green lines out there? Maybe it's that creative design is thought of as unnecessary in a corporate setting. Maybe visualization designers don't realize just how easy it is to set your dashboards apart by introducing some custom color. True, Tableau worked hard to provide standard color options that work very well together, but maybe you need to align your data visualization with your corporate identity or are just looking for a way to help your work stand out. To help you leverage color to improve your data visualization, this chapter shares an introduction to color theory and points you to a tutorial for loading custom color palettes into your own version of Tableau so you can use them every day.

The Color Wheel: Where It All Begins

The color wheel is based off of the primary colors of red, yellow, and blue, with secondary and tertiary colors between each of the three primary colors. This allows a designer to visualize the balance and harmony of colors when they are side by side. The colors on the outside ring of the color wheel shown here are at full saturation, meaning there is no black or white added. These are referred to as hues. Adding lightness (white) or darkness (black) will result in different tints and shades of each primary, secondary, and tertiary hues. These variations can be seen as you move to the center of the color wheel. Note that some color wheel illustrations you find may have the original hues represented on a middle ring, with darker shades as you move to the outside of the wheel and lighter tints as you move to the center of the wheel.

There are ten basic color schemes that can be derived from the color wheel, but for use with Tableau, I am going to recommend starting with complementary and monochromatic schemes.

A *complementary scheme* is created whenever two direct opposites on the color wheel are used. This is a great way to start a custom diverging color palette. For example, from the wheel, you could use blue-violet to represent positive values, and yellow-orange, the color on the exact opposite side of the wheel, to represent negative values. You can also use variations from the inner rings as long as the opposites are on the same ring. A complementary scheme is also a good choice when you are comparing two distinct dimension members, such as East versus West.

A *monochromatic scheme*, created with different shades of the same color, is ideal for a custom sequential palette. To create this, simply pick any column of the color wheel,

then use any or all of the variations in that same column. In Tableau, this could be used to color the intensity of a measure, such as sales, with darker variations of the color representing higher sales. This type of color scheme is ideal for a heatmap. I also like to use these schemes when comparing year-over-year values, with the hue representing the current year, and lighter tints representing past years (the older the year, the lighter the tint). Many monochromatic color schemes come standard with Tableau, but maybe you would like to try a different color or control the intensities of a specific color.

The Psychology of Color

If you are looking for help with choosing a color to use in a design, consider what each color means. Thinking about color in this way will not always be necessary, but if your design is themed, using the psychology of color can help your visualization elicit emotional responses in viewers. Here is a brief psychology of each color according to the book *Color Harmony Compendium* by Terry Marks and Tina Sutton:

- Red is the most vibrant color in the spectrum. Not only does it express emotions such as excitement, power, and passion, it brings objects to the foreground. Red is a good choice if you want to illustrate a clear story in your data.
- Yellow is the most visible color and is synonymous with happiness. Yellow also has the advantage of stimulating clear thinking. Using a yellow background with dark type has been proven to aid in retention. Consider using a combination like this when you want your audience to remember one main point from your visualization.
- Orange, like red, is both a high-energy and high-visibility color. Even in small amounts, orange can help convey warning signals, which makes it a good choice for KPIs that are performing below expectations. Orange is also a favorite of children, teens, and athletes, so consider this color accordingly if you work with any of those audiences.
- Green is said to be the most physically relaxing and calming color in the spectrum. It has emerged as a global symbol for safety, so it is no wonder that green is typically associated with positive values in corporate scorecards.
- Blue is the best-liked of all colors. This color is typically perceived as positive, making it an alternative choice for positive values. Blue has the added advantage of being colorblind friendly. Use darker shades of blue to instill a sense of loyalty, trustworthiness, and integrity.
- Purple, especially darker shades of the hue, exudes luxury. Be careful with this one though—studies have shown that people get less work done around purple because it encourages daydreaming!

- Pink, no surprise, is considered the most feminine color. Pink discourages aggressiveness and has a soothing effect. This color would probably be best-served in a themed visualization, but in certain situations could also be a good choice for conveying that performance is positive, or at least in-line with expectations.
- Brown has been shown to put consumers at ease and is considered timeless. It is also closely associated with the earth, making it ideal for visualizations related to the environment.
- Gray has many similar qualities to brown, but is thought of as less warm when used alone. On the positive side, gray can represent conservative authority. Using it as a secondary color can help instill a feeling of maturity and the associated reliability.

Using Custom Color Palettes in Tableau

Now that you have some guidance on which colors to use in your designs, I will close by sharing a way to help you put the use of color into practice. In addition to the custom colors that can be entered as RGB or HSL values by clicking any color in a color legend, Tableau also makes it possible to load custom color palettes that can be saved for permanent use in your own version of Tableau. Tableau allows you to integrate three different types of custom color palettes: categorical, sequential, and diverging. If you have specific colors that you would like to have available to you while you design, start by looking up the hex values for those colors. Once you have the values, all you need to do is add a short snippet of code that includes the hex values in the *preferences.tps* file in your Tableau Repository. There is a great article in Tableau's Knowledge Base showing you exactly what this looks like.

Use these methods for choosing and using color and your visualizations will be standing out in no time!

CHAPTER 56

Three Creative Ways to Use Dashboard Actions

Dashboard actions, filters, and parameters are three of the best tactics to use in Tableau because they provide a way to hand over 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, we will share three different ways to leverage dashboard actions to improve your user experience.

A Primer on Tableau Dashboard Actions

Before we get to the 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" on page 355.

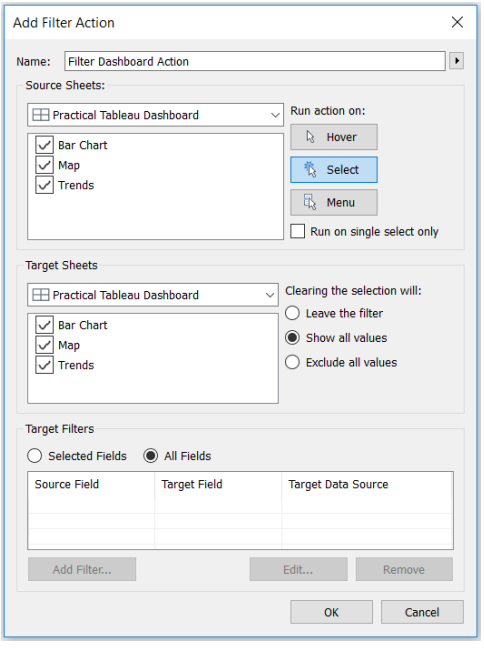
Dashboard actions in Tableau allow you to add logic to dashboard components that create actions somewhere else. For example, you can add logic that says, "If a user clicks on Dashboard Sheet 1, I want something to happen on Dashboard Sheet 2." If you think about dashboard actions this way, their setup 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 lower-left corner, you will be presented with three options for the type of dashboard action that you want to add:

Filter
: If you click sheet 1, sheet 2 will be filtered to whatever you clicked on sheet 1.

Highlight
: If you click sheet 1, sheet 2 will be highlighted by whatever you clicked on sheet 1.

URL
: If you click sheet 1, open a URL (this can either be opened on a web page dashboard object 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 upper-right corner of the dialog box):

Hover
: If you hover over the source sheet, the action will take place on the target sheet.

Select
: If you click the source sheet, the action will take place on the target sheet.

Menu
: If you hover over the source sheet, a menu of dashboard actions will appear in the tooltip. Clicking 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 or clicking away from the chart):

Leave the filter
: Leaves the last dashboard action that happened in place. If you filtered sheet 2 by something clicked on sheet 1, the filter on sheet 2 will stick.

Show all values
: Reverts back to the original view as if no dashboard action took place.

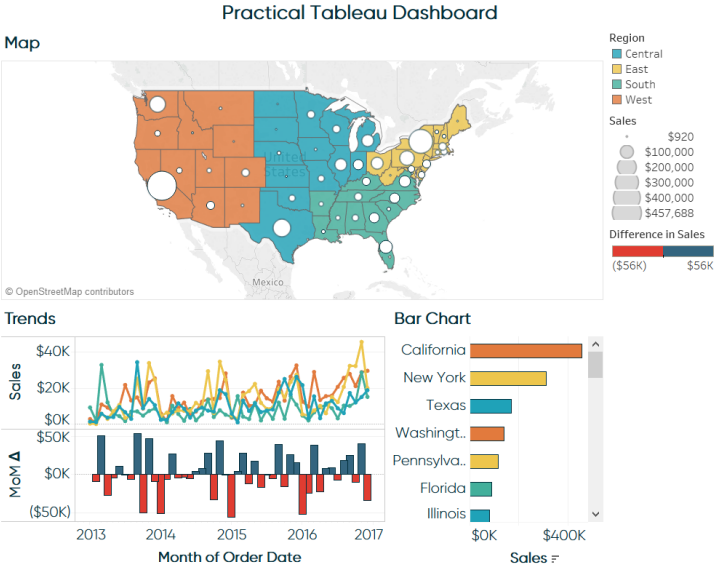
Exclude all values
: Clears everything off of the target sheet, meaning the target sheet will not show unless a dashboard action is executed.

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.

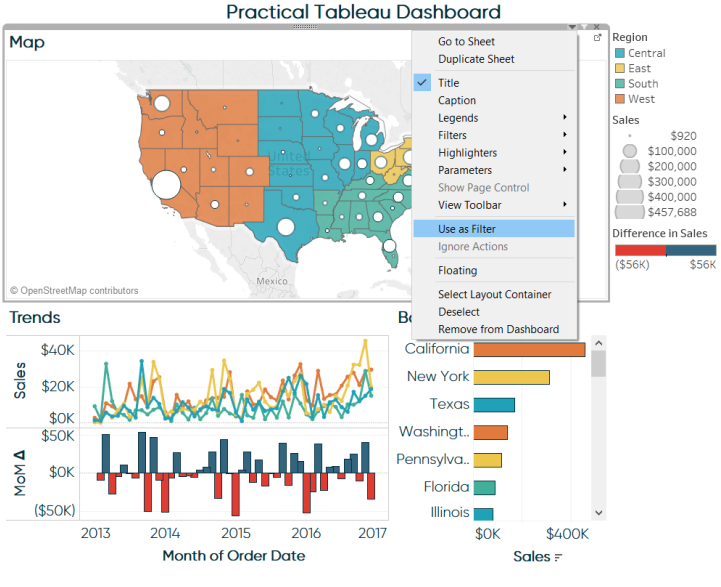
Tableau Dashboard Action #1: Use Every Sheet as a Filter

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 upper-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.

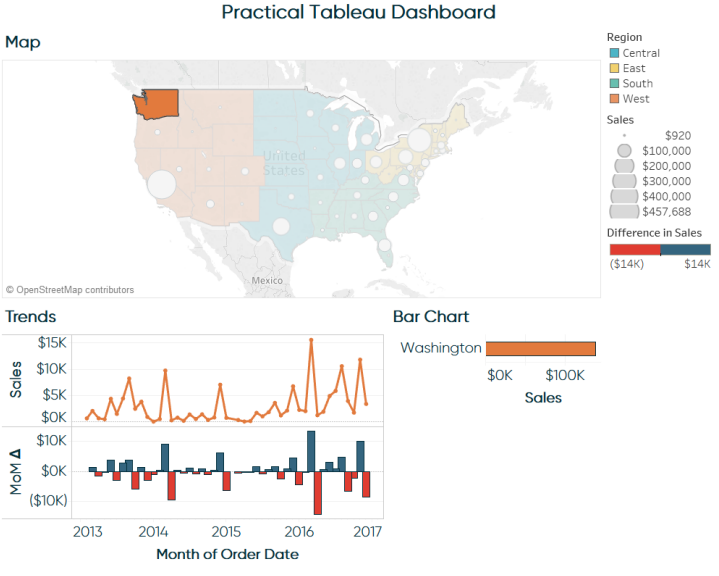
Let's look at how this would work with the sample dashboard created in Chapter 17. Here is the original view:



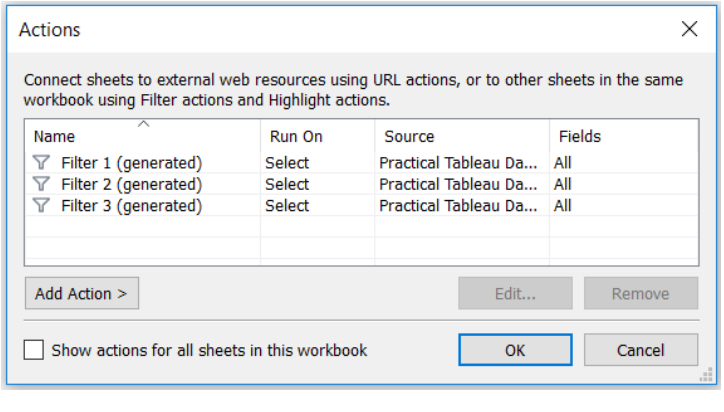
I will now choose to use every individual dashboard sheet as a filter for the entire dashboard by hovering over each sheet, clicking the down arrow that appears in the upper right, and selecting "Use as Filter":



Now when I click any sheet, the other sheets are filtered to whatever I clicked. For example, if I click 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 as shown in the first image in this chapter. You can confirm Tableau created dashboard actions by navigating to Dashboard → Actions:

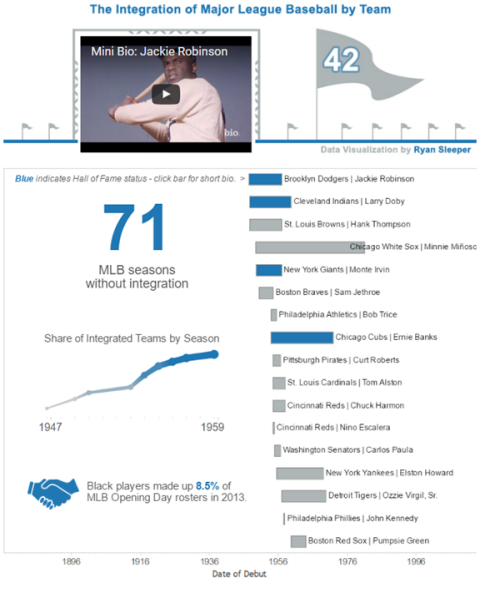


The good thing about setting up each dashboard sheet as a filter instead of sharing one dashboard action between all of the source sheets is that you can refine the

actions independently of each other. For example, the bar chart doesn't add much value when it is filtered to just Washington as pictured here. Maybe it would be better if I added a target filter that filtered the bar chart to region instead of state when I click 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.

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

This is a tip I first picked up from friend and mentor, Ben Jones, at his excellent blog, DataRemixed.com. The first time I embedded a video in a Tableau dashboard was my 2013 viz, MLB Integration by Team. In the interactive version, clicking any hall of fame player, represented by a blue Gantt bar, will load a short biography of that player on the scoreboard:

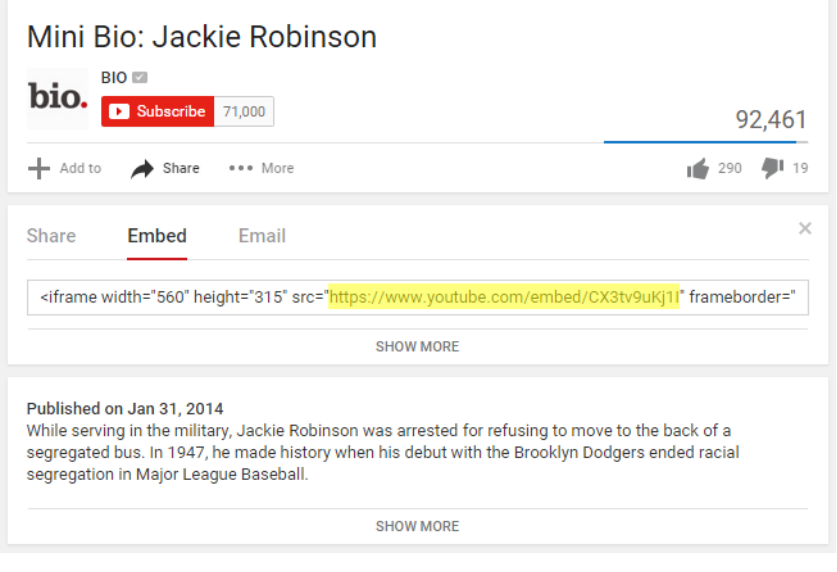


This novel effect was actually quite easy to create in Tableau. The only tricky part is finding the correct YouTube links and having the YouTube links as a field in your underlying data source. Here's a look at the underlying data for this viz:

	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%	https://www.youtube.com/embed/CX3tv9uKj1I
3	Larry Doby	Cleveland Indians	American	7/5/1947	7/26/1959	4404	thirteen	2	13.00%	https://www.youtube.com/embed/m6yq0BR80oY
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%	https://www.youtube.com/embed/K6ZHOKRRI84
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%	https://www.youtube.com/embed/HPL-Z6cst1Q
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:



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 I could make it the exact same dimensions as the scoreboard—

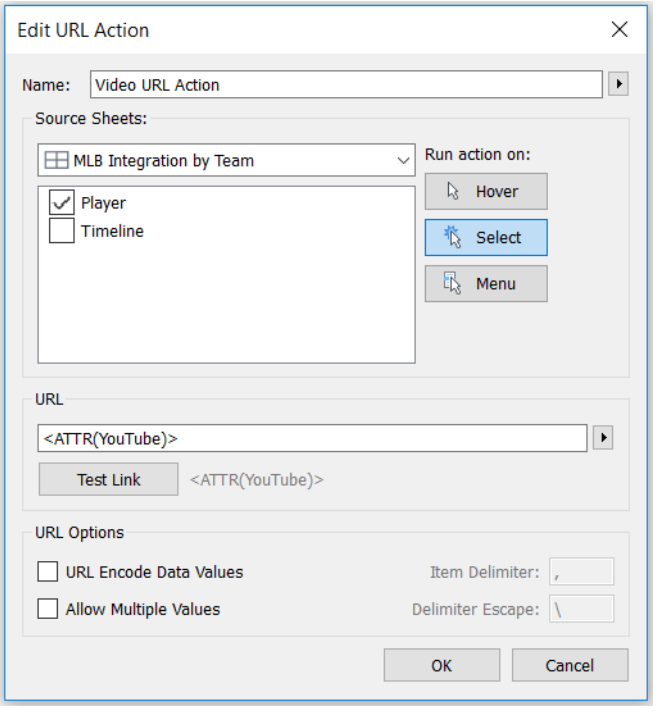
providing the illusion that the video is being displayed on the jumbotron. When you add the Web Page object, Tableau will ask you to enter the URL for the page you want to display; leave this blank for now because we will be using a dashboard action to provide these instructions.

2. Add a dashboard action by navigating to Dashboard → Actions → Add Action.

   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).

3. 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.

   Here's how my final dashboard action looks:



Now if I click a name on the Player sheet 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. In the interactive version, clicking 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:

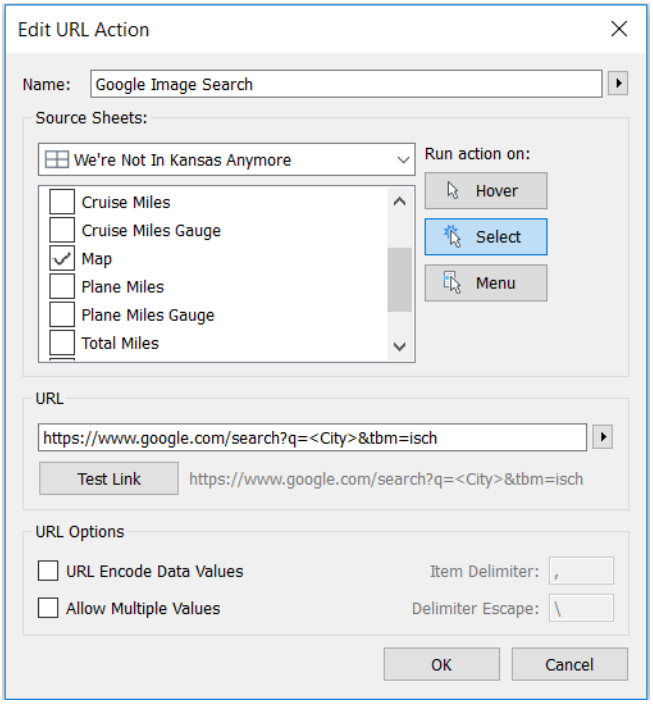
1. Go to Google Images and search for one of the locations.
2. Record the URL that appears in the browser. A search for Kansas City resulted in this URL:

   https://www.google.com/search?q=kansas+city&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiH6eqWhoDKAhVWz2MKHctYBJsQ_AUICSgD&biw=1366&bih=643

   You can keep the entire link, but all that you actually need is: *https://www.google.com/search?q=kansas+city&tbm=isch*—this URL includes the query (Kansas City) and tells Google to show results on the image tab (isch = image search)
3. Add a dashboard action by navigating to Dashboard → Actions → Add Action.

   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, paste the URL that you recorded from the previous search. 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 enabling discovery, one tactic for creating insights and improving retention for you and your audience.

CHAPTER 57

How to Conditionally Format Individual Rows or Columns

Almost every Tableau user has also been, or still is, a heavy Excel user. Being that we almost all learned Excel first, it is natural to approach a tool like Tableau for the first time with some preconceived notions about how we think it should work. While there is some overlap between the reporting outputs of the two software programs, it can take some time to understand how the two programs differ as well as best fit together.

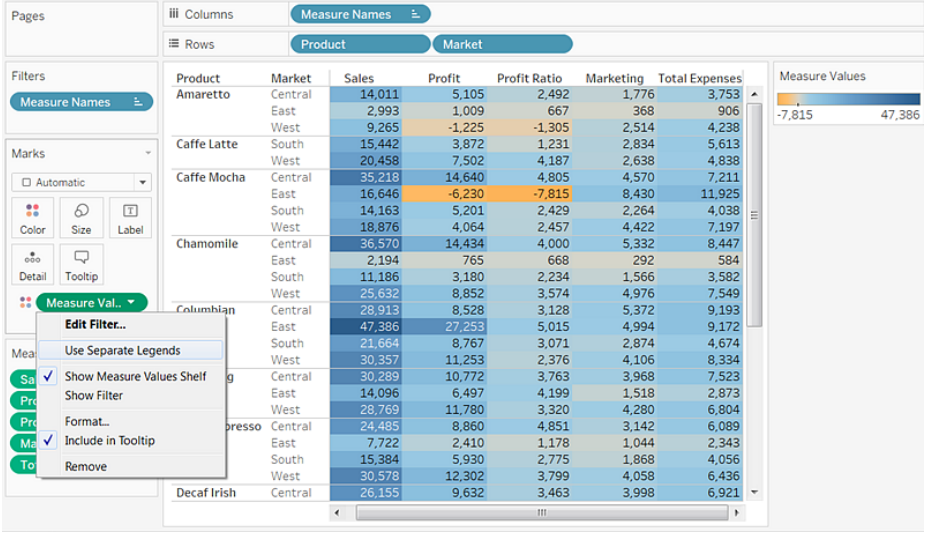
While there are tremendous advantages to using both programs, I've seen firsthand how learned behavior for Excel can become a barrier to adoption of Tableau. One such example relates to conditional formatting, and the ability to modify individual rows or columns. This is very easy to do in Excel because you can modify every single individual cell exactly as you wish. While Tableau certainly has many encoding options available through the Marks Cards, it can seem a little "all or nothing." For example, if you place a measure on the Color Marks Card, then *all* marks on the view are colored.

This chapter shows you how to use the *legends per measure* feature and a trick to conditionally format in Tableau like you can in Excel.

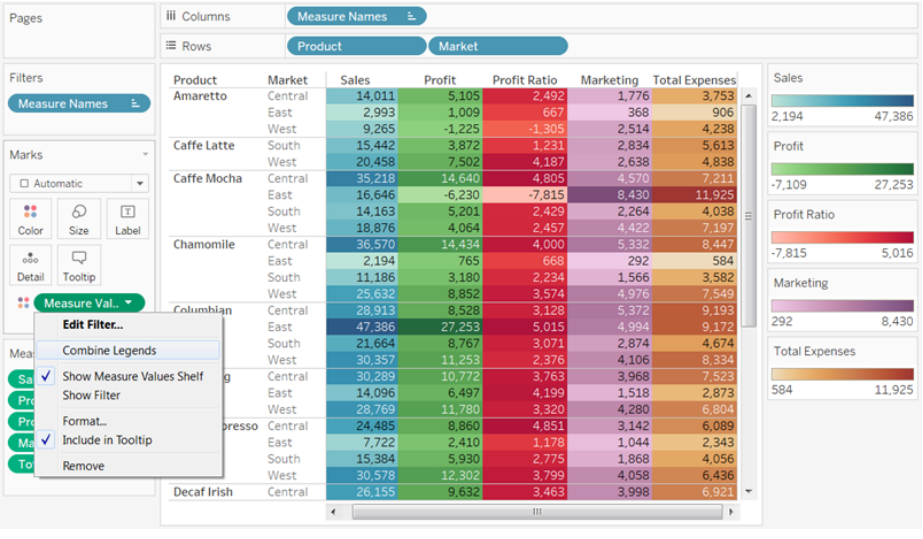
How to Use Legends Per Measure

One of the features released with Tableau version 10.2 helps solve the limitation of "all or nothing" encoding. With legends per measure, if you build a highlight table with multiple measures, then color the marks by the Measure Values field, you can create a color legend for each measure on the view.

The first step to use legends per measure in Tableau is to set up a table that uses the Measure Names field. Next, place the Measure Values field on the Color Marks Card. In the past, this would result in one color legend for the measure values, with the same colors being applied across the entire table of different measure names. With legends per measure, simply clicking the Measure Values field on the Color Marks Card and choosing Use Separate Legends will result in independent legends for each measure in the table:



Here is how the highlight table looks with legends per measure applied:

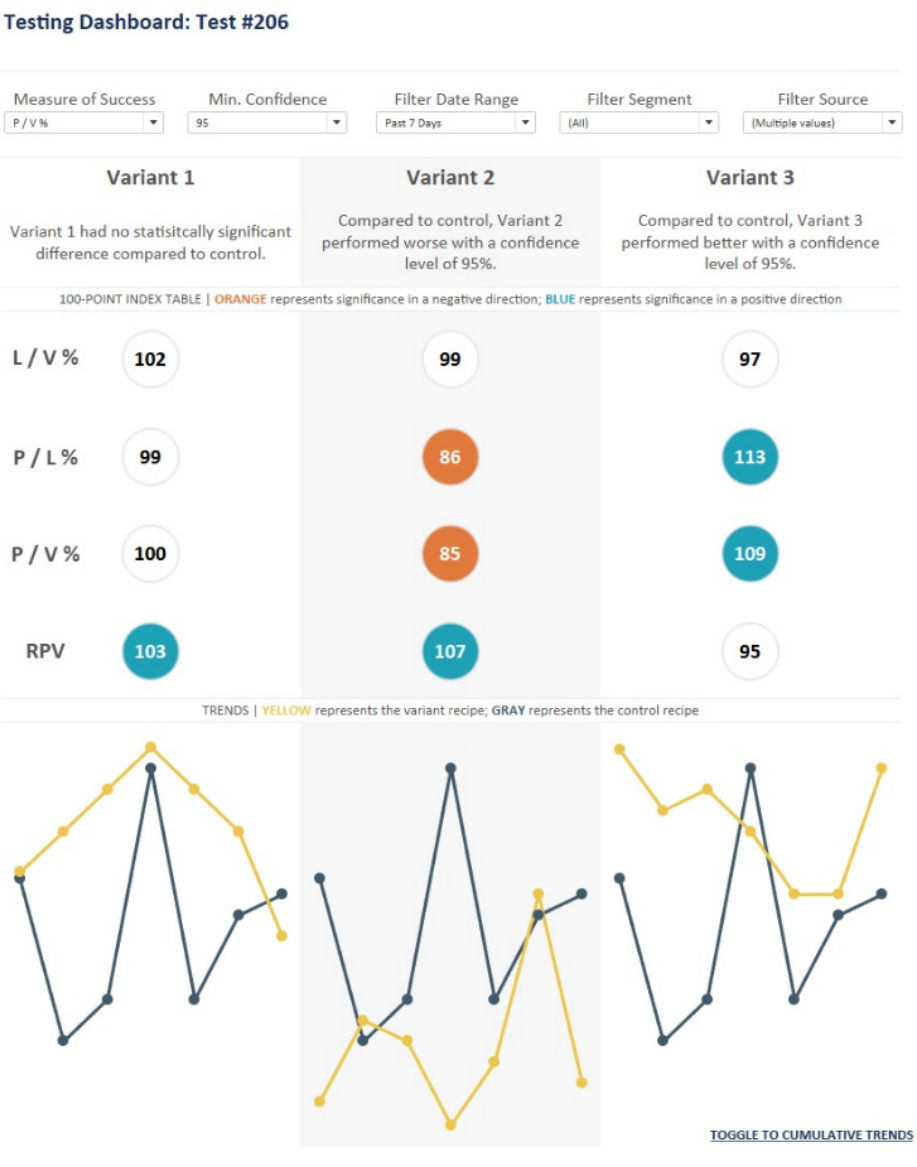


Note you can also recombine the legends by clicking the Measure Values field on the Color Marks Card again and choosing Combine Legends.

How to Conditionally Format in Tableau Like Excel

Legends per measure is a nice step toward providing flexible formatting for highlight tables, but it still doesn't allow you to *conditionally* format the individual columns for measure names. For example, what if you wanted to color the cells for each measure name based on whether a condition was met for each respective measure? Fortunately, there is a trick that leverages a placeholder field to allow you this level of Excel-like conditional formatting in Tableau.

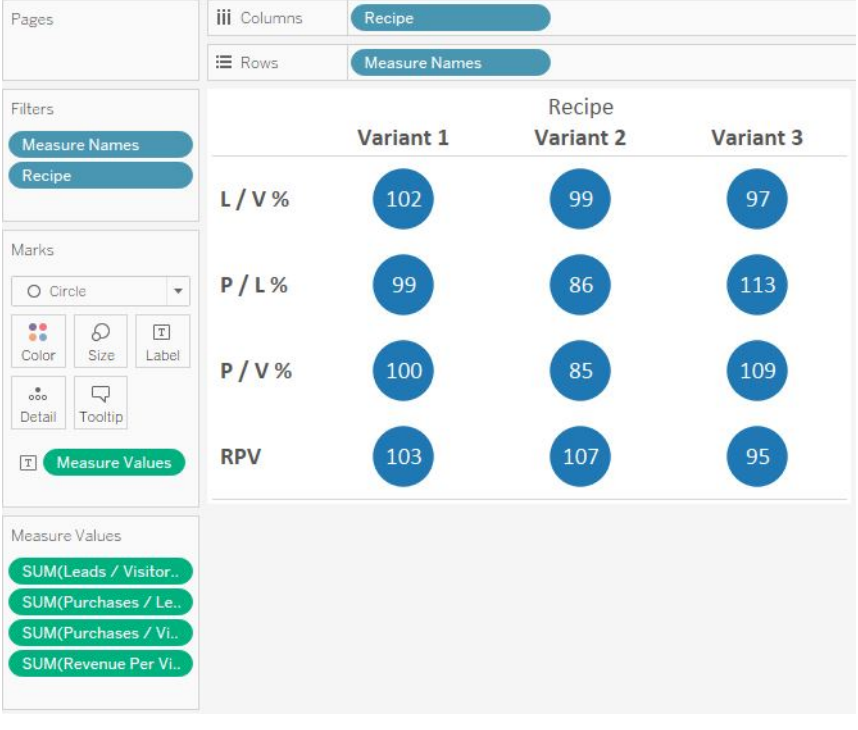
To illustrate the need for this trick and provide a tutorial on how it's used, we will be using this Tableau dashboard that is used to evaluate results for A-B split tests on a company's website:



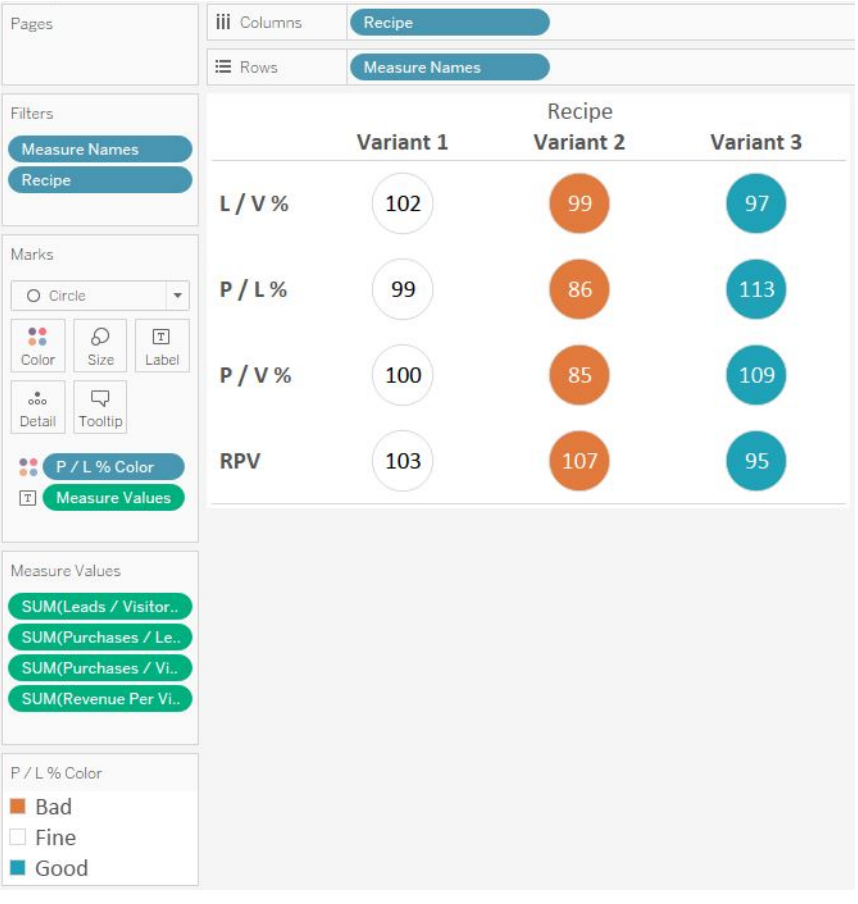
We will be focusing on the 100-point index table in the middle. The table has a column for each variant recipe in a split test and rows for common web KPIs such as Leads/Visits Rate, Purchases/Leads Rate, Purchases/Visits Rate, and Revenue Per Visit. The label on each circle is a 100-point index score with a score of 100 indicating the variant performed the same as the control recipe; scores above 100 indicate higher performance; scores below 100 indicate lower performance.

The special aspect of this table is that the circles are colored by whether or not the variant recipes performed differently than the control recipe with statistical significance. There is a parameter at the top that allows the end users to choose their minimum confidence level, then statistical significance is calculated for each recipe and each KPI.

Normally to create a table with different measures, you would use the generated fields for Measure Names and Measure Values. Here's how this table might normally be constructed (control recipe is filtered out for consistency):



So far so good, but the issue presents itself when I go to conditionally color the circles for statistical significance. With this default setup, I can only color the circles by one field at a time. Further, every row will be colored by that same field. So which one do I pick? Here's what the table looks like if I color the table by statistical significance for P/L % (Purchases/Leads Rate):

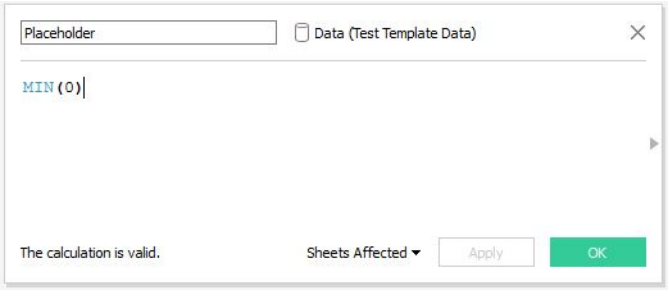


The color is correct for the P/L% row, but because the Color Marks Card is coloring every mark on the view by the same thing, all four KPI rows receive the same color treatment. What I really want is to conditionally format the KPI rows independently of each other so that the color indicators are applied only to their respective KPI.

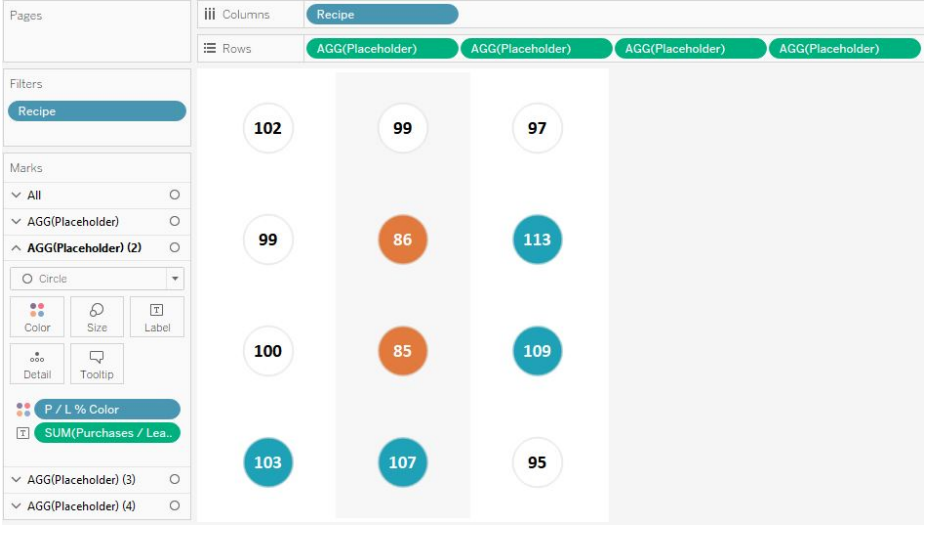
The Solution: A Calculated "Placeholder" Field

The trick for conditionally formatting the rows independently of each other is to create separate Marks Shelves for each row. This is achieved through a simple calculated field that looks like this:

MIN(0)



Now if the table is built with a Placeholder for each of the four rows instead of the Measure Names field, the rows can be modified independently of each other. Here's how my final table looks under the hood:



Notice that there are now four different sets of Marks Cards, one for each Placeholder field, or in this case, each KPI. Since there are independent Marks Cards, you can independently modify each row. This way, the P/L% row can be colored by the statistical significance for P/L%, while the row for L/V% can be colored by the statistical significance for L/V%, and so on!

Note that this example modified rows, but the same exact approach can be used for columns.

The one big drawback to this approach is that you lose the row headers, but it is fairly easy to add text boxes for each row on the dashboard as pictured here.

This is just one example of applying conditional formatting in Tableau like you can in Excel, but you can format the rows with any of the options available to you on the Marks Cards. For example, one row could be just text, while the other rows could be circles. Some rows can have different shapes. You could have different sized marks to highlight a key finding. The list goes on and on…

CHAPTER 58

Five Tips for Creating Efficient Workbooks

One of the most common questions I receive from new Tableau users is, "How much data can Tableau handle?" Further, I commonly undertake Tableau engagements with the exclusive goal of making workbooks run more efficiently. Tableau is capable of handling extremely large datasets, and the software only becomes more powerful with each new release.

That being said, providing an answer on how much data Tableau can handle is a tricky question, and that's because "big data" is a relative term. Whenever a new field is placed on a view, Tableau queries the underlying data to visualize the answer. This works much like when a database query language like SQL is used to ask questions of a database. For this reason, the efficiency can depend on many factors, including not only the number of records, but the processing power of the hardware, the complexity of calculations, the type of data, and so on.

I won't go deep into technical details and optimizations in this chapter, but give five tips anyone can use to make Tableau workbooks run more efficiently. While Tableau likely won't crash on you when dealing with a large dataset, the time it takes your visualization to load should be considered as a measure of good user experience.

Credit: It's hard to pick just five Tableau efficiency tips, but the following was prioritized in part because of this great Tableau post: "6 Tips to Make Your Dashboards More Performant" by Nicholas Hara.

Five Tips for Creating Efficient Workbooks in Tableau

Tip #1: Think Strategically About the Data You Absolutely Need

By far the most powerful efficiency gains that I've witnessed were when the size of the dataset was reduced by removing irrelevant data from the file. This may seem obvi-

ous, but I can't tell you how many times I've seen authors attempting to visualize more data than they need. For example, if the company-wide standard for your analyses is year over year, don't even bring in the data from three years ago. In this scenario, you immediately remove at least one-third of the data before you've even started!

Another piece of low-hanging fruit I often see is the aggregation of dates. For example, your company may be collecting data at the timestamp level, but if your analyses do not require hour/minute/second level views, you can aggregate the dates by day. This will significantly reduce the number of records.

These are just two common examples, but bringing in only the data you need is the first step to creating efficient workbooks in Tableau.

Prepare data before it gets to Tableau

In Chapter 79, I mention that I prefer to prepare my data before it gets to Tableau. While Tableau has many powerful data preparation features, they are not the primary value of the software, and there are better tools for this specific job. I recommend using a tool outside of Tableau to prepare the dataset, so that once you start using it in Tableau, Tableau can do what it's best at.

Use context filters

I've combined the "first" tip into three parts because preparing and limiting the size of the data are that critical to the efficiency of your Tableau workbooks. Another way to limit the data that Tableau is visualizing is to leverage "context filters." *Context filters are processed before anything else and can be thought of as temporary tables for your view*. When a context filter is used, Tableau creates a subset of the dataset limited to the filter selection; then all subsequent filters hit only the subset of the data. Any dimension filter can be used as a context filter by right-clicking the filter from the Filters Shelf and choosing "Add to context."

Tip #2: Limit Filters; Use the "Apply" Button

When you think about filters as queries on the data, it makes logical sense that each incremental filter on a view will add processing time. While I don't have a hard-and-fast rule for how many filters are acceptable, prioritizing which fields should be available as a filter can dramatically improve the efficiency of your views. If fields should be filtered more permanently, consider tip #1 and either filter the data before it gets to Tableau or add the filter to context to speed up processing.

One way to reduce filters on a dashboard is to use dashboard actions instead. Using a sheet as a filter or adding a filter dashboard action that runs on hover or select provides a more efficient means for filtering the rest of the dashboard.

If you *are* using filters, a handy feature of Tableau is that dimension filters that are being shown to end users on the view can be set to process only after all changes to the filter have been selected. For example, if you are showing a dimension filter that includes 10 dimension members, then by default, Tableau will apply the filter every time a dimension member is checked or unchecked on the filter. So if you uncheck one dimension member, the view is reprocessed; uncheck a second dimension member and the view is reprocessed again. If you would prefer that the view only process after both dimension members have been unchecked in this scenario, click the drop-down arrow on the filter, hover over Customize, and select "Show Apply button." Now the filters will be processed only after the end user has made her filter selections and then clicked the Apply button.

Tip #3: Reduce the Number of Marks

When you think of each data point, or mark, as a record that needs to be processed on a view, this tip also makes logical sense: the more marks that need to be processed, the longer it may take for the visualization to appear. The efficiency tips to this point should have a big impact on the number of marks you are left to work with, but this is still good to keep in mind. Depending on your analysis requirements, it may not always be possible to reduce the number of marks on a view, but sometimes there is an opportunity to change the level of detail to improve efficiency. Consider ways you can aggregate data points into hierarchies and/or make the analysis less granular.

Tip #4: ~~Boolean~~ ~~Integer~~ ~~Float~~ ~~Date~~ ~~Date Time~~ ~~String~~

One of the most valuable features of Tableau is its ability to execute custom calculations on the fly. This allows for "discovery" analytics, where you can quickly pivot through different analyses without even necessarily knowing exactly what you are looking for. While the calculations are very powerful, they can come with a cost to the efficiency of the workbook. To help get the most out of the calculated fields functionality and keep your workbooks running smoothly, think about the data types in your calculations. Not all data types are created equal in terms of efficiency; here is a list of data types going in order from most efficient to least efficient, with a short definition:

Boolean
: A true or false binary result

Integer
: Whole numbers

Float
: Any number including decimals

Date
: Date aggregated at the day level

Date Time
: Date including the timestamp level

String
: Text

When possible, use more-efficient data types. Rather than a column with two possible text outcomes, convert it to a Boolean that has the outcomes of True or False. If you have a large dataset of numbers, consider whether it's possible to throw out decimals and deal only with whole numbers as integers. If you have dates with timestamps, consider aggregating them at the date level and leave out the timestamps.

Tip #5: Reduce Sheets, Dashboards, and Data Sources

This tip not only helps with efficiency, it will help you keep your sanity and improve the end user experience. In general, the more sheets, dashboards, and data sources you have in a workbook, the more potential there is for the workbook to run slowly. This is especially true when you are combining many different sheets on a dashboard with views that are blending data from multiple data sources.

As with database architecture, your Tableau workbooks will run more efficiently if you put some thought into how workbooks can be broken down into smaller, individual files. Not only will this help with efficiency, it will be easier for you to manage and help your end users. This is even easier and more efficient if you have Tableau Server or Tableau Online because several smaller workbooks can share the same data source saved to the cloud.

In the case that I truly have several dashboards that are connected, I like to create a navigation dashboard that helps the end user locate the most relevant views for their specific business questions. If you have Tableau Server or Tableau Online, the navigation links can be set up to open new dashboards by simply adding URL dashboard actions with links to the dashboard locations online. This same technique can be used from within specific dashboards (i.e., add a URL action to run on Menu that links the end user to another dashboard/additional information).

While this isn't a comprehensive list of Tableau efficiency tips, I have found that implementing these tactics follows (at least) the 80/20 rule, in that these five tips alone should help capture at least 80 percent of the possible efficiency gains in your workbooks.