**How to make gauges with custom shape palettes in Tableau**

By the end of this post, you will be able to make indicators using custom shapes of your choosing that change based on performance.



In the example, the indicators turn to a green smiley face if the month over month change for a KPI was positive, a yellow neutral face if the month over month change was negative but within 10%, and a red upset face if the month over month change was worse than 10%. You can use any images and performance thresholds that you want.

Once you have all the images you want to use in your visualization, you have to place the files in the Shapes folder within the Tableau Repository on your computer. This can be found at Documents > My Tableau Repository > Shapes. Each folder within the Shapes folder corresponds with a shapes palette within Tableau Desktop that can be used to map shapes to dimension members.

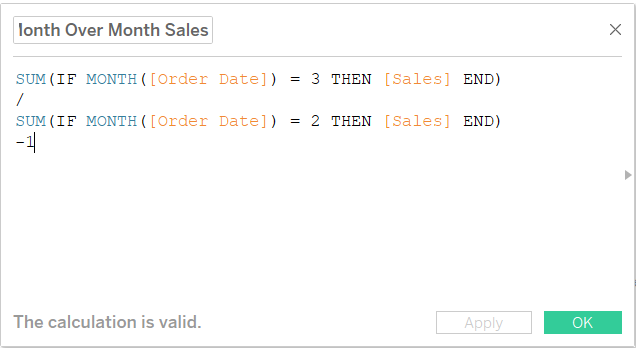
We’re now ready to open Tableau and build the indicator gauges. To create gauges that represent month over month performance or performance compared to a goal, you need two calculated fields for each measure: the first to compute the comparison, and the second to compute the ‘Good’ / ‘Okay’ / ‘Bad’ thresholds (or whatever you are using).

For ease of illustration, we will pretend we’re comparing entire months to each other across all four years of data in the Sample – Superstore dataset, and we want to compute the percent change between the month of March and February. You will need to adjust this calculation to include your two comparison points, but the formula for percent change is:

([Current Performance] / [Comparison Performance])-1

If I’m comparing March to February, the formula is:

SUM(IF MONTH([Order Date]) = 3 THEN [Sales] END)  
/  
SUM(IF MONTH([Order Date]) = 2 THEN [Sales] END)  
-1



For the second calculated field, we will use IF / THEN logic to create three different scores that will eventually be mapped to the custom shapes. We’ll say that if the current performance outperformed the comparison, we’ll call that “Good”; under-performance within 10% will be described as “Okay”, and under-performance of greater than 10% is “Bad”. The formula is:

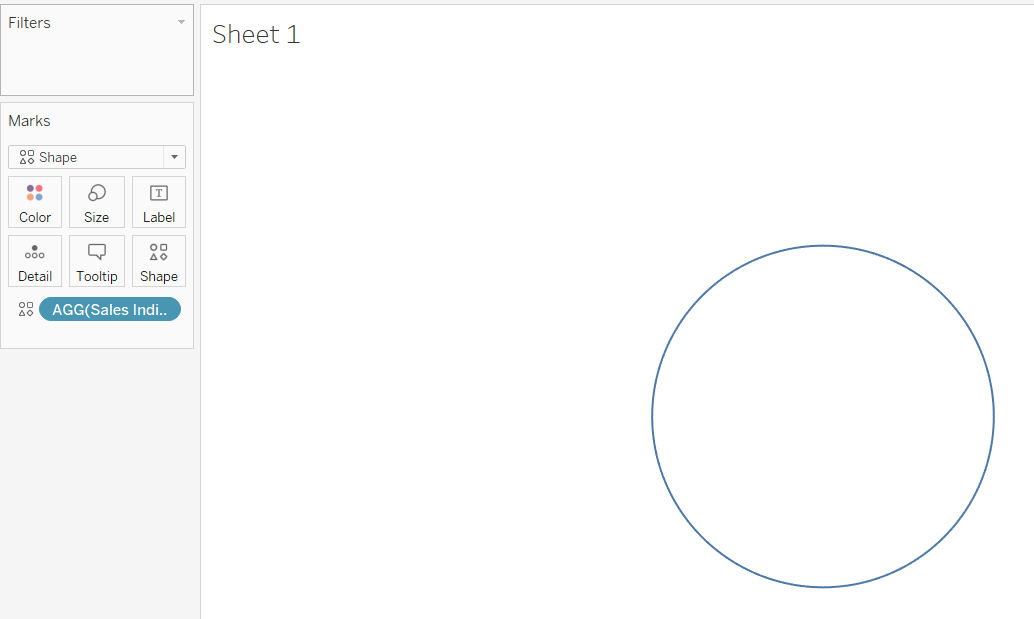
IF [Month Over Month Change] > 0 THEN "Good"

ELSEIF [Month Over Month Change] > -.1 THEN "Okay"

ELSE "Bad"

END

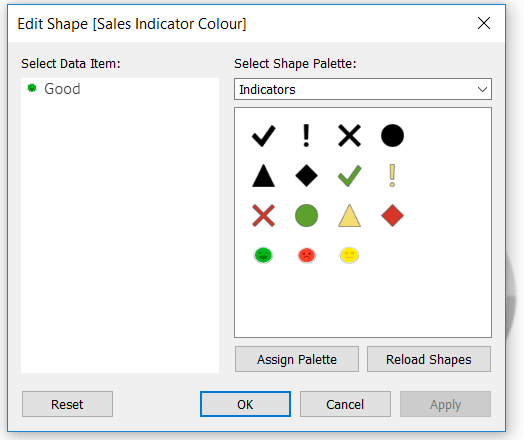
To make the gauge, start a new worksheet, change the mark type to Shape, and put the calculated field that is computing the performance thresholds onto the Shape Marks Card.



In this example, March has outperformed February so the indicator has been classified as “Good”. So far so good, but now we need to map our custom shapes to each of the three performance scores. The first one is easy because it is already on the view. Just click on the Shape Marks Card to access your available shape palettes and use the dropdown to find the custom shape palette installed in your Tableau Repository.

If you installed the custom palette while Tableau was open, you will need to click the “Reload Shapes” button in the bottom right-corner of this interface before the new palette will appear.

To map a performance score to a specific shape, ensure the dimension member is selected from within this interface, then click on the shape you want assigned to the dimension member.



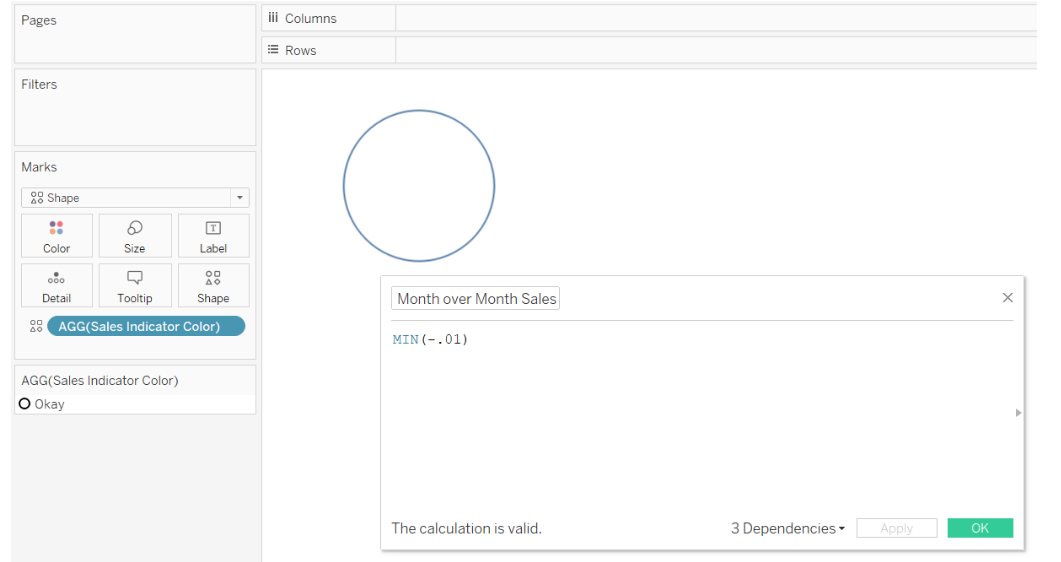


After clicking the “OK” button, we have my first indicator mapped. We will see a green smiley face any time my current performance outperformed the comparison.

We have only mapped one of the three outcomes so should the performance change to Okay or Bad in the future, at this point we would only see default shapes. So how do you map the other two outcomes if they’re not on the view? You could manipulate the dates in the month over month calculated field to attempt to find months that match the other scoring thresholds, but we have an easier way.

To map discrete scoring scenarios (like we have here) that are not on the view or have yet to appear in your data, temporarily hardcode numbers that match the unseen scenarios in the underlying calculated field. We can do this by using the aggregation MIN and typing in numbers that match the additional outcomes. For example, to create the “Okay” scenario, I will edit the calculated field that is computing the comparison, and overwrite what is there with “MIN(-.01)”. Just make sure you copy and paste the original formula back into the calculated field after we have mapped all the outcomes.

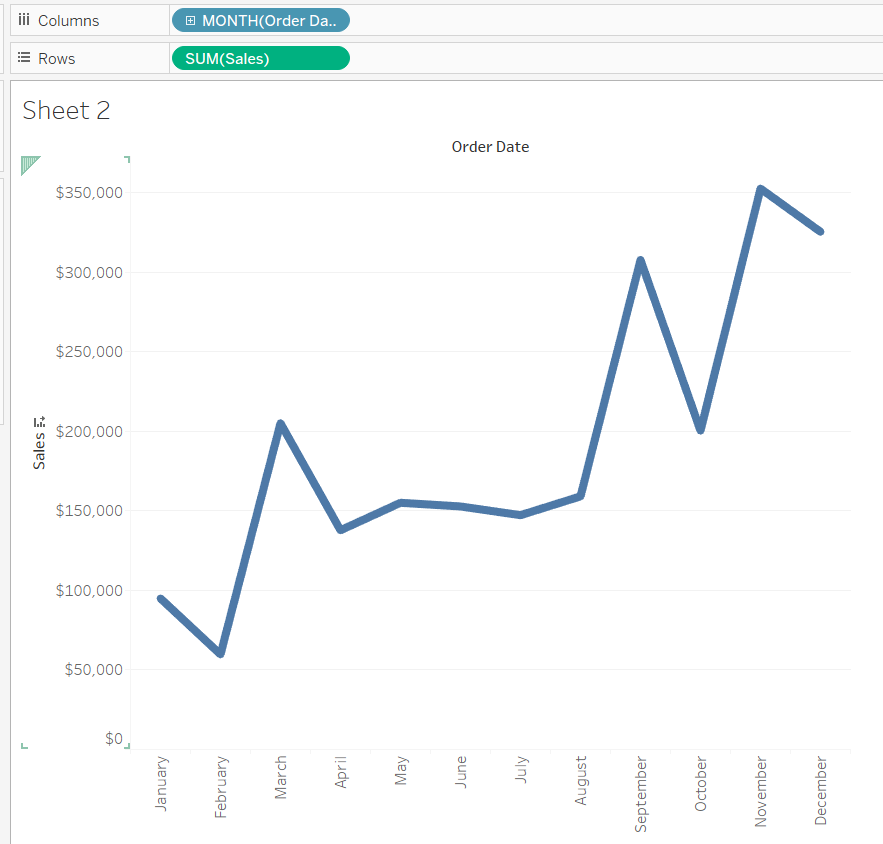
-.01 is an under-performance of 1%, which in this example, results in a scoring outcome of “Okay”. I can now map the neutral yellow face to this scenario.



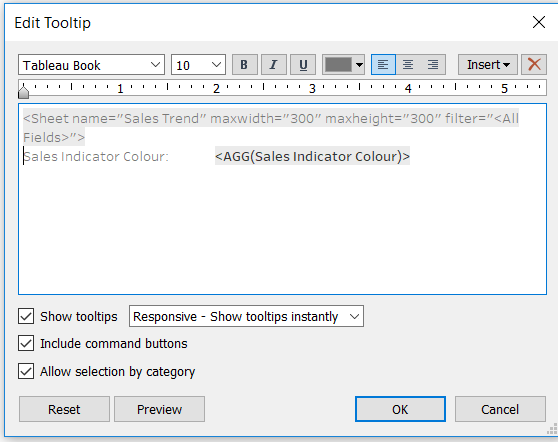
Repeat this step for as many outcomes as you have, then paste your original formula back into the calculated field. The mapping now works for every scenario, even if it hasn’t appeared in your data yet!

#### Add value with Tableau’s viz in tooltip feature and/or dashboard actions

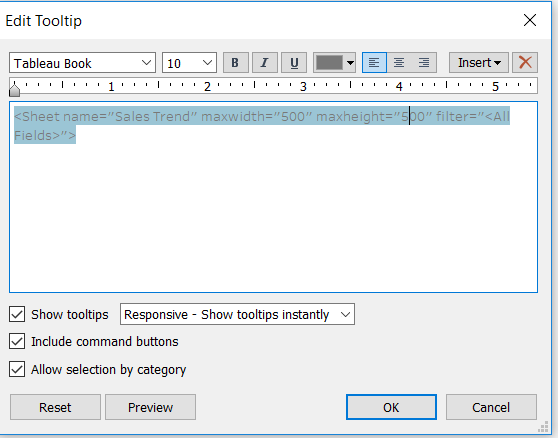
This type of gauge is extremely descriptive in nature, providing only a very high-level overview on which areas of the business are healthy or need attention. One way to add value to this custom shape gauge is to use [Tableau’s viz in tooltip feature](https://www.ryansleeper.com/how-to-add-an-image-to-a-tableau-tooltip/). To display a trend when the end user hovers over one of my custom indicators, we will first make a trend that looks at Sales by Month of Order Date.



This graph can be added to the tooltip of the Sales indicator by navigating back to the sheet for the indicator, clicking on the Tooltip Marks Card, clicking the “Insert” button in the top right-corner of the Tooltip interface, and hovering over Sheets.



After choosing the Sales Trend sheet, it will appear in my tooltip. You may have to adjust the height and width defaults to ensure the graph does not get cut off when being viewed within the tooltip. Here are the settings after changing the height and width to 500 pixels each.



Of course, you can customize this viz in tooltip with whatever is relevant for your business and/or add additional fields to the tooltip such as current performance and comparison performance.