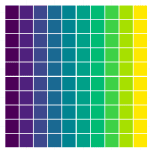
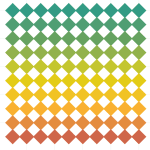
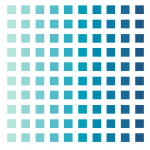
**Waffles are for Breakfast**

It’s been a long time since my last update and I’ve decided to start with Tableau, of all topics! Although open source advocates do not look kindly upon Tableau, I find myself using it frequently and relearning all the stuff I can do in R. For my series of “how-to’s” regarding Tableau, I’d like to start with posting about **how to make a waffle chart in Tableau**.

Without further ado:

**What’s a Waffle Chart?**



There are a gazillion ways to depict an amount. Bar charts and pie charts are common, maybe you’ve even seen an area/tree chart. But variety is the spice of life – repeatedly looking at bar charts becomes tiring. Enter the waffle chart – It has the appearance of a waffle iron. It’s basically a grid of squares (or any other mark). Squares can be colored to represent an amount.

In Tableau, a waffle chart is NOT one of the off-the-shelf charts that you can make by clicking the “Show Me” panel. If you want to create one, you’ll need to be, err, creative.

**Setting Up**

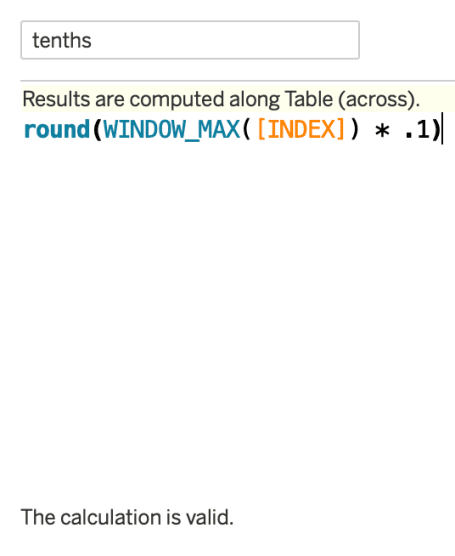
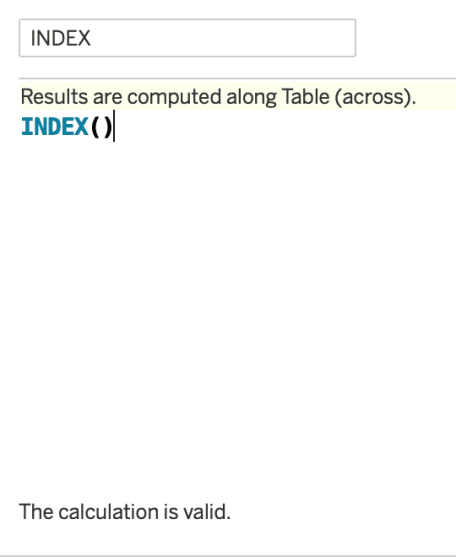
For this tutorial I will use the famous iris dataset. You can find it *anywhere*, but it is primarily hosted at [https://archive.ics.uci.edu/ml/datasets/iris](http://jjohn987.rbind.io/post/tableau-creating-a-waffle-chart/UCI%20machine%20learning%20repository). Download it and get it into Tableau. **Important** – Make sure that your spreadsheet software creates an “ID” variable for each row (a row number for each row), and read it as a dimension.

**Some things to figure out:**

1. How to make a grid with your data (conventionally, a 10 x 10).
2. How to assign color to the squares/marks in that grid.
3. The calculation that will be used to create the color assignment.

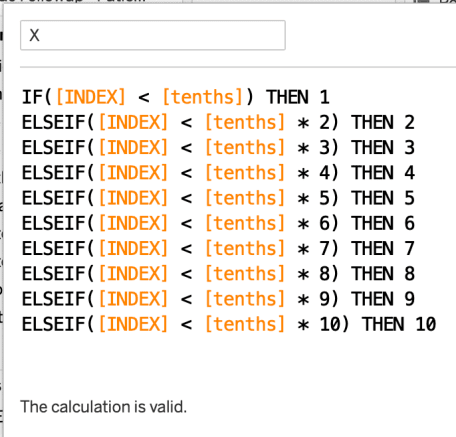
Of course, there’s always more than one way to do things in Tableau. I’ll show my unique way of building a waffle chart which serves my data needs quite well.

We start by creating an index. To do so, all you have to do is make a calculated field with INDEX() in it.

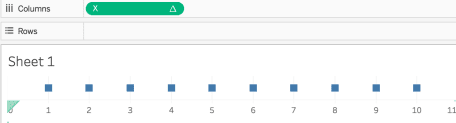


Next up, use INDEX to partition the whole dataset into 10 rows to build out the x-axis. So let’s figure out how many rows are in a “tenth” of the data – divide the maximum value of the index by 10 or multiply by 0.1. **Note** it is important to use ROUND to square off the value. If not, you will have rows of unequal lengths later.

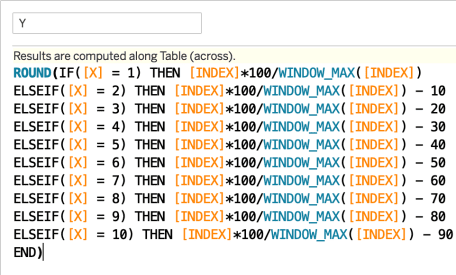
To make the X-axis from here, I simply assign values based off of where their index stands in relation to each tenth percentile.



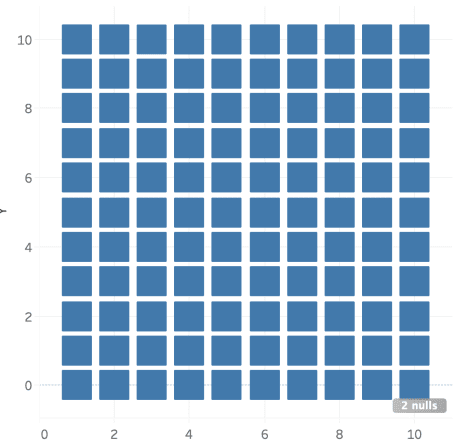
At this point, you place your newly created X variable on the rows shelf and choose to “compute using \_\_\_\_” (your dimension/row-level ID variable), and doing so you can see a neat little row of marks as such:



Now we need to expand this single row into 10 Y-values. If you guessed that we will now partition the X-values into 10 different rows, you are right. To do this I multiply the index value by 100 and divide by the maximum index value. This is exactly what it sounds like – a percentage. Note that rounding the values to the nearest integer is very important. Without doing so we won’t achieve a neatly separated 10 x 10 grid but a continuous line of marks instead.

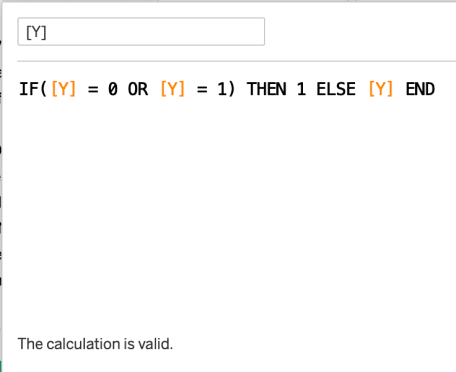
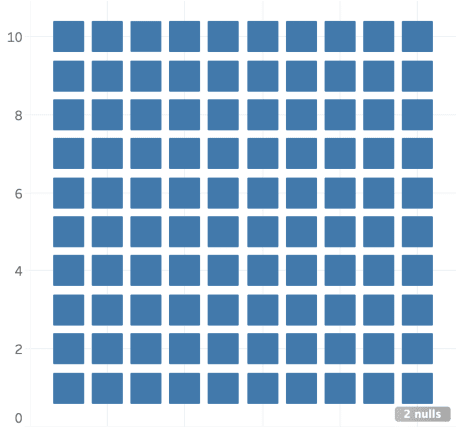


The result is not a beautiful 10×10 grid, but a beautiful 11 x 11 grid.



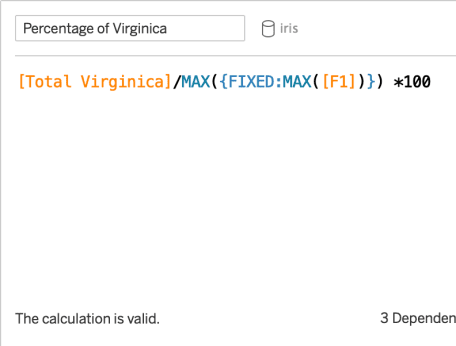
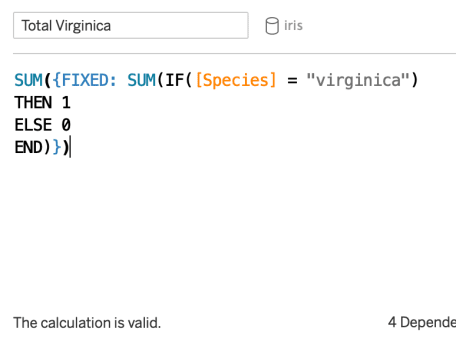
This is because of the rounding we did earlier – for each tenth percentile partiion of X, the bottom portion of the Y values were closer to 0 than 1, which means they were rounded down. Other tutorials may have a workaround, however, they seem to employ other means which require their own unique workarounds as well. At the end of the day, you’re probably going to need a workaround. Here is mine:

Simply fold the 0 and 1 Y values into each other with another calculated field, as such.

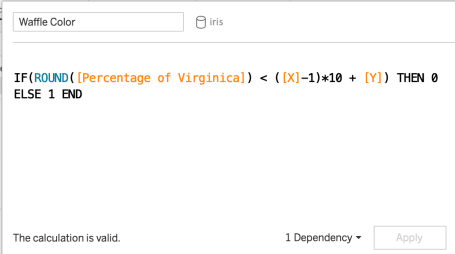
**And we get ====>**

Voila! Now we have our “template” for a waffle chart! Not that it was necessary to fold all the values of our data into the chart.

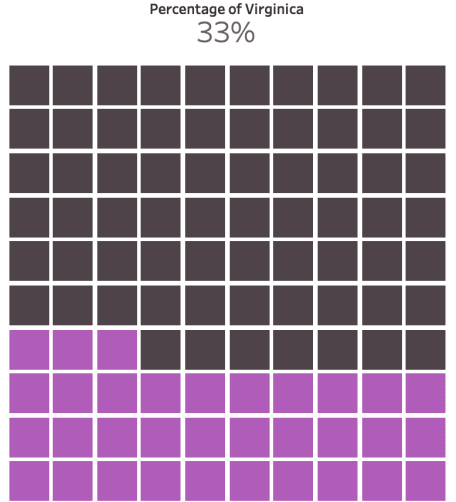
**But we can’t eat waffles yet** – we need to assign color to it first. I suppose I’ll color the waffle chart based off of the percentage of Virginica species in the Iris dataset. Notice that I need to use fixed LOD calculations for this. Since the “View” influences calculations, it is best to work with an LOD which will compute the calculation first in spite of whatever is happening in the view.



Now we are ready to assign a color using the percentage variable. The logic is profoundly simple once you figure out the concept. Essentially, you need to tell Tableau to color a mark based off of the values it is less than. That means, multiplying X values by 10, Y values by 1, and starting by looking at the previous row. This calculation handles all possibilities, except of course doing partial squares.



Presto! Now we can finalize the beautiful waffle chart!



33% of the flowers in the iris dataset belong to the virginica species.

**The End – Enjoy your Waffles!**