Lab 9. Tableau Advanced Visualization

In this lab, we will cover the following recipes:

- Lollipop charts
- Sankey diagrams
- Marimekko charts
- Hex-Tile maps
- Waffle charts

Technical requirements

We will use Tableau 2019.x, and data about mass transit complaints, beer, education, commuter times, and Lord of the Rings.

Introduction

In this lab, we will learn more chart types that go beyond `Show Me`. We will use external data sources to provide a plotting frame and, in many cases, use advanced calculations to draw our visualizations. In these recipes, we will show how data elements relate to each other or how they are related to the whole.

Lollipop charts

Lollipop charts get their name from their shape. These charts can offer visual variety. We would typically use lollipops in place of bar charts or dot plots, which also happen to be construction components.

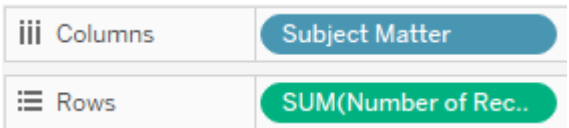
Getting ready

We will learn how to successfully combine a bar chart and a dot plot to create a lollipop chart.

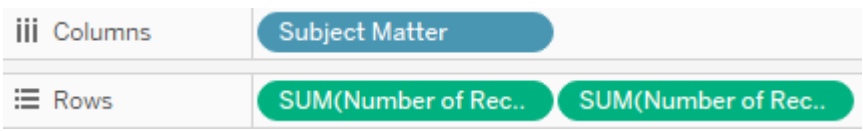
How to do it..

We will open the packaged workbook lollipop to work through the recipe. We will examine the NYC mass transit customer complaints by complaint subject matter; use `MTA_Customer_Feedback_Data_Beginning_2014.csv` and `MTA complaints.twbx`:

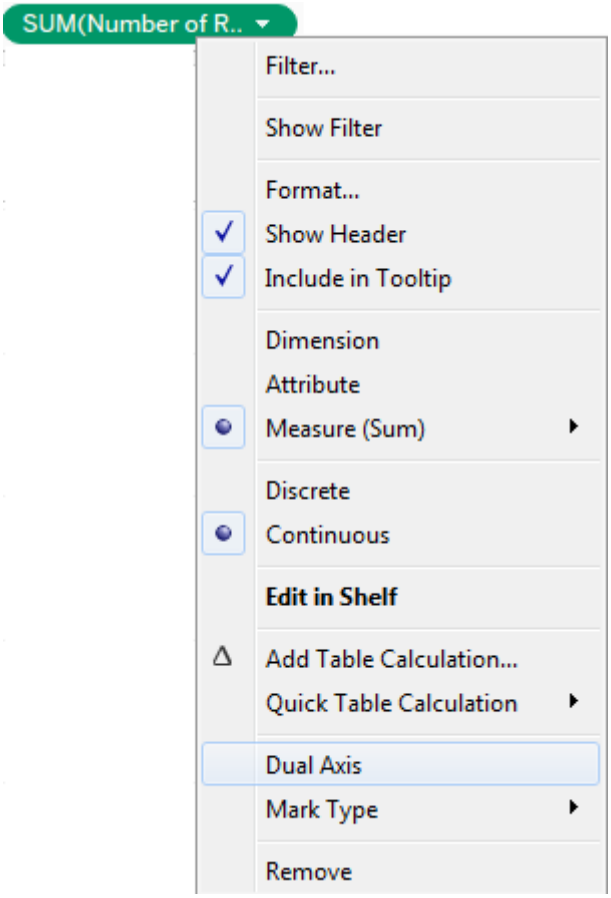
1. Place `Subject Matter` and `Number of Records` on the `Columns` and `Rows` shelves:



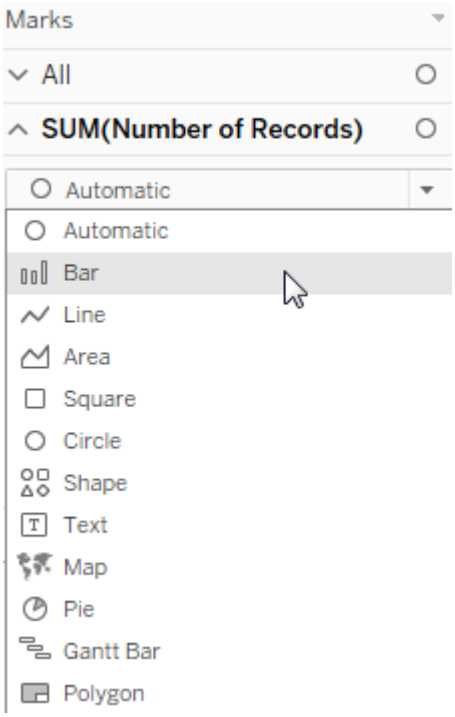
2. Duplicate the `Number of Records` pill on the R`ows` shelf:



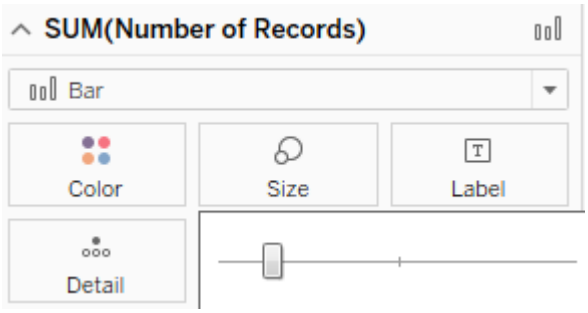
3. Modify the chart to be `Dual Axis`:



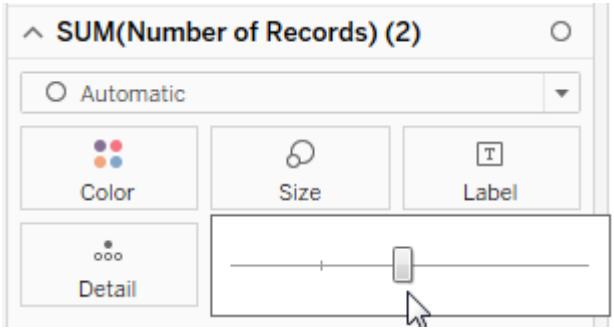
4. Change the mark type of `Sum(Number of Records)` to `Bar`:



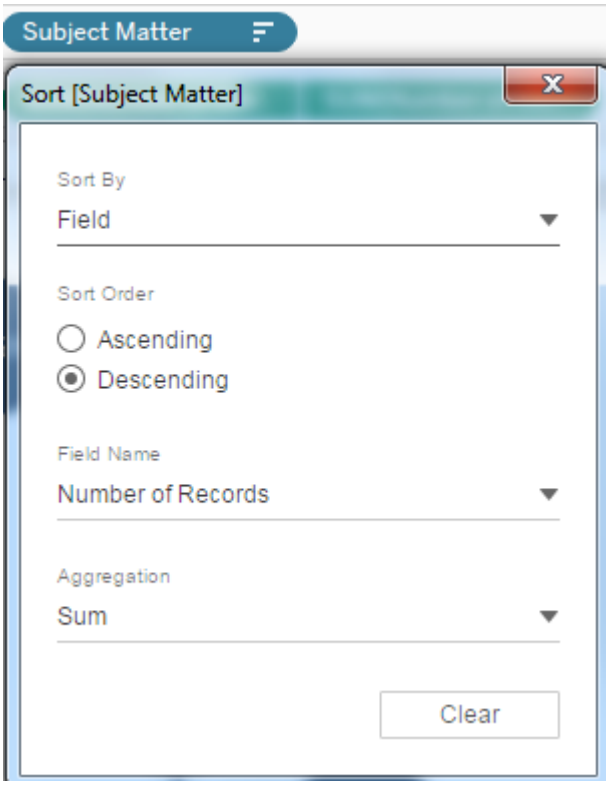
5. Make the `Size` of the bar smaller:



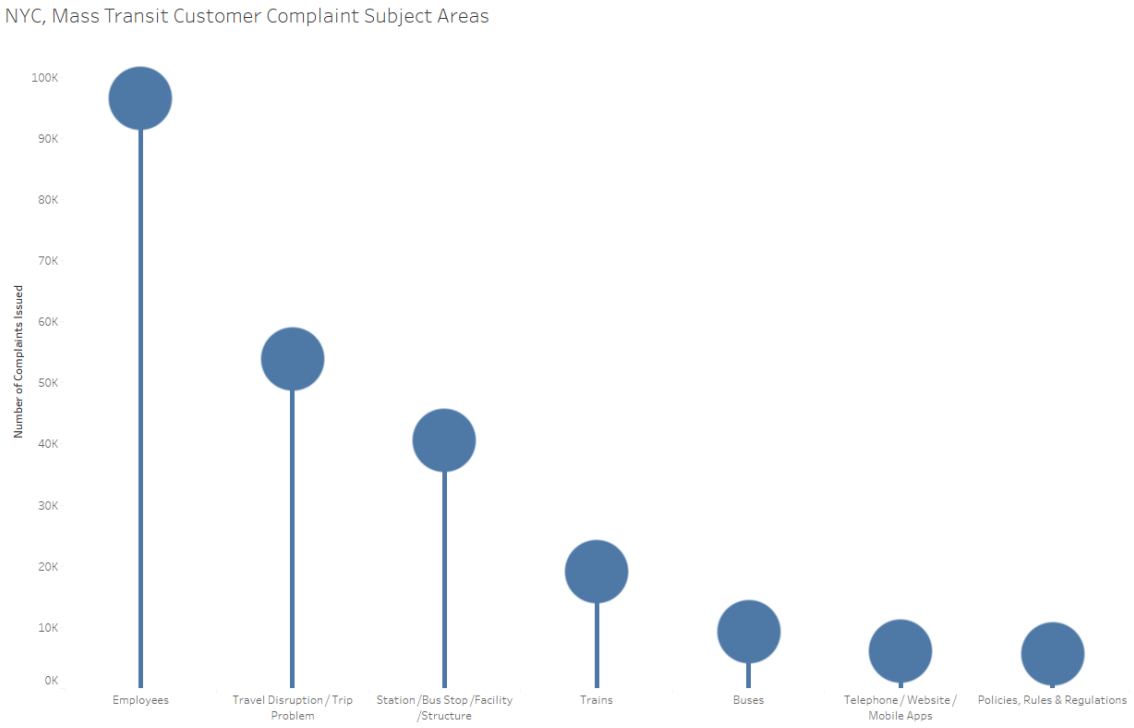
6. Make the `Size` of the circle larger:



7. Sorting `Subject Matter` in descending order by `Number of Records`:



We now have the following example:



Note

Because there were many different `Subject Matter`, we filtered `Subject Matter` to the top seven based on the `Number of Complaints` issued.

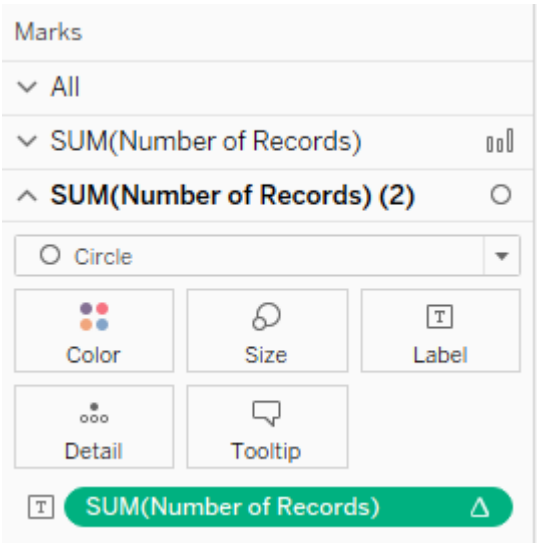
How it works...

We placed the `Subject Matter` on the `Columns` shelf and `Number of Records` on the `Rows` shelf twice. Next, we made this a `Dual Axis` chart. We modified the mark type to be `Bar` and made the size smaller for the first `Number of Records`. We left the mark type for the second `Number of Records` as `Circle` and increased the `Size`. Finally, we combined a skinny bar chart with a dot plot chart.

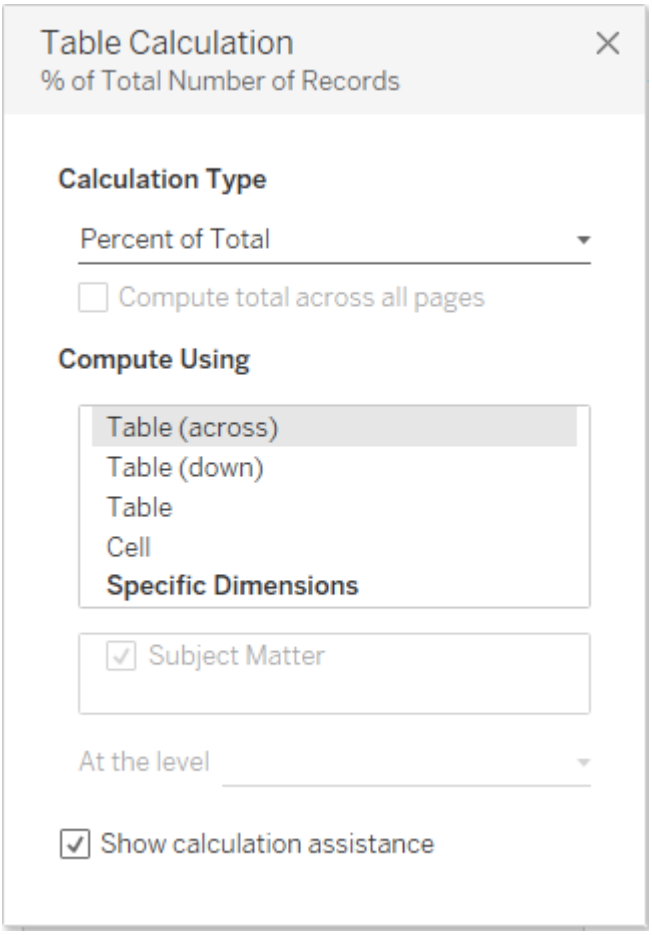
There's more...

We can place a value in the center of the dot plot:

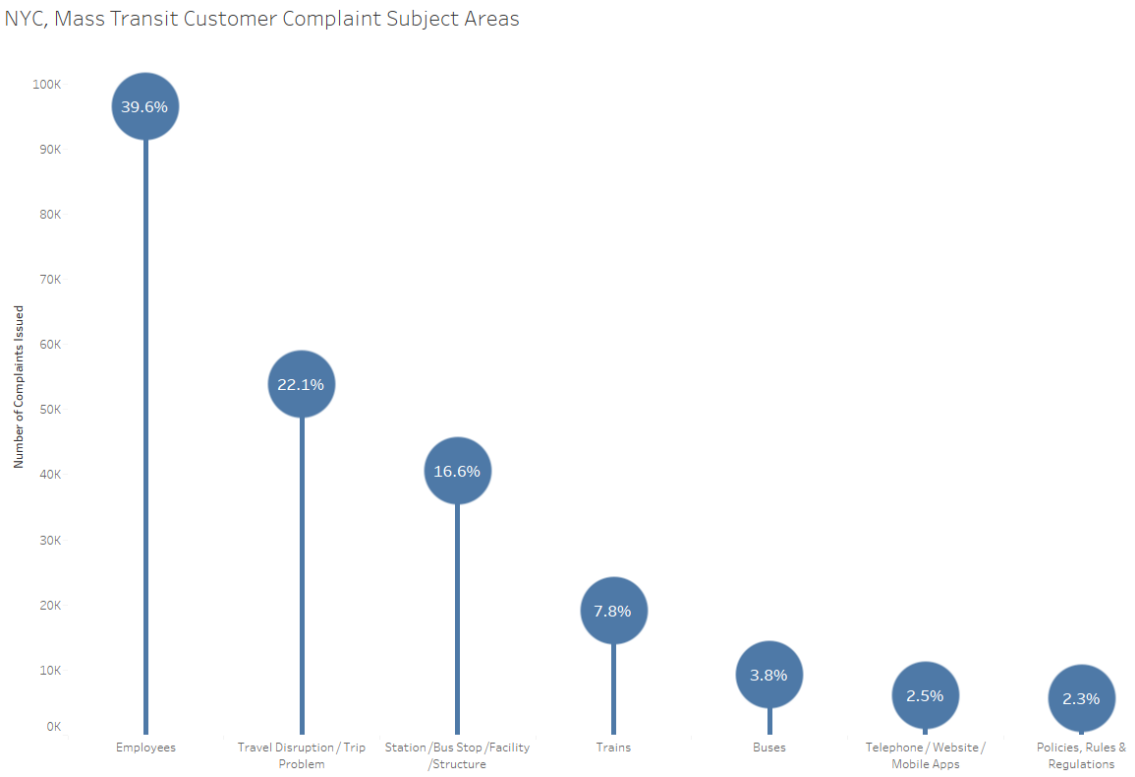
1. Place `Number of Records` onto the `Marks` card to create `Number of Records 2`:



2. Right-click on `SUM(Number of Records)`, choose `Add Table Calculation`, and choose `Percent of Total`:



3. Format the text size, alignment, and color in order to create a similar chart:



See also

To learn more about dot plots, use the following links as a starting point:

- https://uc-r.github.io/cleveland-dot-plots
- https://en.wikipedia.org/wiki/Dot_plot_(statistics)
- http://www.storytellingwithdata.com/blog/2018/8/1/swdchallenge-lets-plot-with-a-dot

Sankey diagrams

A Sankey diagram is a tool used to show the flow from one dimension to another.

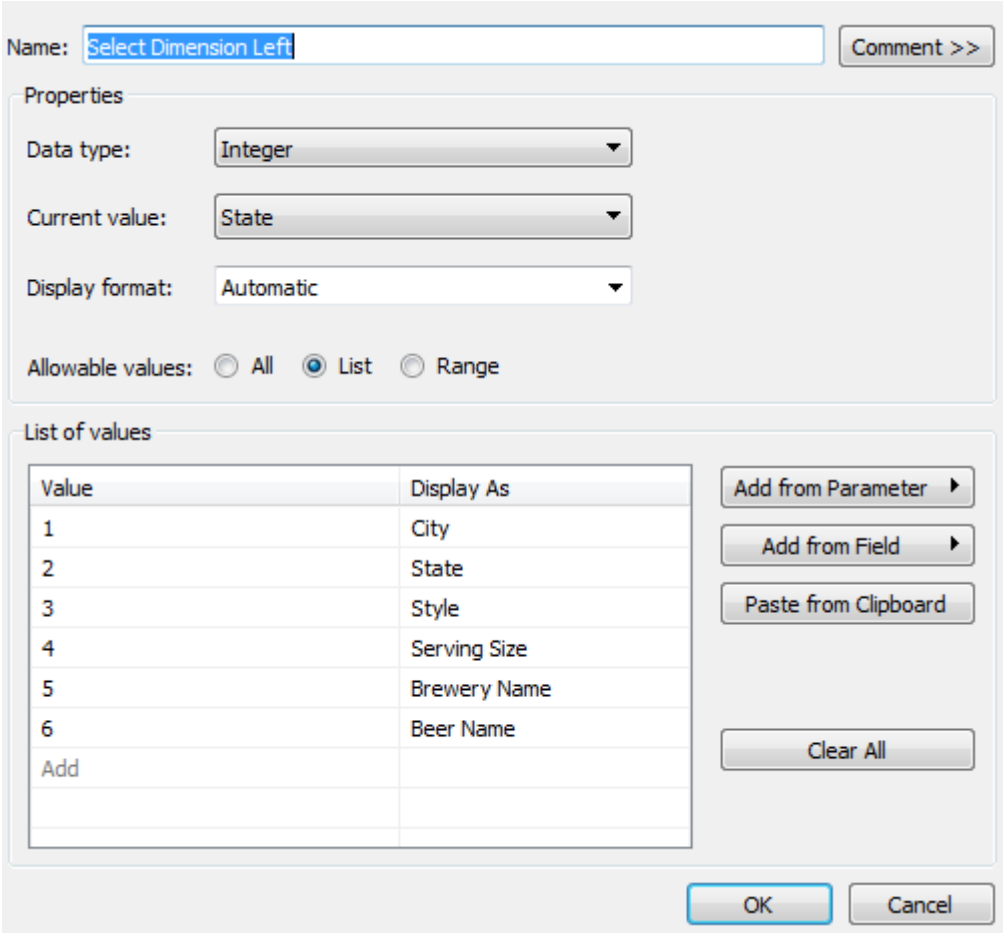
Getting ready

In this recipe, we will create a Sankey diagram using a web extension provided by Infotopics. Web extensions are a new feature in 2018.2.

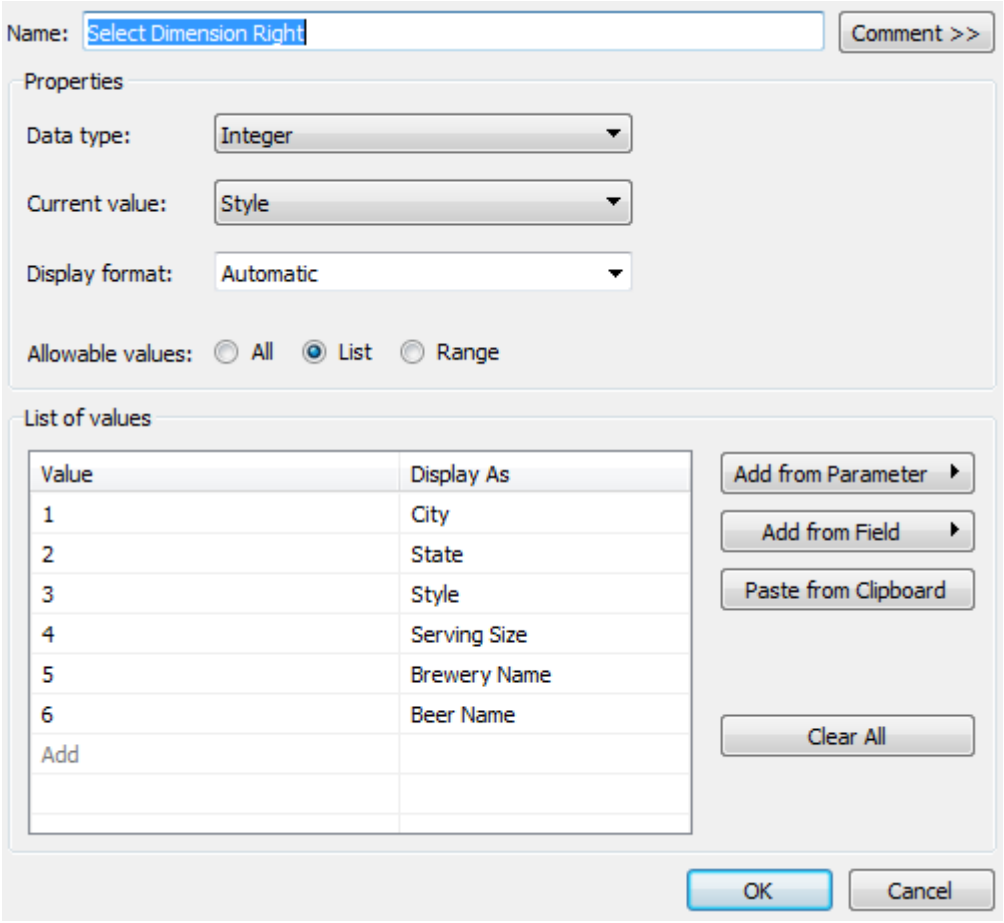
How to do it..

In 2018.2, we can use web extensions to build a Sankey chart. To follow along, use `Beer 2018.2.twbx`, `beers.csv`, and `breweries.csv`.

1. Create the `Select Dimension Left` parameter:



2. Create the `Select Dimension Right` parameter:



3. Create the `Dimension Left` parameter:


```
CASE [Select Dimension Left]
WHEN 1 THEN [City]
WHEN 2 THEN [State]
WHEN 3 THEN [Style]
WHEN 4 THEN [Serving Size]
WHEN 5 THEN [Brewery Name]
WHEN 6 THEN [Beer Name]
END
```

4. Create the `Dimension Right` parameter:

Dimension Right

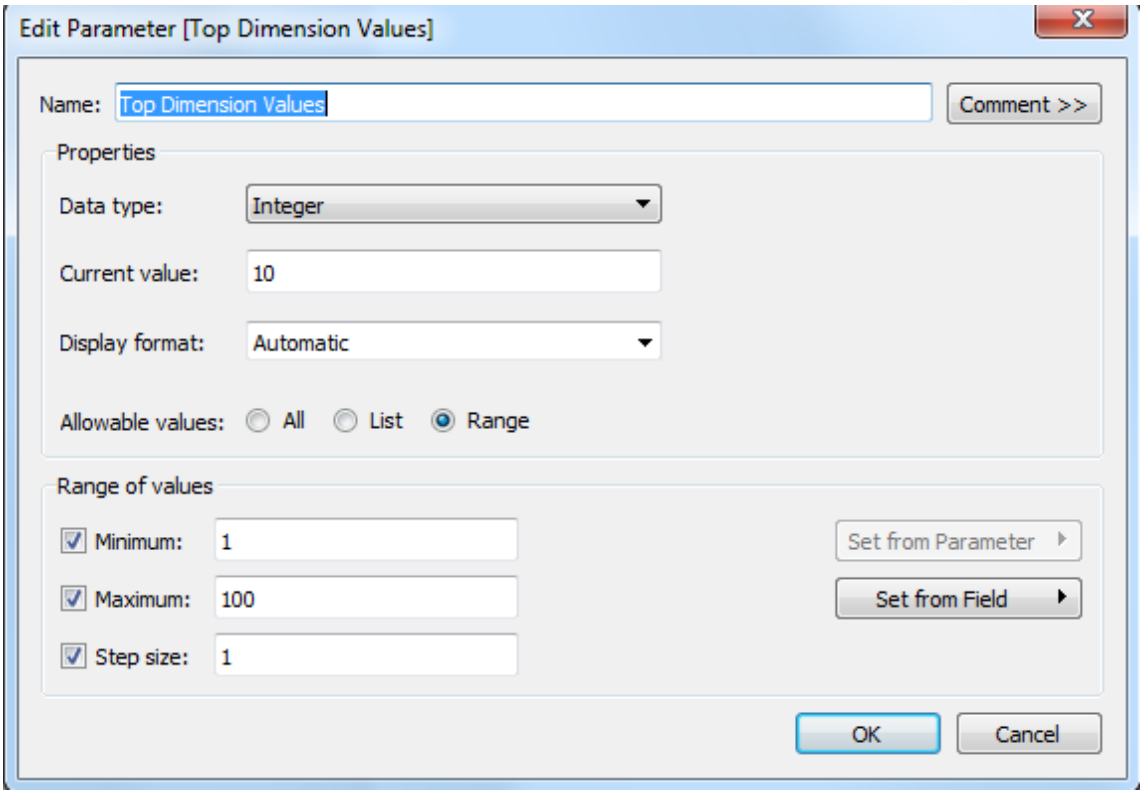
```
CASE [Select Dimension Right]
WHEN 1 THEN [City]
WHEN 2 THEN [State]
WHEN 3 THEN [Style]
WHEN 4 THEN [Serving Size]
WHEN 5 THEN [Brewery Name]
WHEN 6 THEN [Beer Name]
END
```

5. Create the `Chosen Measure` parameter:

Chosen Measure

```
[Fl]
```

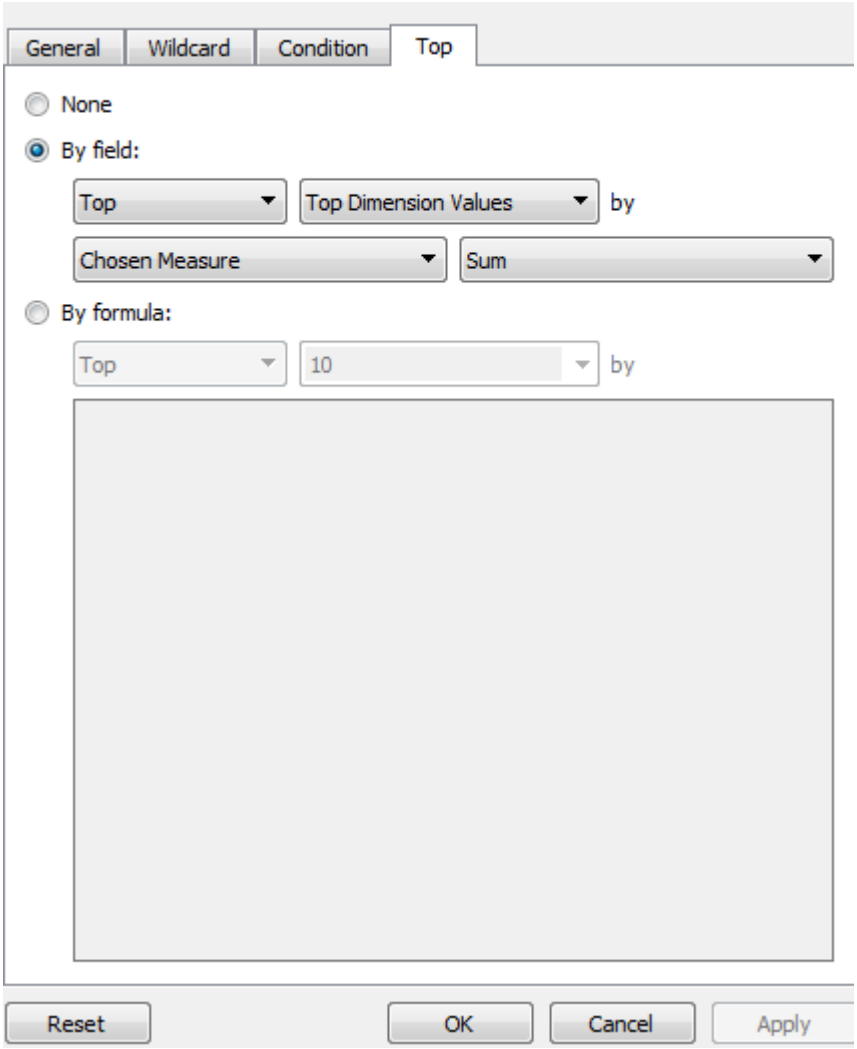
6. Create the `Top Dimension Values` parameter:



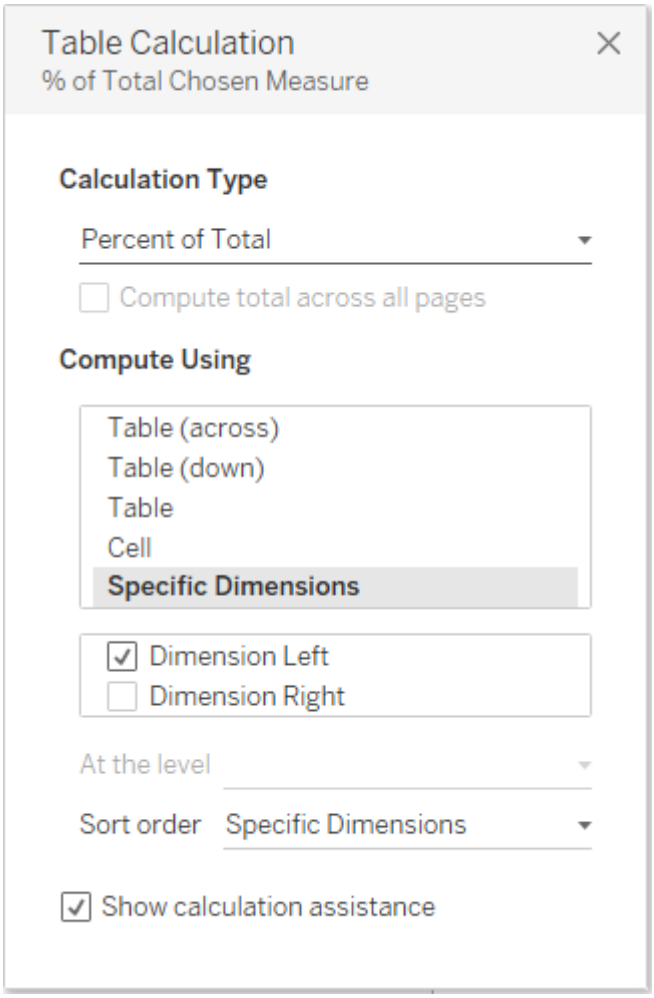
Note

The first six steps are not required, but they give the visualization some fun flexibility. The user can select which categories appear on the left-hand side and the right-hand side of the chart. Also, because there are so many values, which can make the Sankey chart look busy, we've added a top value to filter the dimension values. For this Sankey recipe, we actually only need to create a sheet with two dimensions and one measure.

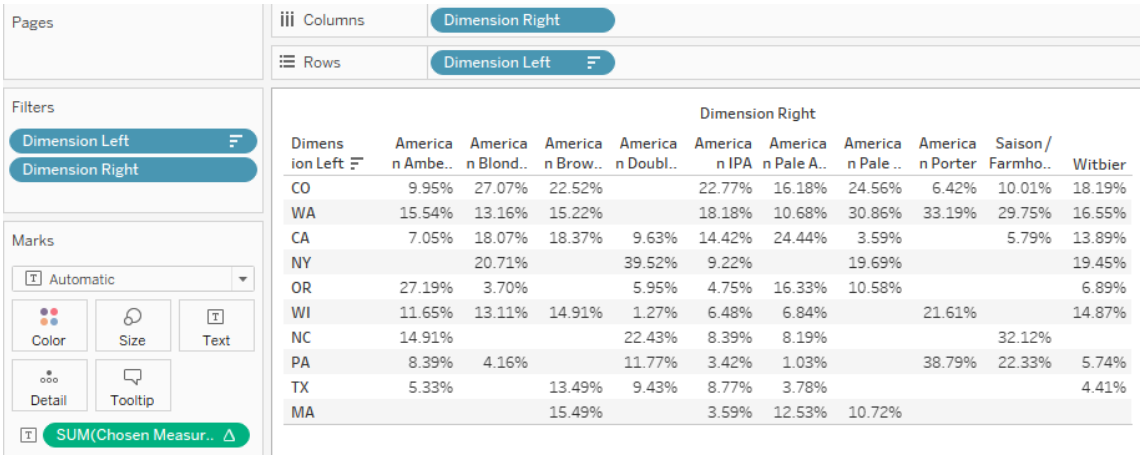
7. Add `Dimension Right` to the `Columns` shelf and the `Filters` shelf.
8. Add `Dimension Left` to the `Rows` shelf and the `Filters` shelf.
9. Right-click `Dimension Right` and `Dimension Left` in the `Filters` shelf, select `Edit Filter`, and set as follows:



10. Add `Chosen Measure` to `Text` in the `Marks` card.
11. Add a `Percent of Total` option in `Table Calculation`:

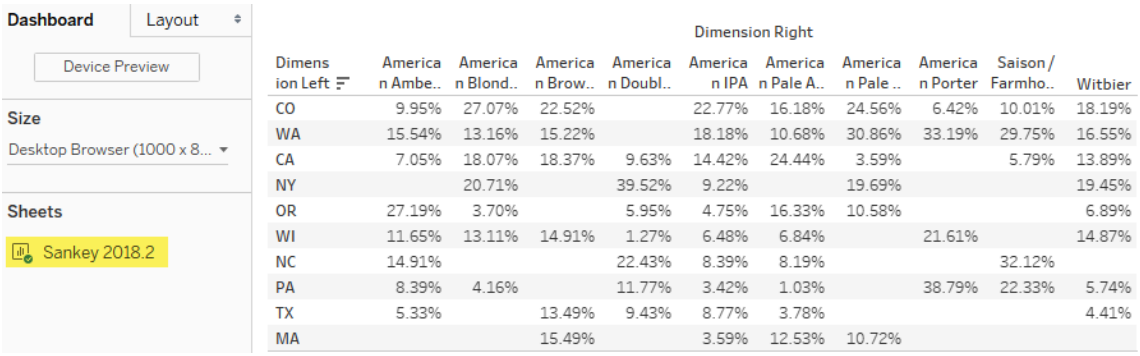


The cross-tab should appear as follows:



	Dimension Right									
Dimension Left	American Ambe..	American Blond..	American Brow..	American Doubl..	American IPA	American Pale A..	American Pale ..	American Porter	Saison / Farmho..	Witbier
CO	9.95%	27.07%	22.52%		22.77%	16.18%	24.56%	6.42%	10.01%	18.19%
WA	15.54%	13.16%	15.22%		18.18%	10.68%	30.86%	33.19%	29.75%	16.55%
CA	7.05%	18.07%	18.37%	9.63%	14.42%	24.44%	3.59%		5.79%	13.89%
NY		20.71%		39.52%	9.22%		19.69%			19.45%
OR	27.19%	3.70%		5.95%	4.75%	16.33%	10.58%			6.89%
WI	11.65%	13.11%	14.91%	1.27%	6.48%	6.84%		21.61%		14.87%
NC	14.91%			22.43%	8.39%	8.19%			32.12%	
PA	8.39%	4.16%		11.77%	3.42%	1.03%		38.79%	22.33%	5.74%
TX	5.33%		13.49%	9.43%	8.77%	3.78%				4.41%
MA			15.49%		3.59%	12.53%	10.72%			

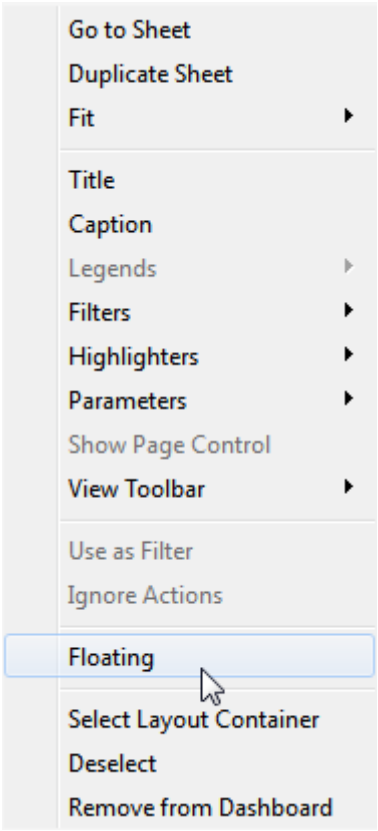
12. Add the sheet to a dashboard, as shown in the following screenshot:



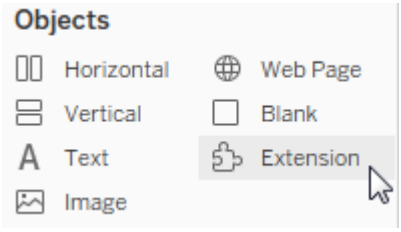
Dimension Right

Dimension Left	American Ambe..	American Blond..	American Brow..	American Doubl..	American IPA	American Pale A..	American Pale ..	American Porter	Saison / Farmho..	Witbier
CO	9.95%	27.07%	22.52%		22.77%	16.18%	24.56%	6.42%	10.01%	18.19%
WA	15.54%	13.16%	15.22%		18.18%	10.68%	30.86%	33.19%	29.75%	16.55%
CA	7.05%	18.07%	18.37%	9.63%	14.42%	24.44%	3.59%		5.79%	13.89%
NY		20.71%		39.52%	9.22%		19.69%			19.45%
OR	27.19%	3.70%		5.95%	4.75%	16.33%	10.58%			6.89%
WI	11.65%	13.11%	14.91%	1.27%	6.48%	6.84%		21.61%		14.87%
NC	14.91%			22.43%	8.39%	8.19%			32.12%	
PA	8.39%	4.16%		11.77%	3.42%	1.03%		38.79%	22.33%	5.74%
TX	5.33%		13.49%	9.43%	8.77%	3.78%				4.41%
MA			15.49%		3.59%	12.53%	10.72%			

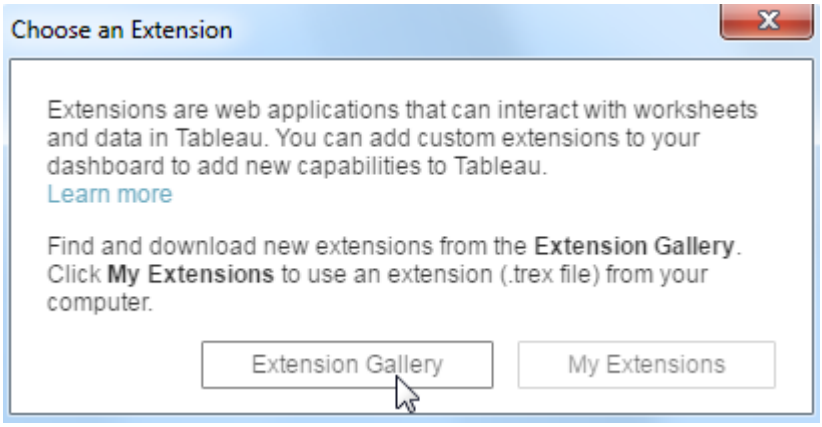
13. Make the sheet `Floa``ting` and then minimize it. In this example, we don't want to see the cross-tab. However, the **`Show Me More`** extension requires it to be on the dashboard in order to work:



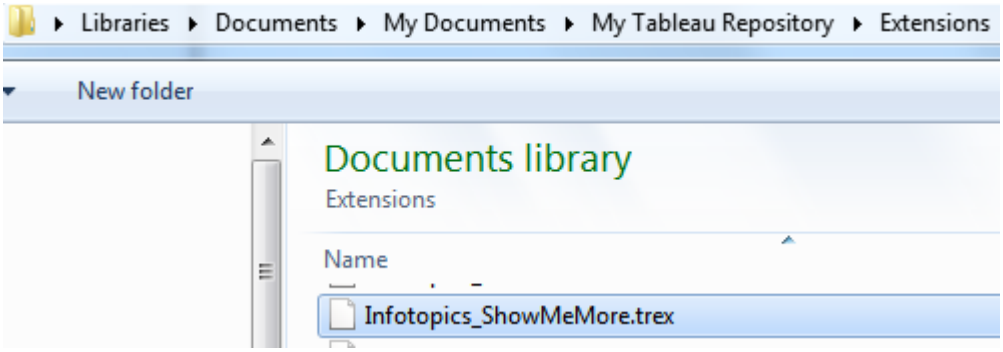
14. Use the **`Extension`** object:



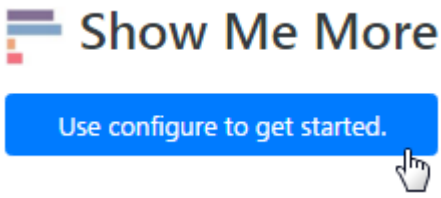
15. When prompted for the first time, download the `Show Me More` web extension by visiting `Extension Gallery`:



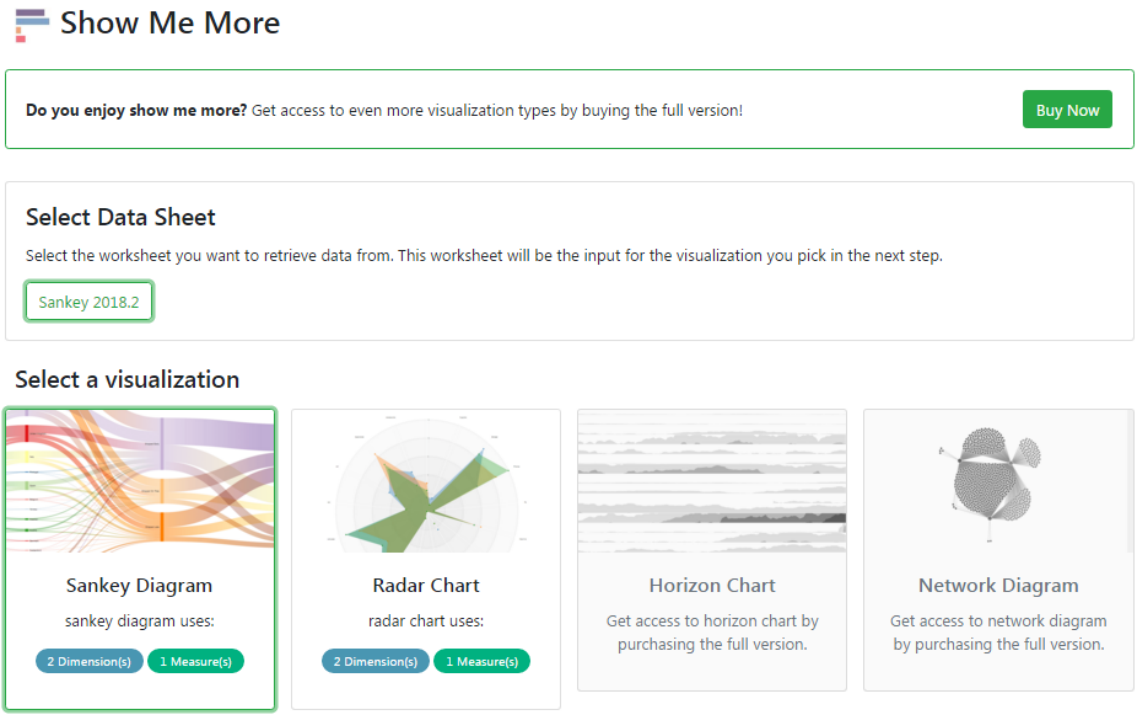
16. You can go to `My Extensions` and navigate to your extensions library to choose `Show Me More`:



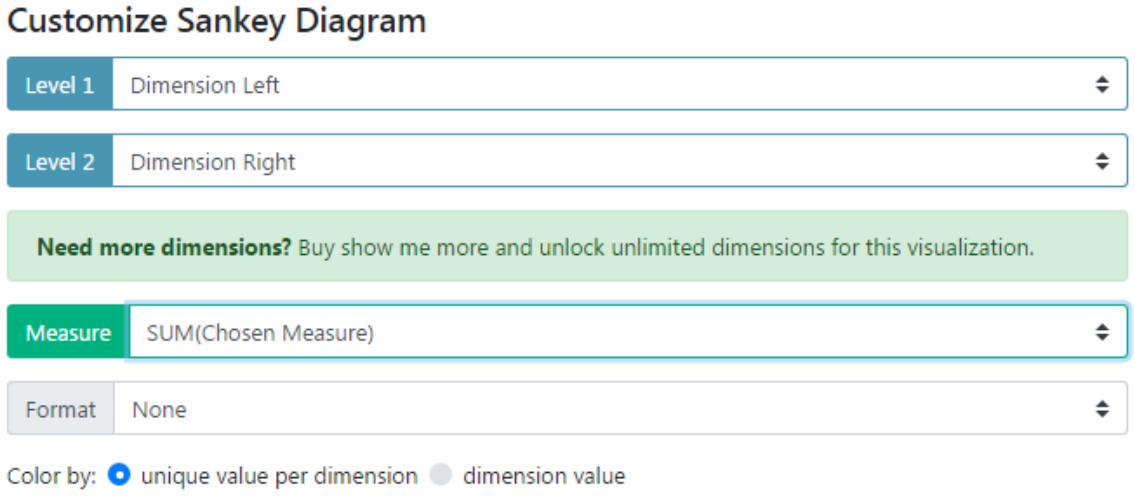
17. Click on `Use configure to get started.` to begin the configuration:



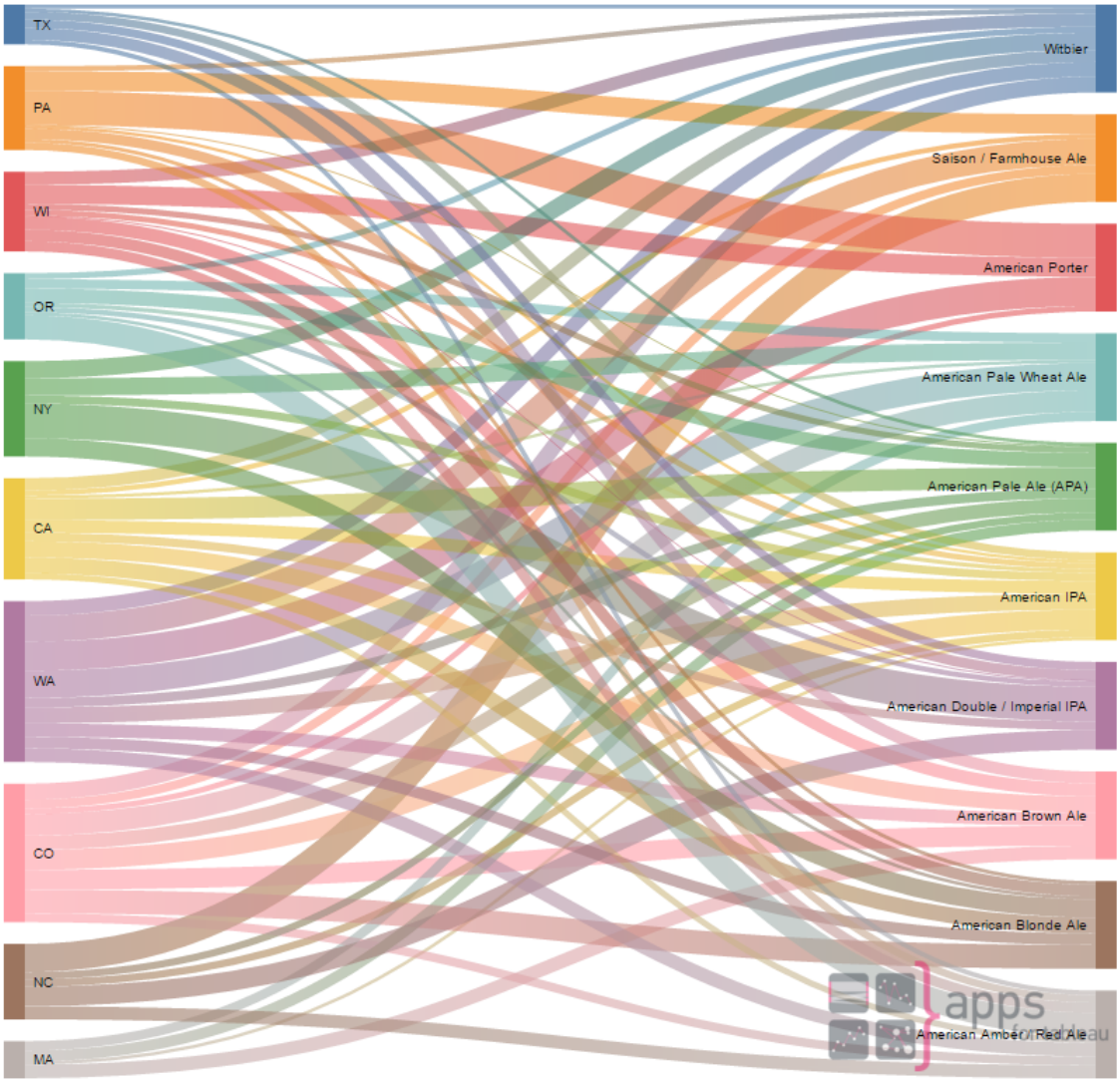
18. Choose `Sankey Diagram`:



19. Select the dimensions and measures as shown in the following screenshot:



We can see the final visualization in the following screenshot:



How it works...

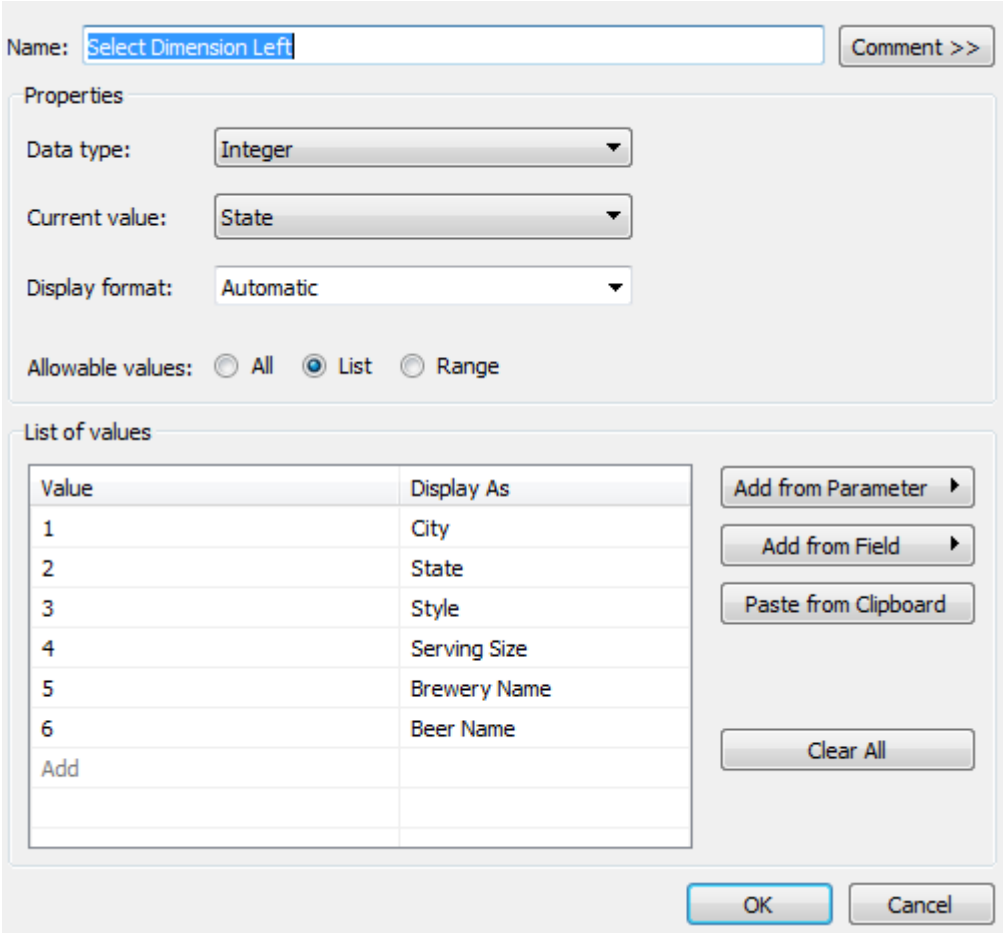
We created a cross-tab with two dimensions and one measure. We then added this to a dashboard. Because we only wanted to see the Sankey chart and the extension requires that the cross-tab be on the dashboard, we "hid it" by making it very small and floating. Using the extension object, we chose the **`Show Me More`** extension. We configured it for **`Sankey Diagram`**.

See also

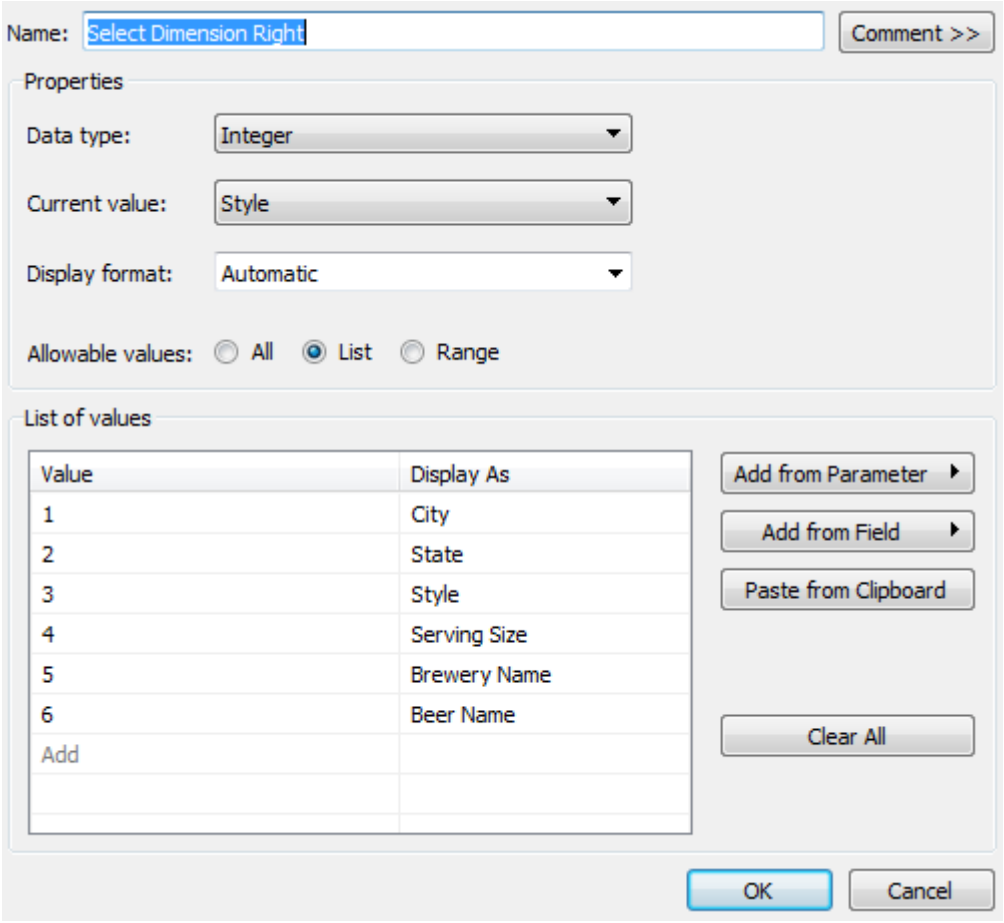
There are several ways to create a Sankey chart published on various blogs. Ian Balwin's post on [*Information Lab*] is excellent because it is flexible and does not require outside data prep. However, there are many calculations, which can make it hard to follow. But once it's set, there is a great deal of flexibility because of how the dimensions and measures have been created. The following directions are taken from his blog post at https://www.theinformationlab.co.uk/2018/03/09/build-sankey-diagram-tableau-without-data-prep-beforehand/{.ulink}.

Use `Beer.twbx`, `beers.csv`, and `breweries.csv` to work through this example:

1. Create the `Select Dimension Left` parameter or use the same one from the previous recipe:



2. Create the `Select Dimension Right` parameter or use the same one from the previous recipe:



3. Create the `Dimension Left` parameter or use the same one from the previous recipe:


```
CASE [Select Dimension Left]
WHEN 1 THEN [City]
WHEN 2 THEN [State]
WHEN 3 THEN [Style]
WHEN 4 THEN [Serving Size]
WHEN 5 THEN [Brewery Name]
WHEN 6 THEN [Beer Name]
END
```

4. Create the `Dimension Right` parameter or use the same one from the previous recipe:

Dimension Right

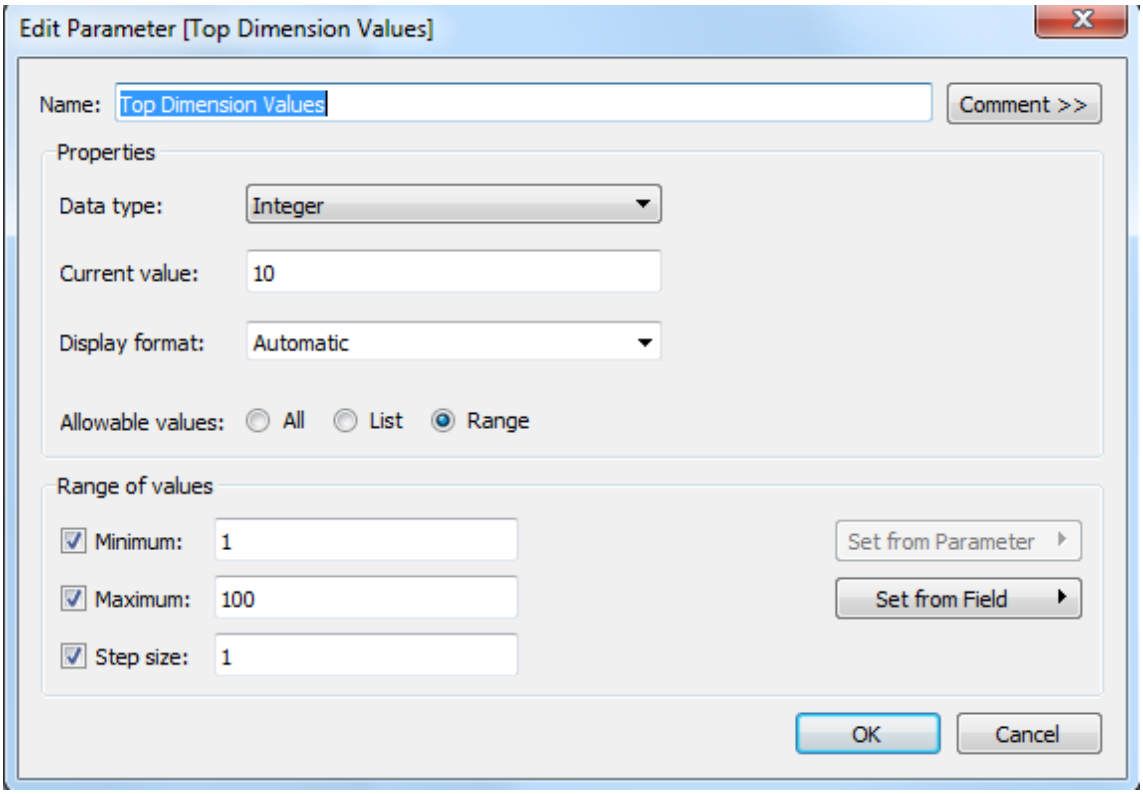
```
CASE [Select Dimension Right]
WHEN 1 THEN [City]
WHEN 2 THEN [State]
WHEN 3 THEN [Style]
WHEN 4 THEN [Serving Size]
WHEN 5 THEN [Brewery Name]
WHEN 6 THEN [Beer Name]
END
```

5. Create the `Chosen Measure` parameter or use the same one from the previous recipe:

Chosen Measure

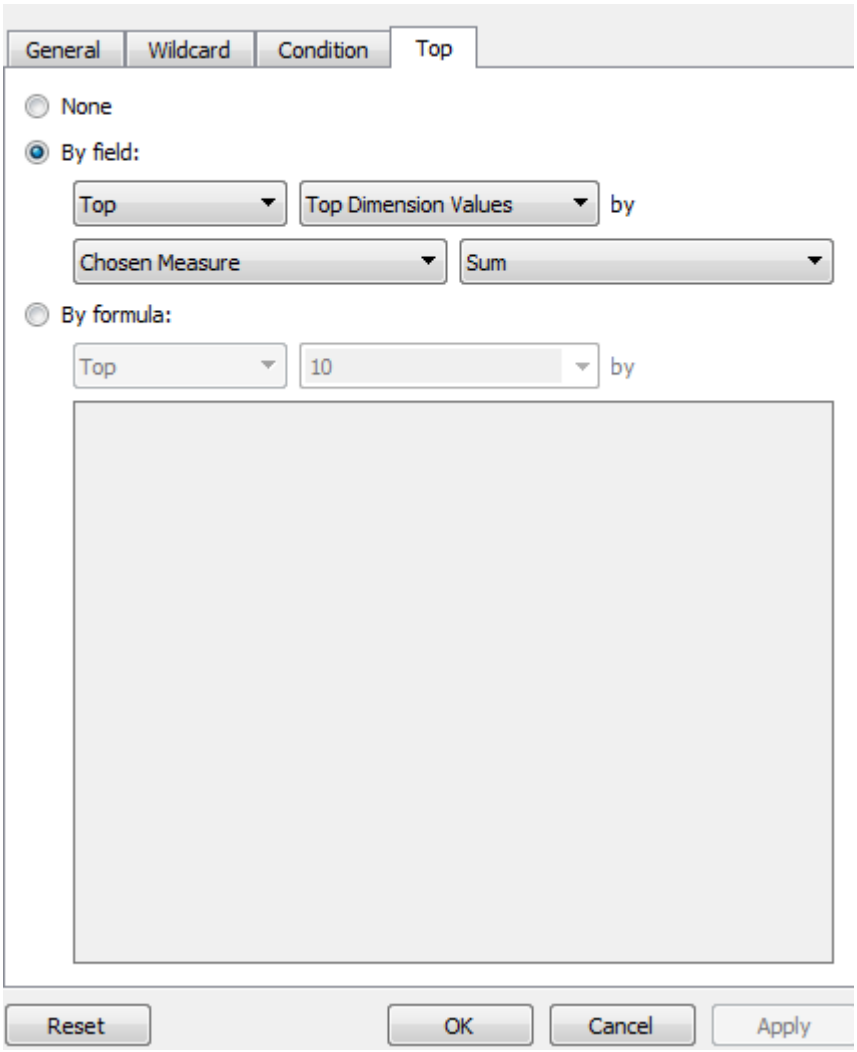
```
[F1]
```

6. Create the `Top Dimension Values` parameter or use the same one from the previous recipe:

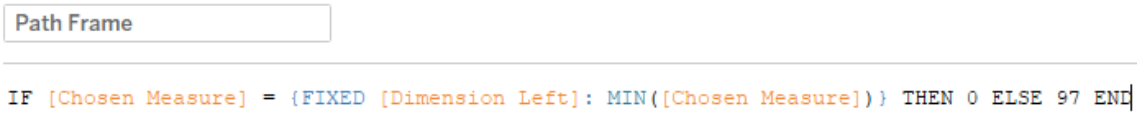


7. Add **`Dimension Right`** and **`Dimension Left`** to the **`Filters`** shelf.

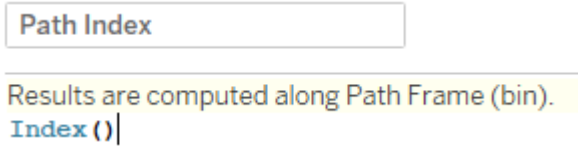
8. Right-click **`Dimension Right`** and **`Dimension Left`** in the **`Filters`** shelf, select **`Edit Filter`**, and set it as follows:



9. Create the `Path Frame` parameter:



10. Create a `Path Index` parameter:



11. Create the `T` parameter:

T

```
IF [Path Index] < 50
THEN (([Path Index]-1)%49)/4-6
ELSE 12 - (([Path Index]-1)%49)/4-6
END
```

12. Create the `Sigmoid` function:

Sigmoid

```
1/(1+EXP(1)^-[T])
```

13. Create the `Sankey Arm Size` parameter:

Sankey Arm Size

Totals summarize values from Dimension Left, Dimension Right.

```
SUM([Chosen Measure])/TOTAL(SUM([Chosen Measure]))
```

14. Create all the following calculations for the top of the Sankey Arm:
    - `Max Position Left`
        - `RUNNING_SUM([Sankey Arm Size])`
    - `Max Position Left Wrap`
        - `WINDOW_SUM([Max Position Left])`
    - `Max Position Right`
        - `RUNNING_SUM([Sankey Arm Size])`
    - `Max Position Right Wrap`
        - `WINDOW_SUM([Max Position Right])`
15. Create all of the following calculations for the bottom of the Sankey Arm:
    - `Max for Min Position Left`
        - `RUNNING_SUM([Sankey Arm Size])`
    - `Min Position Left`
        - `RUNNING_SUM([Max for Min Position Left])-[Sankey Arm Size]`
    - `Min Position Left Wrap`
        - `WINDOW_SUM([Min Position Left])`

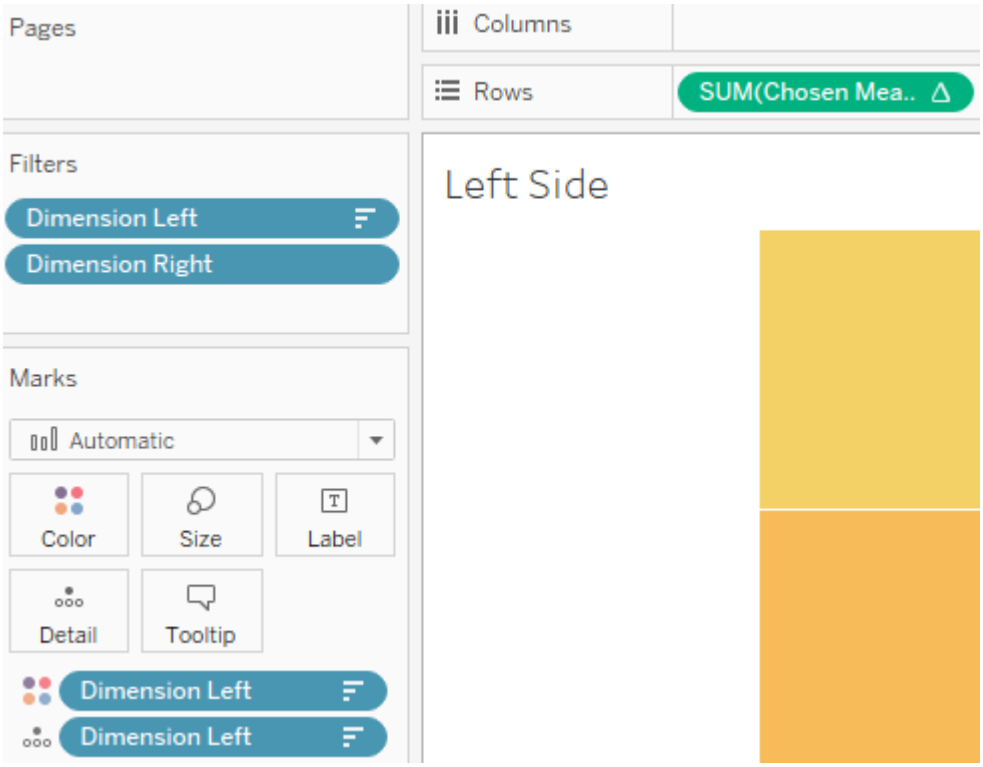
- `Max for Min Position Right`
  - `RUNNING_SUM([Sankey Arm Size])`
- `Min Position Right`
  - `RUNNING_SUM([Max for Min Position Right])-[Sankey Arm Size]`
- `Min Position Right Wrap`
  - `WINDOW_SUM([Min Position 2]`

16. Create the `Sankey Polygons` calculation as follows:

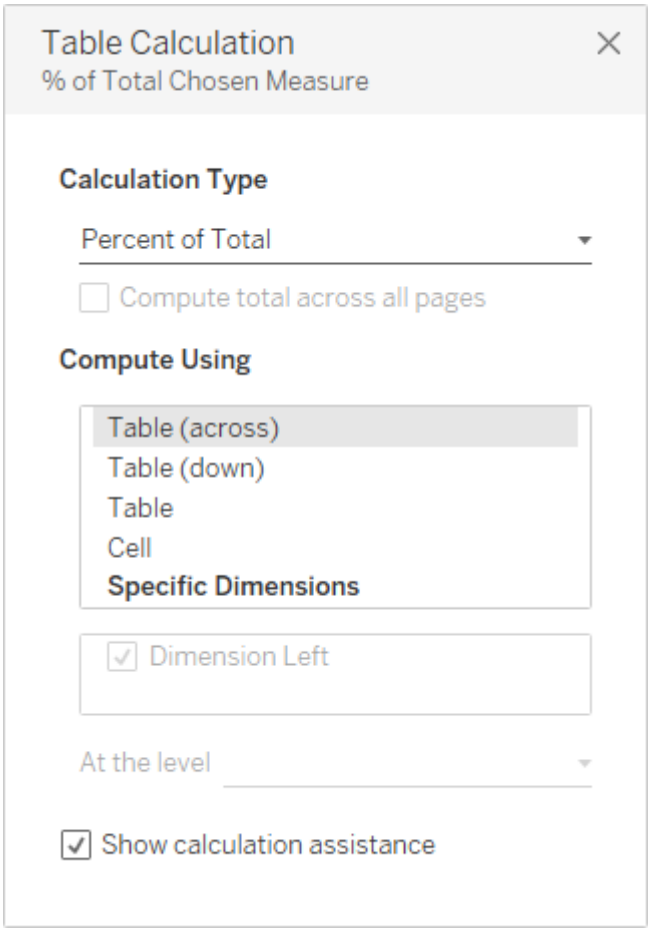
Sankey Polygons

```
IF [Path Index] > 49
THEN [Max Position Left Wrap]+([Max Position Right Wrap]-[Max Position Left Wrap])*[Sigmoid]
ELSE [Min Position Left Wrap]+([Min Position Right Wrap]-[Min Position Left Wrap])*[Sigmoid]
END
```

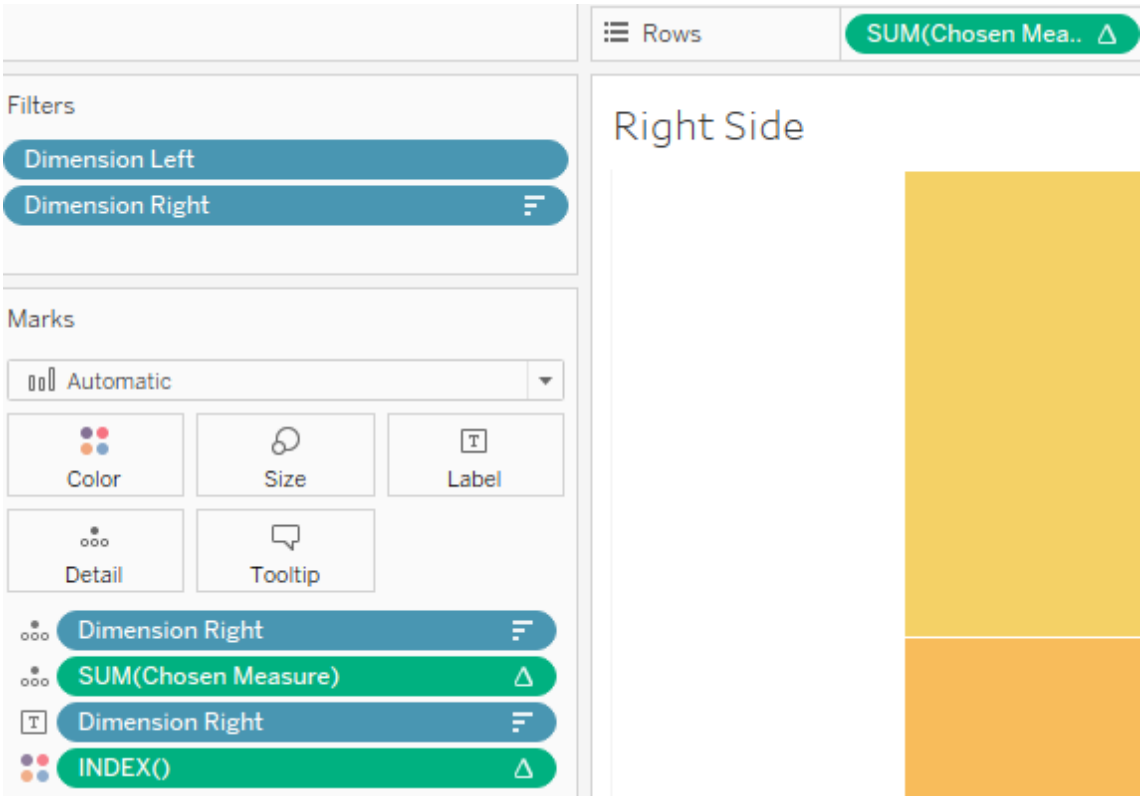
17. Create the **`Left Side`** sheet:



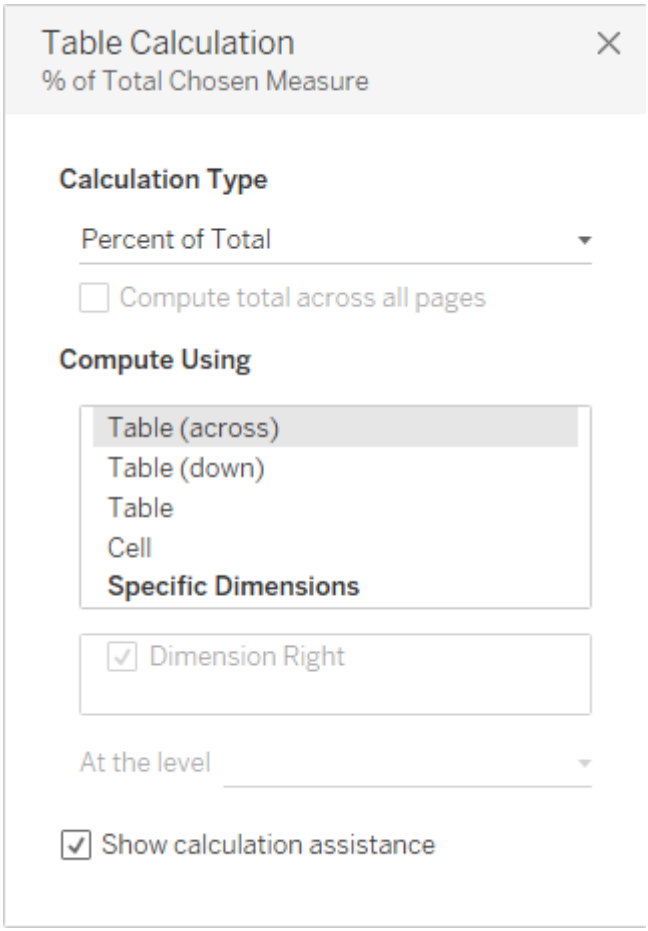
18. Apply a **`Percent of Total`** option in **`Table Calculation`** to **`Chosen Measure`**:



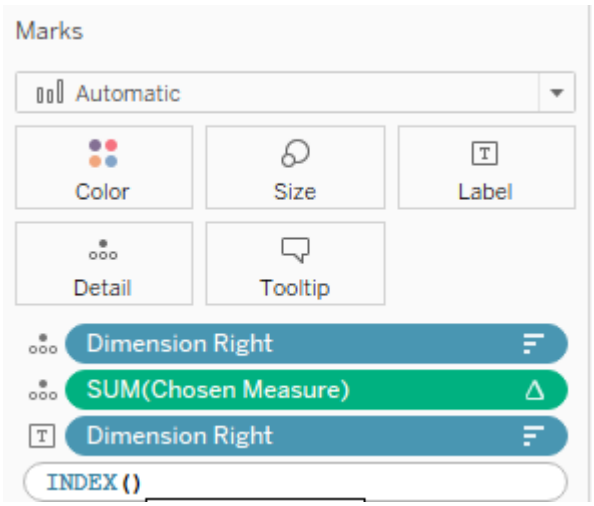
19. Create the `Right Side` sheet:



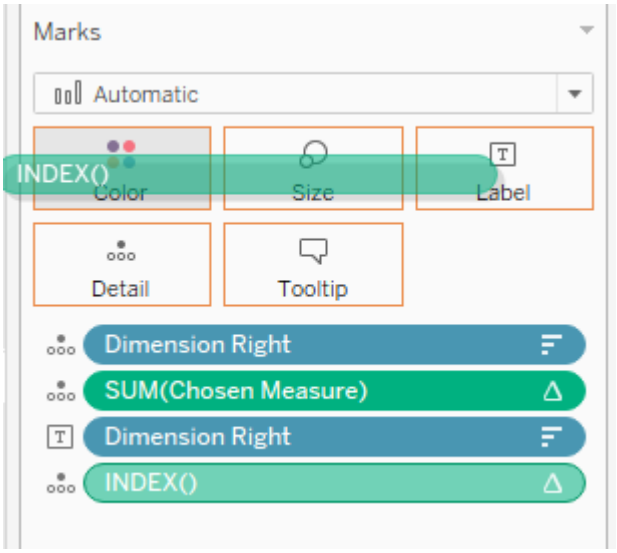
20. Apply a `Percent of Total` option under `Table Calculation` to `Chosen Measure`:



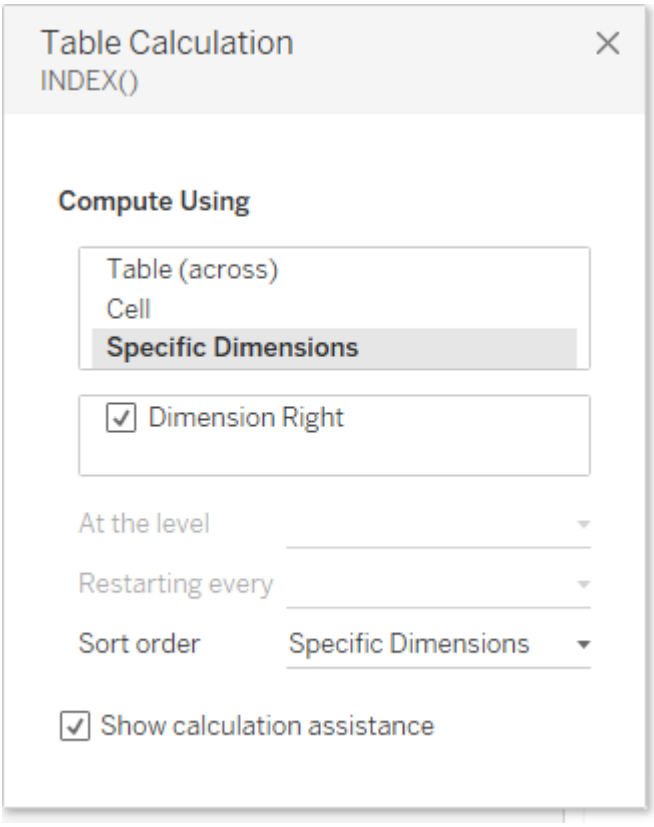
21. Create `INDEX()` by double-clicking in the **`Marks`** card and typing `INDEX()`:



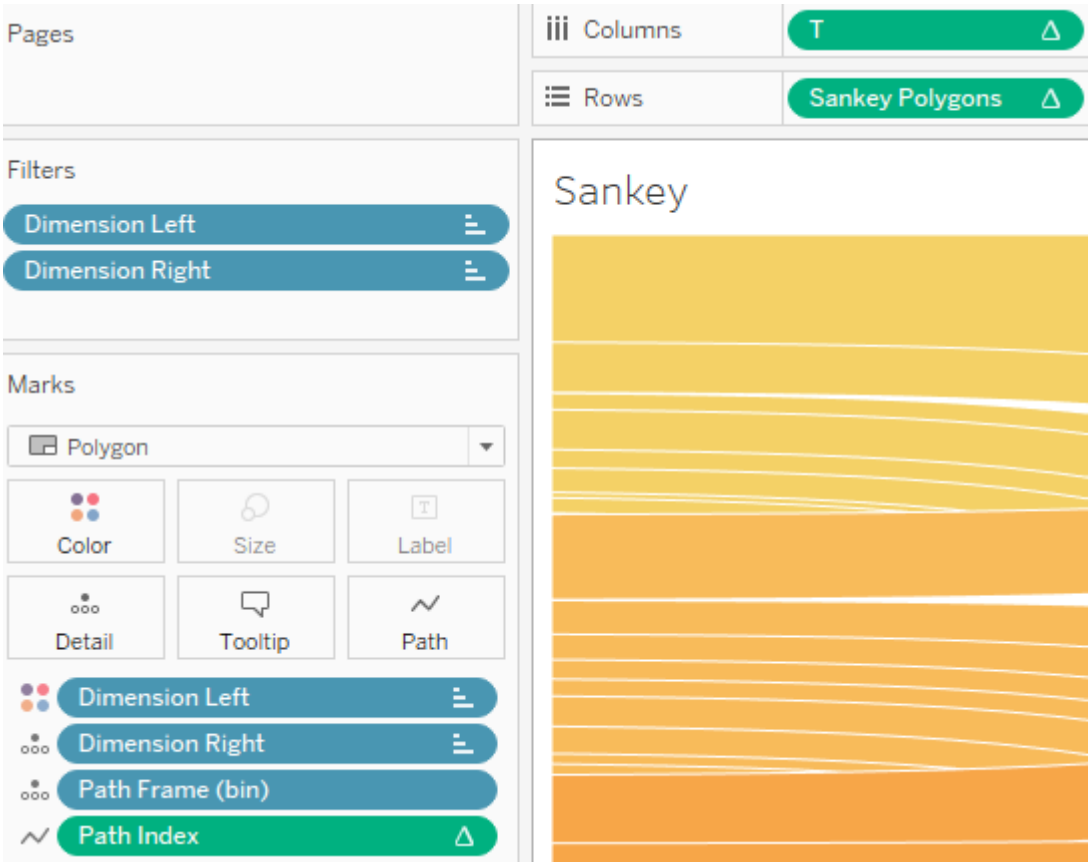
22. Apply **`INDEX()`** to the **`Color`** card:



23. Apply a table calculation to `INDEX()`, as shown in the following screenshot:

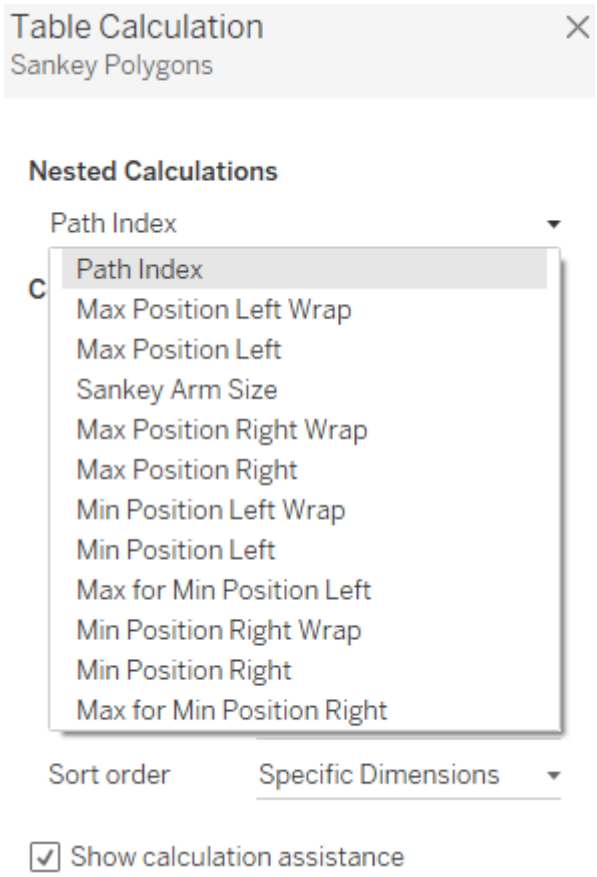


24. Create the `Sankey` sheet:

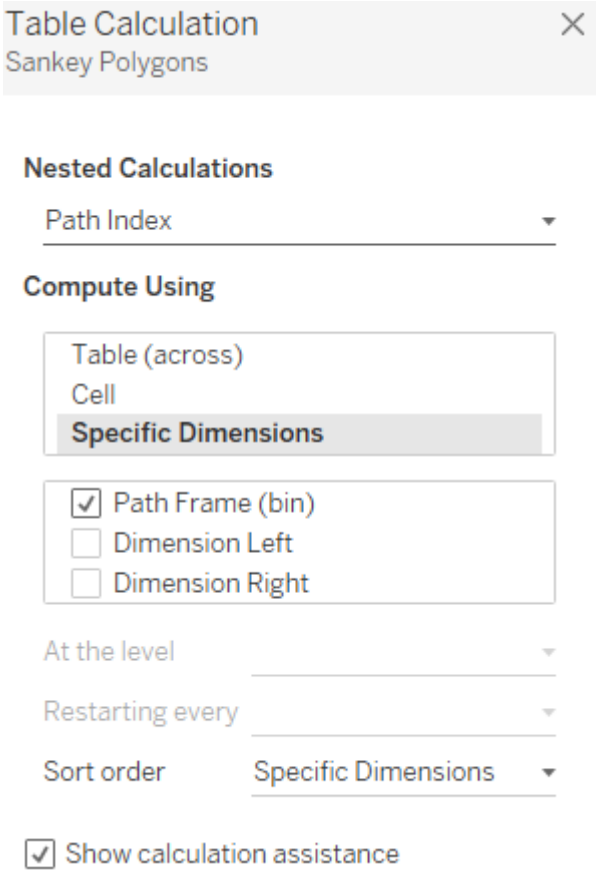


25. `Table Calculation` for `Sankey Polygons` :

- These are all the table calculations we will need to configure for the **`Sankey Polygons`**:

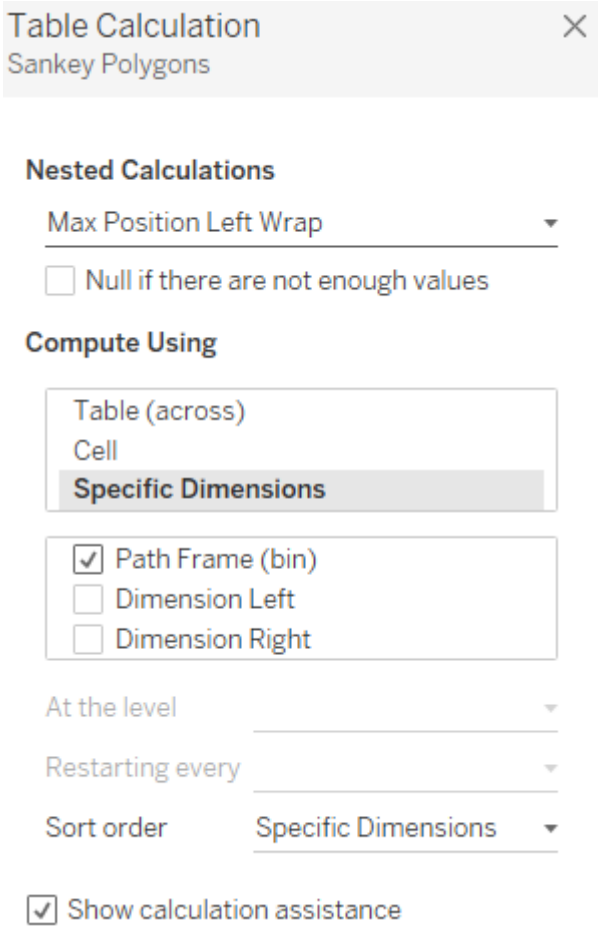


26.



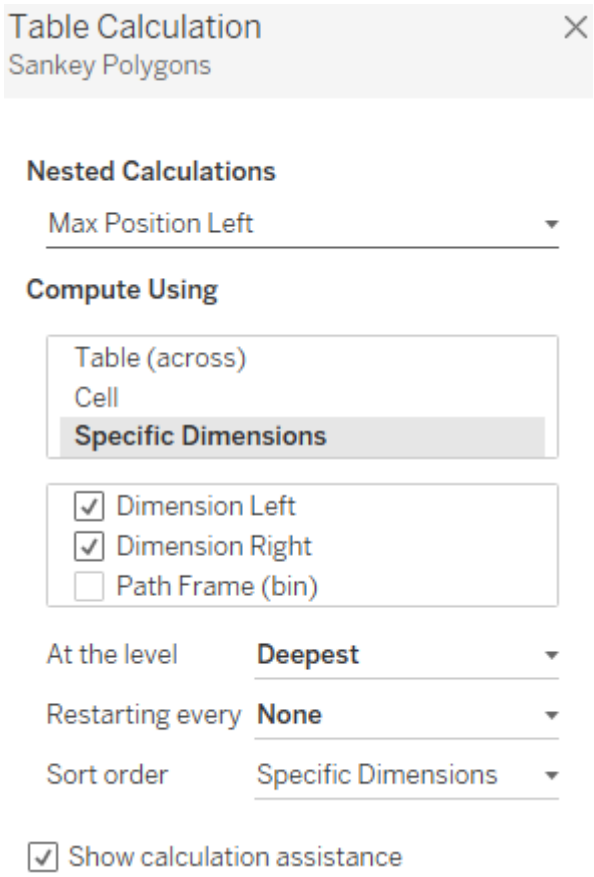
- `Path Index`:

27.



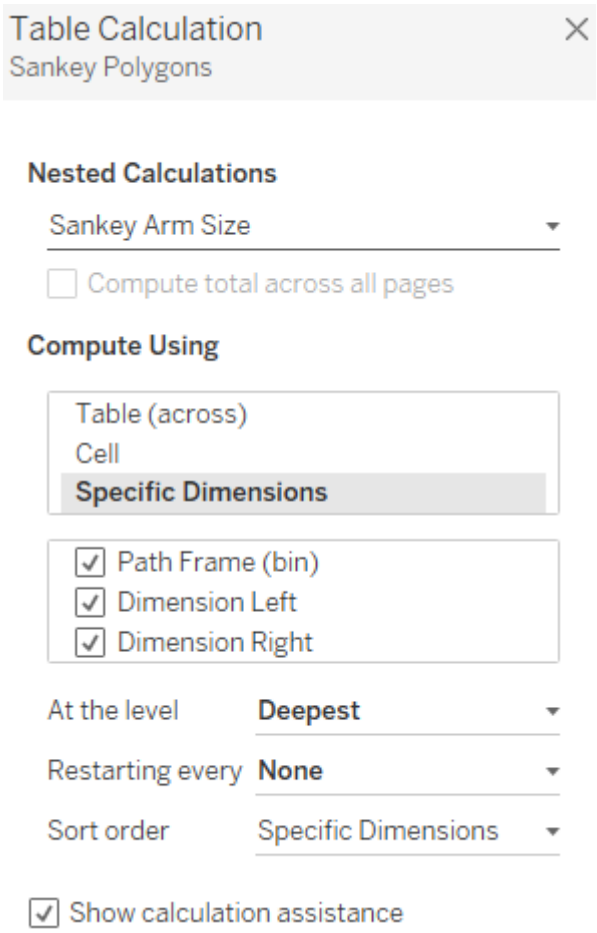
- `Max Position Left Wrap`:

28.



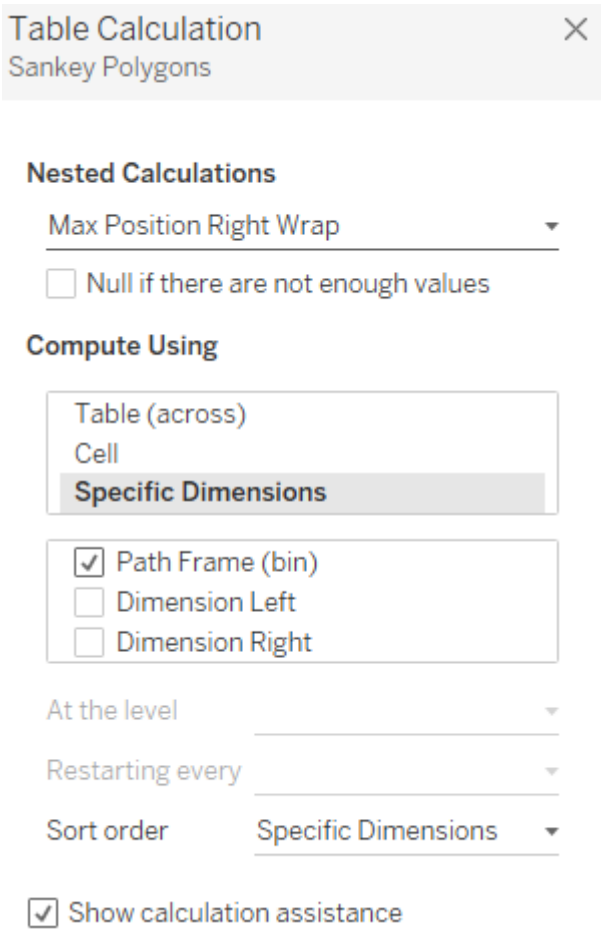
- `Max Position Left`:

29.



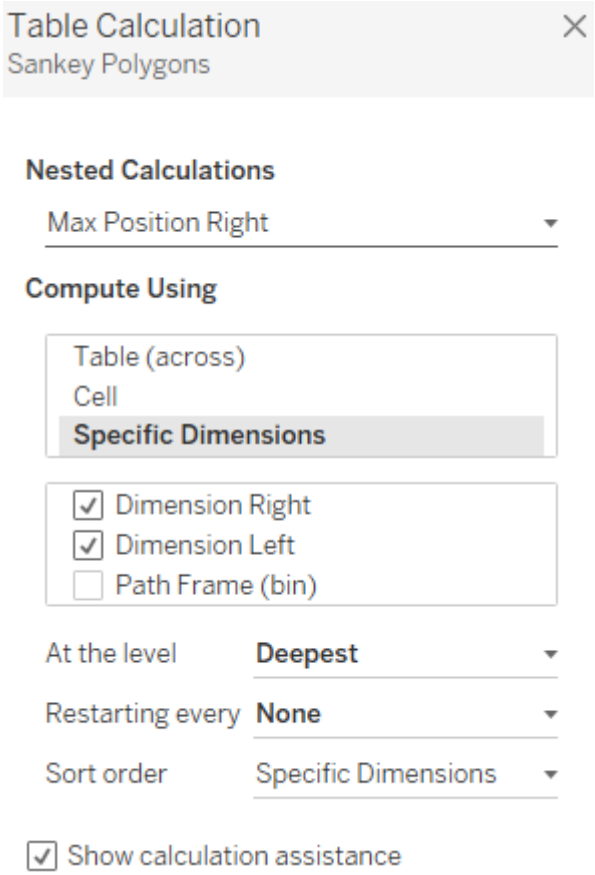
- `Sankey Arm Size`:

30.



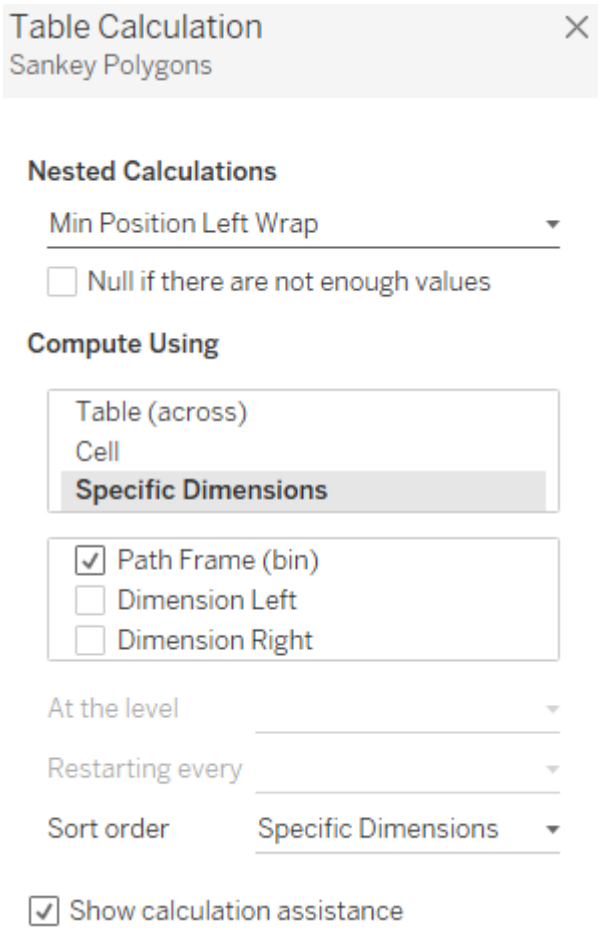
- `Max Position Right Wrap`:

31.



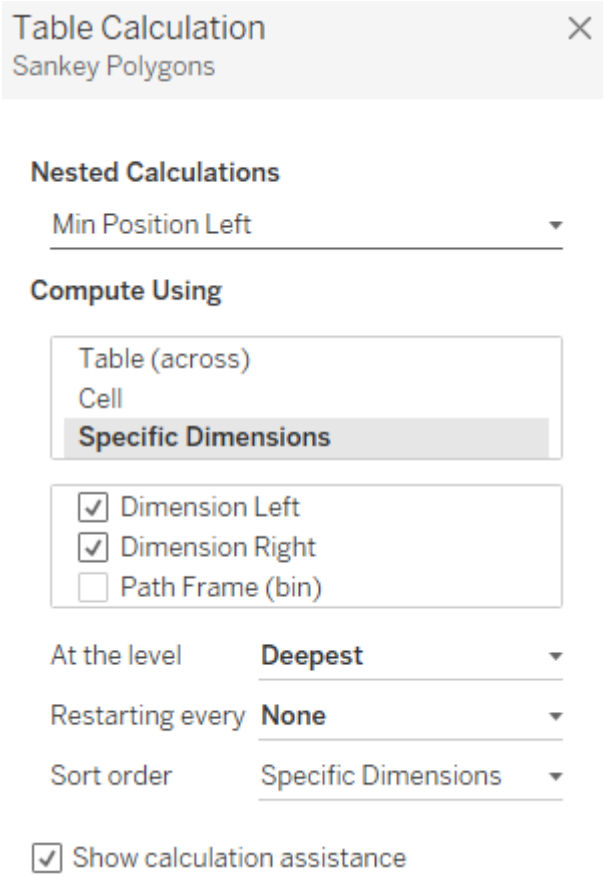
- `Max Position Right`:

32.



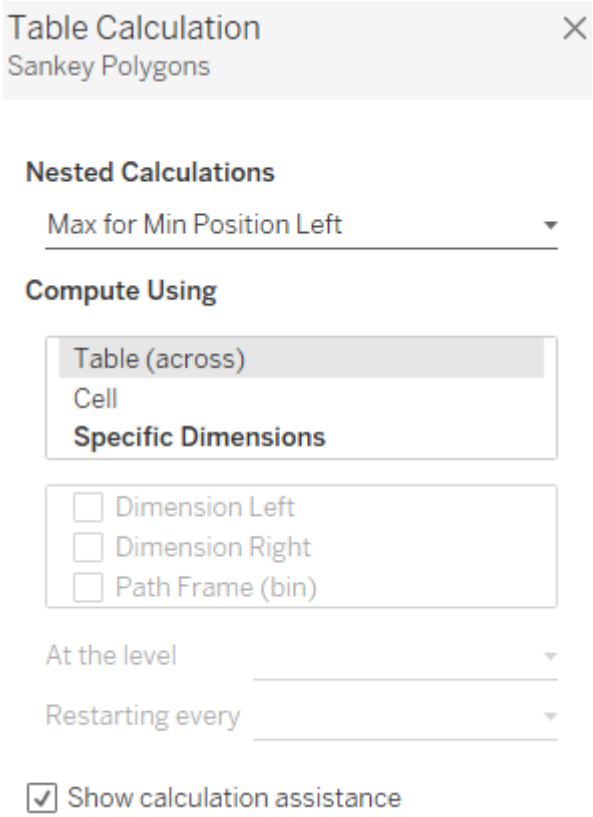
- `Min Position Left Wrap`:

33.



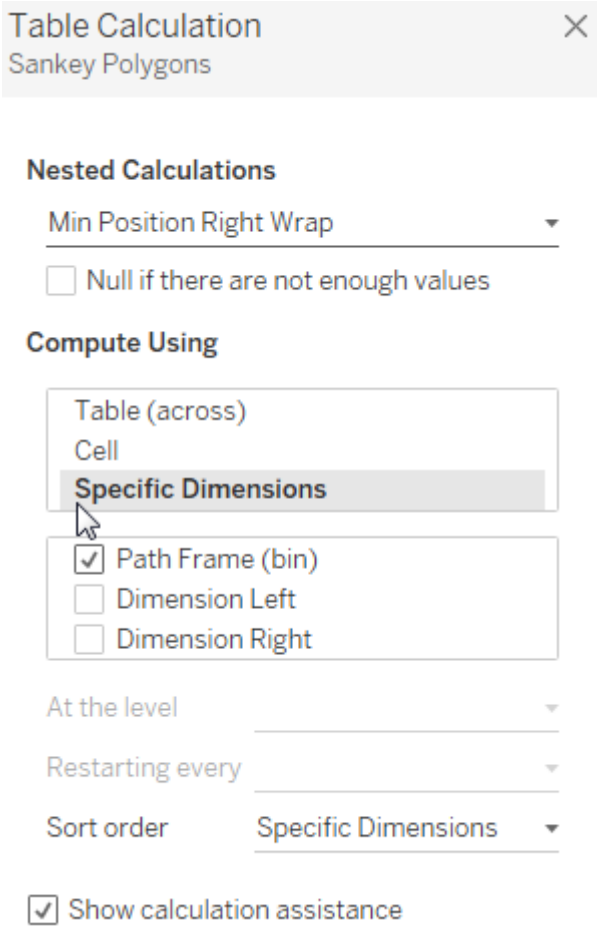
- `Min Position Left`:

34.



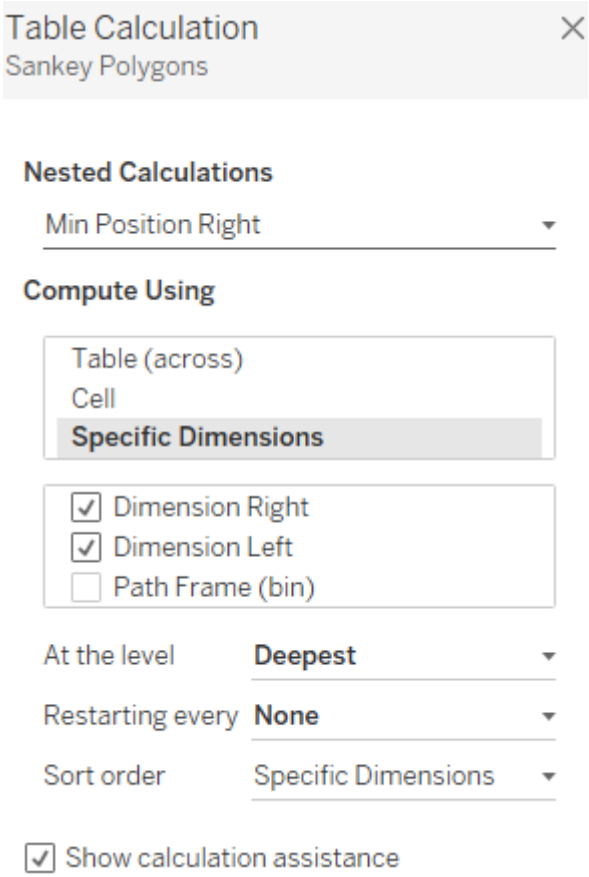
- `Max for Min Position Left`:

35.



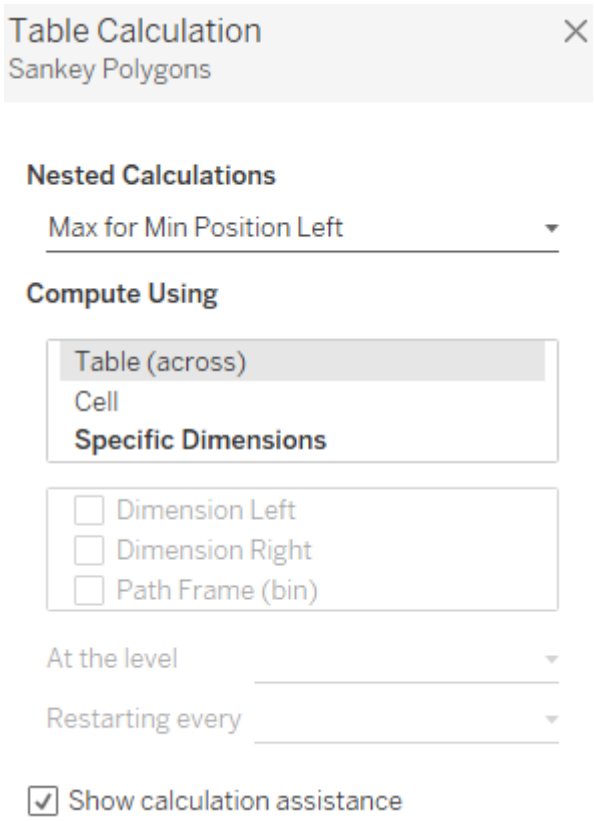
- `Min Position Right Wrap`:

36.



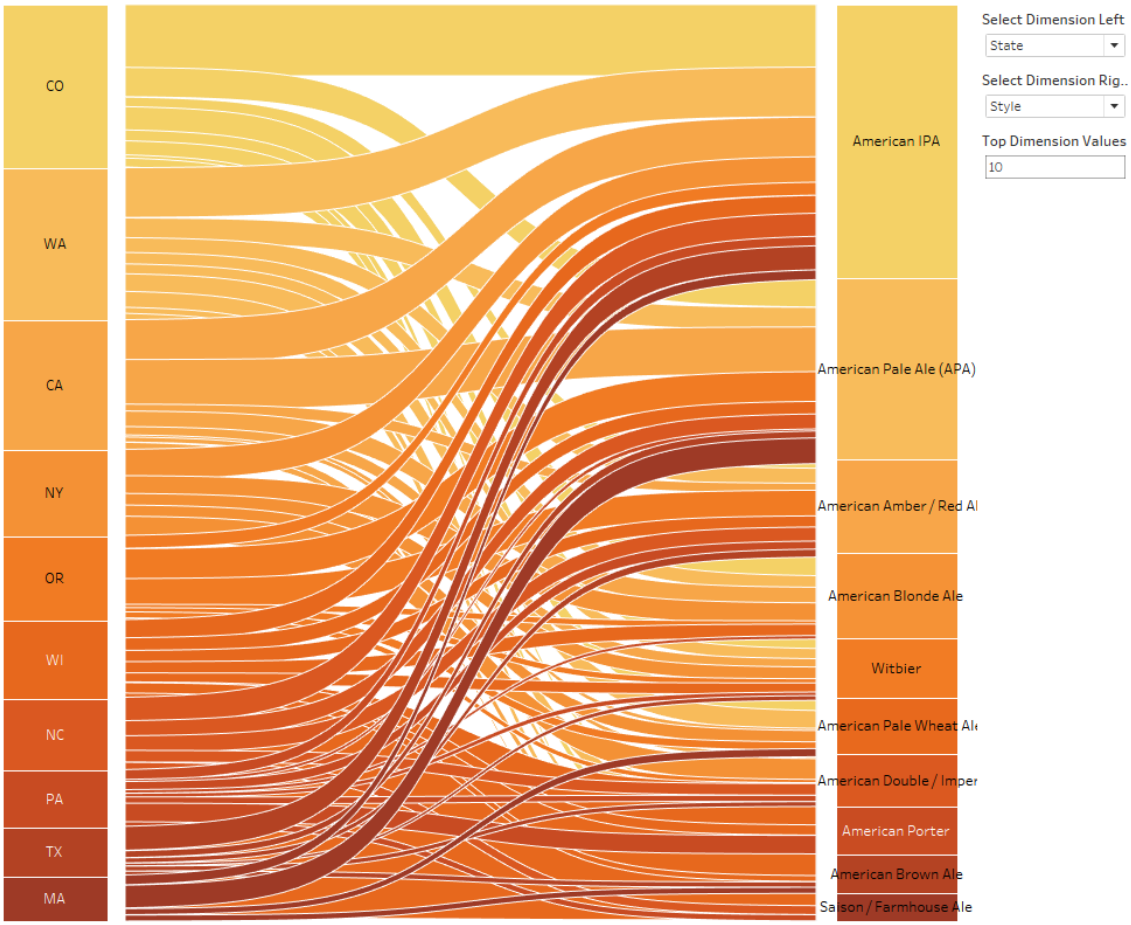
- `Min Position Right`:

37.



- `Max for Min Position Right`:

  You will see the final dashboard in the following screenshot:



Marimekko charts

Marimekko charts go by many names: mekko, mosaic, or matrix, to name a few. It is a two-dimensional stacked chart. They are used to analyze data composition or distribution across two variables at once. Each axis represents 100%.

Getting ready

In this recipe, we use complex calculations to build the Marimekko chart.

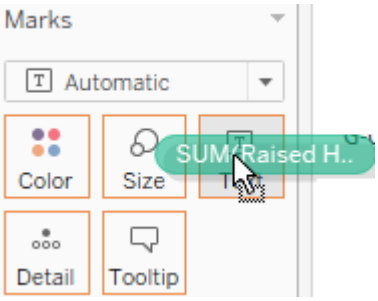
How to do it..

To follow along, open the Marimekko packaged workbook: `Mekko.twbx` and `xAPI-Edu-Data.csv`. We are going to compare gender, parent satisfaction with school, and student participation by raised hands.

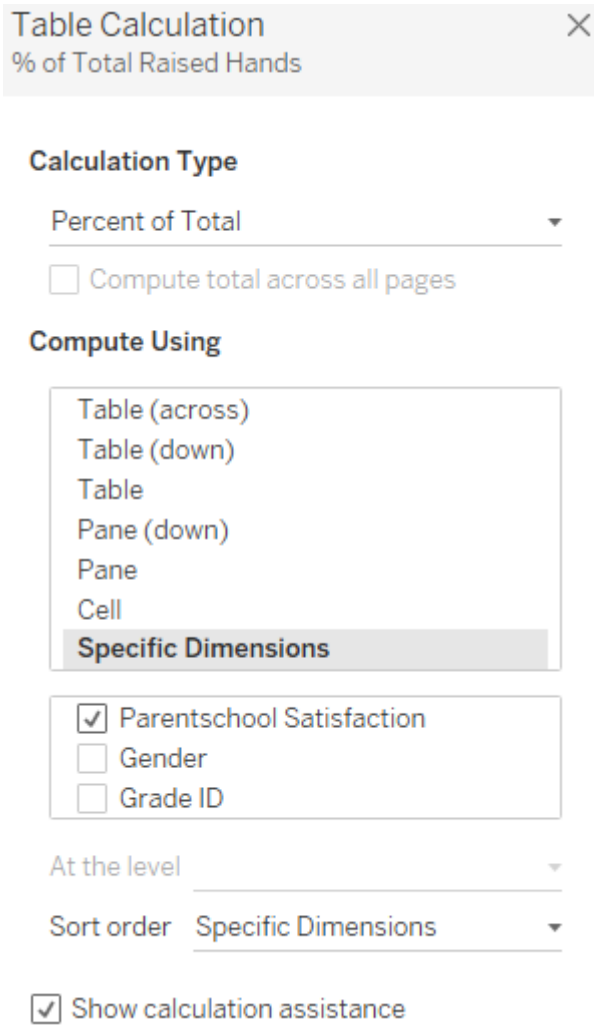
Note

It is recommended you start with a text table to get the calculations correct.

1. Add the dimensions of interest to the **`Rows`** shelf. We are going to add **`Grade ID`**, **`Gender`**, and **`Parentschool Satisfaction`**.
2. Add the **`Raised Hands`** to the **`Text`** shelf in the **`Marks`** card:



3. Add the **`Percent of Total`** option in **`Table Calculation`**, and under **`Compute Using`** select **`Parentschool Satisfaction`**:



4. Double-click on **`Measure Values`**:

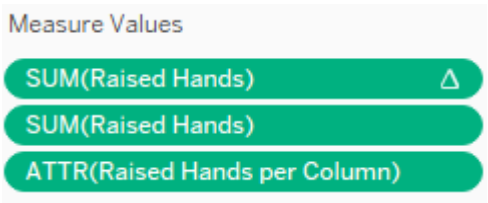
5. Add the **`Raised Hands`** to the **`Measure Values`** card:

6. Create calculation `Raised Hands per Column`:

Raised Hands per Column

```
{EXCLUDE [Parentschool Satisfaction]:SUM([Raised Hands])}
```

7. Add **`Raised Hands per Column`** to the **`Measure Values`** card:



8. Create **`Calculation # of Raised Hands, Grade, Gender ID`**, **`Parentschool Satisfaction`** and add it to the **`Measures Value`** card. This calculation creates somewhat of a running total summary for grades and genders. This value will be the [*x*] axis:

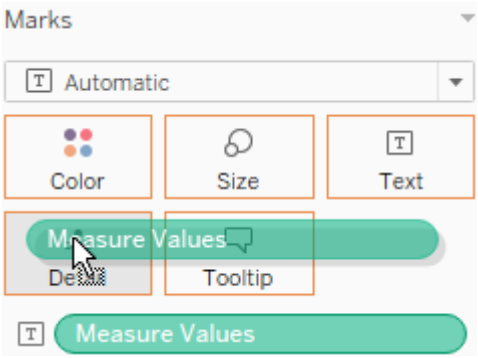
Results are computed along Table (across).

```
//If it's the first row in parition
IF FIRST()==0 THEN
     //return this value
     MIN([Raised Hands per Column])

//check if this grade is NOT the same as the previous one
ELSEIF MIN([Grade ID]) != LOOKUP(MIN([Grade ID]),-1) THEN
     //Add the previous value of raised hands per column to this one
     PREVIOUS_VALUE(0) + MIN([Raised Hands per Column])

//check if gender is NOT the same as the previous one
ELSEIF MIN([Gender]) != LOOKUP(MIN([Gender]),-1) THEN
     //add the previous value of raised hands per column to this one
     PREVIOUS_VALUE(0) + MIN([Raised Hands per Column])

ELSE
     //it's the same grade and gender, show the same raised hands value
     PREVIOUS_VALUE(0)

END
```

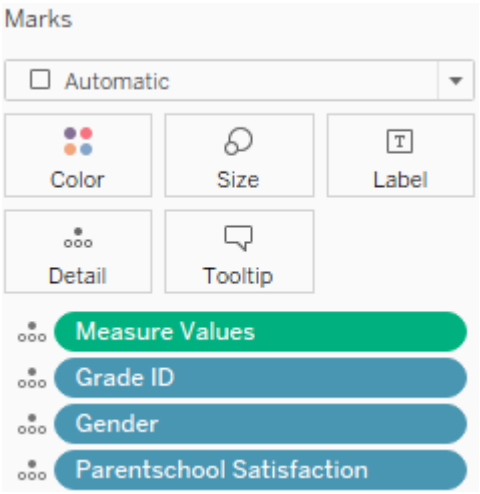
9. We've completed the calculation work in order to make the visualization. Your grid should look as follows:

Grade ID	Gender	Parentschool Satisfaction	Raised Hands	% of Total Raised Hands along Parents..	Raised Hands per Column	# of Raised Hands, Grade, Gender ID, Pa..
G-02	F	Bad	499	20.18%	2,473	2,473
		Good	1,974	79.82%	2,473	2,473
	M	Bad	1,072	37.00%	2,897	5,370
		Good	1,825	63.00%	2,897	5,370
G-04	F	Bad	272	24.46%	1,112	6,482
		Good	840	75.54%	1,112	6,482
	M	Bad	345	30.50%	1,131	7,613
		Good	786	69.50%	1,131	7,613

10. Move **`Measure Values`** from **`Text`** to **`Detail`**:

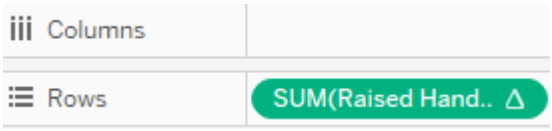


11. Move **`Grade ID`**, **`Gender`**, and **`Parentschool Satisfaction`** to **`Detail`**:

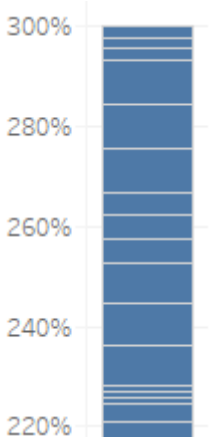


12. Move **`Sum(Raised Hands)`** with the **`Percent of Total`** calculation to the **`Rows`** shelf:

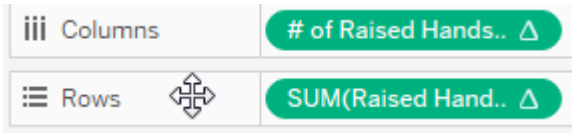
13. Remove **`Measure Names`** from the **`Columns`** shelf:



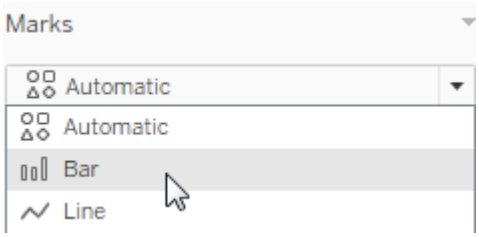
14. We have a stacked bar at this step:



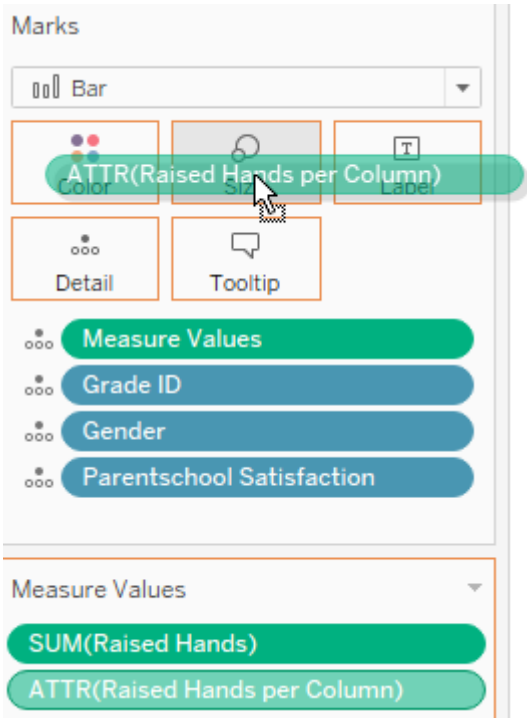
15. Move # `of Raised Hands, Grade, Gender ID` , `Parentschool Satisfaction` to the `Columns` shelf:



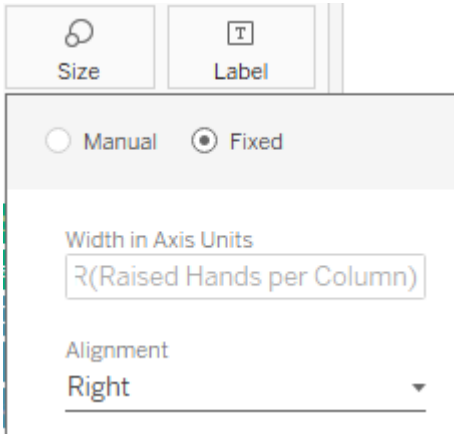
16. Change the `Scatter Plot` mark type to `Bar` :



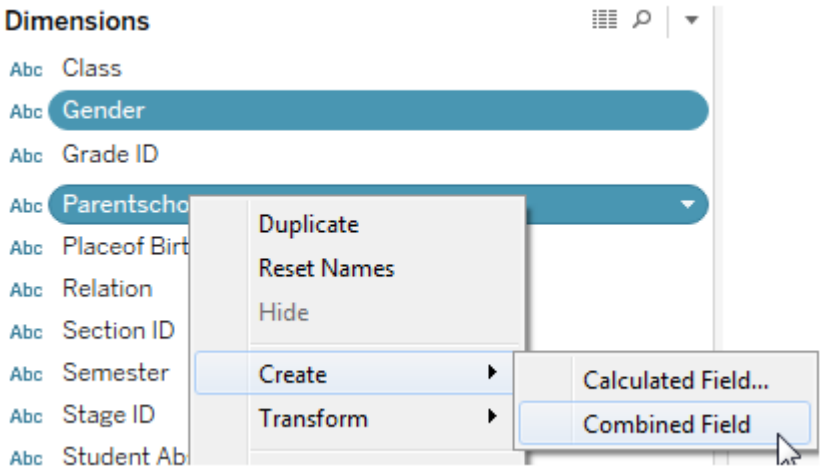
17. Move `Raised Hands per Column` from `Measure Values` to `Size` :



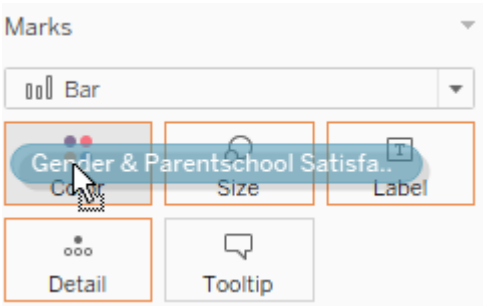
18. Set `Size` to `Fixed` and `Alignment` to `Right`:



19. Create a `Gender` and `Parentschool satisfaction` combined field, as shown in the following screenshot:

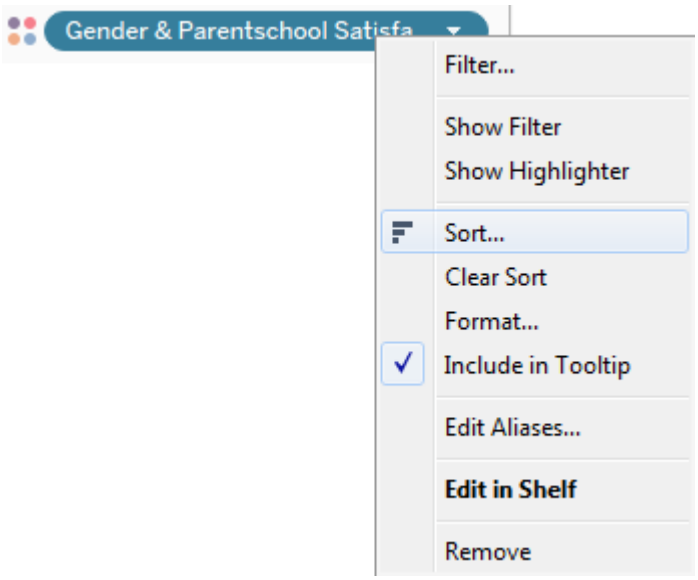


20. Add the combined field to `Color`:



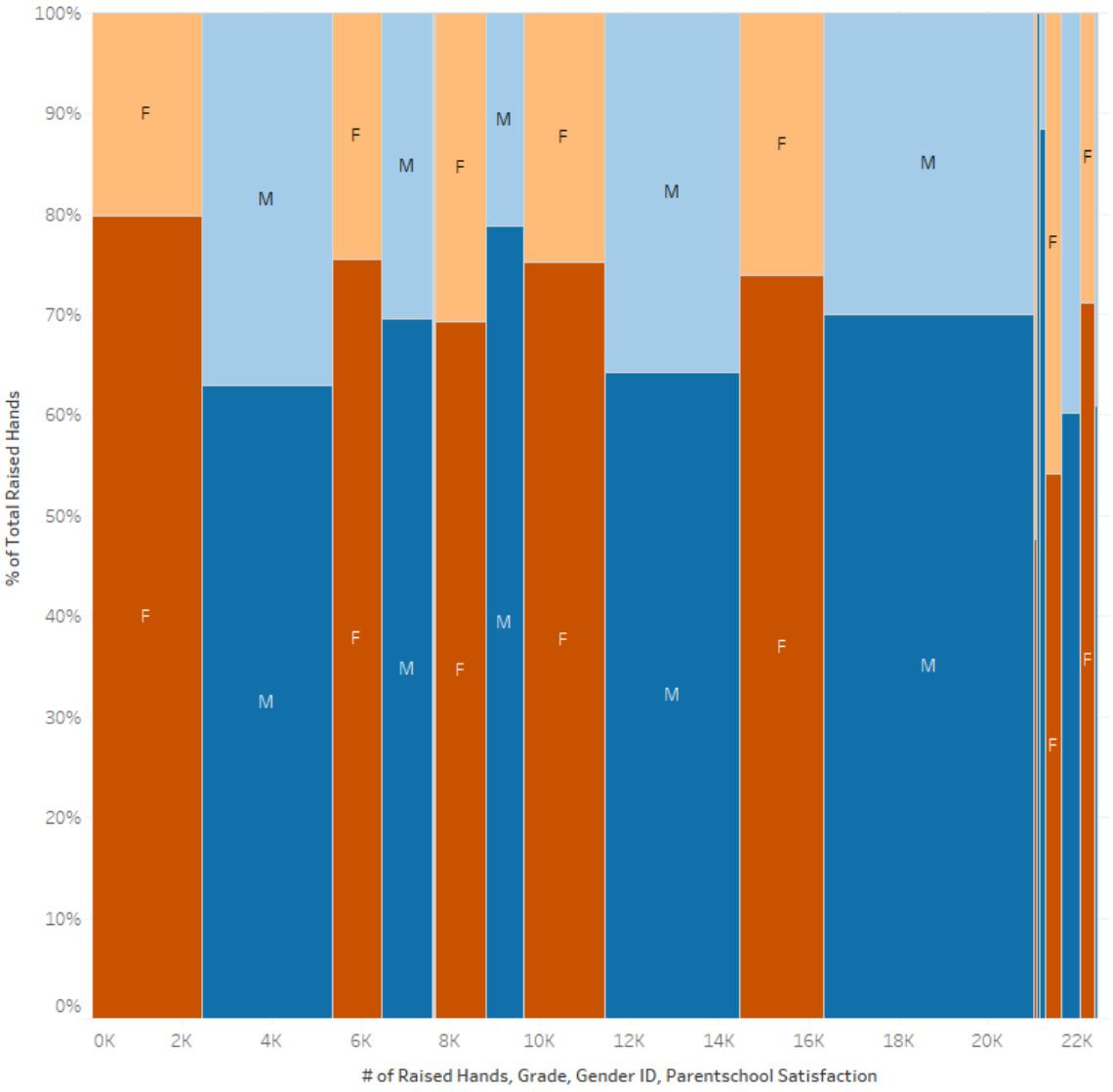
21. Manually sort by `Gender & Parentschool Satisfaction`:

- Right-click on this dimension in the marks card and select `Sort`:



- Choose `Manual Sort` and make it appear as follows:

22. This should produce the following chart. Second, seventh, and eighth graders raise their hands more in class than other grades. Parents with students who raise their hands in class were generally happier with the school. High school students do not raise their hands in class, as shown in the following screenshot:



How it works...

First, we created a text table to work through our calculations. We then added the dimensions of interest to the **`Rows`** shelf and the measure of interest to the **`Text`** shelf.

For column height, we created `Percent of Total for Raised Hands`, which we compute by using **`Parentschool Satisfaction`**. This allows us to see 100% for each **`Gender`** and **`Grade`** combination, which is our column height. We add this calculation to the **`Measure Values`** card.

In order to get our column widths, we created the **`Raised Hands per Column`** calculation, which calculates the sum for all hand raises for each combination of **`Gender`** and **`Grade`**. We then add the calculation to the **`Measure Values`** card.

Next, we created **`# of Raised Hands, Grade, Gender ID, Parentschool Satisfaction`** to help order the columns of each grade and gender correctly along the x-axis. This calculation is building a running total based on raised hands per column.

It does this math by going through the following checks:

- Is it the first row in the partition then return this value?
- Has the grade changed then add this value and the previous value?
- Has the gender changed? If it has, add this value and the previous value; otherwise we have to return the previous value. Then we began to create the visualization. We moved **`Measure Values`** from **`Text`** to **`Detail`**; move **`Grade ID`**, **`Gender`**, and
- *`Parentschool Satisfaction`* * *to* **`Detail`**. *We moved Sum(Raised Hands) with the percent of total calculation to the* **`Rows`** *shelf, and removed* **`Measure Names`** *from the ** `Columns` shelf. At this point, we have a stacked bar.

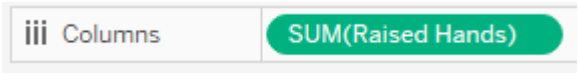
We moved **`# of Raised Hands, Grade, Gender ID, Parentschool Satisfaction`** to the **`Columns`** shelf. Next, we changed the **`Scatter Plot`** mark type to **`Bar`**. To get the column widths, we move **`Raised Hands per Column`** from **`Measure Values`** to **`Size`** and set it to **`Fixed`** and **`Alignment`** to **`Right`**.

In order to color our visualization, we created a **`Gender`** and **`Parentschool Satisfaction`** combined field that we add to **`Color`**. Finally, to make **`Gender`** more visible, we added it to **`Label`**.

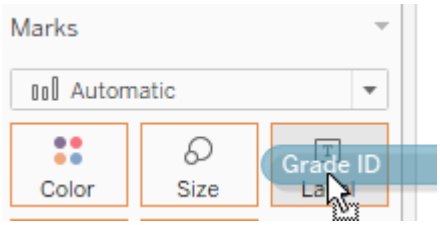
There's more...

We can create a header visualization to use in a dashboard so that the grades are labelled better:

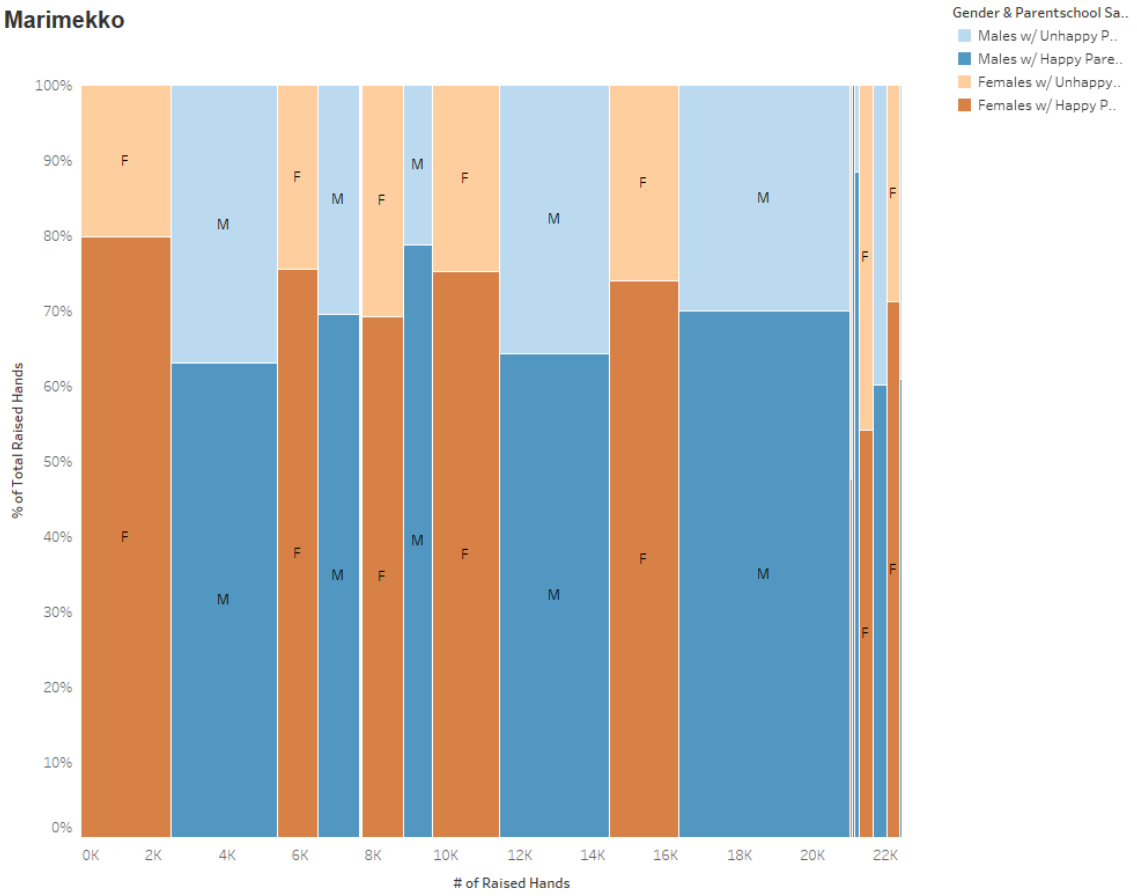
1. Add **`Raised Hands`** to the **`Column`** shelf:



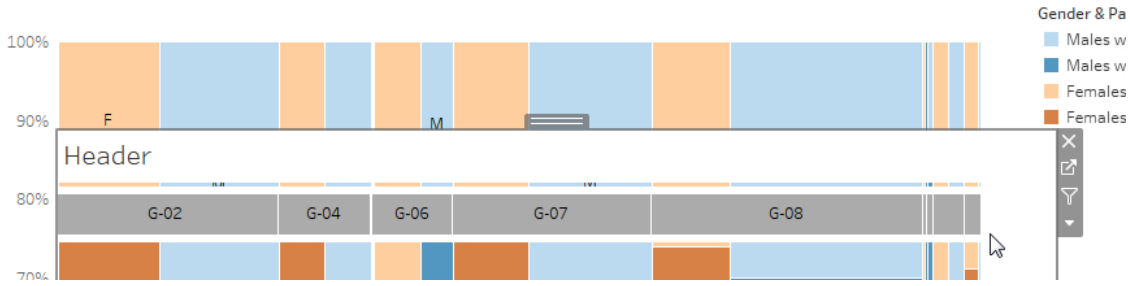
2. Add **`Grade ID`** to **`Label`**:



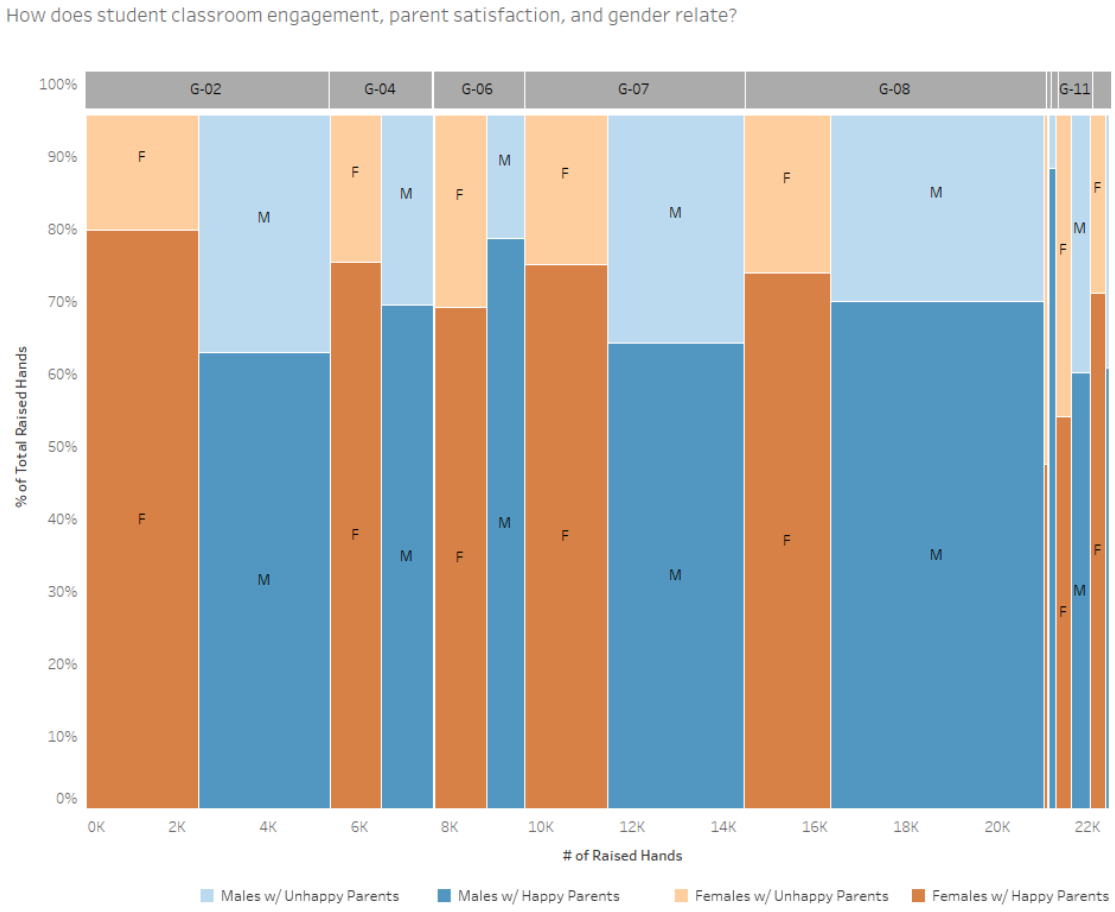
3. Add the **`Marimekko`** chart to a dashboard:



4. Add the `Header` visualization to the dashboard as a floating object and resize it:



5. After some formatting and resizing, we can view our final visualization in the following screenshot:



Hex-Tile maps

Here, you will learn how to eliminate the visual perceptions that occur due to different sizes of different states or countries and focus on showing more actionable trends.

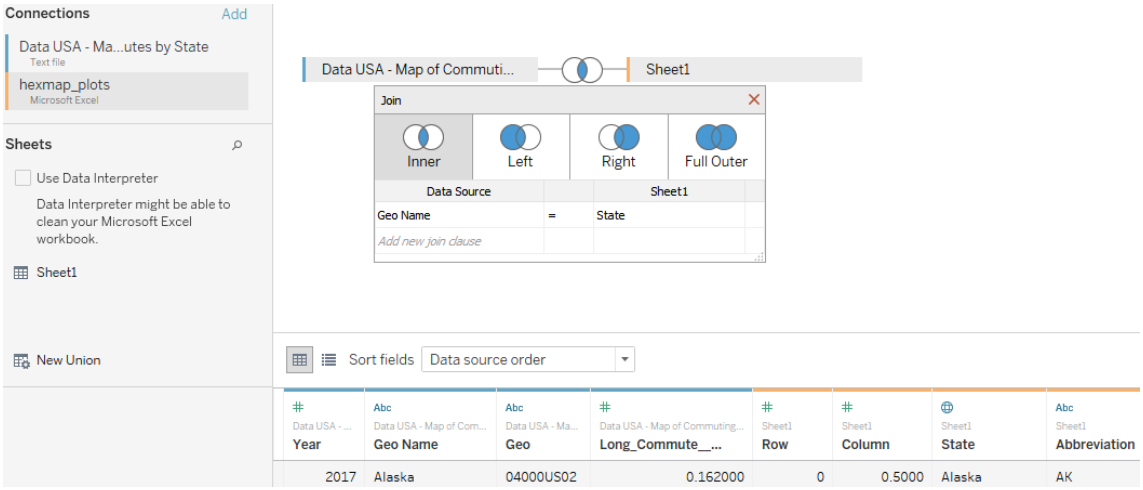
Getting ready

In this recipe, we will create a Hex-Tile map using a scatter plot as a base.

How to do it..

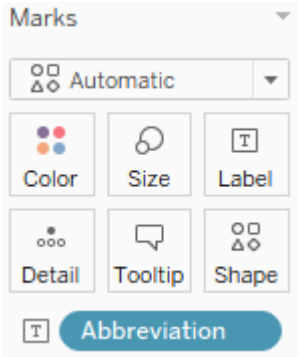
Follow along using the `Hexmap.twbx`, `hexmap_plots.xlsx`, and `Data USA - Map of Commuting Alone over 30 Minutes by State.csv` datasets:

1. Choose the `hexmap_plot.xlsx` file.
2. Join **`Data USA - Map of Commuting Alone over 30 Minutes by State`** to the **`hexmap_plot`** data using the State and Geo Name columns as the join condition:

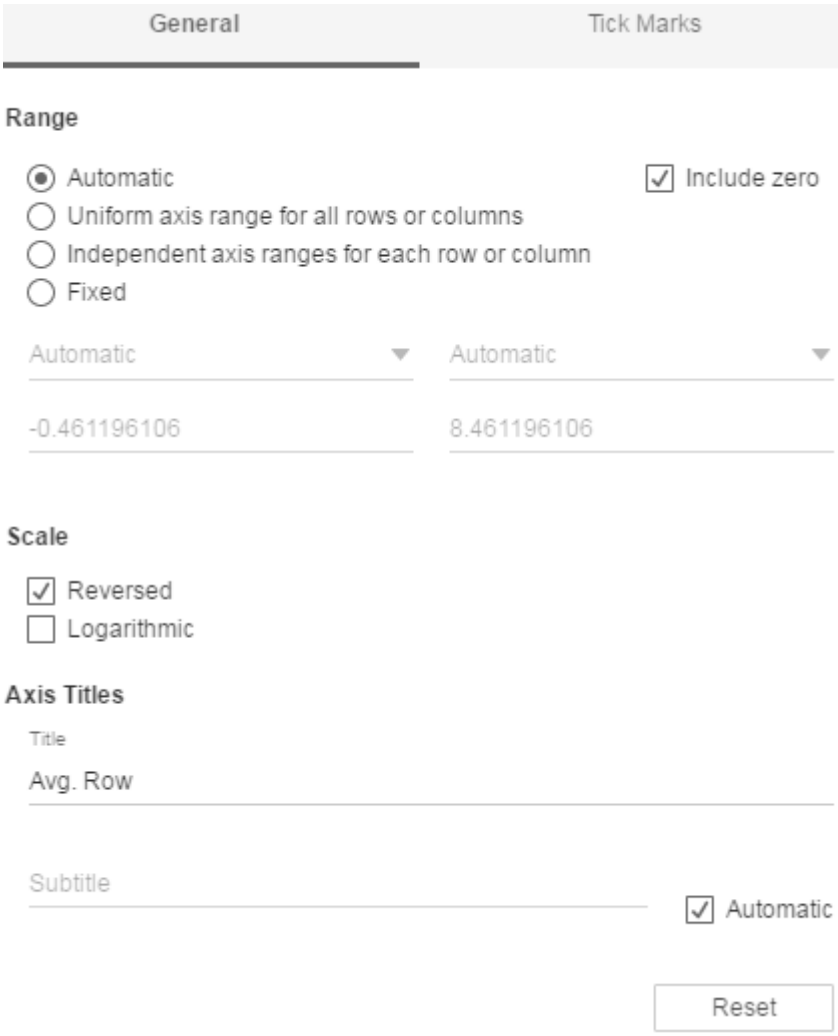


3. Add `Column` to the `Columns` shelf, add `Row` to the `Rows` shelf, and use `AVG` for the aggregation:

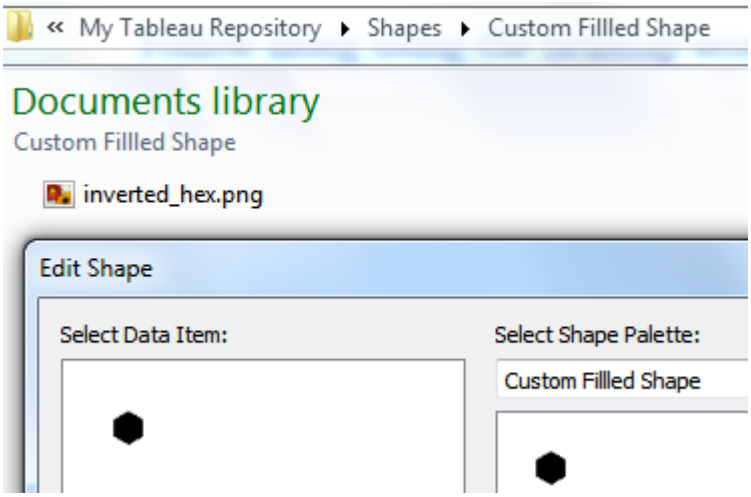
4. Use `Abbreviation` as the `Label`:



5. Edit the row axis and reverse the scale:



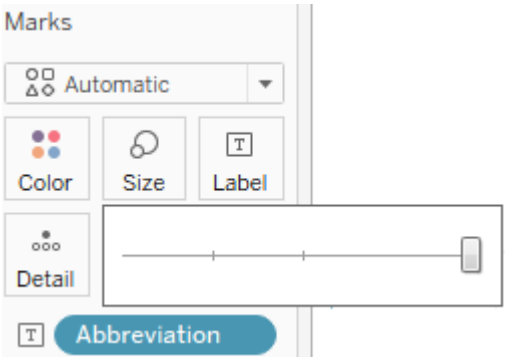
6. Use the `inverted_hex.png` image as a custom shape:



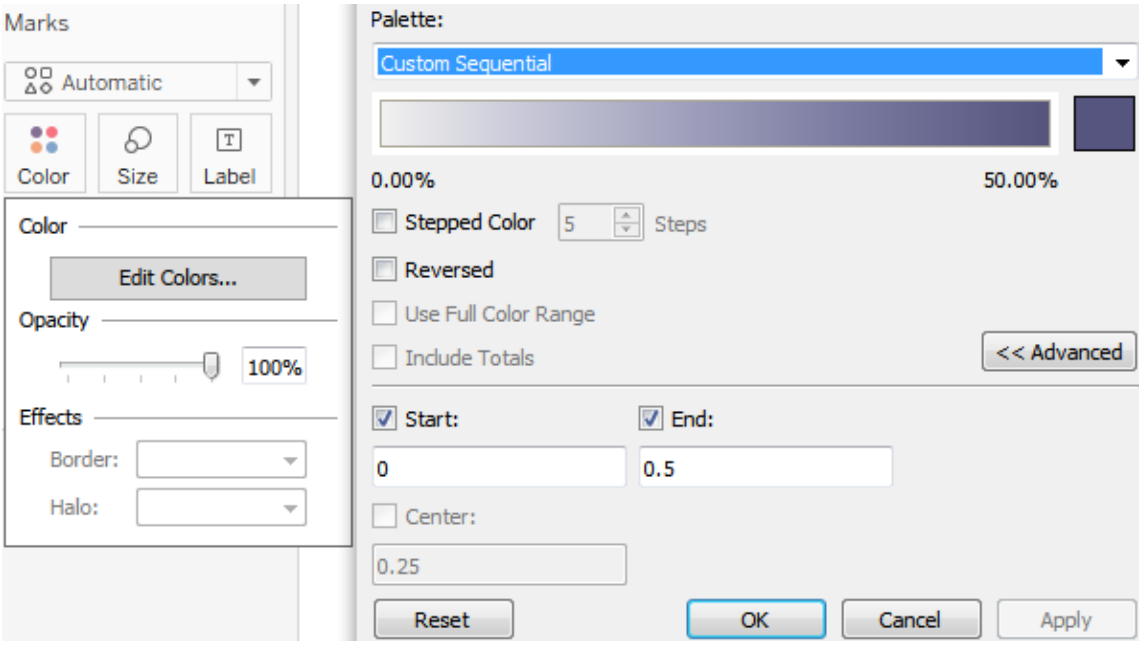
Note

Save the `inverted_hex.png` image to your `Tableau Repository` | `Shapes` | `Custom Filled Shape` folder.

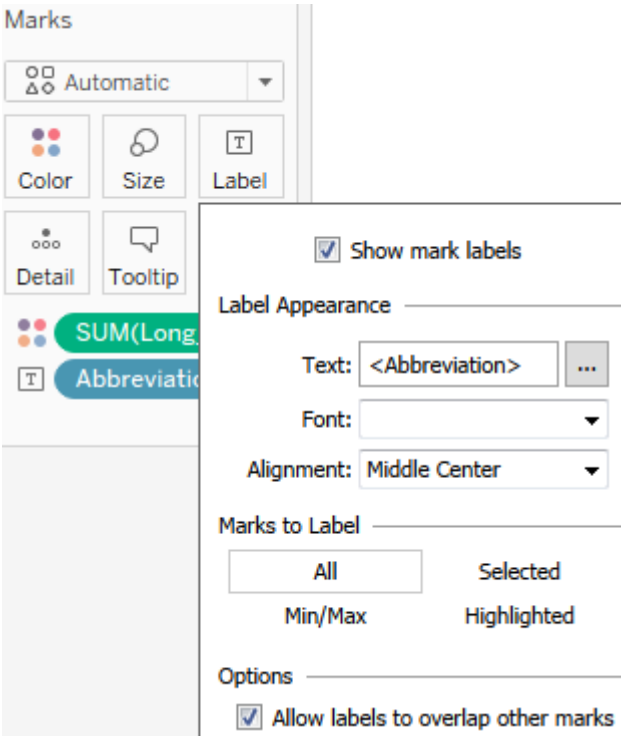
7. Adjust the **`Size`**:



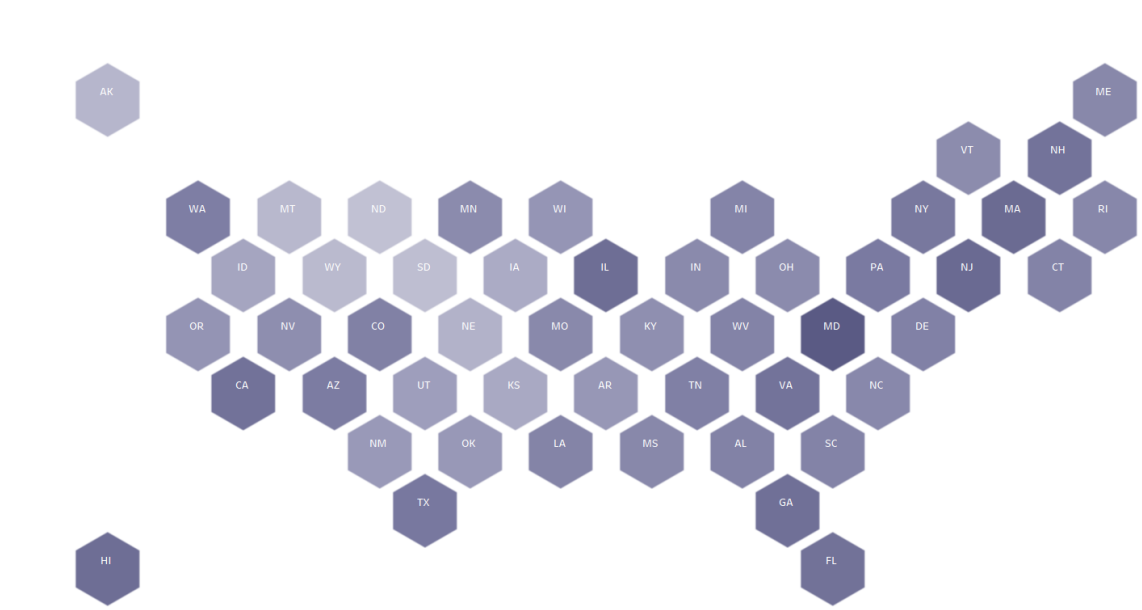
8. Color and shade the tiles by a measure in your data file. In our visualization, we're using **`Longest_Commute_Driving_Alone`**:



9. Add state labels, by putting **`Abbreviation`** on **`Label`**:



After some formatting, we can see the final visualization in the following screenshot:



How it works...

The key to the [*Hex-Tile maps*] recipe was using the `hexmap_plot` data and the tile image. The column and row values position the tiles so they were arranged in a similar, relative place as each state. The custom shape helped tie the visualization together in a compact way.

See also

Here are some examples of user community hexmap plots for other countries:

- https://revizited.com/how-to-create-hex-tile-map-for-india-in-tableau/
- https://www.sportschord.com/single-post/2018/02/12/Maps-in-Tableau-Part-1---UK-Hex-Tile-Map

Waffle charts

Waffle charts have almost similar use cases to Donut charts. They are used to show how items contribute to a whole. They are best used when comparing only a few categories.

Getting ready

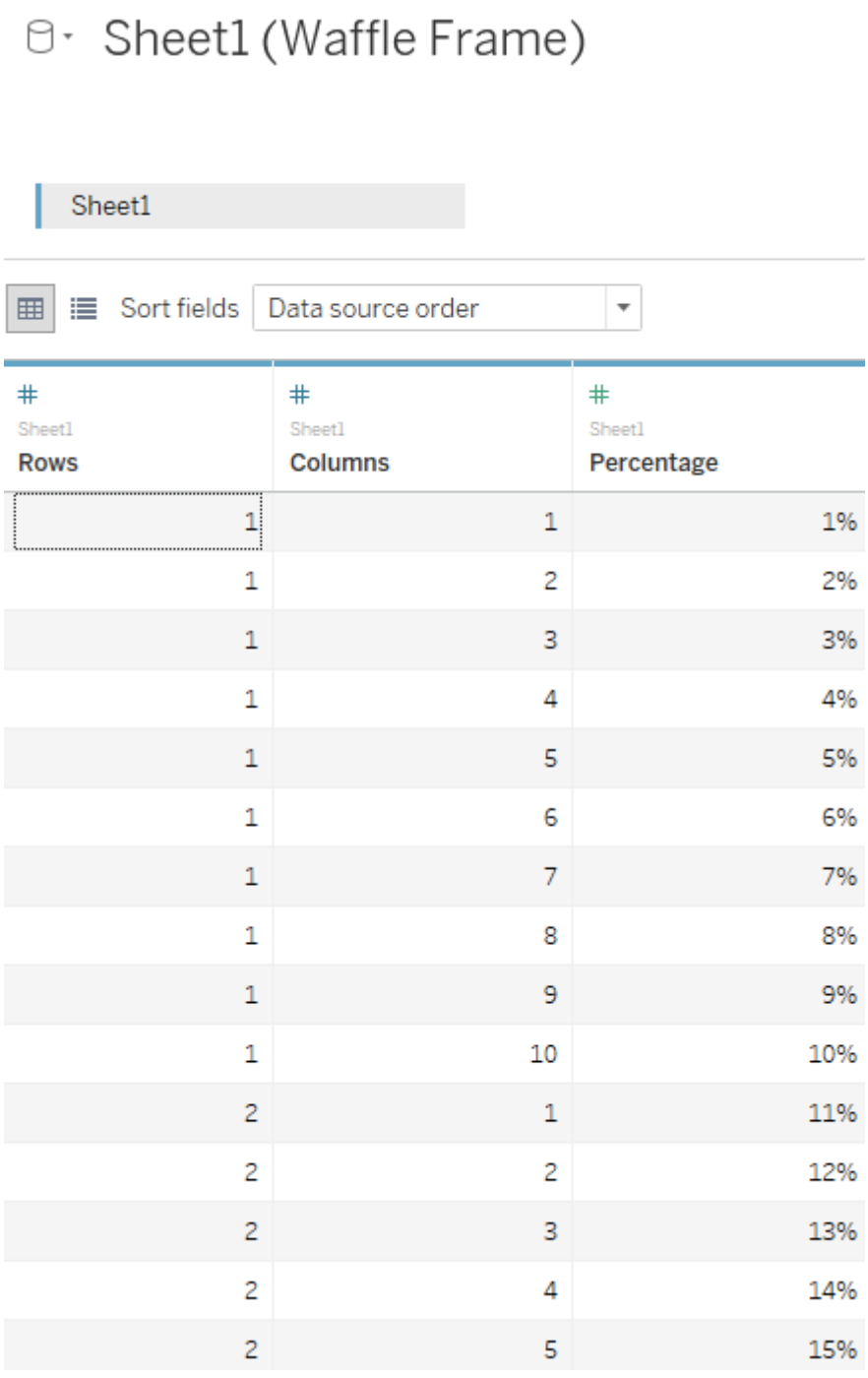
In this recipe, we will create a Waffle chart using a text table as the foundation.

How to do it..

To follow along, open the Waffle chart packaged workbook. In this example, we are looking at word counts for each Lord of the Rings character by race:

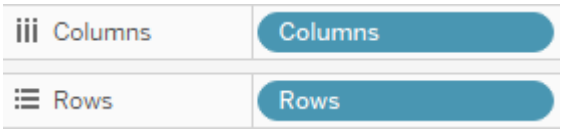
1. Use the `Waffle frame` Excel sheet:
    - This Excel sheet is 100 rows representing each percentage point. Because we want to create a frame of 100 squares in a 10 x 10 frame, we have columns called `Rows`, `Columns`, and `Percentage`. Each row and column has a value of 1 through 10, repeating. This creates a 10 row

x 10 column square. The first 20 are displayed in the following screenshot:



Sheet1 Rows	Sheet1 Columns	Sheet1 Percentage
1	1	1%
1	2	2%
1	3	3%
1	4	4%
1	5	5%
1	6	6%
1	7	7%
1	8	8%
1	9	9%
1	10	10%
2	1	11%
2	2	12%
2	3	13%
2	4	14%
2	5	15%

2. Add `Columns` to the `Columns` shelf and `Rows` to the `Rows` shelf. Because we want these values to be grouped in discrete buckets, we have set these to discrete. This can be achieved by right-clicking and choosing discrete, as opposed to continuous:



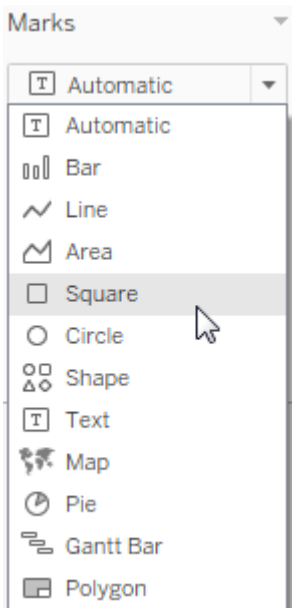
3. Add percentage to `Label` to see the layout:

	Columns									
Rows	1	2	3	4	5	6	7	8	9	10
10	91%	92%	93%	94%	95%	96%	97%	98%	99%	100%
9	81%	82%	83%	84%	85%	86%	87%	88%	89%	90%
8	71%	72%	73%	74%	75%	76%	77%	78%	79%	80%
7	61%	62%	63%	64%	65%	66%	67%	68%	69%	70%
6	51%	52%	53%	54%	55%	56%	57%	58%	59%	60%
5	41%	42%	43%	44%	45%	46%	47%	48%	49%	50%
4	31%	32%	33%	34%	35%	36%	37%	38%	39%	40%
3	21%	22%	23%	24%	25%	26%	27%	28%	29%	30%
2	11%	12%	13%	14%	15%	16%	17%	18%	19%	20%
1	1%	2%	3%	4%	5%	6%	7%	8%	9%	10%

4. Order the `Rows` column in descending order:

	Columns						
Rows	1	2	3	4	5	6	7
10	91%	92%	93%	94%	95%	96%	97%
9	81%	82%	83%	84%	85%	86%	87%

5. Remove the percentage from `Label` and change the mark type to `Square`:



6. Create the actual share of words spoken by the Hobbits. Go the `WordsByCharacter` dataset and create a `Hobbit` calculation based on `Race = Hobbit` for percent of total spoken words:

Hobbit | WordsByCharacter

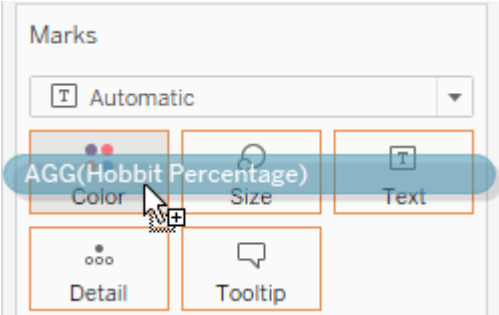
```
SUM(IF [Race] = "Hobbit" THEN [Words] END)/SUM([Words])
```

7. Create a true/false calculation to indicate whether the `Hobbit` is greater than or equal to each square in the Waffle chart. The **`Hobbit Percentage`** calculation allows us to color each square:

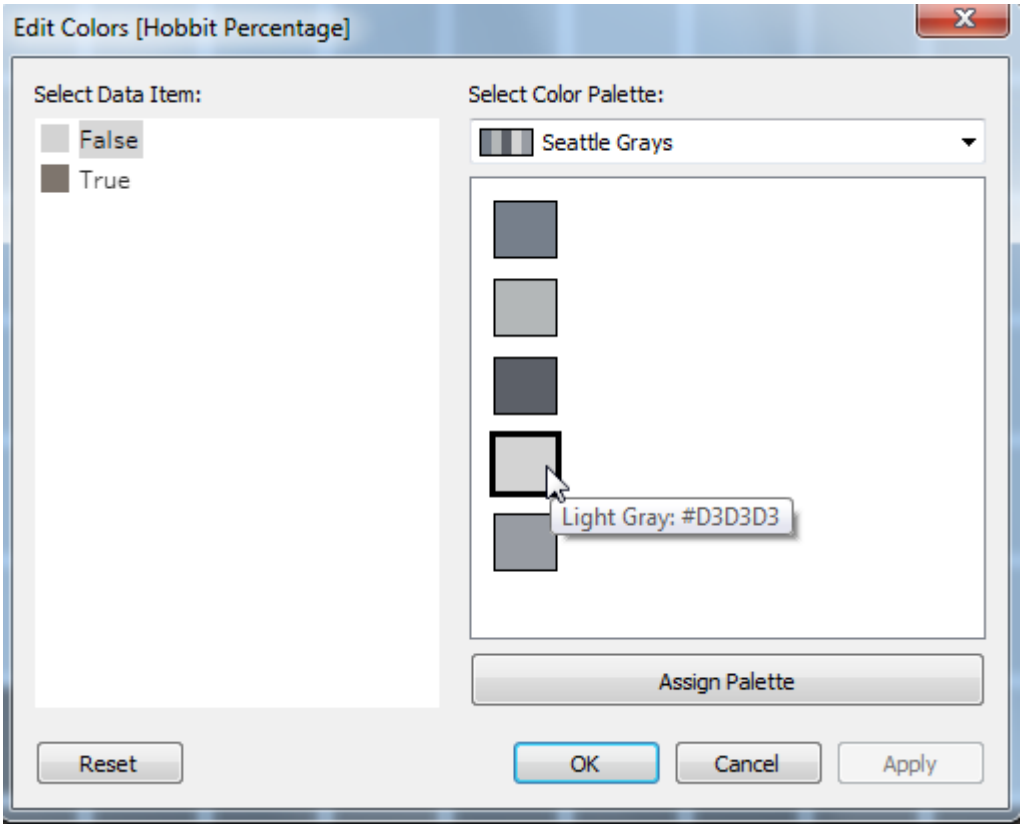
Hobbit Percentage | Sheet1 (Waffle Frame)

```
[WordsByCharacter].[Hobbit] >= sum([Percentage])
```

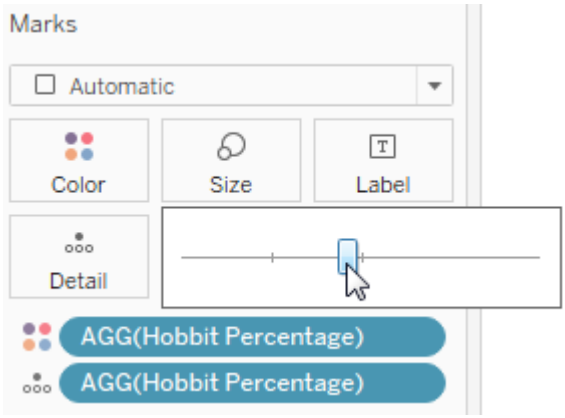
8. Apply **`Hobbit Percentage`** to **`Color`**:



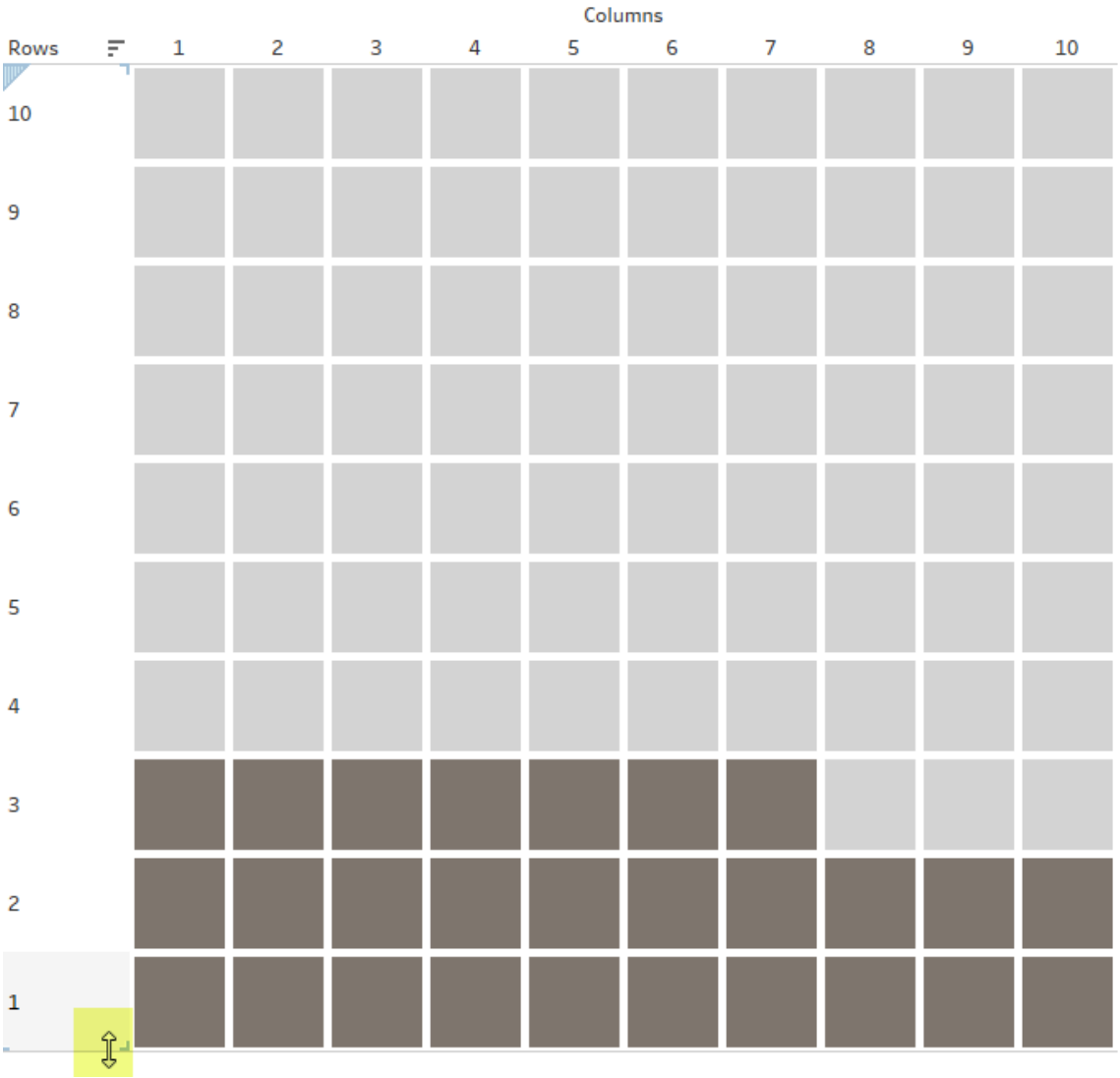
9. Adjust the color so that false is faint and subtle:



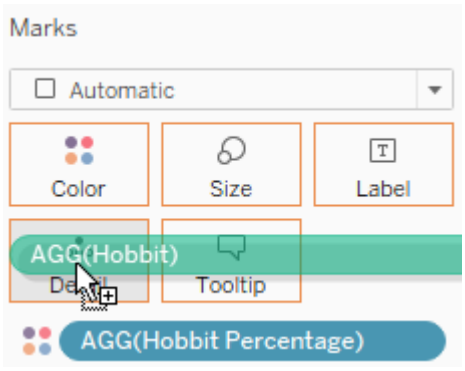
10. Adjust the mark size as follows:



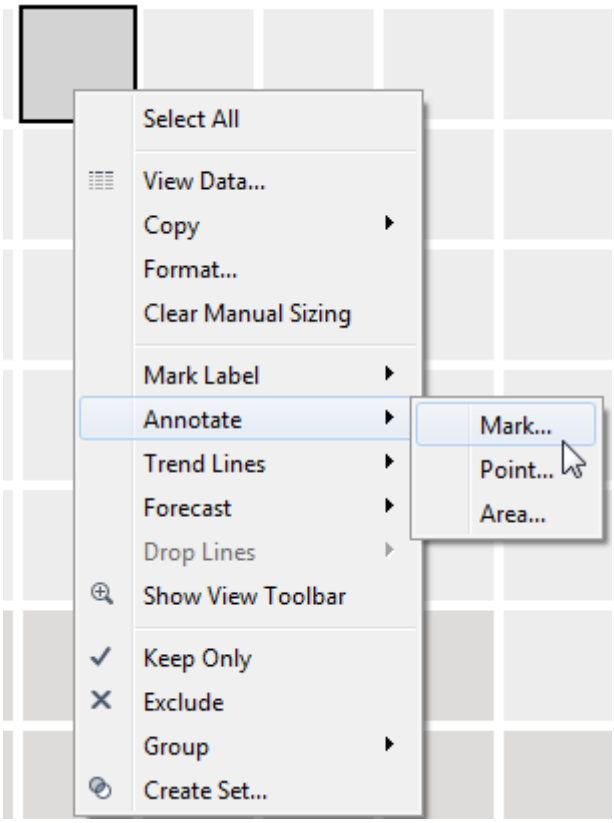
11. Adjust the chart size by manually adjusting the columns and rows:



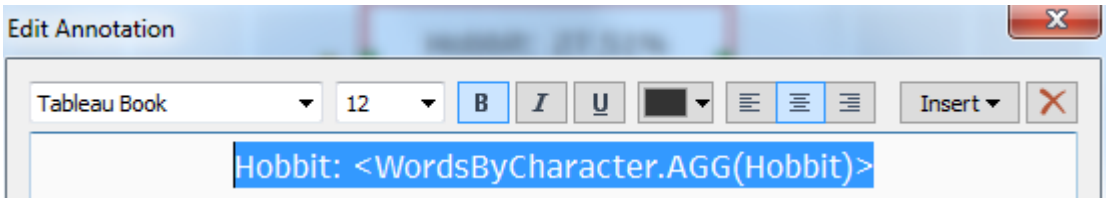
12. Add `Hobbit` to details, so we can annotate the chart:



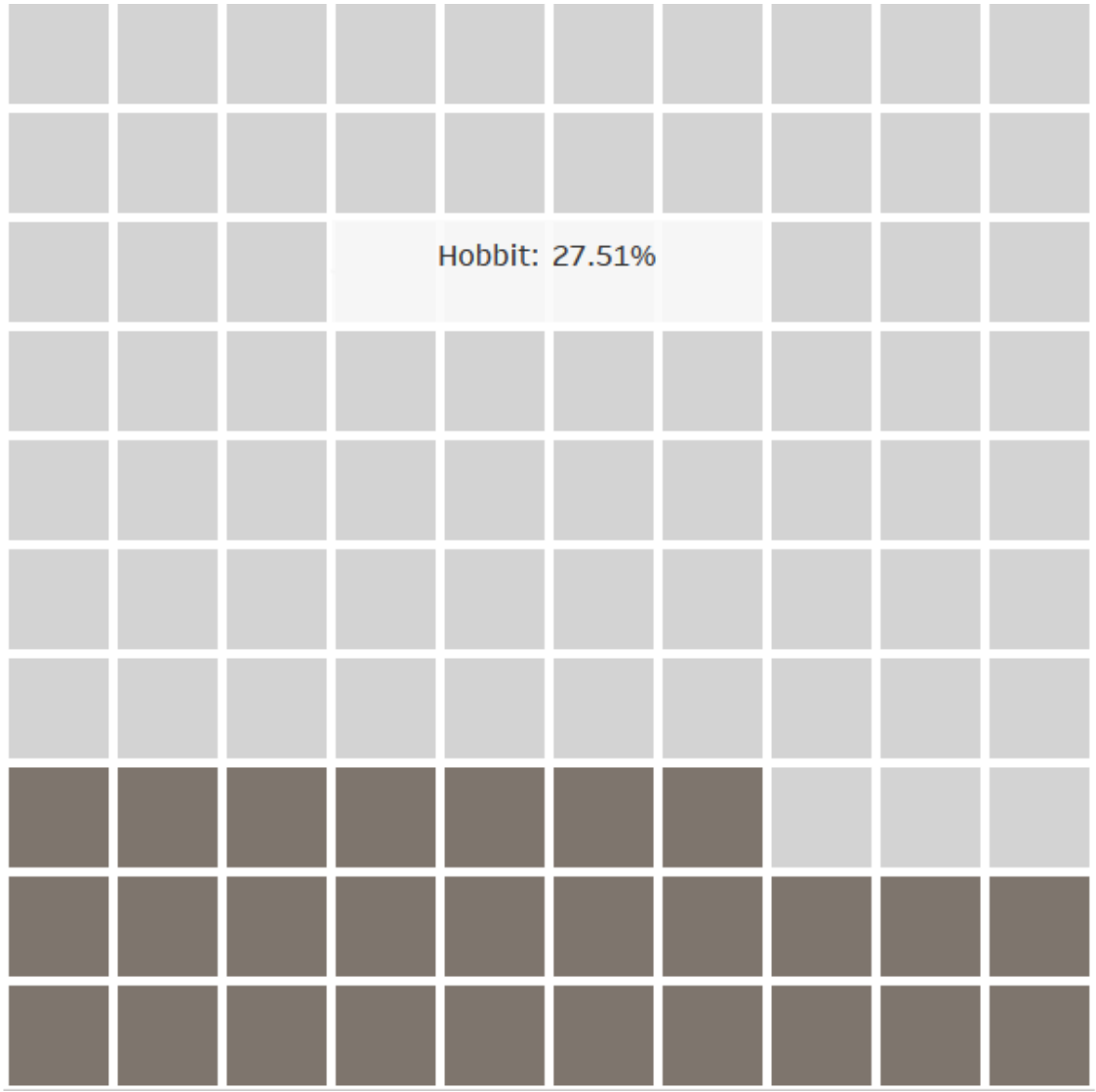
13. Annotate the Waffle chart as follows:



14. Modify the annotation as shown in the following screenshot:



15. After formatting and hiding headers, the final Waffle chart should look like this:



How it works...

We used the Excel 10 row x 10 column grid to represent a 100% grid, and put columns and rows on their respective shelves. We then changed the mark type to `Square`. We colored the tiles by creating percent of total for each category, in this case for race. In order to emphasize each category, we checked whether the value is less than the percentage value in our Waffle frame, which drives the color. We also changed the size of each square to get a nice waffle shape. Finally, we annotated the Waffle chart to make it easier to read.

There's more...

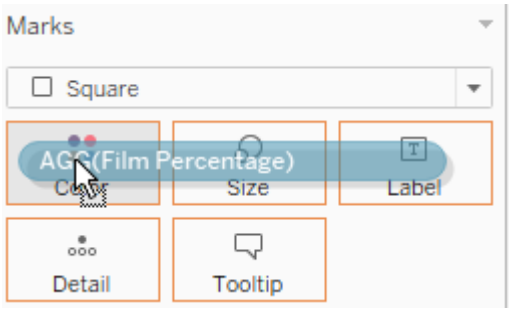
We can create more complicated calculations to represent more than one category in each Waffle chart. In a new sheet, starting with columns and rows in their respective shelves, continue with the following steps:

1. Create a calculation that will color every square in the Waffle chart depending on what percentage of words came from which movie:

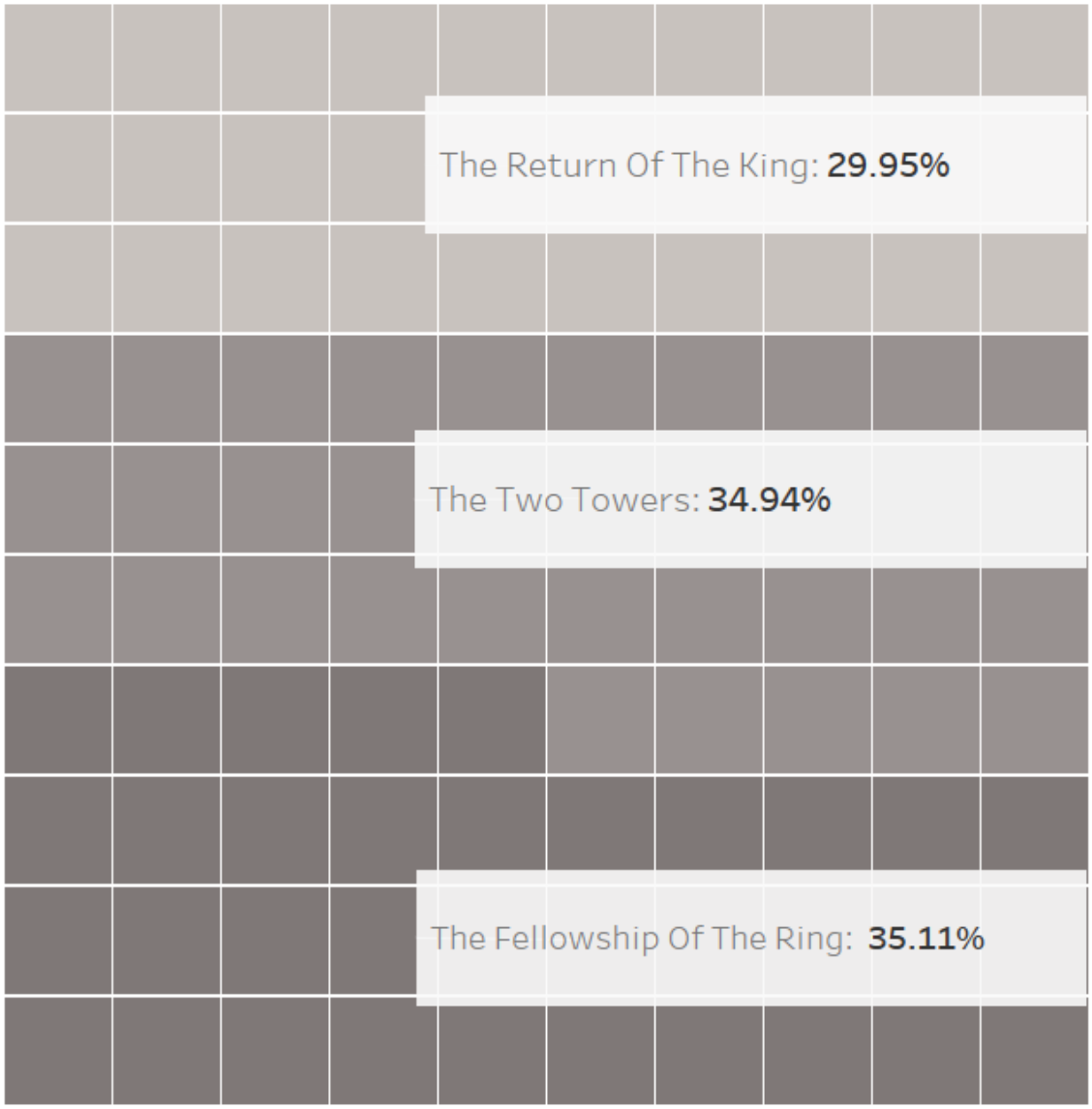
Film Percentage | WordsByCharacter

```
IF AVG(([Sheet1 (Waffle Frame)].[Rows]*10)-(10-[Sheet1 (Waffle Frame)].[Columns]))
    <= ROUND(([Film, Fellowship])*100) THEN "The Fellowship Of Ring"

ELSEIF AVG(([Sheet1 (Waffle Frame)].[Rows]*10)-(10-[Sheet1 (Waffle Frame)].[Columns]))
    <= ROUND(([Film, Fellowship] + [Film, Towers])*100) THEN "The Two Towers"

ELSE "The Return Of The King"

END
```

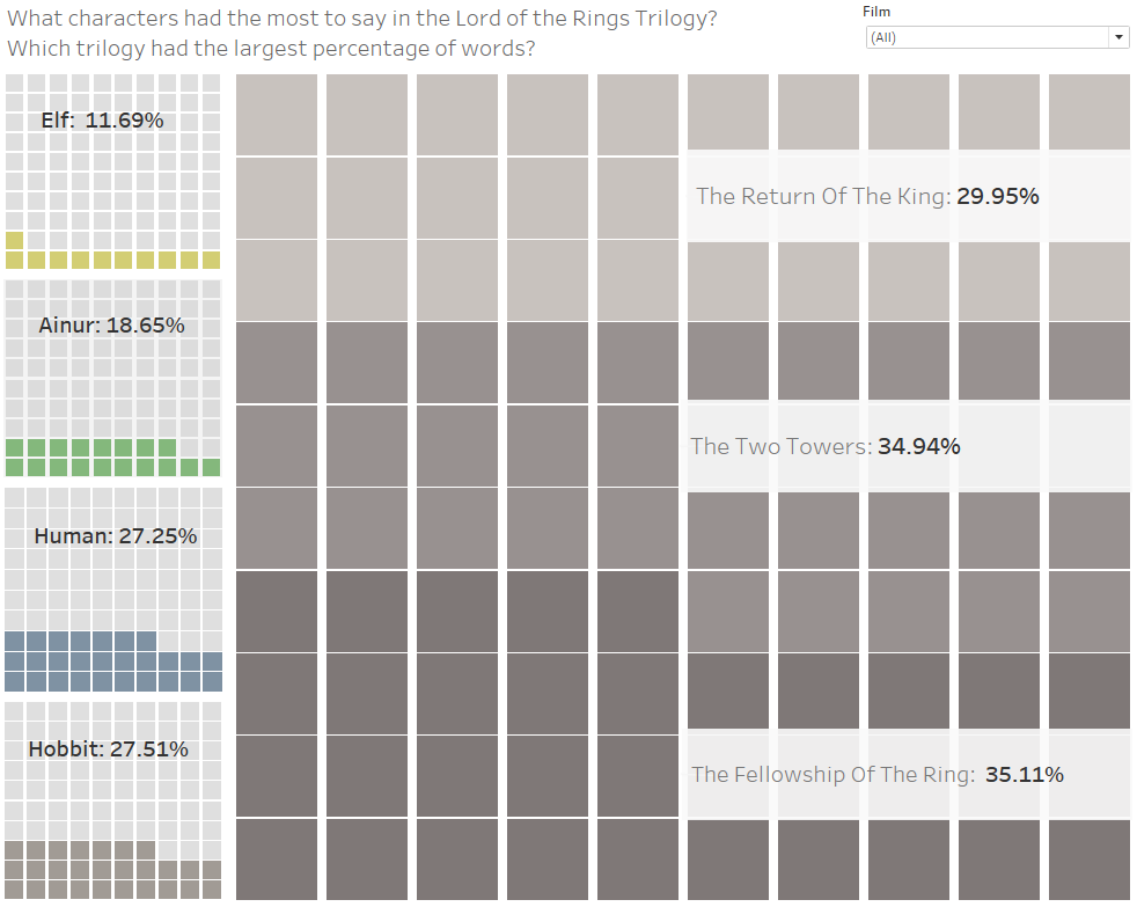
2. Add this calculation to `Color`:



3. Annotate the chart:



An example of a dashboard, after some cleanup and formatting, can be seen in the following screenshot:



See also

- See use cases for Pie charts (https://onlinehelp.tableau.com/current/pro/desktop/en-us/buildexamples_pie.htm) and Bar charts (https://onlinehelp.tableau.com/current/pro/desktop/en-us/buildexamples_bar.htm), or the lab for Donut charts. They are used in a similar manner to Waffle charts.