



# BLUE GRANITE ACADEMY

*Tableau is an incredibly versatile tool; unlike many other business intelligence tools, Tableau does not constraint users too much, and instead gives them power to exercise their creativity when developing visually rich reporting.*

**There are many tips & tricks** the Tableau user community has collected over the last few years. The 10 tips below are some of my own favorite, given they deal with very common business scenarios.

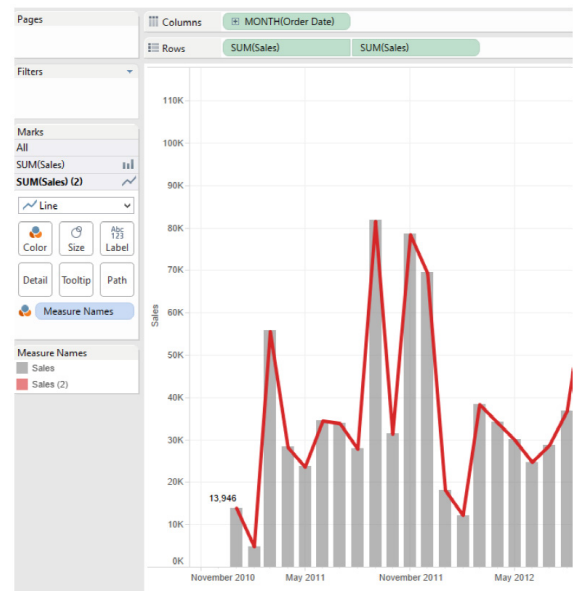
Javier Guillen  
*Solution Architect, BlueGranite*



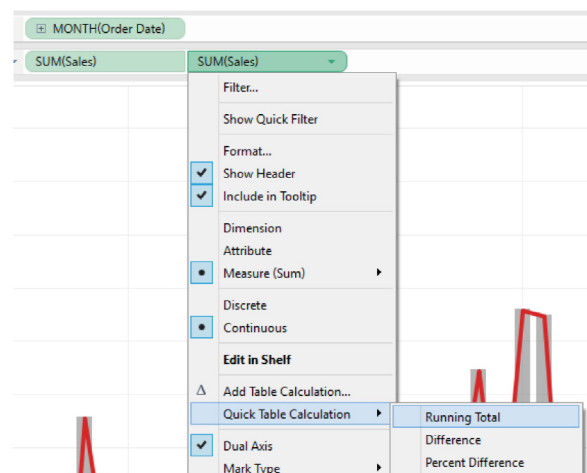
**Note:** We will use a modified Sample Superstore Excel source for these examples. All tips, except # 7, use only one table of data: **Orders**.

## TIP # 1: RUNNING TOTALS

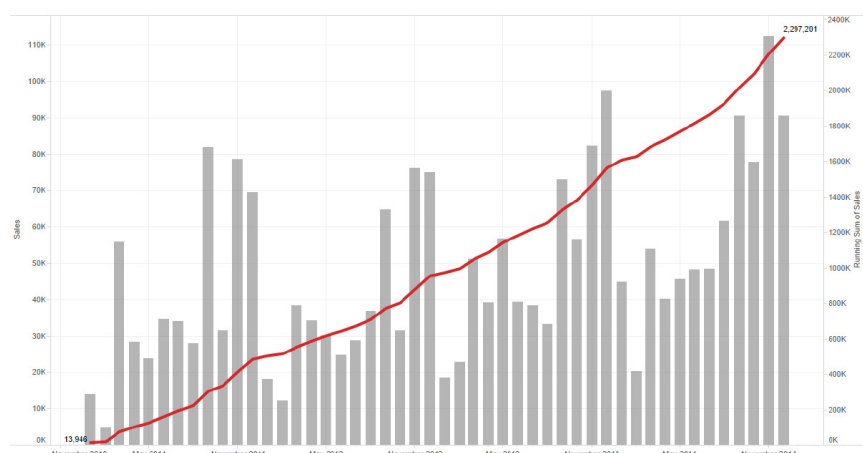
Create a visualization using the Year/Month level of the Date Fields. Add the Sales measure twice, and use a Bar for one mark, and Line for the other one.



To see a “running total since inception”, simply add a “Running Total” quick table calculation to the second sales measure:

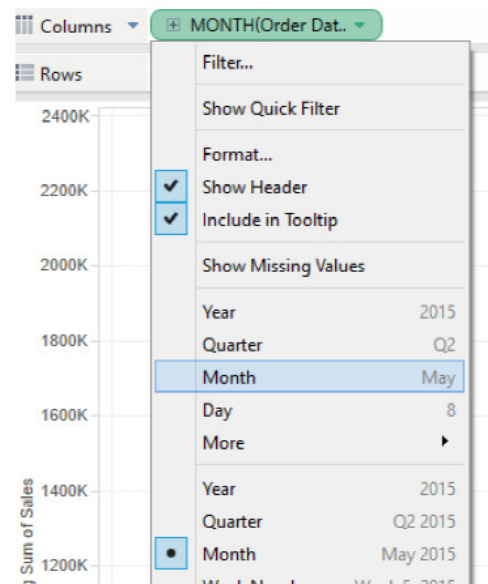


With the following result:

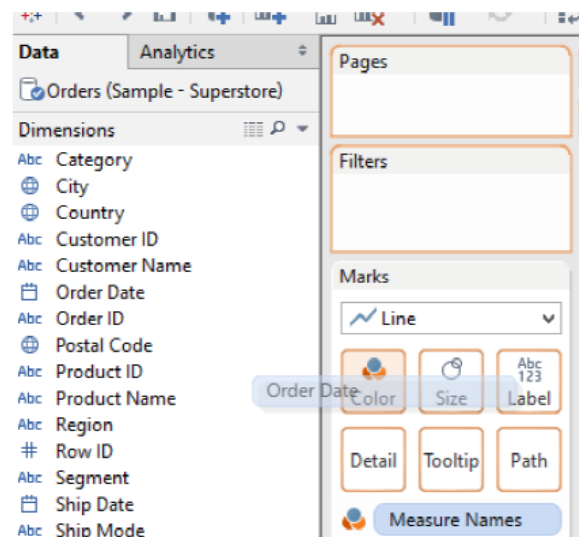




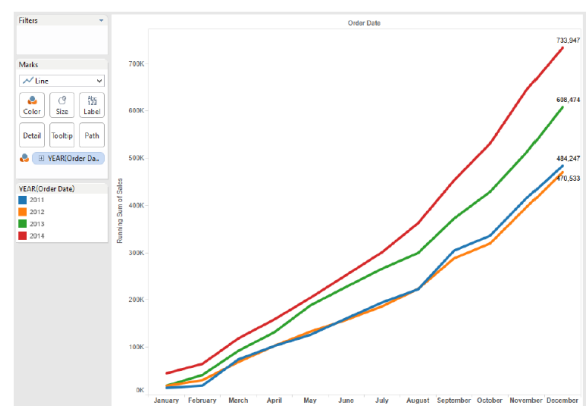
Note this is a dual axis chart. If we want to see YTD running total, and compare all years, we can remove the first Sales measure and change the Order Date detail from “Year/Month” to “Month”:



This will show one line that is the aggregate of all years. To split the line by year, drag a new Order Date field into the color property of the mark:



With the following result, which allows you to compare overall YTD performance across all years:



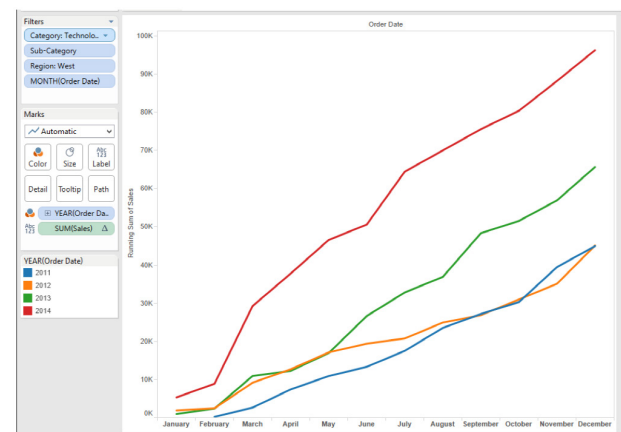


## TIP # 2: COMMON BASELINE

What if you want to compare YTD performance but not all your data elements start on a common date? Take a look at this example:

Region	Year of Order	January	February	March	April	May	June	July	August	September	October	November	December
West	2011		240	2,850	7,411	10,913	13,332	17,518	23,522	27,178	30,300	39,457	44,954
	2012	1,920	2,475	9,179	12,685	17,111	19,402	20,755	24,931	26,965	30,979	35,084	45,090
	2013	998	2,399	10,937	12,277	16,936	26,440	32,815	36,890	48,333	51,547	56,915	65,679
	2014	5,326	8,905	29,257	37,753	46,563	50,590	64,461	70,062	75,577	80,389	86,197	96,262

Notice data for 2011 only starts in February. By applying a running total, we would create a viz that looks like this:



Notice years 2011 and 2012 end up very close on December. If we are only interested in comparing product performance since market release, this chart could be misleading.

Create a new calculated field with a simple definition:

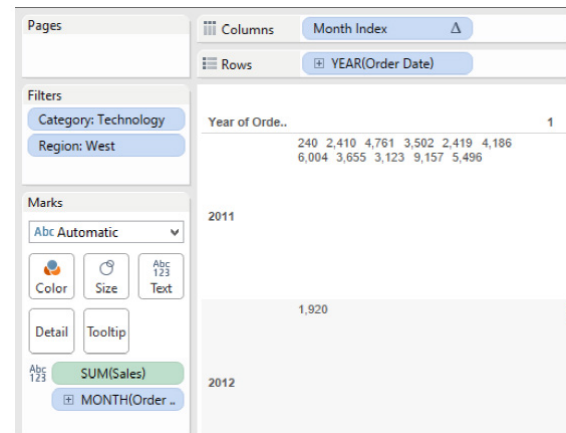
Month Index

Results are computed along Table (Across).  
`INDEX()`



This calculation will be used to give a “common” number on the chart axis such that the running total line starts on the same “period” for all years.

To see this in action, place a Year(Order Date) field on rows, Month(Order Date) field as the viz details, and the “Month Index” field on columns:



Notice the triangle on the “Month Index” pill on the columns shelf. This means Tableau automatically detected a table calculation.

We must edit the table calculation, and configure it to compute using Order Date and restart every year:

Table Calculation [Month Index]

Calculation Definition

Compute using: Order Date

At the level: Deepest

Restarting every: Year of Order Date

Description

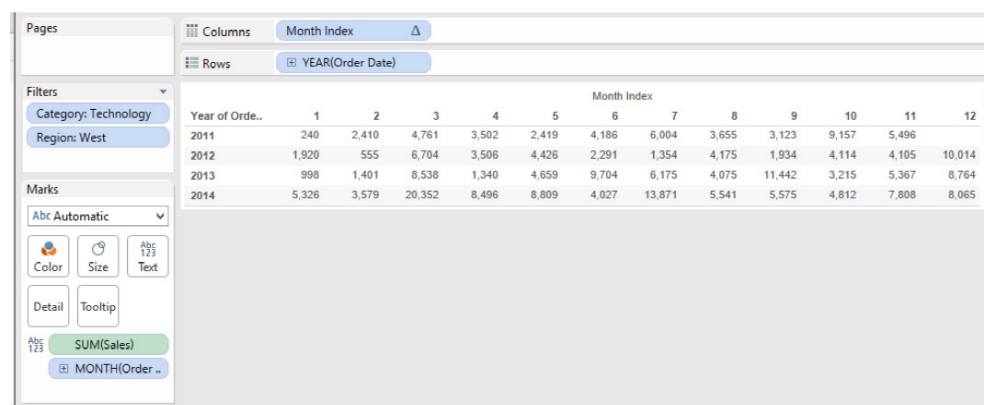
Results are computed along Month of Order Date, for each Year of Order Date.

Formula

INDEX()

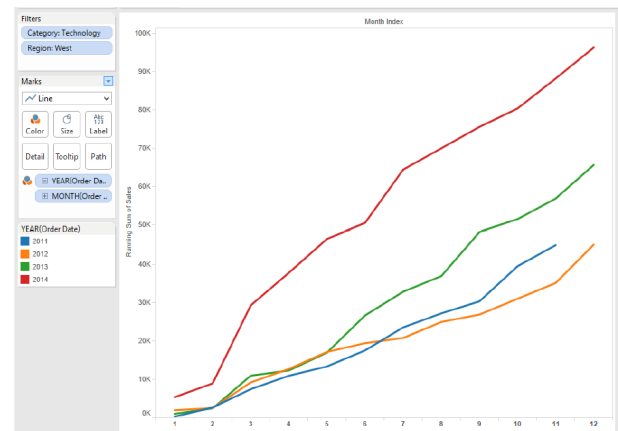
OK Cancel Apply

Notice how the table now shows 2011 starting at the same data point than every other year:



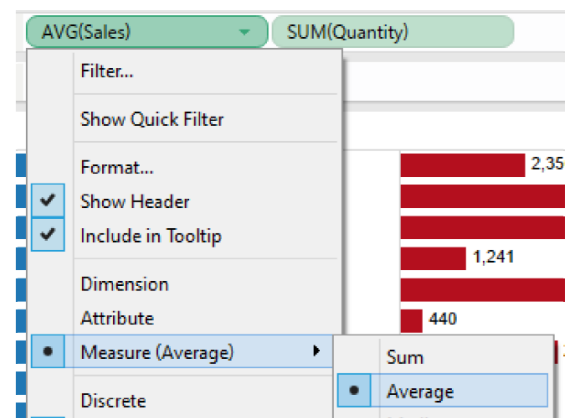


We can now compare the years based on a common baseline, and notice that in fact 2011 and 2012 did not converge on month 11:

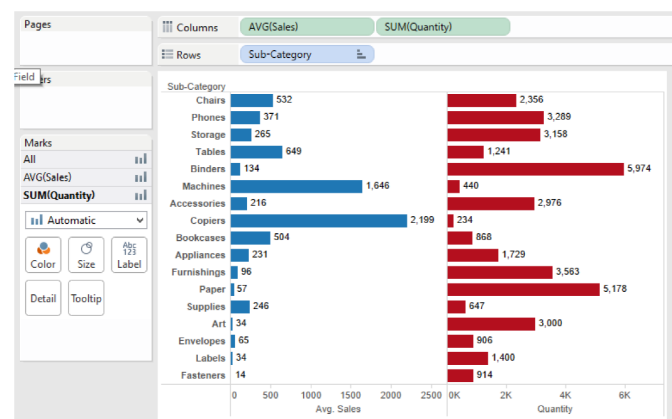


## TIP # 3: WEIGHTED AVERAGES

We can see what the average order sale was by placing the sales measure on columns and changing the aggregation from "Sum" to "Average":



If we compare to the amount of orders we notice that although some sub-categories had a high average order sale amount, they didn't move a lot of product (Copiers for example):





To get a better representation of average order sale amount, we can use a weighted average instead. The weight can be created as a new calculated field:

Weight

Results are computed along Table (Across).  
`SUM([Quantity])/ WINDOW_SUM( SUM([Quantity]) , FIRST(), LAST() )`

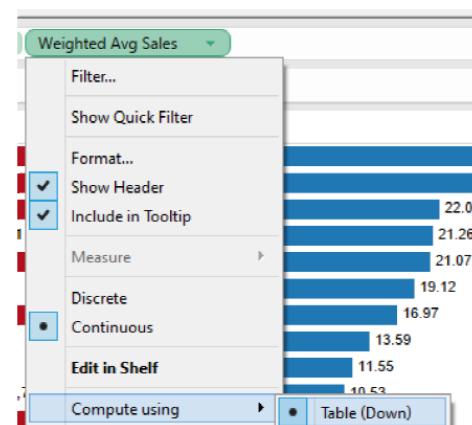
This calculation gives us a weighted factor comparing the quantity of any given sub-category (the numerator) against the quantity for all sub-categories (the denominator).

A second calculated field can be used to multiply the average sale amount by the weight factor:

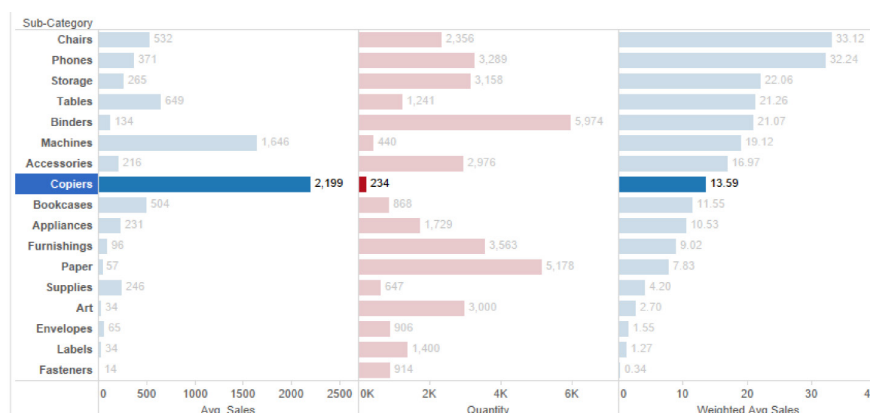
Weighted Average Order Sale Amount

`AVG([Sales]) * [Weight]`

Ensure that when placing it in the chart, the weighted average computes as "Table (Down)":



This calculation gives us a weighted factor comparing the quantity of any given sub-category (the numerator) against the quantity for all sub-categories (the denominator).

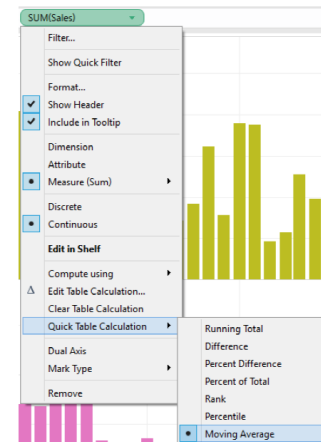




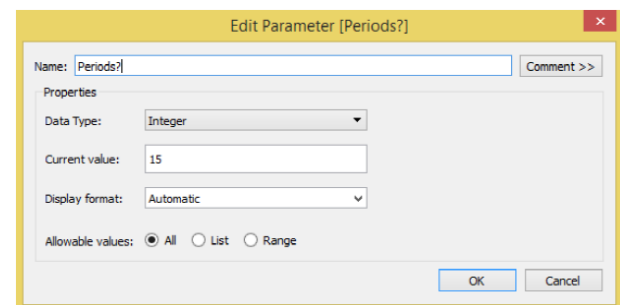


## TIP # 4: MOVING AVERAGE

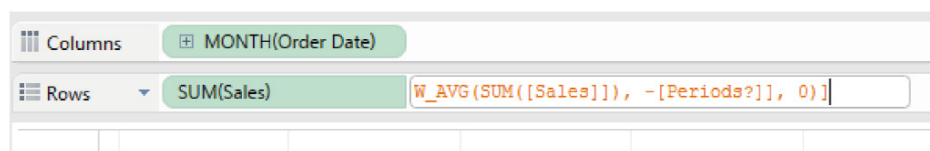
To get a better representation of average order sale amount, we can use a weighted average instead. The weight can be created as a new calculated field:



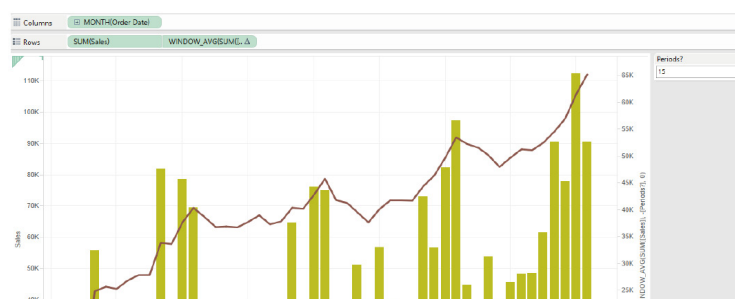
By default, it will assume three periods when averaging but you can change that by editing the table calculation. However, what if we want to give the user the ability to change the periods used? We can create a parameter:



Given we had originally created a moving average, we can leverage the automatically created calculation. In Tableau 9.0, we can now double click the pill to edit any pill on the shelf:



By changing the default "2" value with the parameter, we have now given users to control the moving average look-back periods:







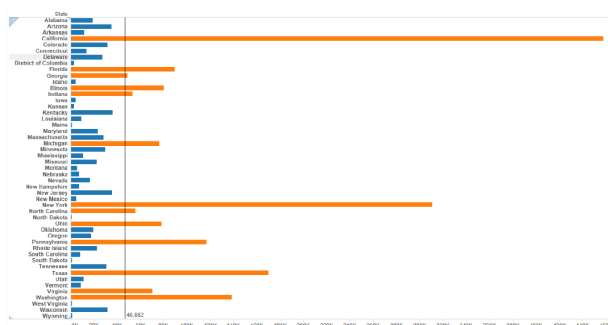
## TIP # 5: GROUPING BY AGGREGATES

Although we can add reference lines denoting averages, it is sometimes useful to be able to group dimensions based on their comparison against the overall average for all data points. To do this, we can again leverage a window calculation:

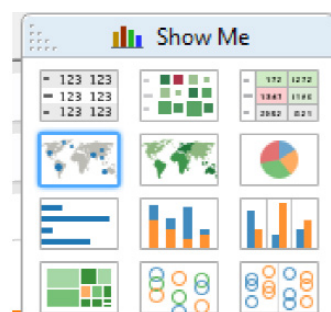
Essentially, we are comparing the sales for any given field against the average sales for all values. We can then use this calculated field, for example, in the color property to make cities with sales above average stand out:

Above Avg?

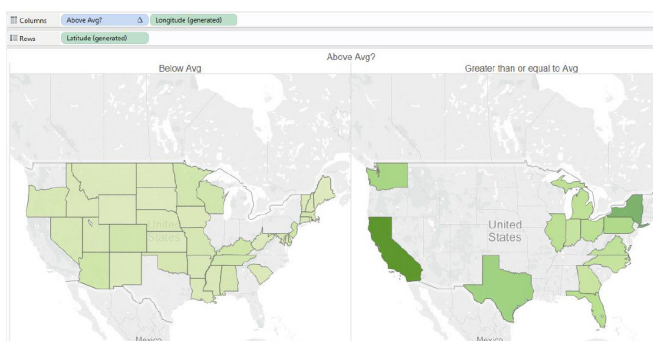
Results are computed along Table (Across).  
 IF  
 SUM([Sales]) < WINDOW\_AVG(SUM([Sales]), FIRST(), LAST())  
 THEN "Below Avg"  
 ELSE "Greater than or equal to Avg"  
 END



Given we are looking at cities, we could display this as a map too by using the "Show me" menu on the top right of the screen:



By moving the "Above Average?" calculated field to the columns shelf we can now clearly see which states had sales above or below average which helps with identifying patterns.

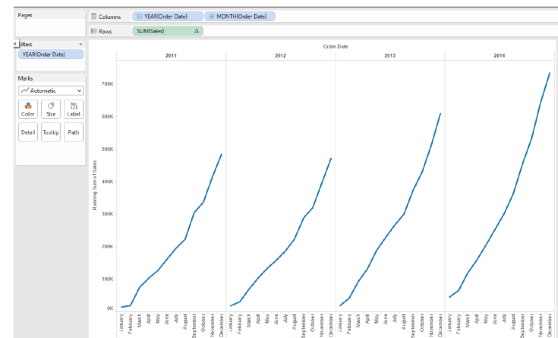




## TIP # 6: THIS YEAR VS LAST YEAR ANALYSIS

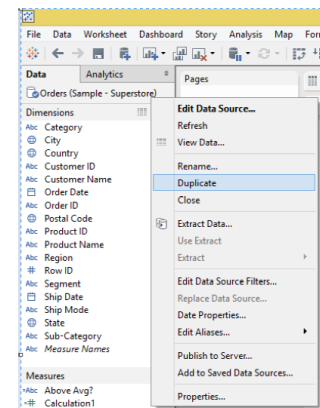
A very frequent comparison is TY vs LY. Although there are many ways to compute this in Tableau, the following tip is relatively simple and very elegant. The trick is to use **blending**.

Let's start with the viz below, using a running total by year –



Now, by right clicking the data source we can select “duplicate”. We do this because we want to have two reference to the data – one is the ‘regular’ reference, the other one is shifted by a year.

Notice this won't duplicate the actual data, it is only a new metadata reference:



It's a good idea to rename the new dataset, and the sales measure on it, by appending a descriptor, so it easy to keep track of fields in the viz.

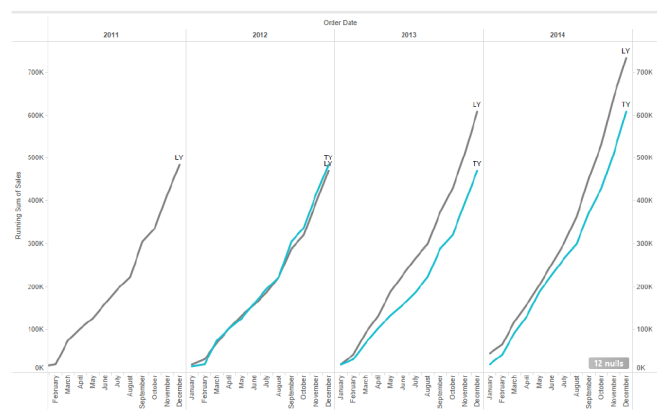
Notice the two data source references are linked by the Order Date, given we are using it in the canvas. We want to rename the Order Date in the new data source reference to “Order Date (shifted)”.



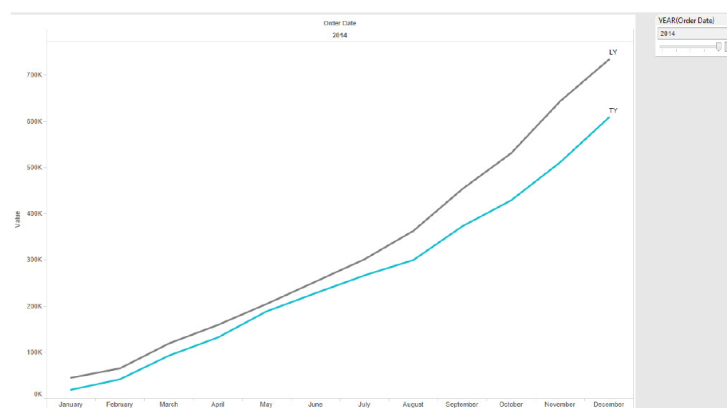
We can now create a new calculated field with the following definition:

Order Date   
`dateadd('year',1,[Order Date (shifted)])`

Given we are giving it the “original” Order Date name, the two data source reference blend back again automatically. With that we can now see the two sales measures (one for each data source reference) in the viz, denoting the current year and the last year for each pane:



To make the experience more user friendly, we can add a quick filter on Year and configure it as a slider:

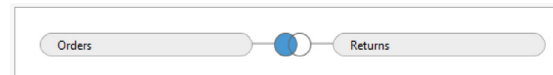




## TIP # 7: APPENDING EXCEL SHEETS

While creating Self-Service BI reports, it is common for data analysts to use Excel data sources. And in some cases, they must append data from multiple sheets: for example, when consolidating a dimension.

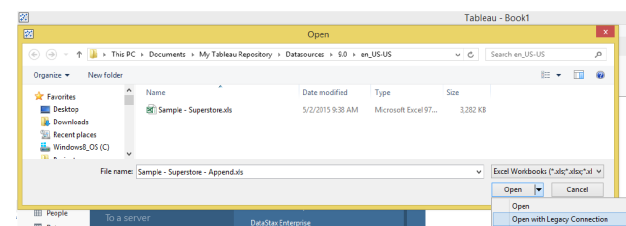
Typically, we want to join sheets and that is what Tableau does very well by default:



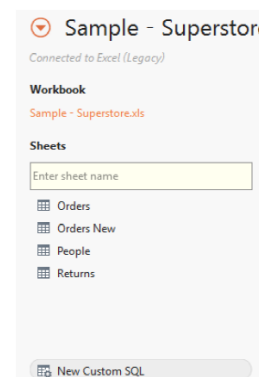
However, as an example, notice there is a 4th sheet in the Superstore datasource we have been using: "Orders new". This represents new orders coming in that did not exist on our original Excel extract.

To work with a comprehensive list of orders, old and new, we want to **append** (instead of **joining**)

To do this we can re-connect to the Excel source but this time we want to open it using a "legacy connection":



This gives the ability to execute queries against the data in the file. You will see a "add custom query" pill enabled:

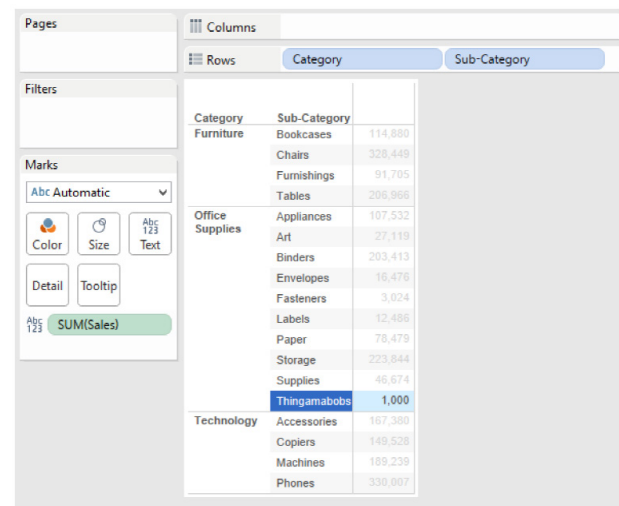




By dragging it into the query pane, we can now union (append) the two Order sheets:

```
SELECT * FROM [Orders$]
UNION
SELECT * FROM [Orders New$]
```

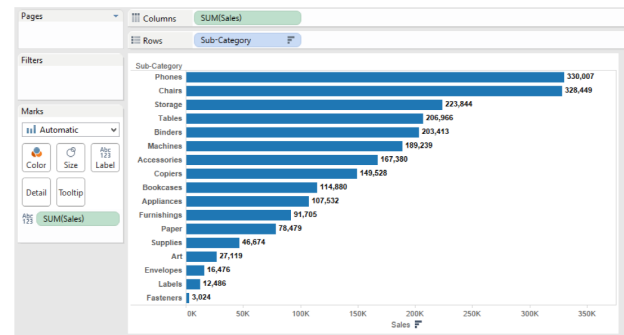
In the viz, we can now see we have a new 'Thingamabob' sub-category which wasn't part of the original Orders dataset. This allows to manage all Order dimension data in one place:



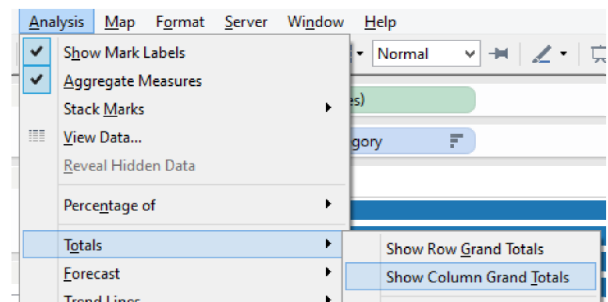


## TIP # 8: BAR CHART TOTALS

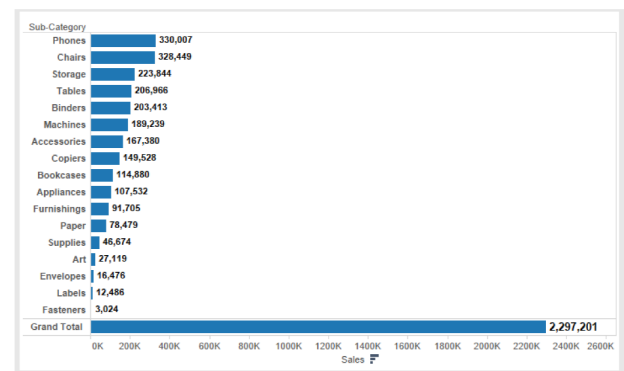
Bar charts are very intuitive and easy to read:



Sometimes, though, you want to see the grand total in the chart. You can turn these on using the Analysis menu:



Given the grand total is the highest value, the chart would become a bit 'cluttered' with a very large bar:



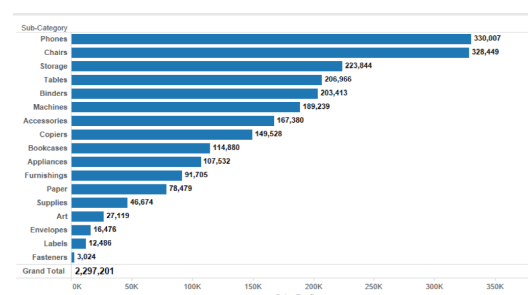
Likely, we don't want to show a bar in the total – only the total value. To do this, we can create the following calculated field:

The **SIZE** function returns all the rows in a current partition. Given that the total is technically not on the same partition as the details, we can detect how many rows are there. If we only find 1, we know we are in the grand total and can display a zero instead of sales amounts. Using this new measure in the viz will look like this:

Sales 2  Orders (Sample - Superstore)

Results are computed along Table (Across).

```
IF SIZE() = 1 THEN 0 ELSE SUM(Sales) END
```

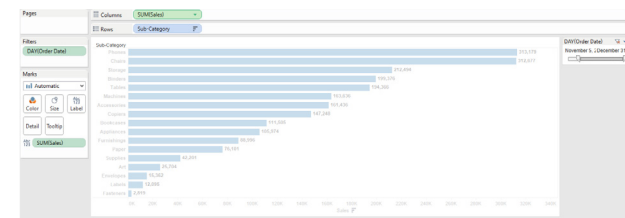
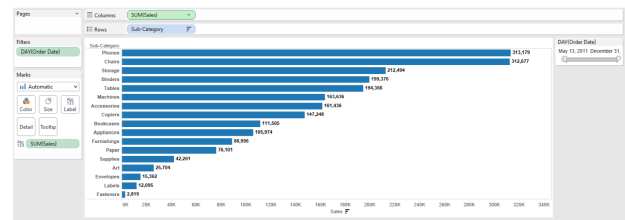




## TIP # 9: FIXED AXIS WHEN RE-DRAWING CHARTS

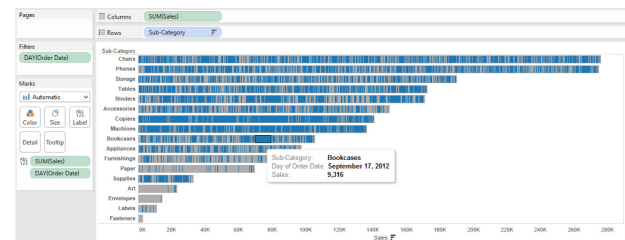
Using quick filters with a slider, for example on Order Date, are a great way to navigate date ranges:

As users select a new range in the quick filter, however, the chart blanks out:

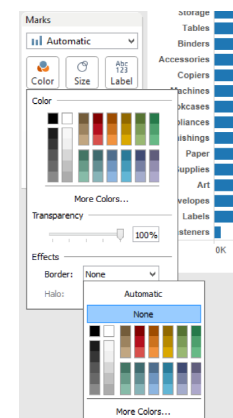


We can change the behavior to something a little more interesting: redrawing the chart as we make our selection. The added advantage of this technique is that it will "fix" the axis, so you can see the relative performance for any given date range against the rest of dates.

To do this, we can simply add the DAY (Order Date) field into the details. Doing so will break the bars and show total by day:



To return to the prior appearance we can turn off borders:



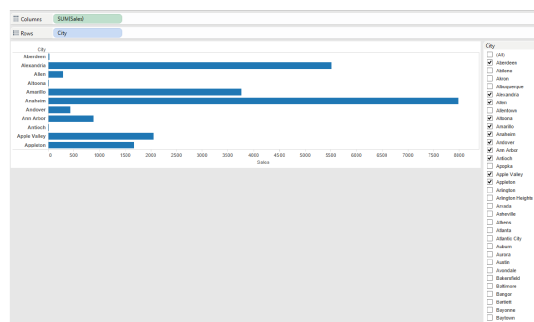
Now, every time a user moves the slider they will see an animated viz that can help understand the data as date ranges become narrower. Note: this effect will finish when the user releases the mouse.



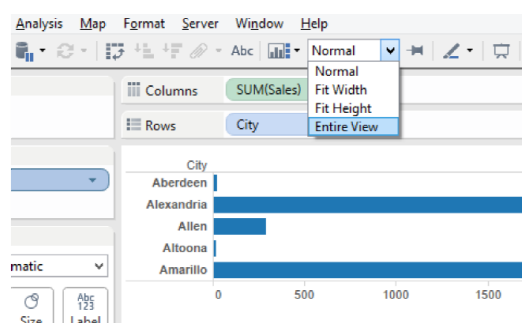


## TIP # 10: AUTO-FITTING SCREEN BEHAVIOR DEPENDING ON DATA SELECTION

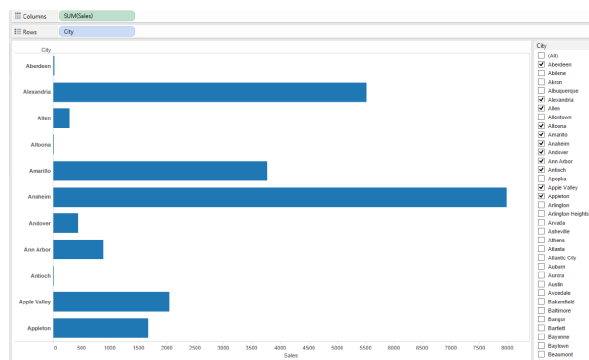
Let's say we are looking at sales by city:



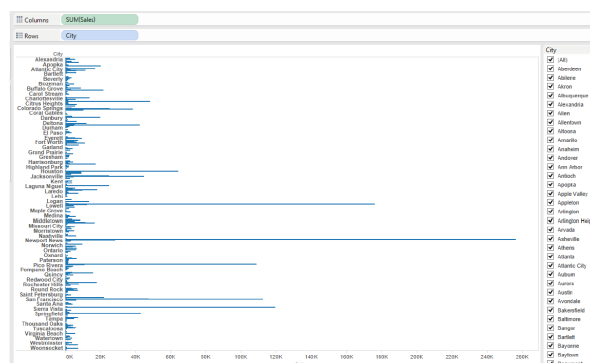
If are only looking at a few cities, there would be a lot of grey space at the bottom. It may be best to have the viz fit the entire view:



With the following result:



However, if somebody selects all cities they will now see a bit of a cluttered chart:

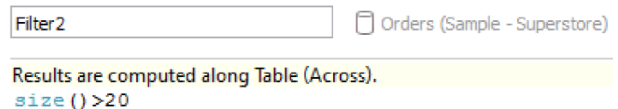
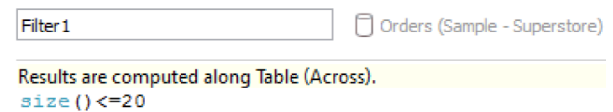




Wouldn't it be great if we could switch between views depending on how many items the user selects? Well, we can! If they select, say, less than 20 we can fit the entire view. Otherwise, we will keep the view as 'normal'.

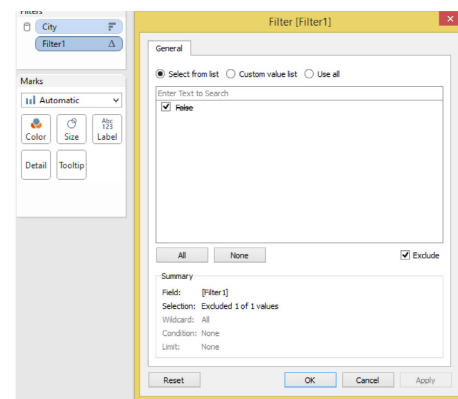
To do this we have to create two simple filters using the same technique we used before invoking the SIZE function:

*And*

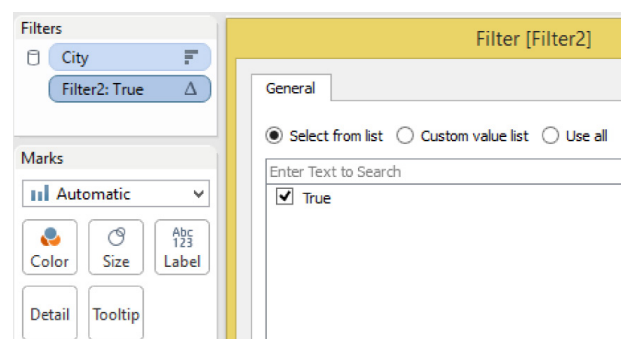


We then create two exact sheets, with the viz in question: one first the entire view, the other one fits normally.

For the entire view sheet we must use "Filter 1". We select the default "False" value and exclude it:

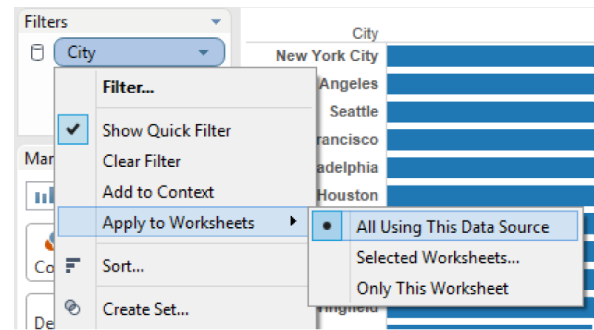


On the normally fitted view, we add "Filter 2" and select the "True" value (don't exclude it):

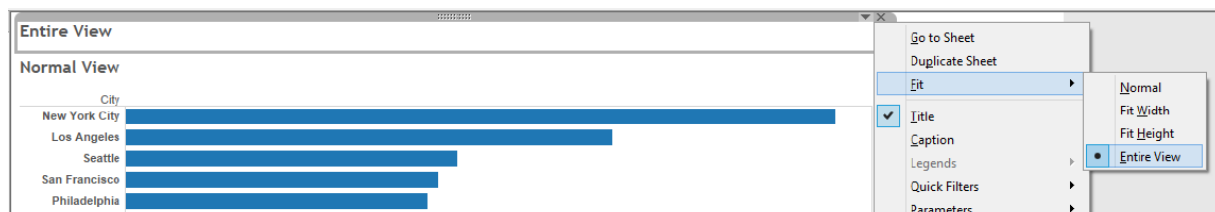




Notice the city filter is filtering both views.  
We can configure that on the filter pill:



Now we can add both sheets to a dashboard, and display the city quick filter.  
For the “entire view” sheet, change it to fit the entire view in the dashboard:



Notice that when all cities are selected the “normally fitted” sheet will show data, however when only a few cities are selected the “entire view” sheet will show data instead:

You could remove the headers at this point for a better user experience:



*The dashboard now will dynamically shift from a nicely fitted viz when only a few cities are selected, to displaying a toolbar to better scroll when many or all cities are selected.*