CHAPTER 5

Table Calculations

The purpose of visualization is insight, not pictures.

—Ben Shneiderman, computer scientist, distinguished university professor in the Department of Computer Science, University of Maryland, College Park, and founding director (1983-2000) of the University of Maryland Human-Computer Interaction Lab

Chapter 4 introduced us to two new fields, namely, measure names and measure values. We learnt to blend multiple measures on a single axis and to use a dual axis to enhance our presentation of data. We were also introduced to some new chart forms, such as slope graphs, combination charts such as bar and line graphs together in a view, lollipop charts, etc. This chapter will help us learn about table calculations that will be performed on measures plotted on the view. In this chapter we will explore the following table calculations:

- Running total of sales
- Percent of total
- Moving average
- Rank
- Level of detail (LOD)
- Percentile
- Year-over-year growth

5.1 What is a table calculation?

Consider a Tableau view (see Fig. 5-1a). For every Tableau view, there is a virtual table determined by the dimensions used in the view. The dimensions can be on the rows shelf, columns shelf, pages, and the marks card (color, size, label, detail and path), in other words, the dimensions within the level of detail.

S. Acharya and S. Chellappan, *Pro Tableau*, DOI 10.1007/978-1-4842-2352-9_5

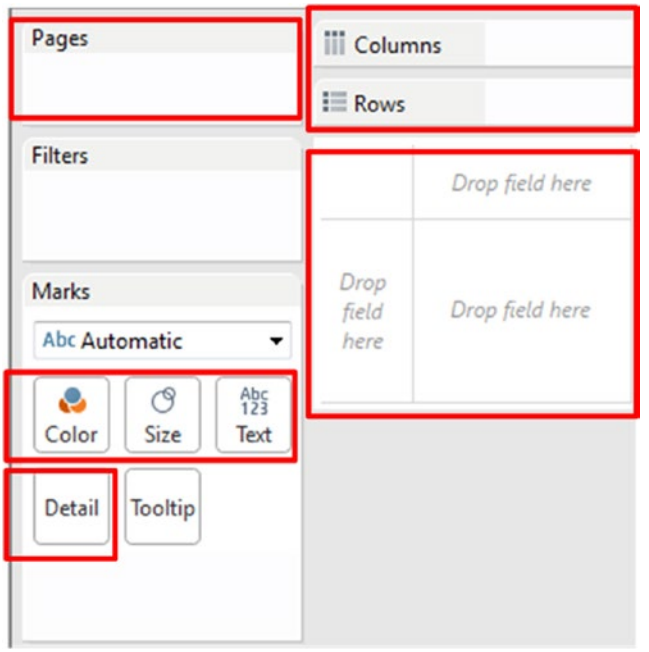


Figure 5-1a. *Dimensions can be placed on the level of details*

A table calculation is a calculation / computation that is applied to all values of a single measure in a view. Table calculations are the computational workhorse of Tableau. They calculate values outside the traditional realm of "Slice by X Dimension". Table calculations allow the user to extend their data. Table calculations are computations that are applied to all values in the entire table and are often dependent on the table structure itself. Example: Table calculations can be used to compute each month's contribution to annual profit.

There are two easy ways to work with table calculations:

- Use quick table calculation. Quick table calculations are a collection of commonly used table calculations (such as running total, difference, percent difference, rank, percentile, etc.).
- Create your own table calculations from scratch using table calculation functions.

Refer to Fig. 5-1b.

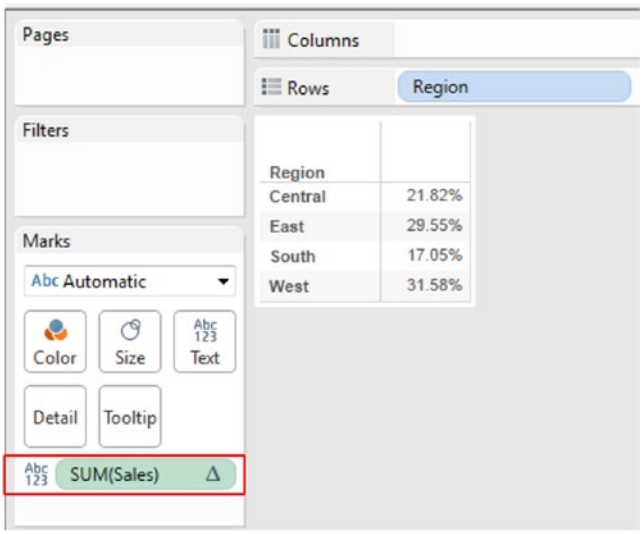


Figure 5-1b. *Table calculation applied on the measure "Sales". Notice the triangular mark next to SUM(Sales)*

In Figure 5-1b, **dimension** "Region" is placed on the rows shelf and m**easure** "Sales" is placed on the "Label" on the marks card. The table calculation, "Percent of Total" is applied to the m**easure** "Sales". The table calculation is "Percent of Total"; therefore, when all the cell values are added up, it aggregates to 100%.

A table calculation makes use of two fields: Partitioning and Addressing fields. In order to understand table calculations, it is important to understand how these fields work. They essentially define "what" a table calculation is and "how" they are performed.

Partitioning field: this field is used to partition the data into buckets. These data buckets are then acted upon by the calculations. In other words, they define the scope or grouping of the calculation. The scope can be the entire table, a pane, a cell, a dimension or it can be customized even further for more advanced calculations.

Addressing field: this field provides the direction in which we want our calculation to proceed. It defines the anchor or the source of each partition. It defines the root of the calculation.

Example: Compute the running total over a period of time (over years) partitioned by a segment. Here "segment" is the partitioning field and "date" is the addressing field.

■ **Note** The sequence in which tableau processes calculated fields, filters and table calculations:

1. Tableau generates a query and sends it for processing to the database.

2. The database processes the query. Tableau takes into consideration all calculated fields, including the level of detail calculations.

3. Lastly, the table calculations are applied.

5.2 Running Total of Sales

A running total is a summation of a sequence of numbers that is updated every time a number is added to the sequence. It is also referred to as "partial sum".

For example: we have a sequence of numbers "5, 2, 4, 7, 8". To get the running total, start by adding 5+2 to get 7. To this 7 add 4 to get 11, to 11 add 7 to get 18, to 18 add 8 to get 26.

5.2.1 Demo 1

Objective**:** To compute the "Running Total of Sales" (compute using Table Down).

Data set used**:** "Sample – Superstore.xls"

Expected Output**:** Shown in Fig. 5-2.

		Order Date							
		2011		2012		2013		2014	
Region	Quarter of O..	Sales	Running Sum	Sales	Running Sum	Sales	Running Sum	Sales	Running Sum
Central	Q1	8,601	8,601	11,768	11,768	20,212	20,212	40,278	40,278
	Q2	17,407	26,008	23,979	35,748	25,709	45,921	26,606	66,884
	Q3	44,171	70,179	24,486	60,233	33,428	79,349	34,042	100,926
	Q4	33,659	103,838	42,641	102,874	68,080	147,429	46,172	147,098
	Total	103,838	103,838	102,874	102,874	147,429	147,429	147,098	147,098
East	Q1	6,579	110,418	17,146	120,020	24,134	171,563	17,341	164,439
	Q2	21,064	131,482	22,703	142,723	52,807	224,371	29,978	194,417
	Q3	33,443	164,925	50,777	193,501	37,528	261,899	67,712	262,129
	Q4	67,594	232,519	65,706	259,206	66,060	327,959	98,209	360,338
	Total	128,680	232,519	156,332	259,206	180,529	327,959	213,239	360,338
South	Q1	44,262	276,781	16,444	275,651	23,934	351,892	9,882	370,219
	Q2	22,524	299,305	16,254	291,905	17,079	368,971	33,137	403,357
	Q3	16,061	315,366	21,460	313,364	22,939	391,910	23,894	427,250
	Q4	20,998	336,364	17,202	330,566	29,588	421,498	56,064	483,314
	Total	103,846	336,364	71,360	330,566	93,539	421,498	122,977	483,314
West	Q1	15,006	351,370	23,493	354,059	24,317	445,815	51,395	534,710
	Q2	25,543	376,913	26,188	380,247	39,774	485,589	44,302	579,011
	Q3	49,957	426,871	33,537	413,784	50,720	536,309	74,786	653,797
	Q4	57,377	484,247	56,748	470,533	72,165	608,474	80,150	733,947
	Total	147,883	484,247	139,966	470,533	186,976	608,474	250,633	733,947

Figure 5-2. *Quarterly sales by region*

5.2.1.1 Steps

Follow the steps as provided.

5.2.1.1.1 Step 1

Read in the data from "Sample – Superstore.xls" into Tableau (Shown in Fig. 5-3).

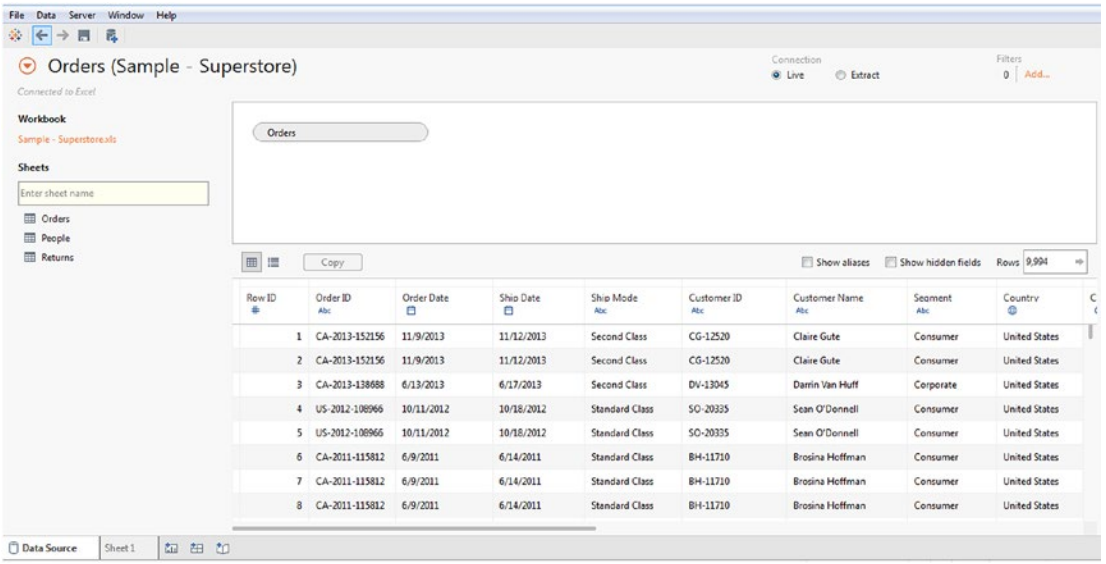


Figure 5-3. *Data read from "Sample - Superstore.xls" into Tableau*

5.2.1.1.2 Step 2

Table 5-1. *Tasks to be performed in the view*

Columns Shelf	Order Date: Set it to "Discrete". The granularity should be "Year". Measure names: Sum(Sales) Sum (Sales): Add a table calculation, "Running Total" and have it compute the running total, "Table Down".
Rows Shelf	Region Order Date: Set it to "Discrete". The granularity should be "Quarter".

Drag the **dimension** "Order Date" from the dimensions area under data pane to the columns shelf. By default it is discrete. This is also evident from the visual cue. It is blue in color. By default dates in tableau have hierarchies defined on it. The hierarchy is set to the highest level, i.e. "Year" (Shown in Fig. 5-4).

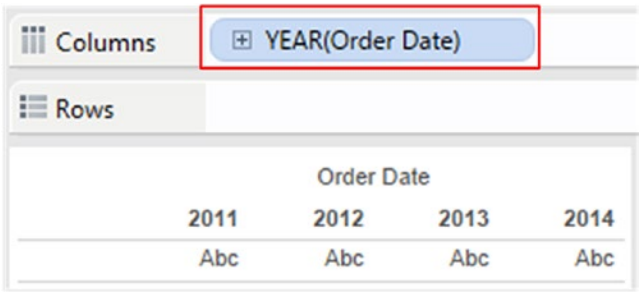


Figure 5-4. *Dimension "Order Date" placed on the columns shelf*

5.2.1.1.3 Step 3

Drag the **dimension** "Region" from the dimensions area under data pane to the rows shelf (Shown in Fig. 5-5).

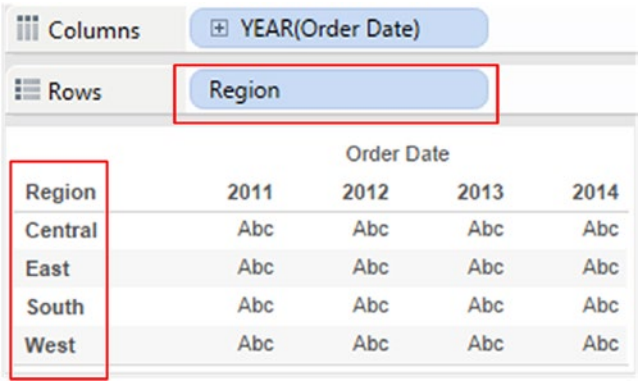


Figure 5-5. *Dimension "Region" placed on the rows shelf*

5.2.1.1.4 Step 4

Drag the **dimension** "Order Date" from the dimensions area under the data pane to the rows shelf, to the right of "Region". By default it is discrete. This is also evident from the visual cue. It is blue in color. Set the hierarchy to "Quarter" (Shown in Fig. 5-6).

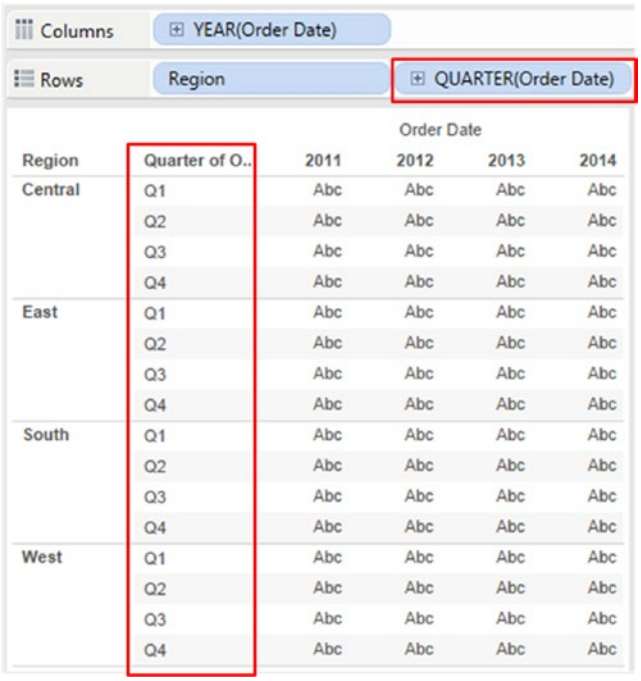
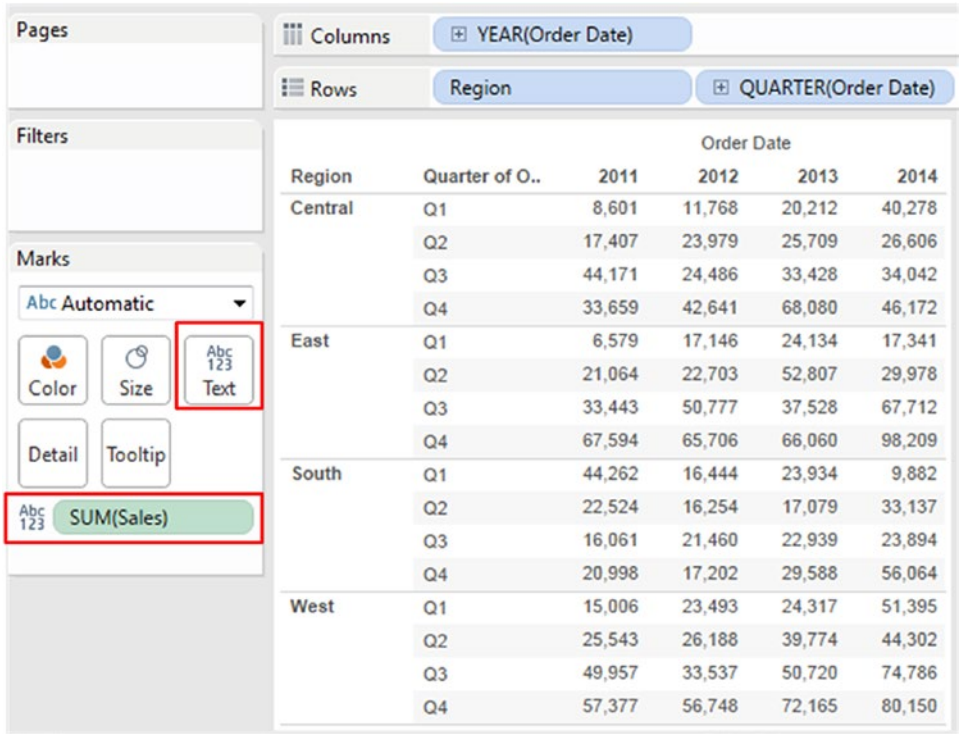


Figure 5-6. *Dimension "Order Date" placed on the rows shelf, to the right of "Region"*

5.2.1.1.5 Step 5

Drag the **measure** "Sales" from the measures area under the data pane to "Label" on the marks card. The default aggregation is "Sum" (Shown in Fig. 5-7).



		Order Date			
Region	Quarter of O..	2011	2012	2013	2014
Central	Q1	8,601	11,768	20,212	40,278
	Q2	17,407	23,979	25,709	26,606
	Q3	44,171	24,486	33,428	34,042
	Q4	33,659	42,641	68,080	46,172
East	Q1	6,579	17,146	24,134	17,341
	Q2	21,064	22,703	52,807	29,978
	Q3	33,443	50,777	37,528	67,712
	Q4	67,594	65,706	66,060	98,209
South	Q1	44,262	16,444	23,934	9,882
	Q2	22,524	16,254	17,079	33,137
	Q3	16,061	21,460	22,939	23,894
	Q4	20,998	17,202	29,588	56,064
West	Q1	15,006	23,493	24,317	51,395
	Q2	25,543	26,188	39,774	44,302
	Q3	49,957	33,537	50,720	74,786
	Q4	57,377	56,748	72,165	80,150

Figure 5-7. *Measure "Sales" placed on "Label" on the marks card*

5.2.1.1.6 Step 6

Add a quick table calculation, "Running Total" to the **measure** "Sum (Sales)" and compute the "Running Total" as "Table Down" (Shown in Fig. 5-8).

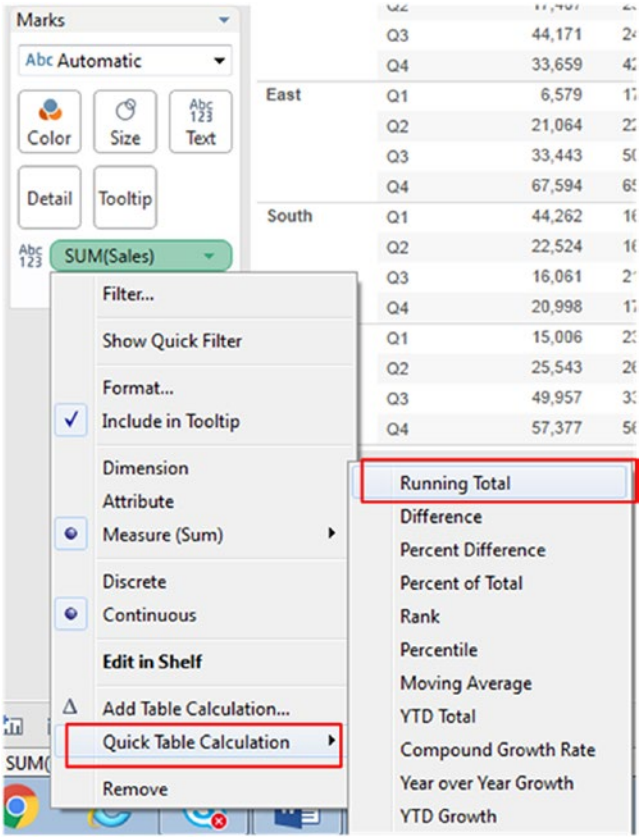


Figure 5-8. *Add "Quick Table Calculation" to the measure "Sales"*

Fill in the values into the "Table Calculation dialog box" as shown in Fig. 5-9.

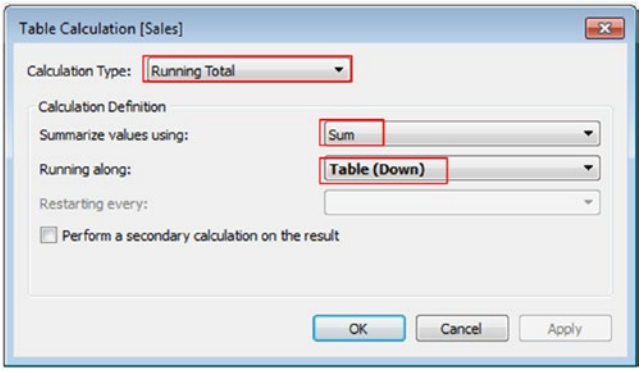
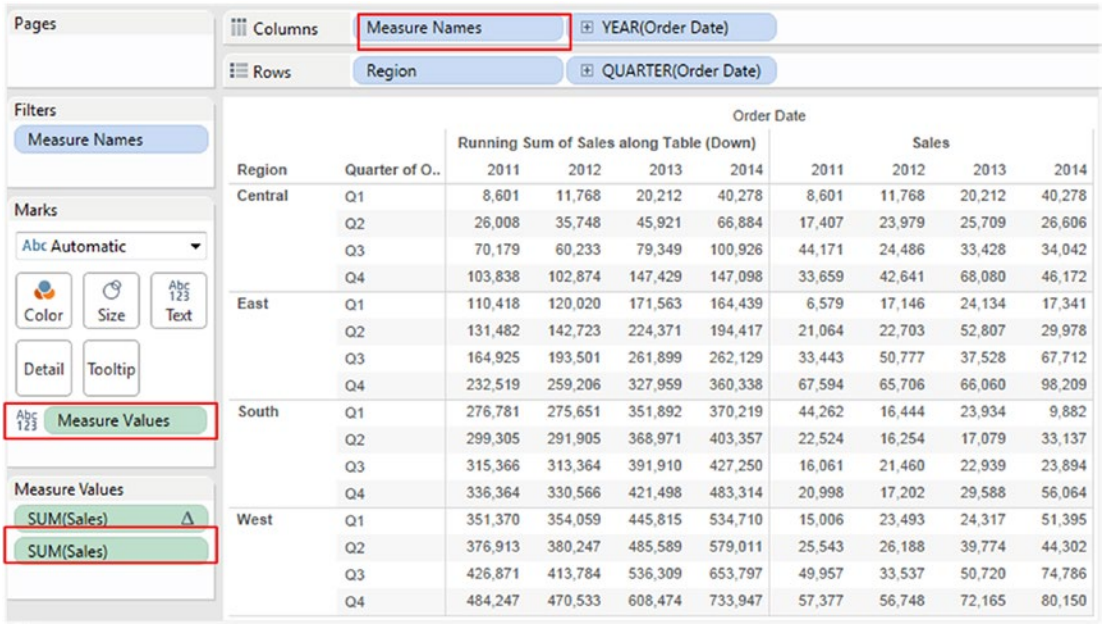


Figure 5-9. *Table Calculation dialog box for measure "Sales"*

Click on “Apply” and then on “OK”.

5.2.1.1.7 Step 7

Once again drag the **measure** “Sales” from the measures area under the data pane and drop it into the view area. The default aggregation is “Sum” (Shown in Fig. 5-10).



		Order Date							
		Running Sum of Sales along Table (Down)				Sales			
Region	Quarter of O..	2011	2012	2013	2014	2011	2012	2013	2014
Central	Q1	8,601	11,768	20,212	40,278	8,601	11,768	20,212	40,278
	Q2	26,008	35,748	45,921	66,884	17,407	23,979	25,709	26,606
	Q3	70,179	60,233	79,349	100,926	44,171	24,486	33,428	34,042
	Q4	103,838	102,874	147,429	147,098	33,659	42,641	68,080	46,172
East	Q1	110,418	120,020	171,563	164,439	6,579	17,146	24,134	17,341
	Q2	131,482	142,723	224,371	194,417	21,064	22,703	52,807	29,978
	Q3	164,925	193,501	261,899	262,129	33,443	50,777	37,528	67,712
	Q4	232,519	259,206	327,959	360,338	67,594	65,706	66,060	98,209
South	Q1	276,781	275,651	351,892	370,219	44,262	16,444	23,934	9,882
	Q2	299,305	291,905	368,971	403,357	22,524	16,254	17,079	33,137
	Q3	315,366	313,364	391,910	427,250	16,061	21,460	22,939	23,894
	Q4	336,364	330,566	421,498	483,314	20,998	17,202	29,588	56,064
West	Q1	351,370	354,059	445,815	534,710	15,006	23,493	24,317	51,395
	Q2	376,913	380,247	485,589	579,011	25,543	26,188	39,774	44,302
	Q3	426,871	413,784	536,309	653,797	49,957	33,537	50,720	74,786
	Q4	484,247	470,533	608,474	733,947	57,377	56,748	72,165	80,150

Figure 5-10. *Measure “Sales” again placed on the view*

Notice that “Measure Names” appears on the columns shelf. Also, “Measure Values” appears on Label on the marks card.

Drag “Measure Names” and pull it to the right of “Year (Order Date)” on the columns shelf (Shown in Fig. 5-11).

Columns: YEAR(Order Date) | Measure Names

Rows: Region | QUARTER(Order Date)

		Order Date							
		2011		2012		2013		2014	
Region	Quarter of O..	Running Sum	Sales	Running Sum	Sales	Running Sum	Sales	Running Sum	Sales
Central	Q1	8,601	8,601	11,768	11,768	20,212	20,212	40,278	40,278
	Q2	26,008	17,407	35,748	23,979	45,921	25,709	66,884	26,606
	Q3	70,179	44,171	60,233	24,486	79,349	33,428	100,926	34,042
	Q4	103,838	33,659	102,874	42,641	147,429	68,080	147,098	46,172
East	Q1	110,418	6,579	120,020	17,146	171,563	24,134	164,439	17,341
	Q2	131,482	21,064	142,723	22,703	224,371	52,807	194,417	29,978
	Q3	164,925	33,443	193,501	50,777	261,899	37,528	262,129	67,712
	Q4	232,519	67,594	259,206	65,706	327,959	66,060	360,338	98,209
South	Q1	276,781	44,262	275,651	16,444	351,892	23,934	370,219	9,882
	Q2	299,305	22,524	291,905	16,254	368,971	17,079	403,357	33,137
	Q3	315,366	16,061	313,364	21,460	391,910	22,939	427,250	23,894
	Q4	336,364	20,998	330,566	17,202	421,498	29,588	483,314	56,064
West	Q1	351,370	15,006	354,059	23,493	445,815	24,317	534,710	51,395
	Q2	376,913	25,543	380,247	26,188	485,589	39,774	579,011	44,302
	Q3	426,871	49,957	413,784	33,537	536,309	50,720	653,797	74,786
	Q4	484,247	57,377	470,533	56,748	608,474	72,165	733,947	80,150

Figure 5-11. *Dimension "Measure Names" placed to the right of "Order Date" on the columns shelf*

Notice the change in display.

In the measure values section, place Sum (Sales) above the Sum (Sales) on which we have defined the Table Calculation (Shown in Fig. 5-12).

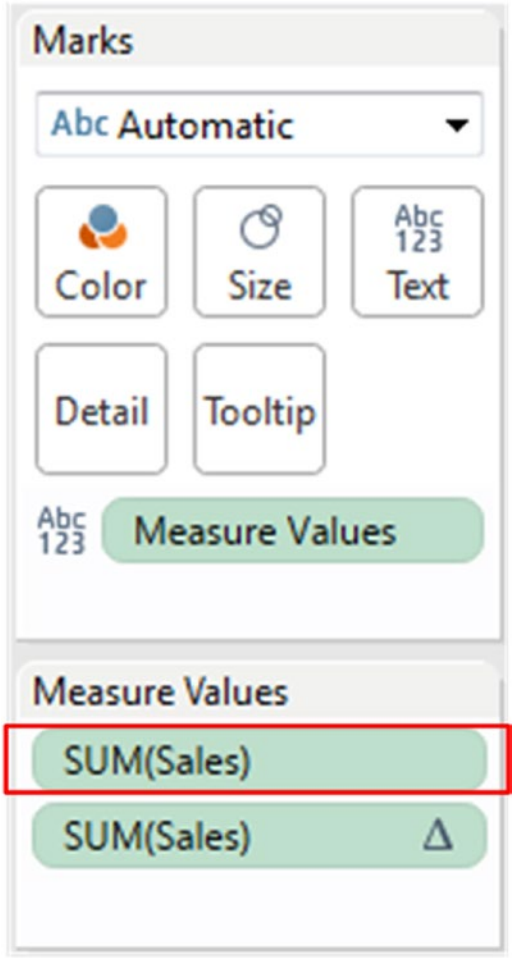


Figure 5-12. *Alter the sequence of the measures on "Label" on the marks card*

As can be seen from the display, first the measure "Sales" is aggregated and then its "Running Total" is displayed.

Let us verify the calculation. For verification, we have picked up the data only for "2011" and for "Central Region".

		2011	
Region	Quarter of O..	Sales	Running Sum
Central	Q1	8,601	8,601
	Q2	17,407	26,008
	Q3	44,171	70,179
	Q4	33,659	103,838

The Sales made in Q1 of 2011 is 8,601. To this we add the sales of Q2, i.e. 17, 407 to give the running sum as 26,008. To this running sum of 26, 008, we add the Q3 sales of 44,171 to give the new running sum of 70,179. To this we add the Sales for Q4, i.e. 33,659 to give the running sum of 103, 838.

Note that the running sum of "Sales" is computed "Table Down".

Let us add Subtotals to the view. For this, select "Analysis" from the menu bar, then click on "Totals" and select "Add All Subtotals" (Shown in Fig. 5-13).

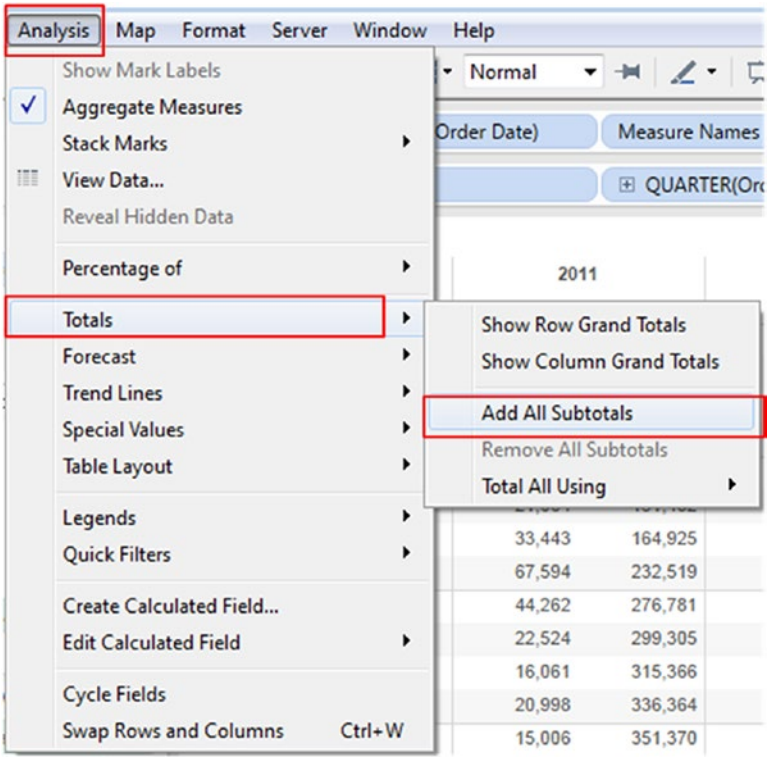


Figure 5-13. *Apply "Add All Subtotals" to measures in the view*

Analysis ➤ Totals ➤ Add All Subtotals

The final output (Shown in Fig. 5-14):

		Order Date							
		2011		2012		2013		2014	
Region	Quarter of O..	Sales	Running Sum	Sales	Running Sum	Sales	Running Sum	Sales	Running Sum
Central	Q1	8,601	8,601	11,768	11,768	20,212	20,212	40,278	40,278
	Q2	17,407	26,008	23,979	35,748	25,709	45,921	26,606	66,884
	Q3	44,171	70,179	24,486	60,233	33,428	79,349	34,042	100,926
	Q4	33,659	103,838	42,641	102,874	68,080	147,429	46,172	147,098
	Total	103,838	103,838	102,874	102,874	147,429	147,429	147,098	147,098
East	Q1	6,579	110,418	17,146	120,020	24,134	171,563	17,341	164,439
	Q2	21,064	131,482	22,703	142,723	52,807	224,371	29,978	194,417
	Q3	33,443	164,925	50,777	193,501	37,528	261,899	67,712	262,129
	Q4	67,594	232,519	65,706	259,206	66,060	327,959	98,209	360,338
	Total	128,680	232,519	156,332	259,206	180,529	327,959	213,239	360,338
South	Q1	44,262	276,781	16,444	275,651	23,934	351,892	9,882	370,219
	Q2	22,524	299,305	16,254	291,905	17,079	368,971	33,137	403,357
	Q3	16,061	315,366	21,460	313,364	22,939	391,910	23,894	427,250
	Q4	20,998	336,364	17,202	330,566	29,588	421,498	56,064	483,314
	Total	103,846	336,364	71,360	330,566	93,539	421,498	122,977	483,314
West	Q1	15,006	351,370	23,493	354,059	24,317	445,815	51,395	534,710
	Q2	25,543	376,913	26,188	380,247	39,774	485,589	44,302	579,011
	Q3	49,957	426,871	33,537	413,784	50,720	536,309	74,786	653,797
	Q4	57,377	484,247	56,748	470,533	72,165	608,474	80,150	733,947
	Total	147,883	484,247	139,966	470,533	186,976	608,474	250,633	733,947

Figure 5-14. *"Running Total" – Demo 1 – final output*

5.3 Profitability as Percent of Total

Percent of total is also called as percent distribution. It is computed using the formula that divides an amount by the total. Example: To find the percent of total for each of the following numbers: 100, 400 and 600, first determine the total by adding up the numbers 100, 400 and 600. The total is (100+400+600) = 1100. Then find what percent of total, 1100 is the number 100. This can be computed as (100 / 1100) *100 = 9.090%.

5.3.1 Demo 1

Objective**:** To demonstrate "Profitability as Percent of Total" for categories of products per segment per region across several years (2011, 2012, 2013 and 2014).

Data set used**:** Sample – Superstore.xls

Expected output**:** (Shown in Fig. 5-15).

Figure 5-15. *"Profitability as percent of total" – Demo 1 – expected output*

5.3.1.1 Steps

Follow the steps as provided.

5.3.1.1.1 Step 1

Read in the data from "Sample – Superstore.xls" (Shown in Fig. 5-16).

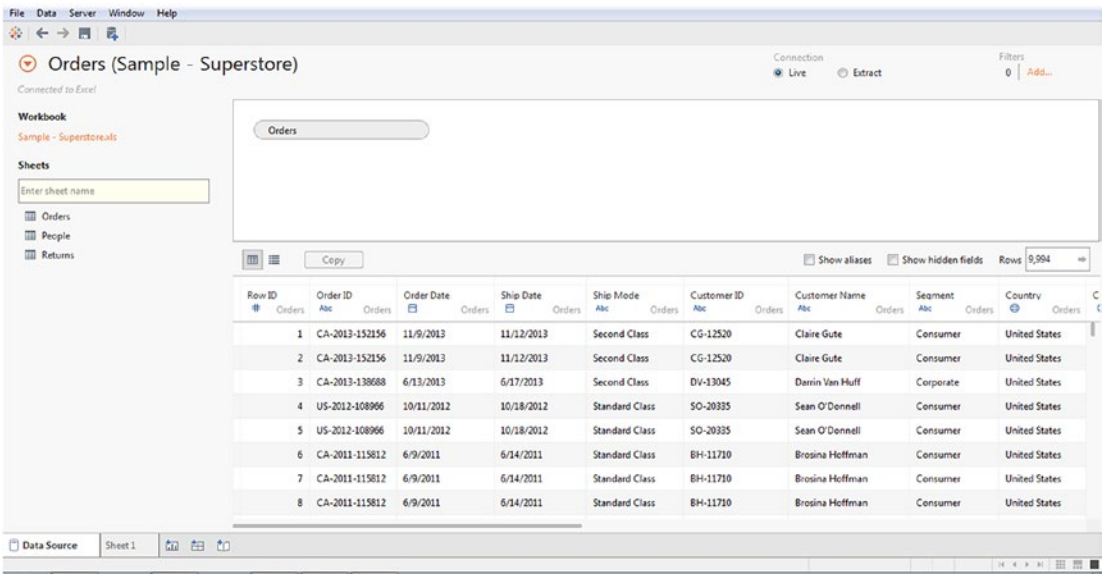


Figure 5-16. *Data read from "Sample - Superstore.xls" into Tableau*

5.3.1.1.2 Step 2

Table 5-2. *Tasks to be performed in the view*

columns shelf	Order Date ("Discrete" with the granularity set to "Year") Sales – Aggregation set to "SUM" Add a Table Calculation, "Percent of Total" to the measure "Sales" and summarize the values from – "Cell"
Rows Shelf	Region, Segment and Category

Drag the **dimension** "Order Date" from the dimensions area under the data pane to the columns shelf. Set it to "Discrete". Also set the granularity to "Year" (Shown in Fig. 5-17).

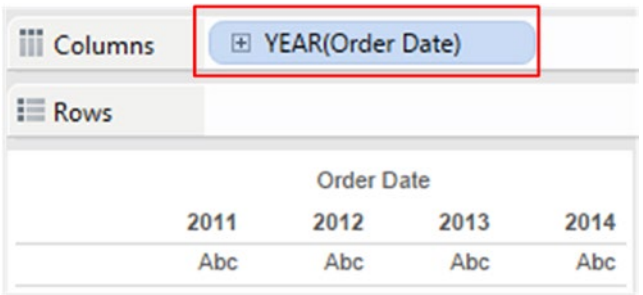


Figure 5-17. *Dimension "Order Date" placed on the columns shelf*

The visual cue ("Order Date" appears blue in color on the columns shelf) indicates that the dimension is discrete. By default, date type fields have a hierarchy defined on it and the default is the highest level in the hierarchy, which in this case is "Year".

Drag the **measure** "Sales" from the measures area under the data pane and drop it to the right of "Order Date". Let it be set to the default aggregation of "SUM" (Shown in Fig. 5-18).

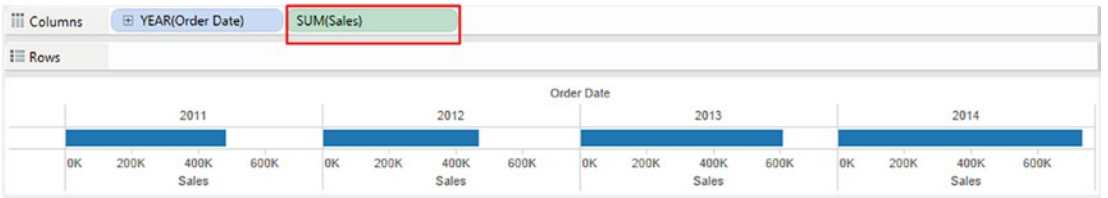


Figure 5-18. *Measure "Sales" placed on the columns shelf*

5.3.1.1.3 Step 3

Drag the **dimension** "Region" from the dimensions area under the data pane and drop it on the rows shelf. The data is available for four regions, namely, "Central", "East", "South" and "West" (Shown in Fig. 5-19).

Figure 5-19. *Dimension "Region" placed on the rows shelf*

5.3.1.1.4 Step 4

Drag the **dimension** "Segment" from the dimensions area under the data pane and drop it on the rows shelf to the right of the **dimension** "Region". The data is available for three Segments, namely, "Consumer", "Corporate", and "Home Office" (Shown in Fig. 5-20).

Figure 5-20. *Dimension "Segment" placed on the rows shelf*

5.3.1.1.5 Step 5

Drag the **dimension** "Category" from the dimensions area under the data pane and drop it on the rows shelf to the right of the **dimension** "Segment". The data is available for three Categories, namely, "Furniture", "Office Supplies", and "Technology" (Shown in Fig. 5-21).

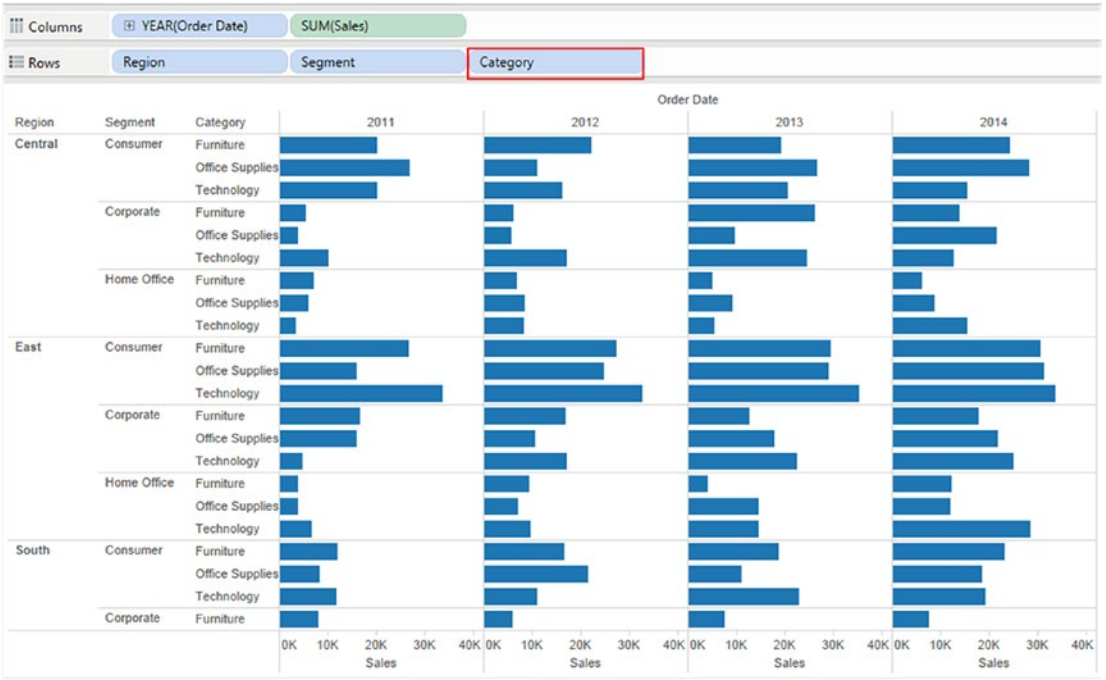


Figure 5-21. *Dimension "Category" placed on the rows shelf*

5.3.1.1.6 Step 6

Add a table calculation, "Percent of Total".

Click on the drop down of the **measure** "Sum (Sales)" and select "Add Table Calculation" (Shown in Fig. 5-22).

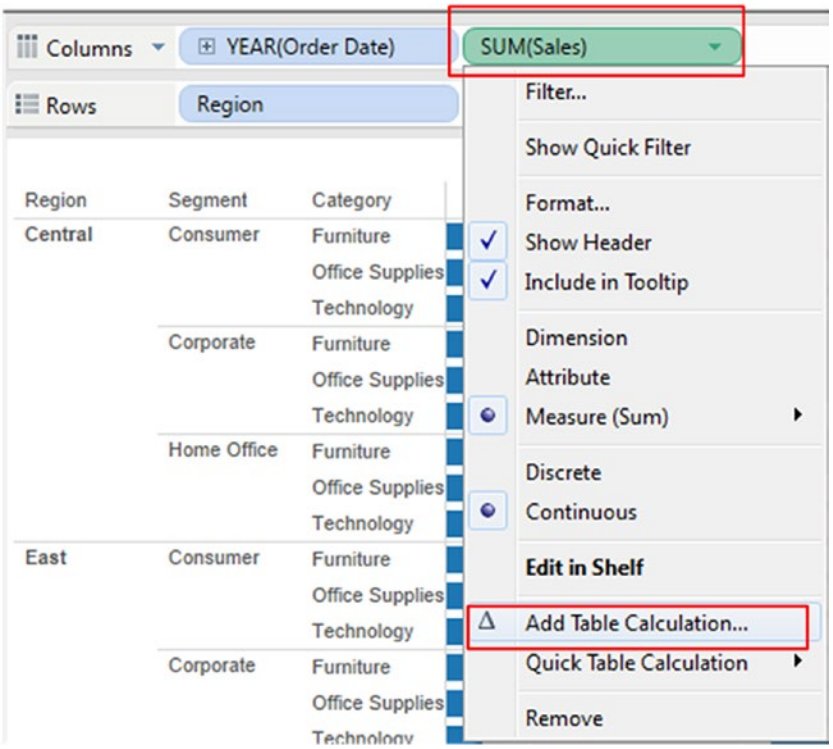


Figure 5-22. *"Add Table Calculation" to the measure "Sales"*

Fill in the values in the "Table Calculation" dialog box as shown in Fig. 5-23.

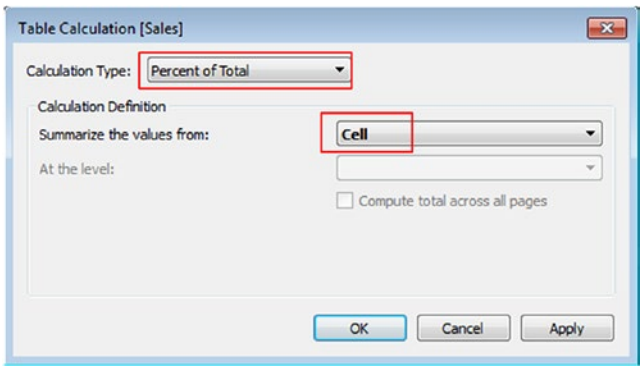


Figure 5-23. *"Table Calculation" dialog box for measure "Sales"*

Click on “Apply” and then click on “OK”.
The output will be as shown in Fig. 5-24.

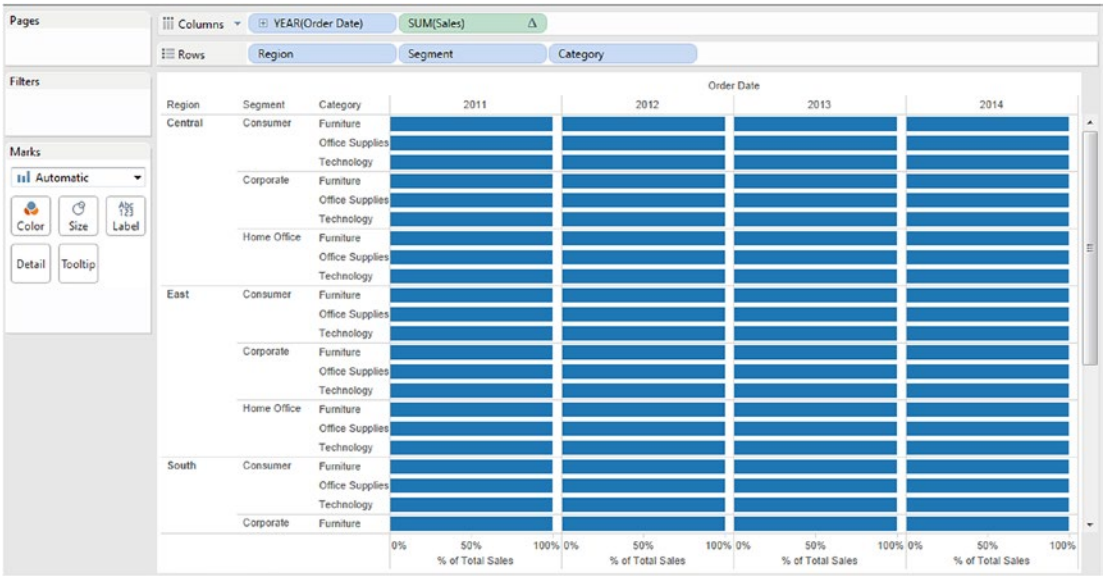


Figure 5-24. *Output after applying the “Table Calculation - Percent of Total” to the measure “Sales”*

5.3.1.1.7 Step 7

Create a calculated field, “Profit or Loss”, as shown in Fig. 5-25.

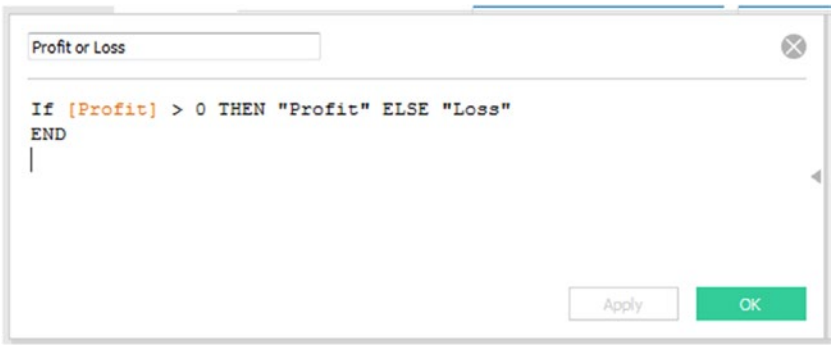


Figure 5-25. *“Calculated Field - Profit or Loss” being created*

Note that a new **dimension** “Profit or Loss” is added to the dimensions area under the data pane.

5.3.1.1.8 Step 8

Drag the newly created d**imension** “Profit or Loss” and drop it on “Color” on the marks card (Shown in Fig. 5-26).

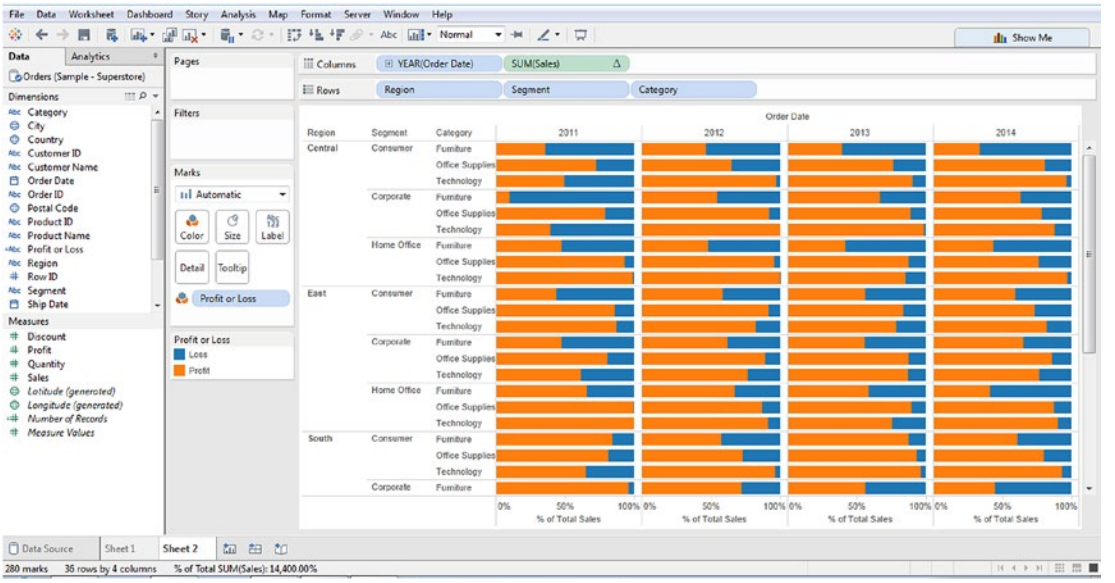


Figure 5-26. *"Calculated Field - Profit or Loss" placed on "Color" on the marks card*

5.3.1.1.9 Step 9

Press control key (CTRL) and drag the measure "Sum (Sales) (where we have added the table calculation)" from the columns shelf and drop it **on the "Label"** on the marks card (Shown in Fig. 5-27).

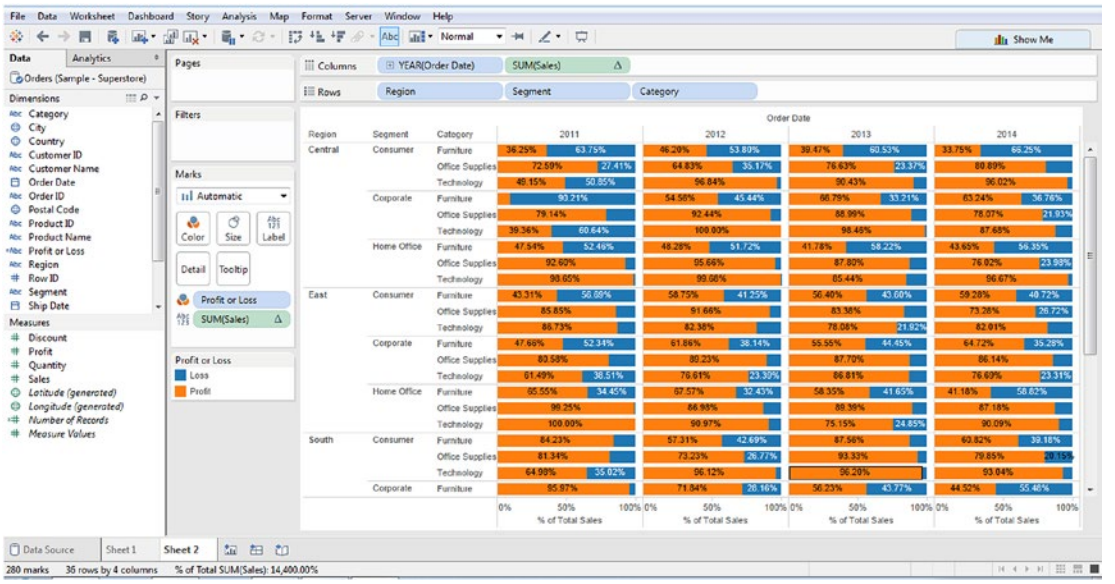


Figure 5-27. *Measure "Sales" with Table Calculation placed on the "Label" on the marks card*

The loss is shown in blue color and the profit in orange color.

5.4 Moving average

It is called by various names, such as rolling average, running average, rolling means, or running means.

5.4.1 Where is it used?

- In the technical analysis of financial data, such as stock prices, returns etc.
- To determine the market conditions. It is used with time-series data to iron out short-term price fluctuations or noises and highlight longer-term trends.
- To identify trends and reversals. Moving averages are lagging indicators. They are never used to predict new trends but confirm trends once they have been established. Example: a stock is termed uptrend when its price is above the moving average and the average slopes upwards. Likewise a stock is considered downtrend when its price is below the moving average and the average is sloping downward.
- To measure the strength of an asset's momentum. This has to do with the time period chosen for computing the moving average.
    - Short-term momentum: <=20 days
    - Medium-term momentum: between 20 to 100 days
    - Long-term momentum: > 100 days
- A valuable tool in planning trading strategy.
- To help with creation of a number of other technical indicators such as moving average convergence divergence (MACD) or Bollinger's bands.
- To help with stochastic measurements.

5.4.2 Types of moving average

Simple Moving Average: This is computed by taking arithmetic mean of a given set of values.

Weighted Moving Average: This is used to ensure that the most recent values have the most impact on the average. It uses values that are linearly weighted. Example: the oldest value is given a weight of 1, the next oldest value a weight of 2, and so on … all the way up to the most recent value which gets the highest weight.

Exponential Moving Average: This is similar to the simple moving average. The difference lies in the fact that while a simple moving average will remove the older values as the new values become available, the exponential moving average calculates the average of all historical ranges, starting at the points that one specifies.

■ **Points to Note** Moving Averages are Lagging Indicators. They are based on events that have already occurred in the market.

They are not predictive indicators.

5.4.3 Demo 1

Objective: To demonstrate the "Moving Average" of the m**easure** "Sales" across several years (2011, 2012, 2013 and 2014).

Input Data Set: "Sample - Superstore.xls"

Expected output: Shown in Fig. 5-28.

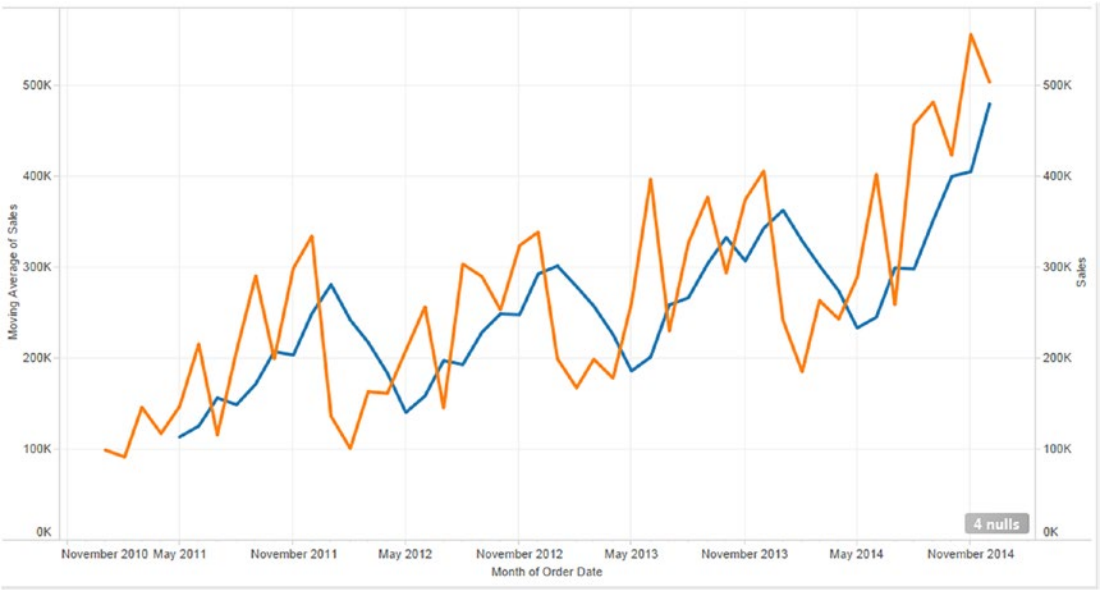


Figure 5-28. *Moving average – Demo 1 – expected output*

5.4.3.1 Steps

Table 5-3. *Tasks to perform in the view*

Columns shelf	Month(Order Date) : Continuous Date
Rows shelf	Sum(Sales) with calculation type – "Moving Calculation", summarize values using "Average"
Rows shelf	Sum(Sales)

5.4.3.1.1 Step 1

Drag the **dimension** "Order Date" from the dimensions area under data pane and drop it on the columns shelf (Shown in Fig. 5-29).

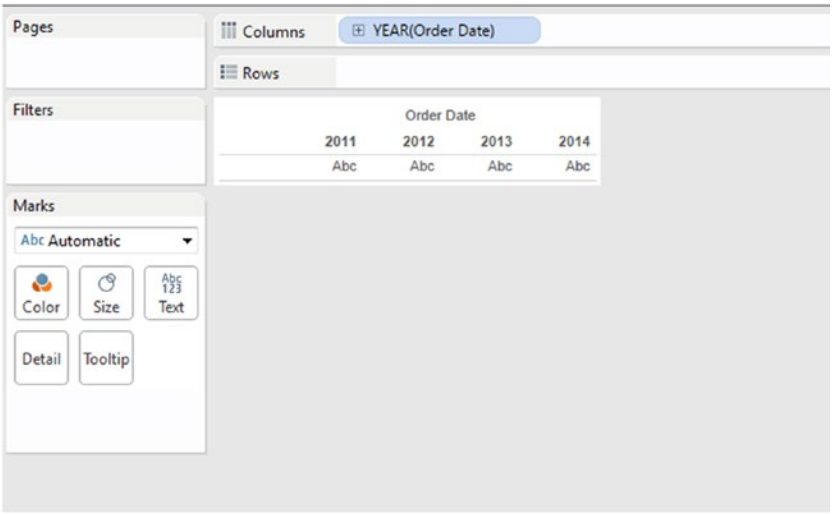


Figure 5-29. *Dimension "Order Date" placed on columns shelf*

5.4.3.1.2 Step 2

Change the "Order Date" to Continuous Month (Order Date) (Shown in Fig. 5-30).

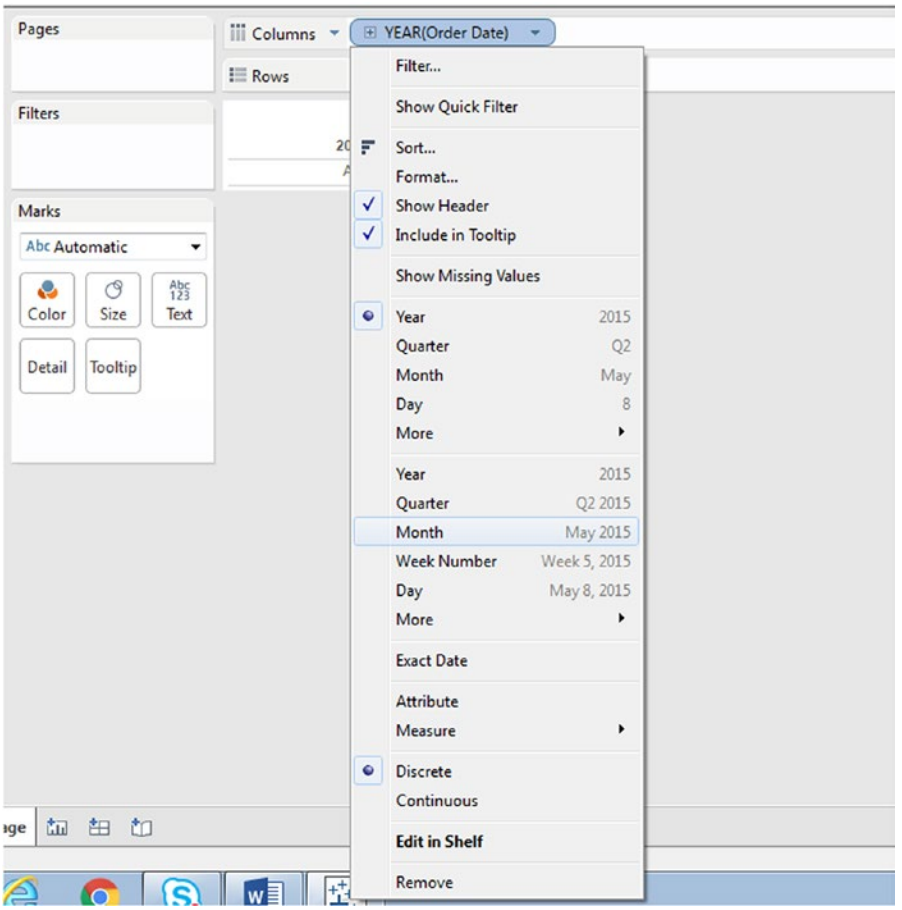


Figure 5-30. *Dimension "Order Date" being changed from discrete to continous*

Notice the visual cue. The Month (Order Date) color is changed to green (Shown in Fig. 5-31).

Figure 5-31. *Dimension "Order Date" changed to "Continous - Month"*

5.4.3.1.3 Step 3

Drag the **measure** "Sales" from the measures area under data pane and drop it on the rows shelf (Shown in Fig. 5-32).

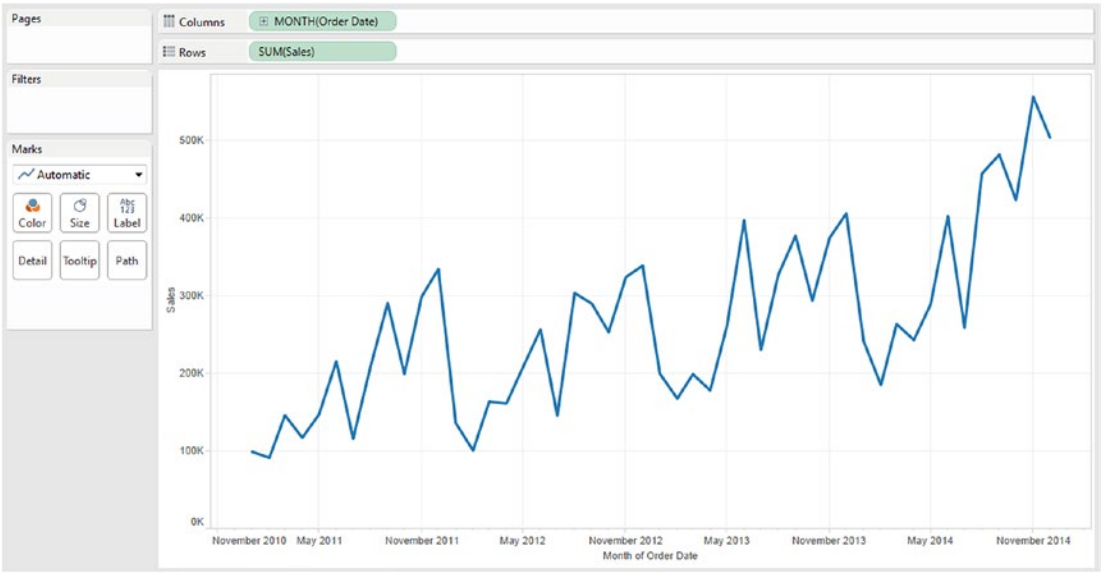


Figure 5-32. *Measure "Sales" placed on the rows shelf*

5.4.3.1.4 Step 4

Click on the drop down of Sum (Sales) to select "Add Table Calculation" (Shown in Fig. 5-33).

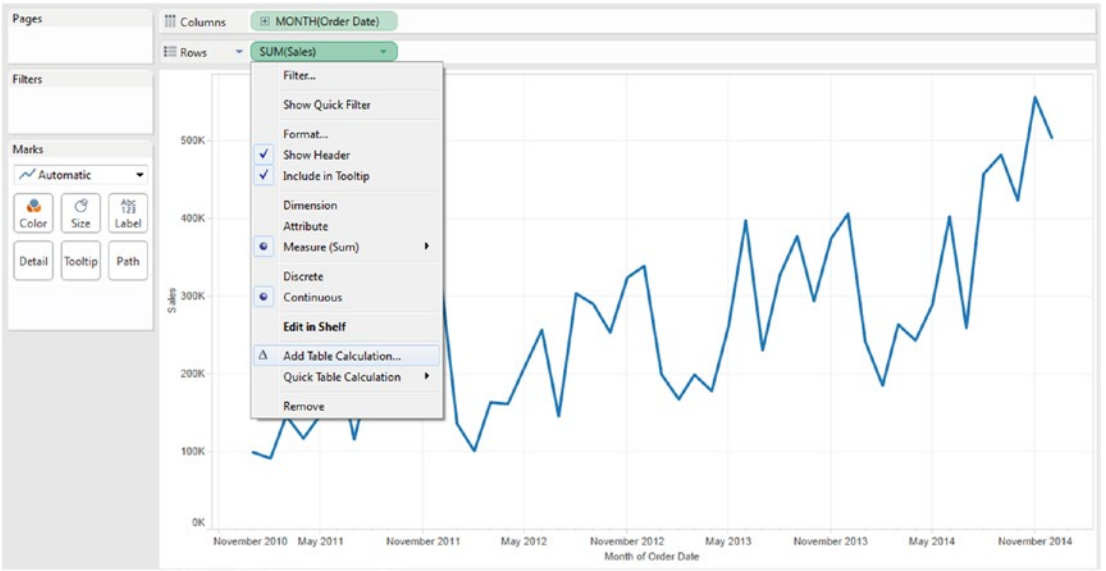


Figure 5-33. *"Add Table Calculation" to the measure "Sales"*

5.4.3.1.5 Step 5

Fill in the values in the "Table Calculation" dialog box as shown in Fig. 5-34.

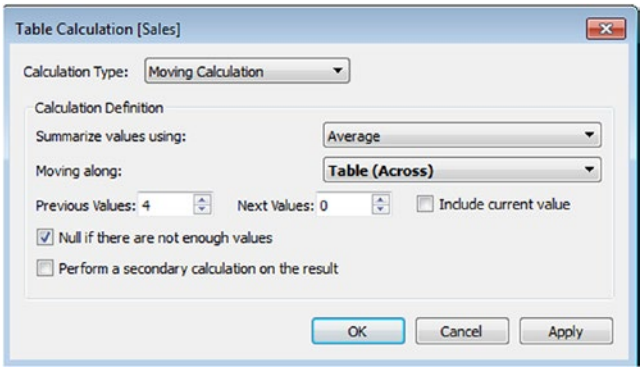


Figure 5-34. *"Table Calculation" dialog box for the measure "Sales"*

The output is as shown in Fig. 5-35.

Figure 5-35. *The output after applying the "Table Calculation" to the measure "Sales"*

Click on “4 nulls”. It brings up the “Special Values for [Moving Average of Sales]” dialog box (Shown in Fig. 5-36).

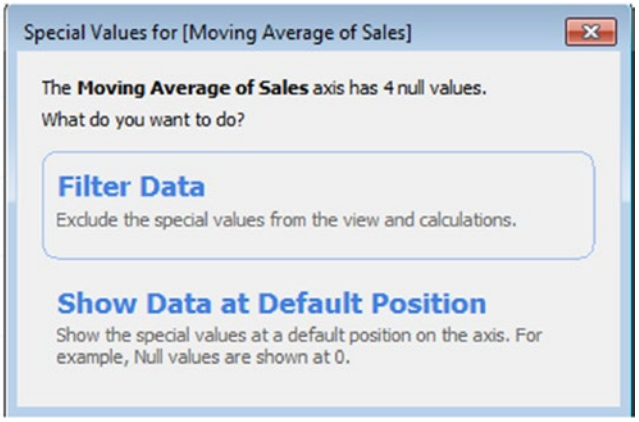


Figure 5-36. *“Special Values for [Moving Average of Sales]” dialog box*

Click on “Show Data at Default Position”.
The output changes to the below (Shown in Fig. 5-37).

Figure 5-37. *The output after setting the null values*

5.4.3.1.6 Step 6

Drag the **measure** "Sales" from the measures area under the data pane and drop it on the opposite axis (Shown in Fig. 5-38).

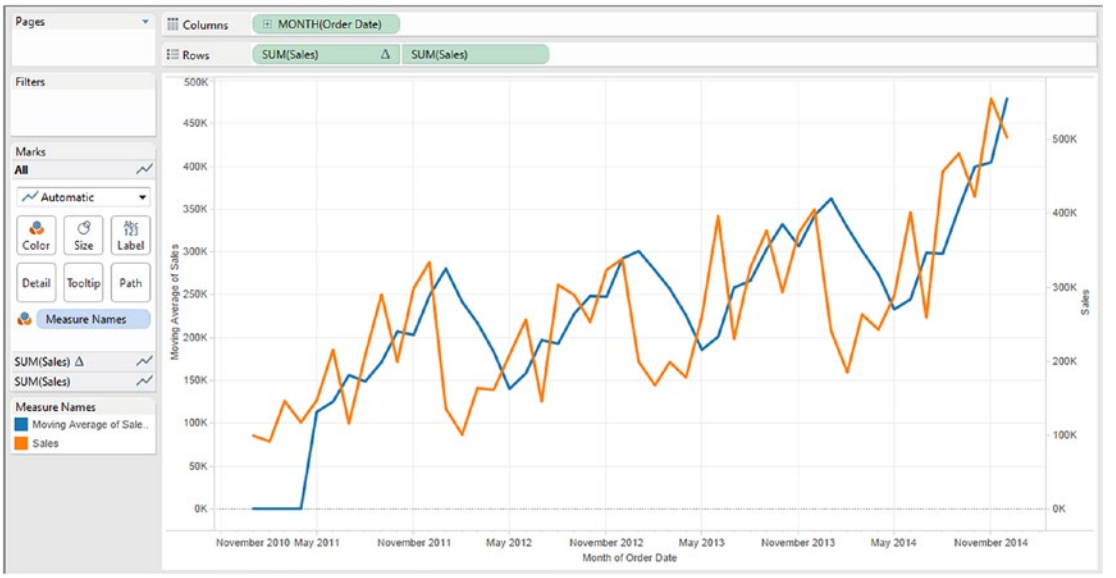


Figure 5-38. *Measure "Sales" placed on the rows shelf*

Synchronize the secondary axis to the primary one (Shown in Fig. 5-39).

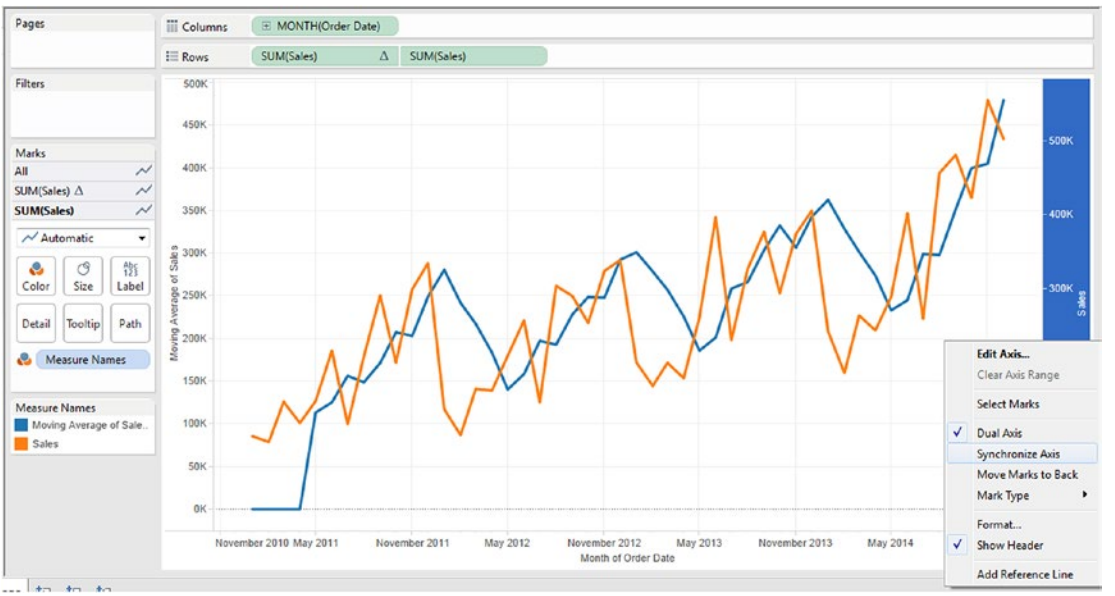


Figure 5-39. *Synchronize the secondary axis to the primary axis*

The output will be as shown in Fig. 5-40.

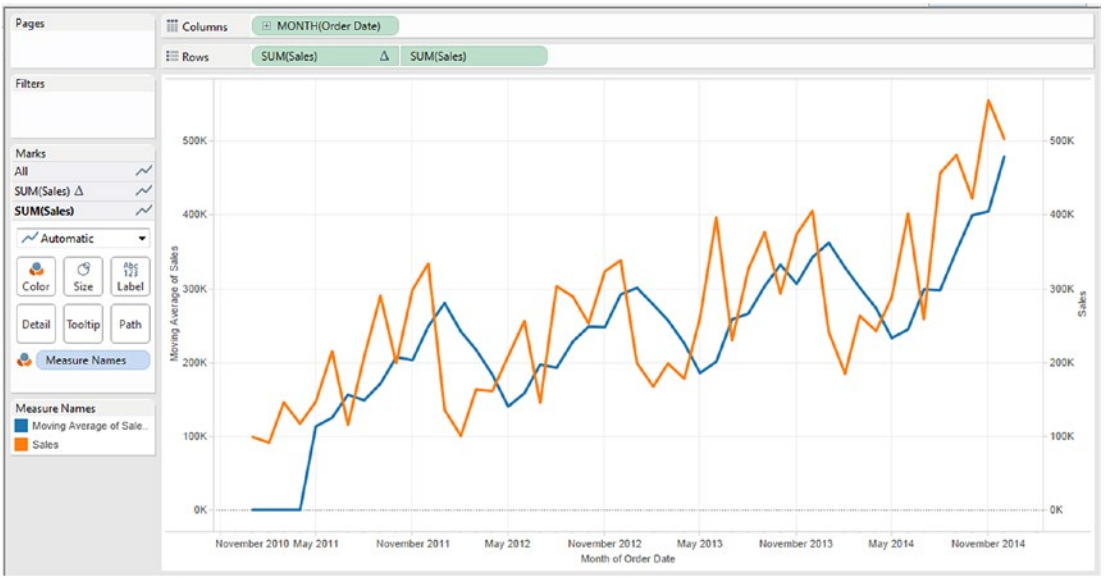


Figure 5-40. *The output after the secondary axis is synchronized with the primary axis*

Let us verify the result:

Month	Sum(Sales)
Jan 2011	98898
Feb 2011	91152
Mar 2011	145729
Apr 2011	116916
May 2011	146748
June 2011	215207
July 2011	115510
....	....

The moving average is calculated moving along, "Table Across" using the previous four values and NOT including the current value. If there are not enough values, it will use Null.

Month	Sum(Sales)	Moving Average
Jan 2011	98898	
Feb 2011	91152	
Mar 2011	145729	
Apr 2011	116916	

Month	Sum(Sales)	Moving Average
May 2011	146748	113174
June 2011	215207	125136
July 2011	115510	156150
Aug 2011	207581	148595
....	...	....

To compute the first data point for moving average:

= (98898 + 91152 + 145729 + 116916) / 4

= (452695)/ 4

= 113174

Likewise to compute the second data point for Moving Average:

= (91152 + 145729 + 116916 + 146748) / 4

= (500545) / 4

= 125136

The final output of moving averages is shown in Fig. 5-41.

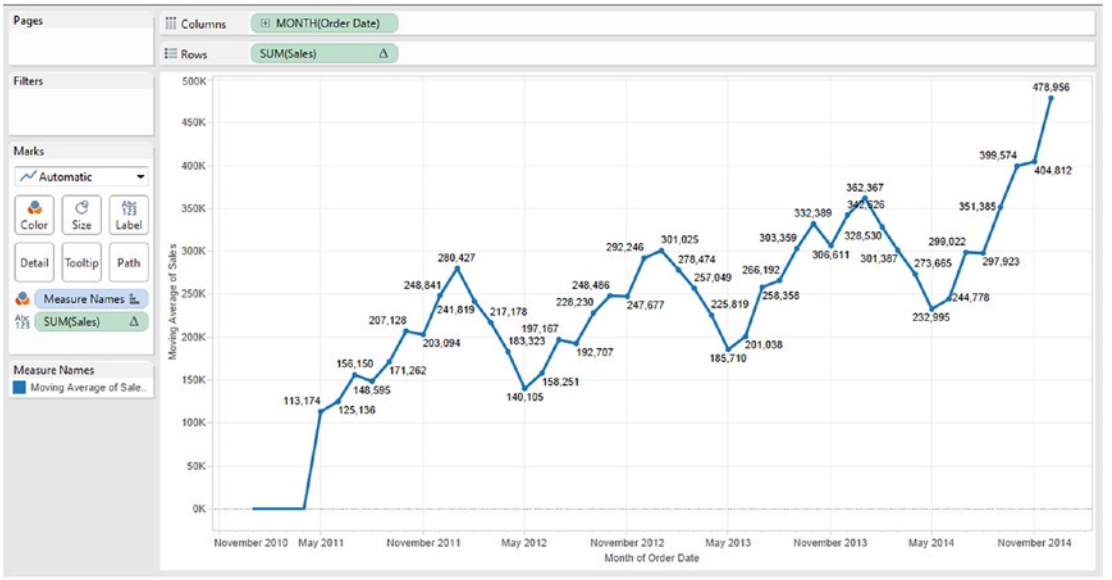


Figure 5-41. *Moving average – Demo 1 – final output*

5.5 Rank

Tableau ignores "nulls" in rank calculations. They appear as blank rows.

The following choices are available on the type of ranking that one can apply:

Table 5-4. *Type of Ranking*

Type of Ranking	Example
1,2,2,2,5 (Competition)	Below is the input set and the ranks assigned in ascending order Input set Rank 50 1 60 2 60 2 65 4 78 5 90 6
1,4,4,4,5 (Modified Competition)	Below is the input set and the ranks assigned in ascending order Input Set Rank 50 1 60 3 60 3 65 4 78 5 90 6
1,2,2,2,3 (Dense)	Below is the input set and the ranks assigned in ascending order Input set Rank 50 1 60 2 60 2 65 3 78 4 90 5
1,2,3,4,5 (Unique)	Below is the input set and the ranks assigned in ascending order Input set Rank 50 1 60 2 60 3 65 4 78 5 90 6

5.5.1 Demo 1

Objective: To rank the "Sub-Category" based on the "Sales Amount". The sub-Category with the highest sales amount is ranked one followed by the Sub-Category with the next highest sales amount, which is ranked two and so on…

Input data Set: "Sample – Superstore.xls"

Expected Output: Shown in Fig. 5-42.

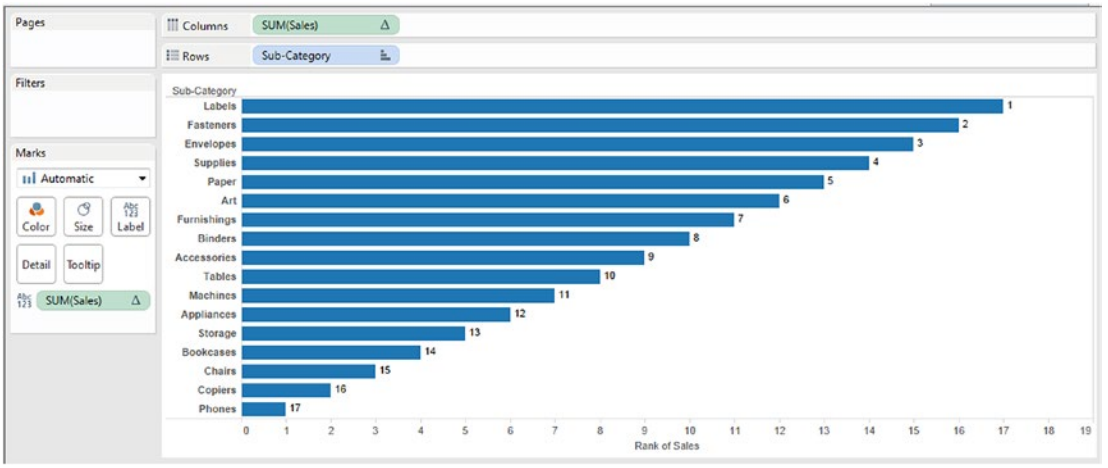


Figure 5-42. *"Sub-Category" ranked in descending order of "Sales" amount*

Table 5-5. *Tasks to perform in the view*

Columns shelf	Sum(Sales)
Row shelf	Sub-Category
Table calculation	Rank Running Along: Table (Down) Sort Order: Descending Rank duplicate Values as: Competition (1,2,2,4)

5.5.1.1 Steps

5.5.1.1.1 Step 1

Drag the **dimension** "Sub-Category' from the dimensions area under data pane and drop it on the rows shelf (Shown in Fig. 5-43).

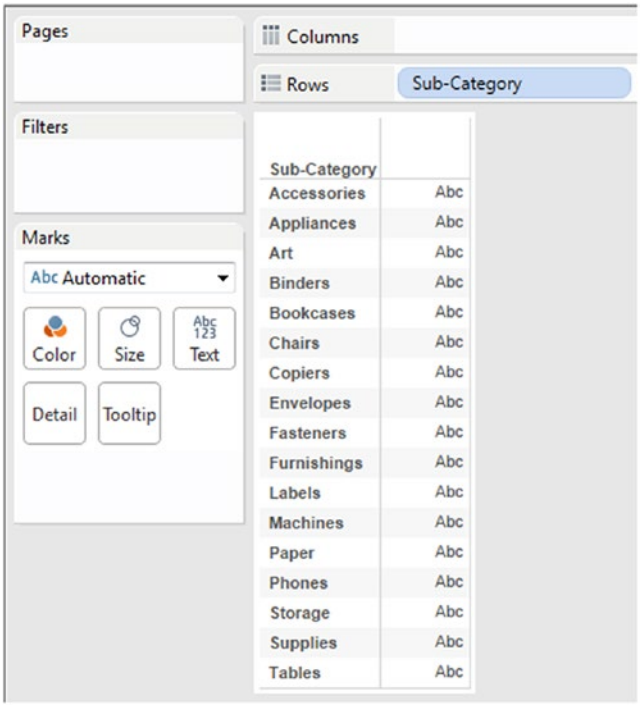


Figure 5-43. *Dimension "Sub-Category" placed on the rows shelf*

5.5.1.1.2 Step 2

Drag the **measure** "Sales" from the measures area under the data pane and drop it on the columns shelf (Shown in 5-44).

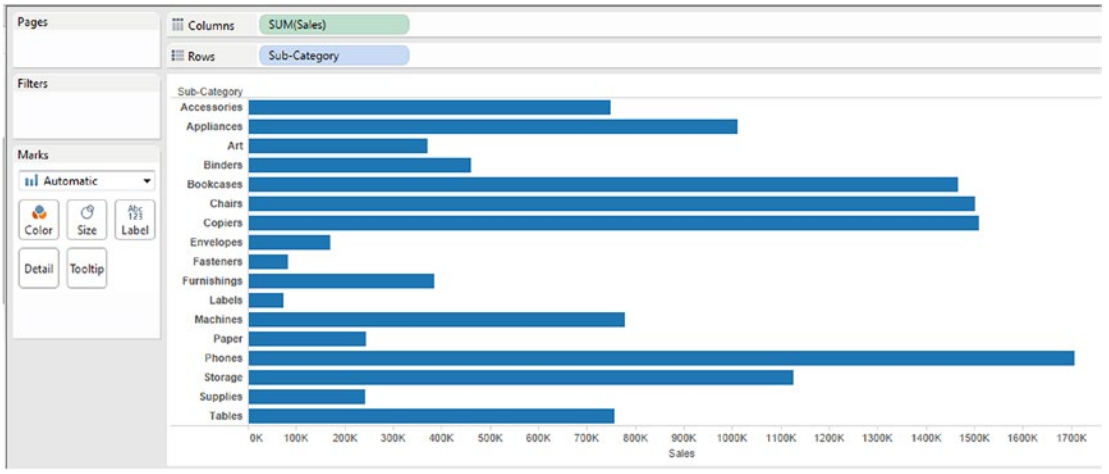


Figure 5-44. *Measure "Sales" placed on the columns shelf*

5.5.1.1.3 Step 3

Sort “Sub-Category” as per “Sales” in “Descending Order” (Shown in Fig. 5-45).

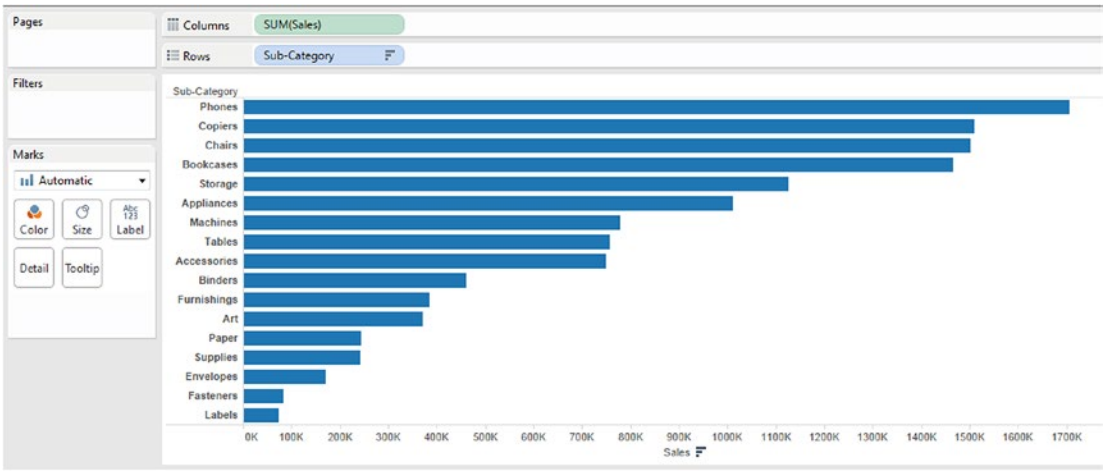


Figure 5-45. *Sort, “Sub-Category” as per “Sales” in “Descending Order”*

5.5.1.1.4 Step 4

Drag “Sales” on the **“Label”** on the marks card (Shown in Fig. 5-46).

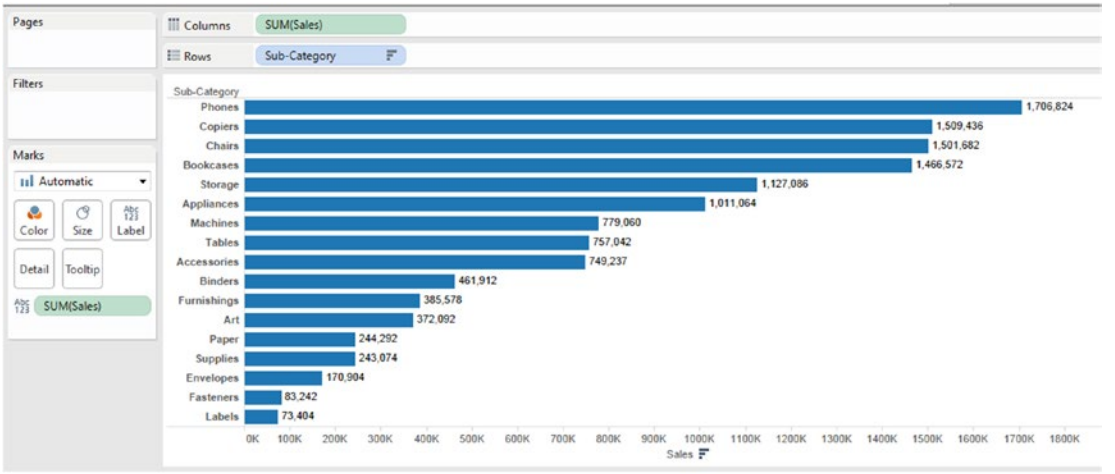


Figure 5-46. *Measure “Sales” placed on “Label” on the marks card*

Click on the drop down button on the Sum (Sales) to bring up the "Add Table Calculation" (Shown in Fig. 5-47).

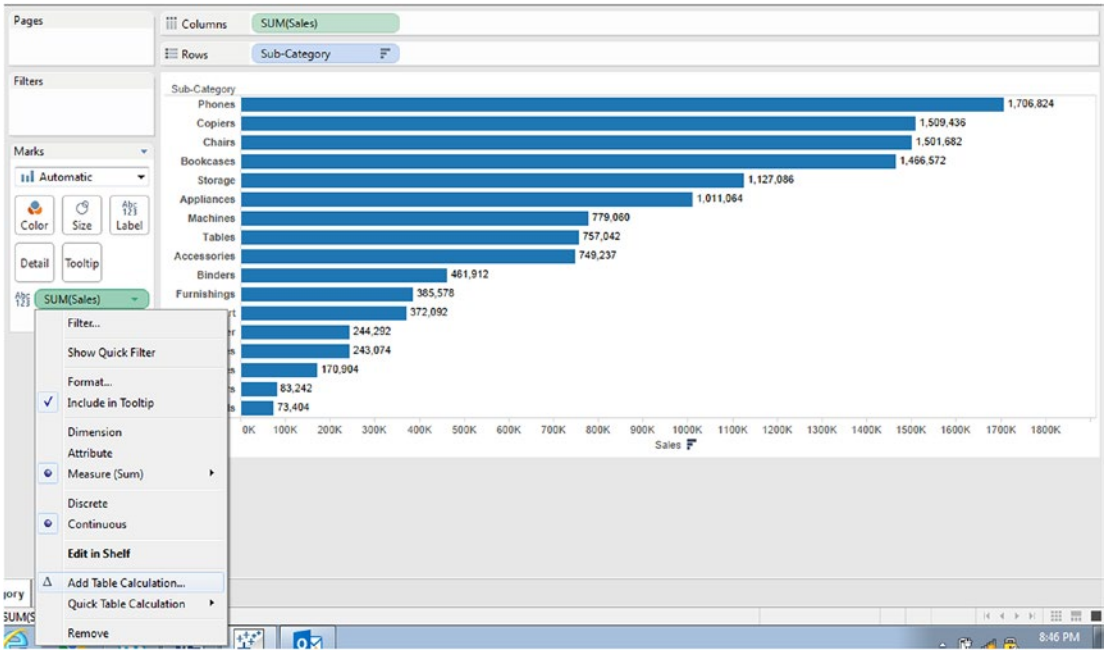


Figure 5-47. *Apply "Add Table Calculation" to the measure "Sales"*

Fill in the values in the "Table Calculation" dialog box as shown in Fig. 5-48.

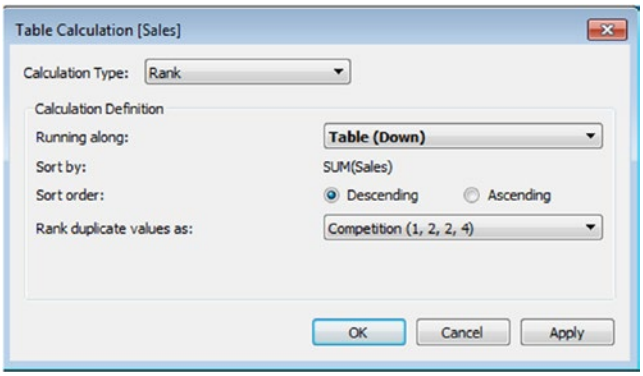


Figure 5-48. *"Table Calculation" dialog box for the measure "Sales"*

The output is as shown in Fig. 5-49.

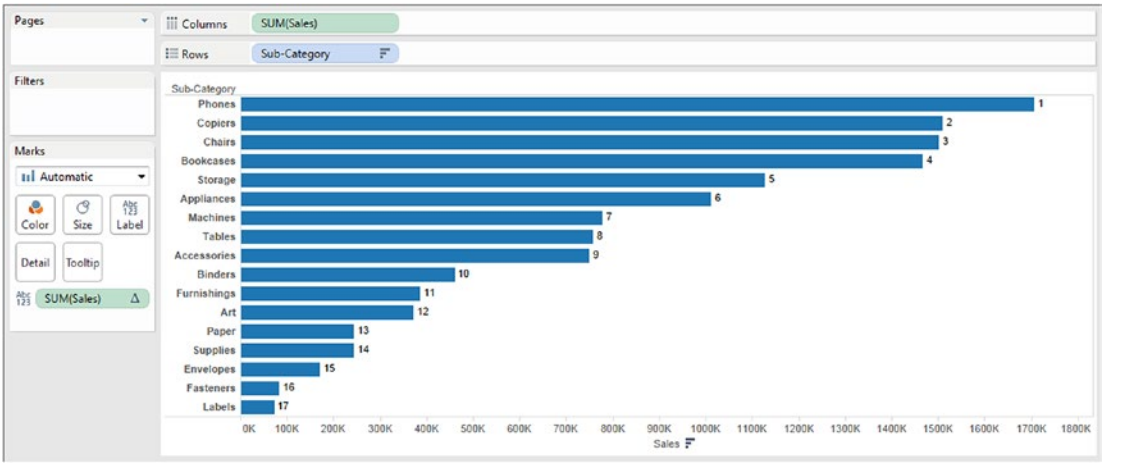


Figure 5-49. *Output after applying "Table Calculation - Rank" to the measure "Sales"*

5.5.1.1.5 Step 5

Let us display Rank as the leftmost column in the worksheet / view (Shown in Fig. 5-50).

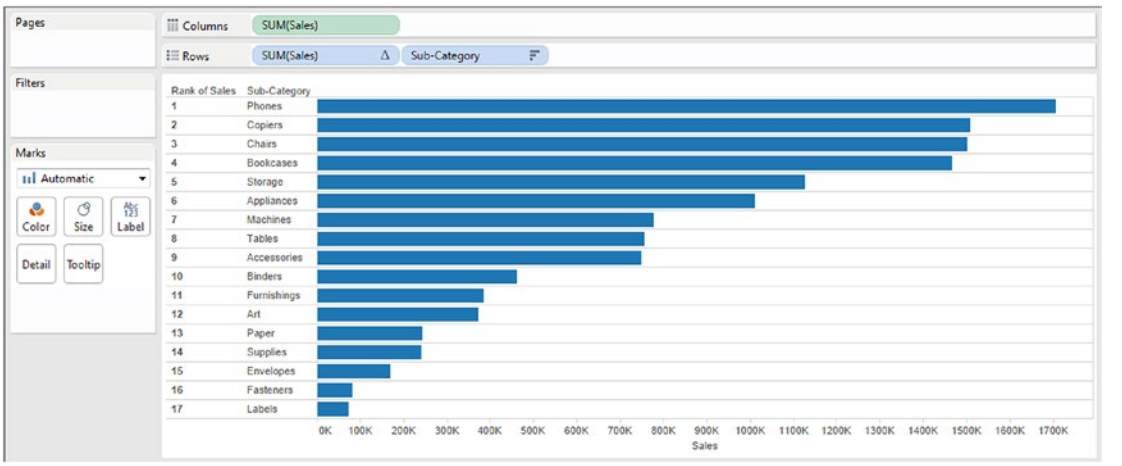


Figure 5-50. *"Rank" displayed as the leftmost column (Expected Output)*

How to achieve the above?

It is required to convert the Sum (Sales) **on the "Label"** on the marks card to "Discrete" (Shown in Fig. 5-51).

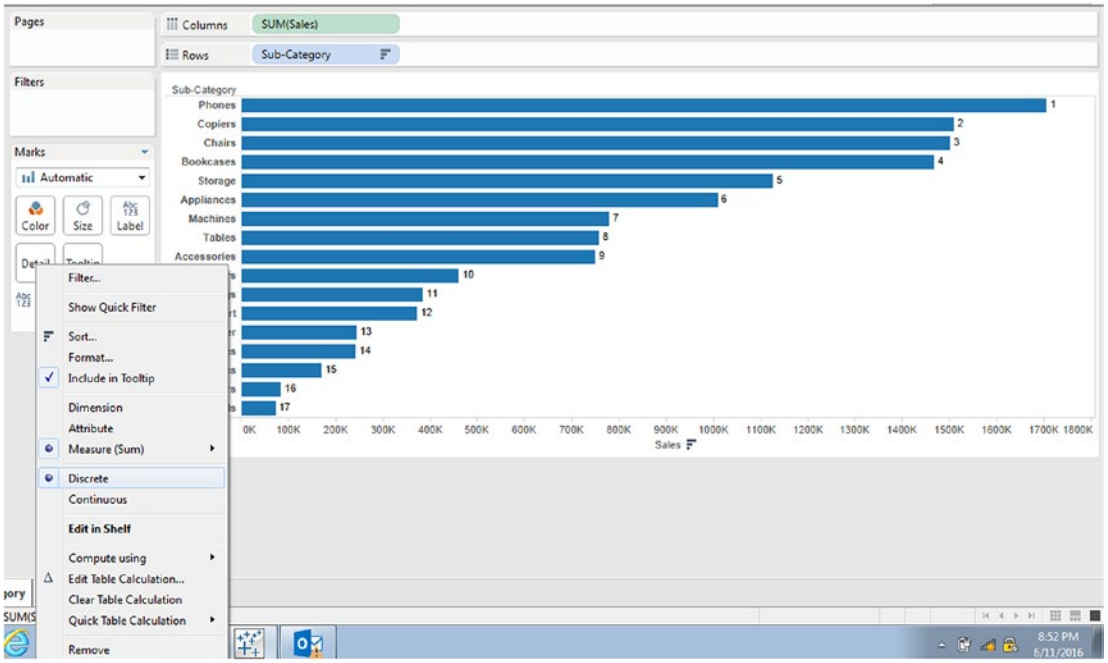


Figure 5-51. *Measure "Sales" converted to "Discrete"*

5.5.1.1.6 Step 6

Now drag the Sum (Sales) which is placed on the **"Label"** on the marks card and place it to the left of "Sub-Category" (Shown in Fig. 5-52).

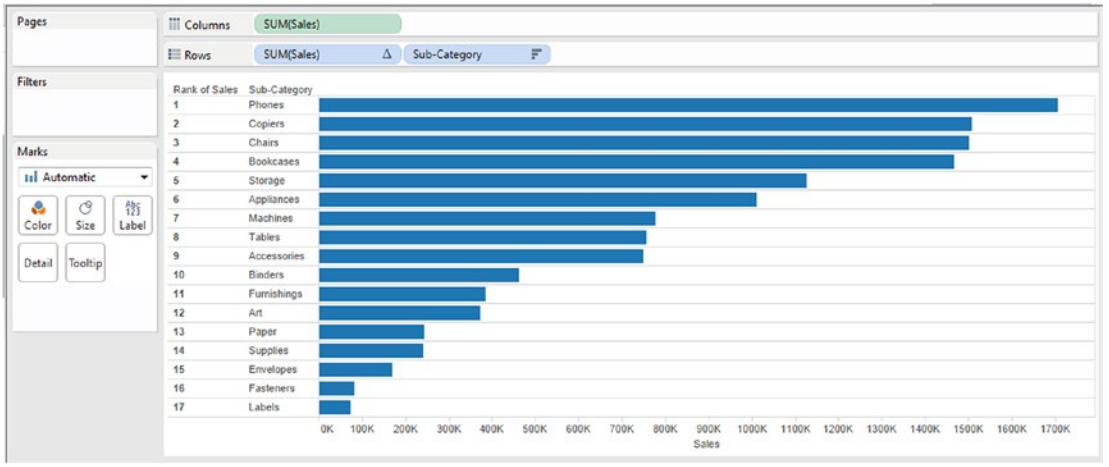


Figure 5-52. *Discrete measure "Sales" placed to the left of dimension "Sub-Category"*

5.5.1.1.7 Step 7

Next, let us add a "Category" to be the leftmost column on the rows shelf (Shown in Fig. 5-53).

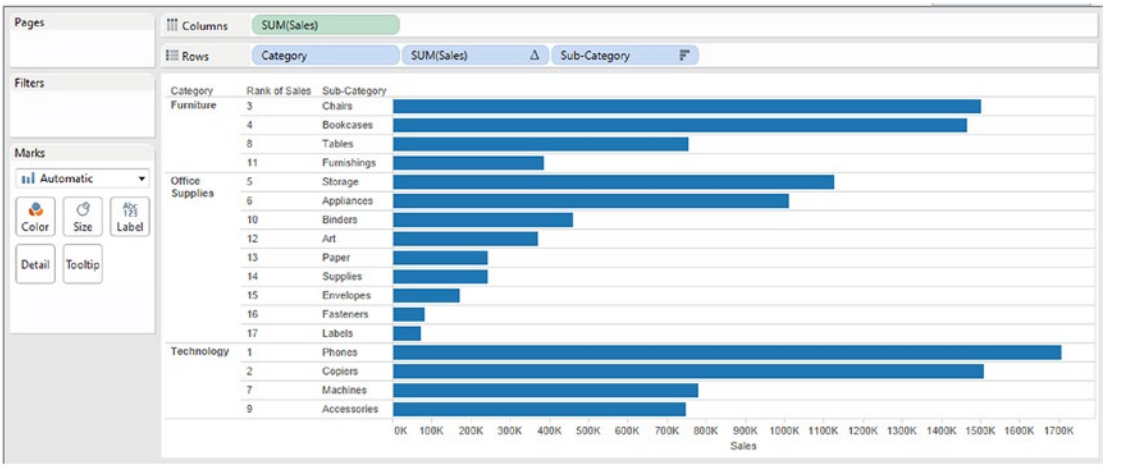


Figure 5-53. *Dimension "Category" placed as the leftmost column on the rows shelf*

The rank for each category should begin at 1 (Shown in Fig. 5-54).

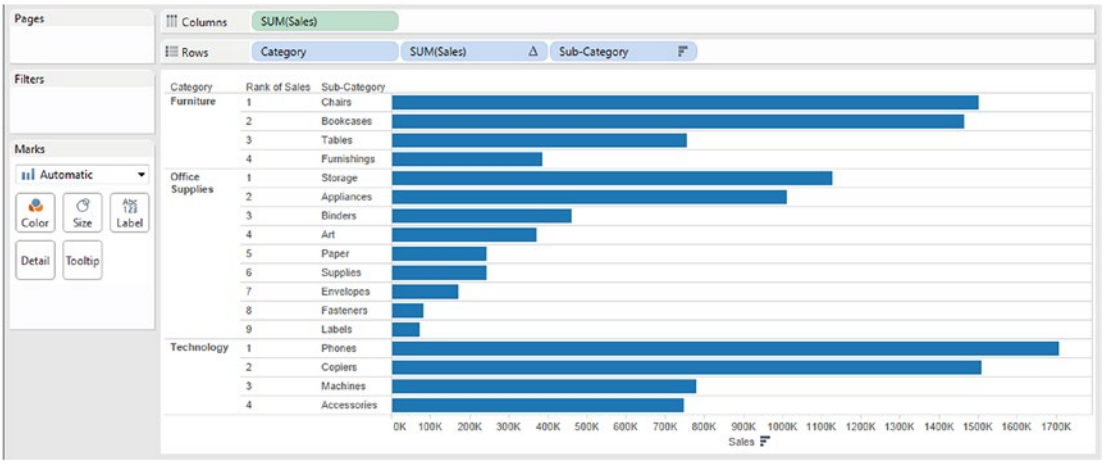


Figure 5-54. *Expected output: Rank for each category to begin at 1*

How to achieve the above?

5.5.1.1.8 Step 8

Click on the drop down on Sum (Sales) to bring up the "Edit Table Calculation" (Shown in Fig. 5-55).

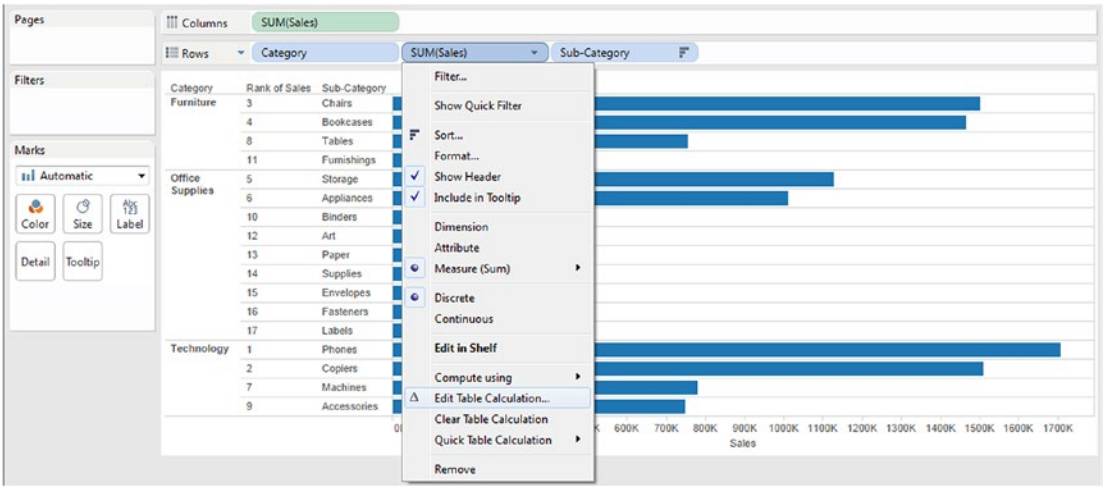


Figure 5-55. *Perform "Edit Table Calculation" to the measure "Sales"*

Change the "Running Along" to "Pane Down" (Figure 5-56).

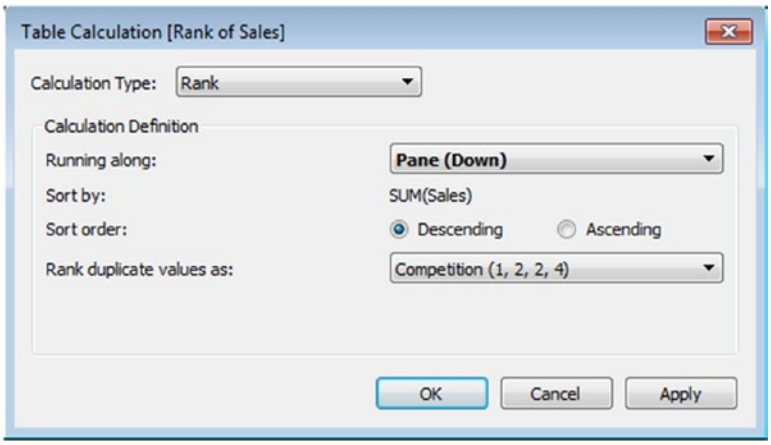


Figure 5-56. *"Running Along" for rank of sales changed to pane (Down)*

Click on “Apply” and then “OK”.
The final output is shown in Fig. 5-57.

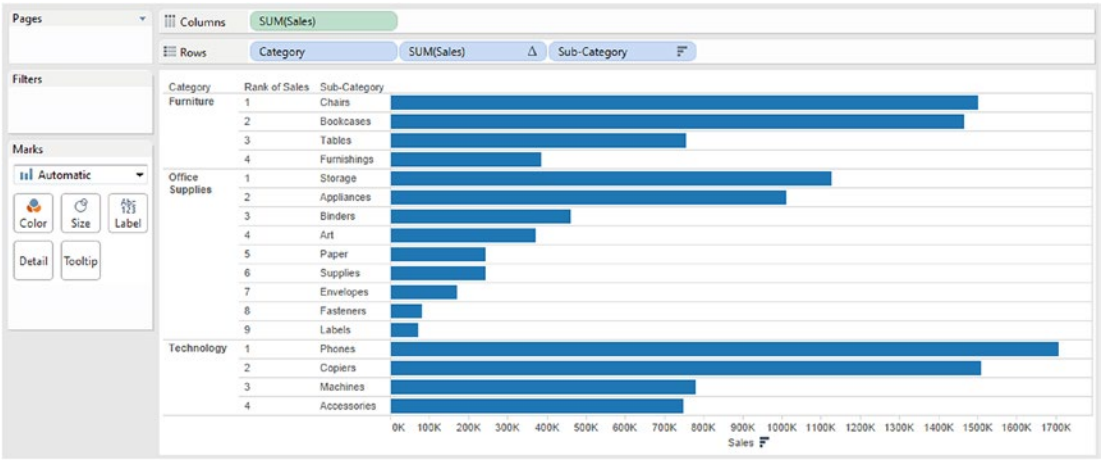


Figure 5-57. *Rank – Demo 1 – final output*

5.6 LOD (Level of Detail)

Level of detail (LOD) is a very important concept. An understanding of the idea of details helps with understanding the level of detail. Let us look at the areas where one can add details to the view / worksheet.

- Columns shelf
- Rows shelf
- Detail on the marks card

Details are defined by the dimensions that are used to segment the measures.

Example:

Drag the **dimension** “Region” from the dimensions area under the data pane and place it on the rows shelf.

Drag the **measure** “Sales” from the measures area under the data pane and place it on the columns shelf (Shown in Fig. 5-58).

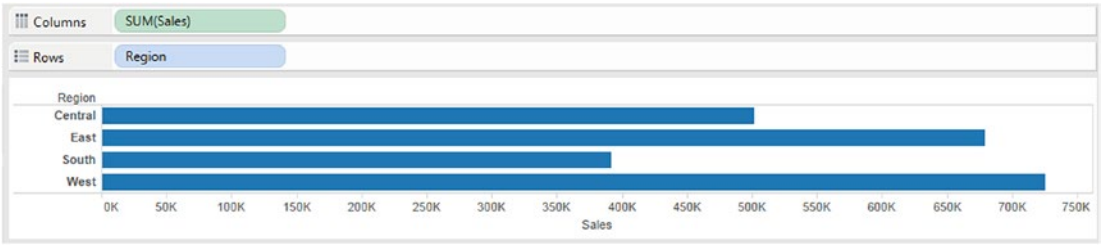


Figure 5-58. *Dimension “Region” & measure “Sales” placed on the rows shelf and the columns shelf, respectively*

The **measure** "Sales" (default aggregation is SUM) is aggregated by the **dimension** "Region".
Let us add another **dimension** "Category" to the rows shelf (Shown in Fig. 5-59).

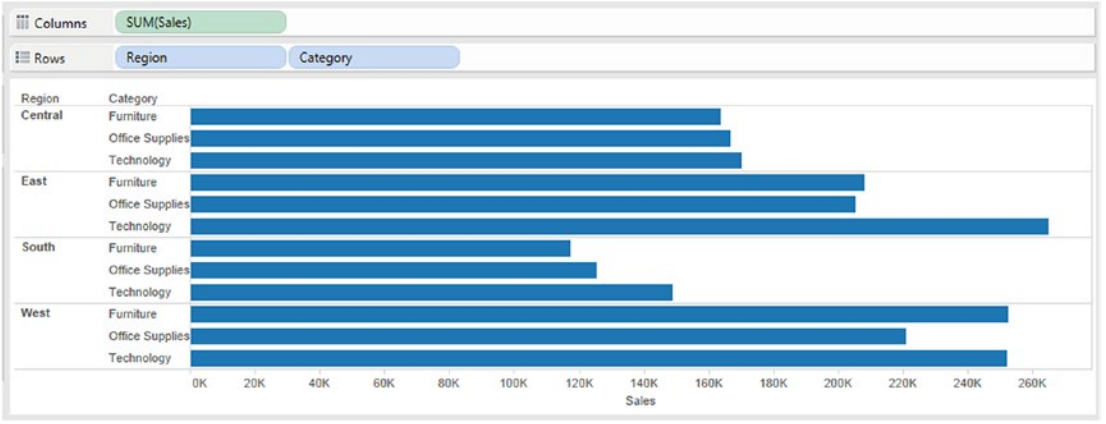


Figure 5-59. *Dimension "Category" placed on the rows shelf*

By placing the **dimension** "Category" on the rows shelf, we have added more granularity and less aggregation to the view / worksheet. It implies that we are adding to our level of detail.

Another area that one can add details is the Details on the marks card. The dimension or dimensions when added to the detail button or the detail shelf affects the visualization in different ways depending on the type of graph in the view or worksheet.

Example:

Drag the **measure** "Sales" from the measures area under the data pane and place it on the columns shelf. Drag the **measure** "Profit" from the measures area under the data pane and place it on the rows shelf (Shown in Fig. 5-60).

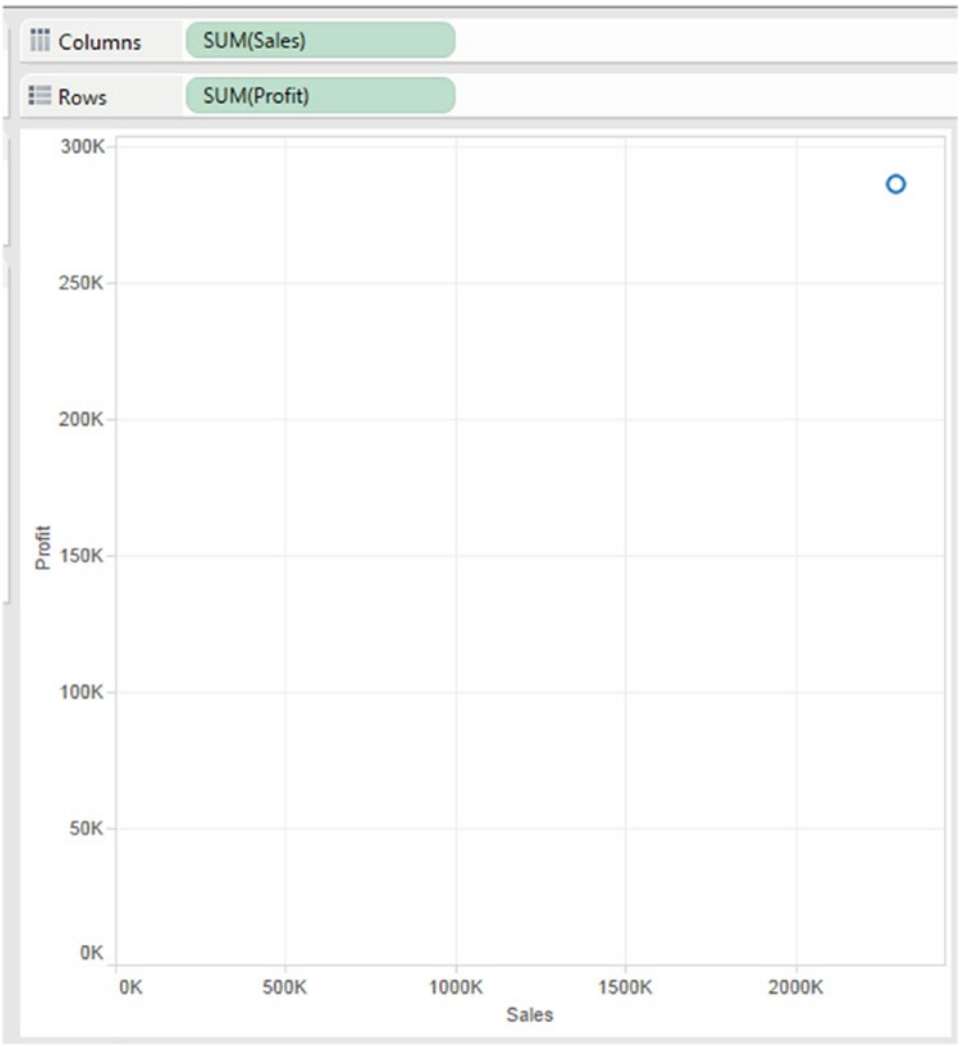


Figure 5-60. *Measures, "Sales" and "Profit" placed on columns and rows shelf, respectively*

The output is as shown in Fig. 5-60. The output is a scatter plot with a single mark on the view. The reason behind the single mark is that the measure is not yet segmented as per any dimension. Drag the **dimension** "Customer Name" from the dimensions area under the data pane and place it on "Details" on the marks card (See Fig. 5-61).

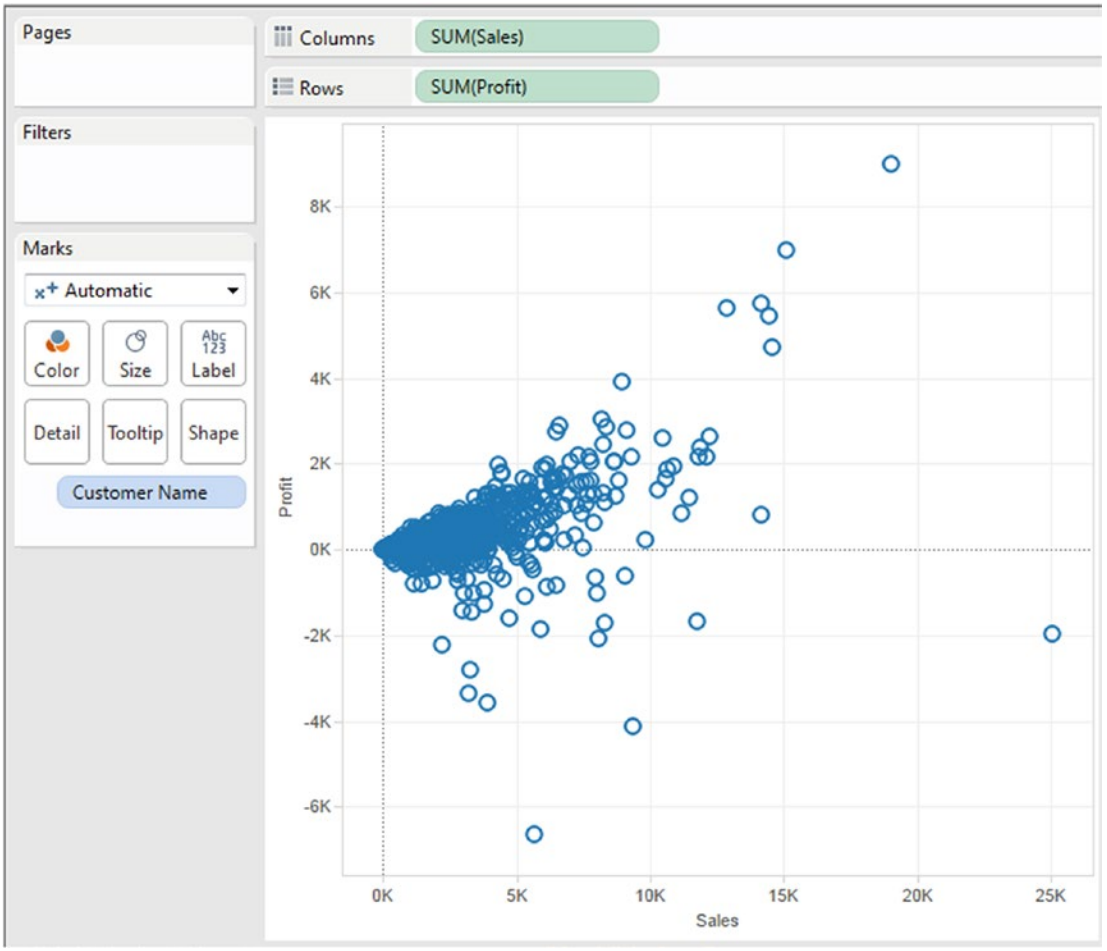


Figure 5-61. *Dimension "Customer Name" placed on "Detail" on the marks card*

The output in Fig. 5-61 shows the measure being segmented by "Customer Name". The view displays mark for every single customer. It makes the view more granular and less aggregated.

Yet another way to add details to the view is by using LOD (Level of Details). LOD expression represents an elegant and powerful way to answer questions involving multiple levels of granularity in a single visualization. Level of detail expressions provide a way to easily compute aggregations that are not at the level of detail of the visualization. You can then integrate those values within visualizations in arbitrary ways. From Tableau v9 onwards, a new concept called level of detail (LOD) expressions has been introduced. LOD expressions can be used to represent the data in different ways at different levels.

When can we use LOD expressions?

Consider using LOD expressions when:

- There is a requirement to show the data at a level different from the dimensions / level present in the view.
- There is a need to obtain some static calculated value that is not affected by any filters that are applied to the view.
- Refer for further reading: `http://www.tableaulearners.com/2016/level-detail-expressions-tableau/`

There are three options available with LOD. They are:

- Include
- Exclude
- Fixed

5.6.1 Demo 1

Objective: To demonstrate the "Level of Detail – Exclude".

Input: "Sample – Superstore.xls"

Expected Output: Shown in Fig. 5-62.

Region	State	City	Region_Sales	State_Sales	City_Sales
Central	Iowa	Dubuque	501,240	4,580	1,687
		Iowa City	501,240	4,580	10
		Marion	501,240	4,580	358
		Urbandale	501,240	4,580	149
		Waterloo	501,240	4,580	30
	Kansas	Garden City	501,240	2,914	312
		Manhattan	501,240	2,914	274
		Olathe	501,240	2,914	896
		Overland Park	501,240	2,914	607
		Wichita	501,240	2,914	825
	Michigan	Ann Arbor	501,240	76,270	889
		Canton	501,240	76,270	818
		Dearborn	501,240	76,270	1,603
		Dearborn Heights	501,240	76,270	1,052
		Detroit	501,240	76,270	42,447
		Grand Rapids	501,240	76,270	526
		Holland	501,240	76,270	138
		Jackson	501,240	76,270	15,420
		Lansing	501,240	76,270	1,610
		Lincoln Park	501,240	76,270	388
		Midland	501,240	76,270	5,292
		Mount Pleasant	501,240	76,270	17
		Oak Park	501,240	76,270	581
		Rochester Hills	501,240	76,270	133
		Roseville	501,240	76,270	638

Figure 5-62. *"LOD Exclude" – Demo 1 – expected output*

Explanation of the output:

The view / worksheet should display the sales by region, by state and by city alongside the dimensions, "Region", "State" and "City".

Let us split the output:

Sales by Region: Shown in Fig. 5-63.

Region	
Central	501,240
East	678,781
South	391,722
West	725,458

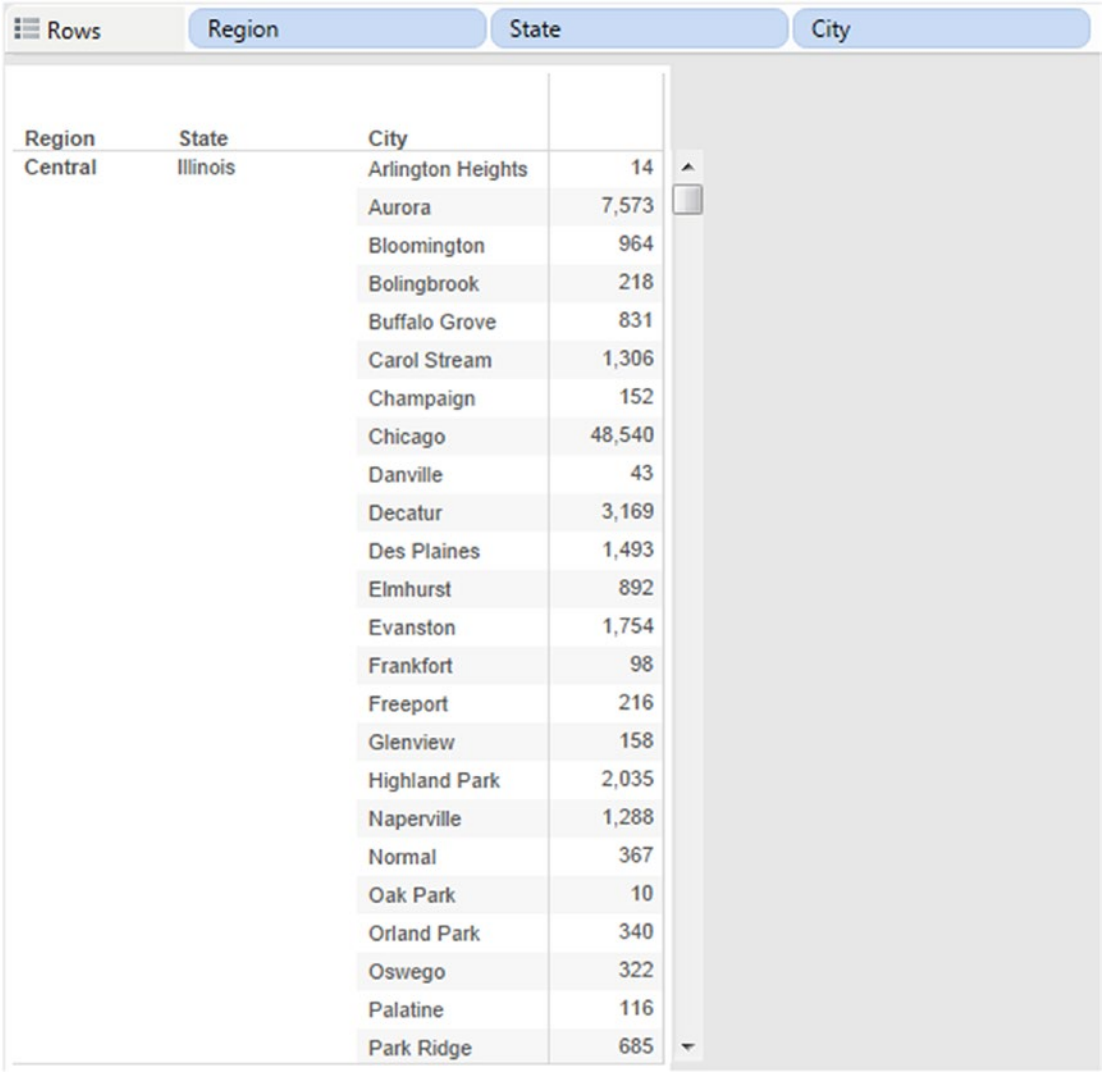
Figure 5-63. *Sales by Region*

Sales by State: Shown in Fig. 5-64.

Region	State	
Central	Illinois	80,166
	Indiana	53,555
	Iowa	4,580
	Kansas	2,914
	Michigan	76,270
	Minnesota	29,863
	Missouri	22,205
	Nebraska	7,465
	North Dakota	920
	Oklahoma	19,683
	South Dakota	1,316
	Texas	170,188
	Wisconsin	32,115
East	Connecticut	13,384
	Delaware	27,451
	District of Columb..	2,865
	Maine	1,271
	Maryland	23,706
	Massachusetts	28,634
	New Hampshire	7,293
	New Jersey	35,764
	New York	310,876
	Ohio	78,258
	Pennsylvania	116,512

Figure 5-64. *Sales by State*

Sales by City: Shown in Fig. 5-65.



Region	State	City	
Central	Illinois	Arlington Heights	14
		Aurora	7,573
		Bloomington	964
		Bolingbrook	218
		Buffalo Grove	831
		Carol Stream	1,306
		Champaign	152
		Chicago	48,540
		Danville	43
		Decatur	3,169
		Des Plaines	1,493
		Elmhurst	892
		Evanston	1,754
		Frankfort	98
		Freeport	216
		Glenview	158
		Highland Park	2,035
		Naperville	1,288
		Normal	367
		Oak Park	10
		Orland Park	340
		Oswego	322
		Palatine	116
		Park Ridge	685

Figure 5-65. *Sales by city*

The challenge is to combine all the three outputs stated above in a single view / worksheet. Let us look at "LOD – Exclude" to accomplish the above output.

5.6.1.1 Steps

5.6.1.1.1 Step 1

Read in the data from "Sample – Superstore.xls" into Tableau (Shown in Fig. 5-66).

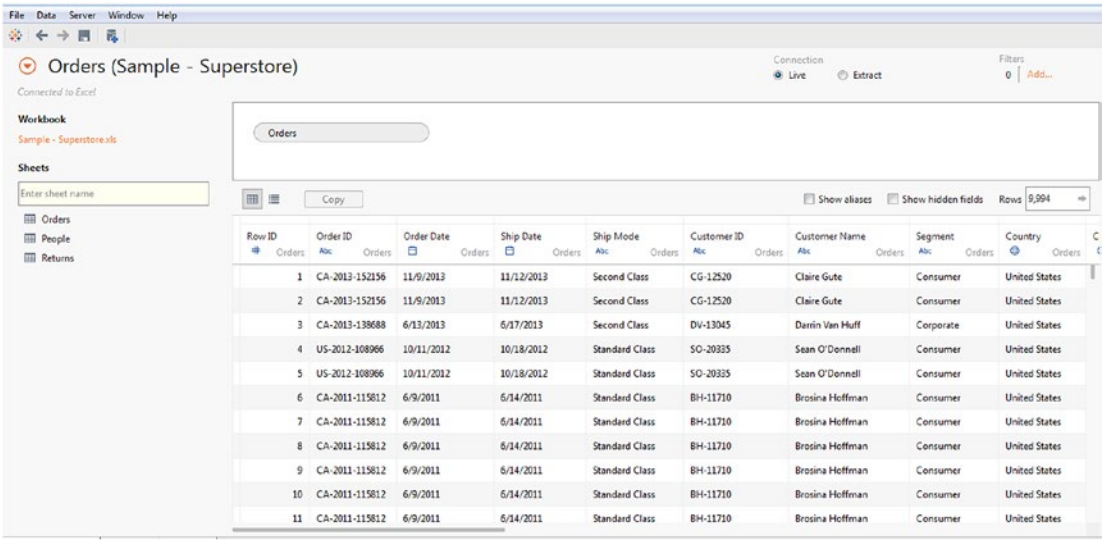


Figure 5-66. *Data from "Sample - Superstore" read into Tableau*

5.6.1.1.2 Step 2

Drag the **dimension** "Region" from the dimensions area under the data pane to the rows shelf (Shown in Fig. 5-67).

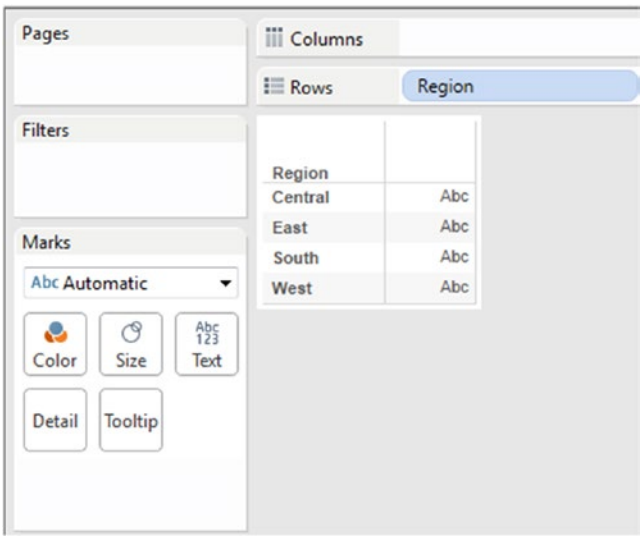


Figure 5-67. *Dimension "Region" placed on the rows shelf*

Drag the **dimension** "State" from the dimensions area under the data pane to the rows shelf (Shown in Fig. 5-68).

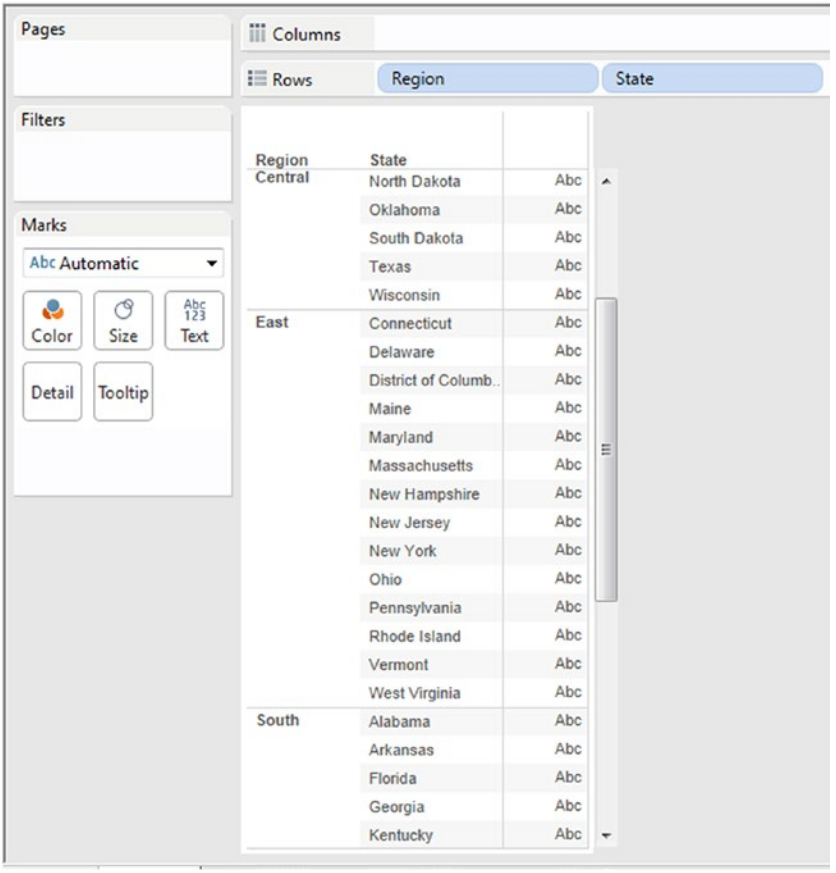


Figure 5-68. *Dimension "State" placed on the rows shelf*

Drag the **dimension** "City" from the dimensions area under the data pane to the rows shelf (Shown in Fig. 5-69).

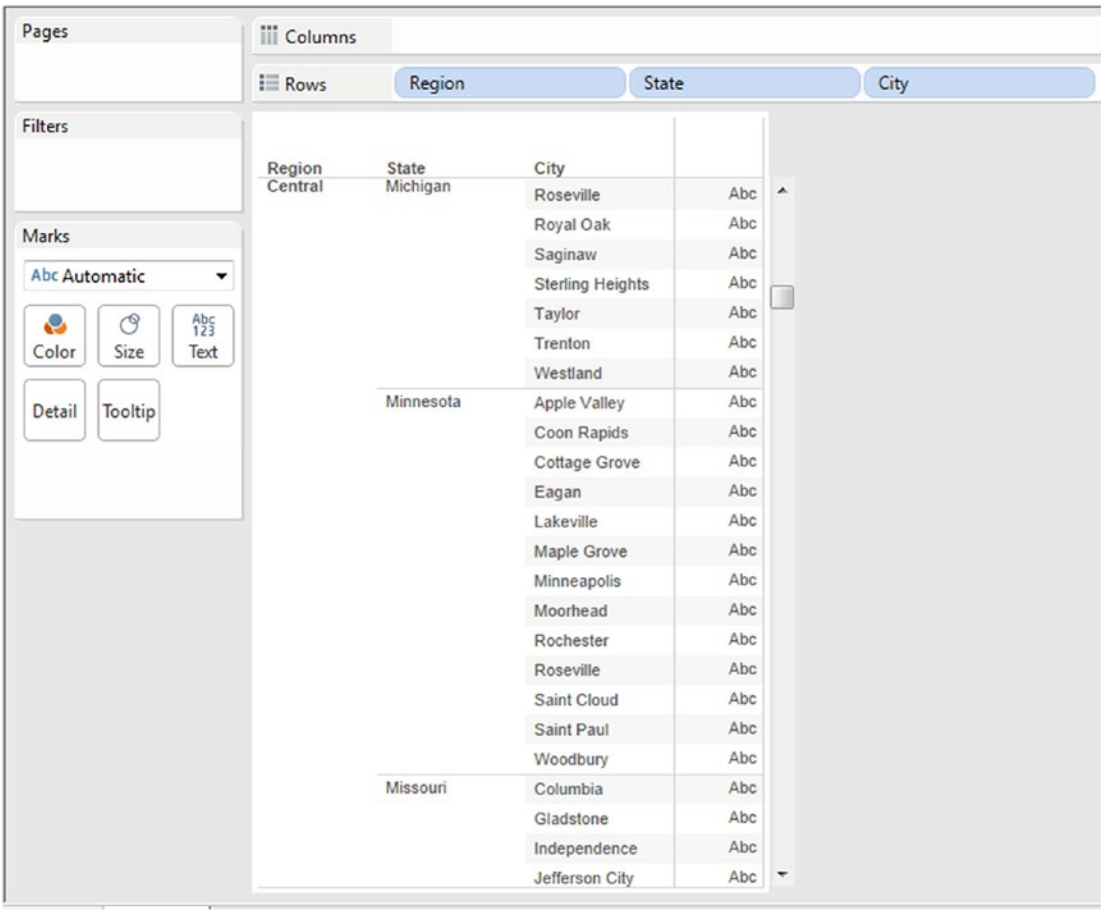


Figure 5-69. *Dimension "City" placed on the rows shelf*

5.6.1.1.3 Step 3

Create two calculated fields, "Exclude_Region_Sales" and "Exclude_State_Sales".

Exclude_Region_Sales: To get the sales by region, we will exclude the "State" and "City" dimensions (Shown in Fig. 5-70).

Figure 5-70. *Calculated field, "Exclude_Region_Sales" being created*

Exclude_State_Sales: To get the sales by state, we will exclude the "City" dimension (Shown in Fig. 5-71).

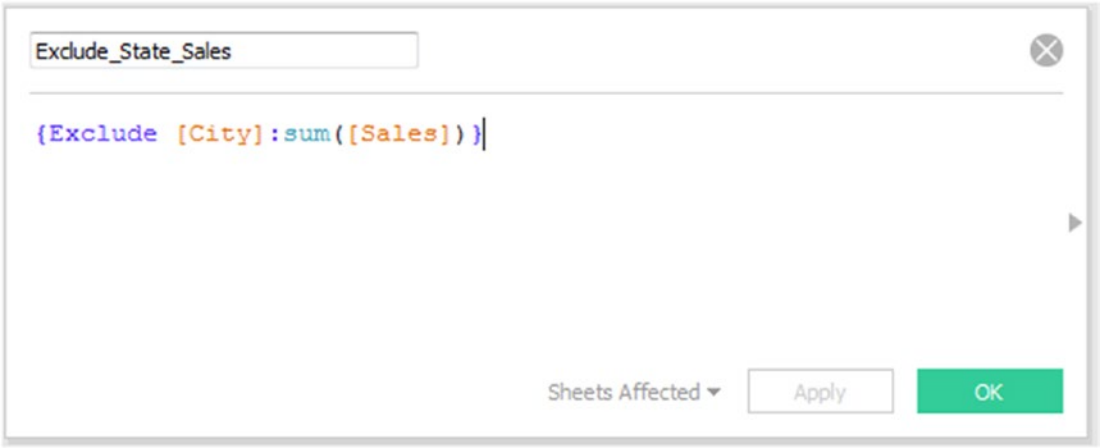


Figure 5-71. *Calculated field field "Exclude_State_Sales" being created*

5.6.1.1.4 Step 4

Drag the **measure** "Exclude_Region_Sales" from the measures area under the data pane and drop it into the view/worksheet (Shown in Fig. 5-72).

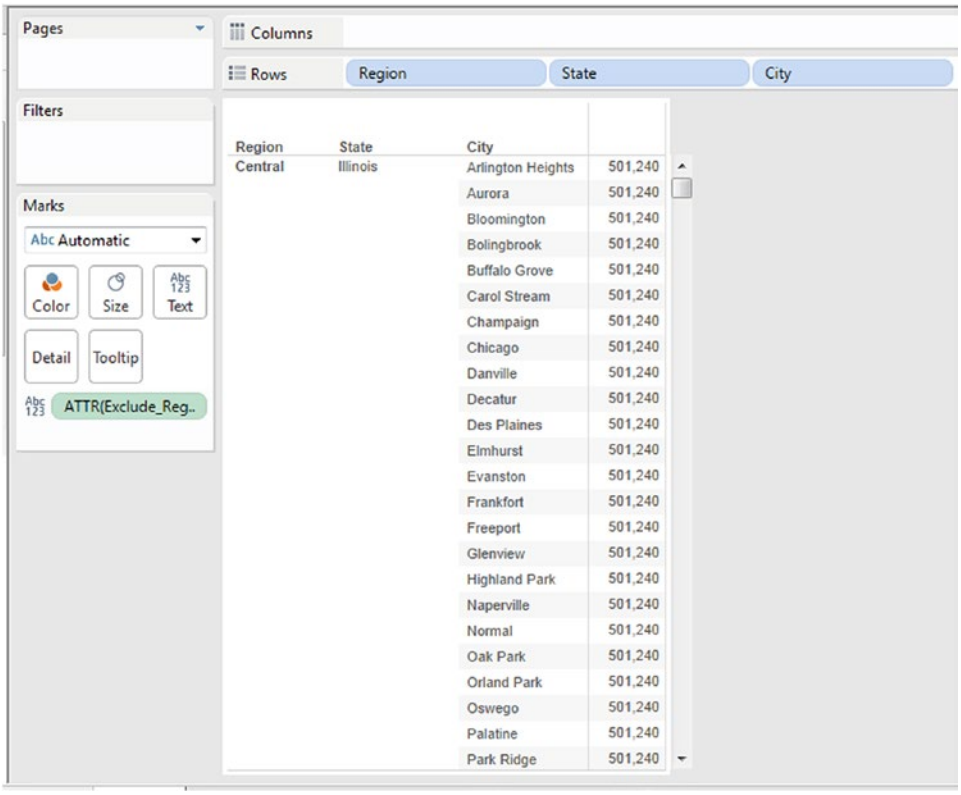
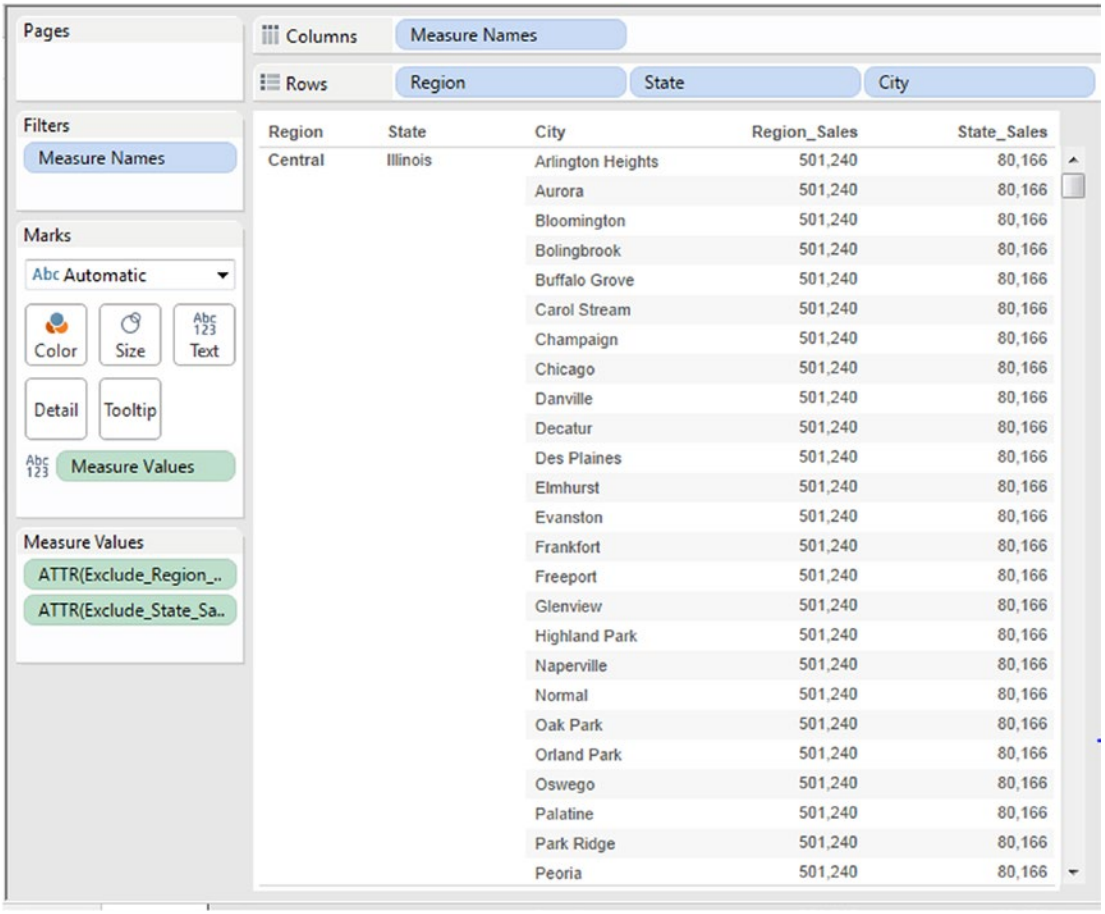


Figure 5-72. *Calculated field, "Exclude_Region_Sales" placed on the view*

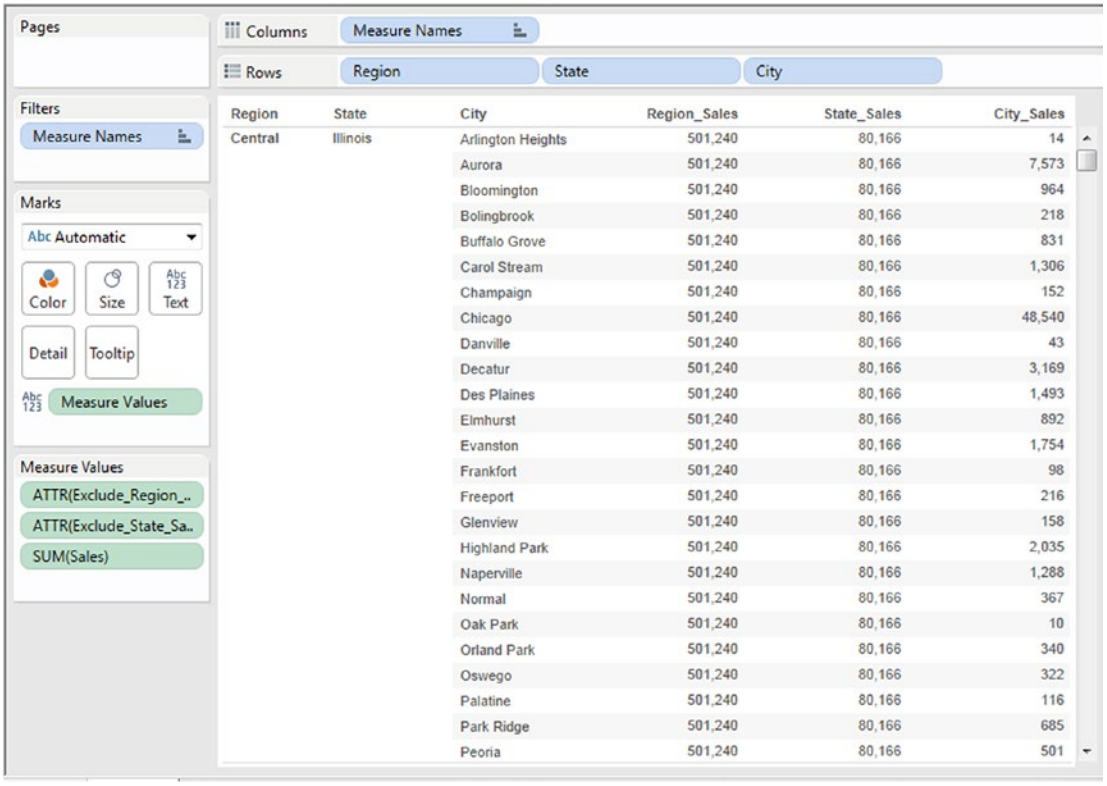
Drag the **measure** "Exclude_State_Sales" from the measures area under the data pane and drop it into the view/worksheet (Shown in Fig. 5-73).



Region	State	City	Region_Sales	State_Sales
Central	Illinois	Arlington Heights	501,240	80,166
		Aurora	501,240	80,166
		Bloomington	501,240	80,166
		Bolingbrook	501,240	80,166
		Buffalo Grove	501,240	80,166
		Carol Stream	501,240	80,166
		Champaign	501,240	80,166
		Chicago	501,240	80,166
		Danville	501,240	80,166
		Decatur	501,240	80,166
		Des Plaines	501,240	80,166
		Elmhurst	501,240	80,166
		Evanston	501,240	80,166
		Frankfort	501,240	80,166
		Freeport	501,240	80,166
		Glenview	501,240	80,166
		Highland Park	501,240	80,166
		Naperville	501,240	80,166
		Normal	501,240	80,166
		Oak Park	501,240	80,166
		Orland Park	501,240	80,166
		Oswego	501,240	80,166
		Palatine	501,240	80,166
		Park Ridge	501,240	80,166
		Peoria	501,240	80,166

Figure 5-73. *Calculated field "Exclude_State_Sales" placed on the View*

Drag the **measure** "Sales" from the measures area under the data pane and drop it into the view/ worksheet (Shown in Fig. 5-74).



Region	State	City	Region_Sales	State_Sales	City_Sales
Central	Illinois	Arlington Heights	501,240	80,166	14
		Aurora	501,240	80,166	7,573
		Bloomington	501,240	80,166	964
		Bolingbrook	501,240	80,166	218
		Buffalo Grove	501,240	80,166	831
		Carol Stream	501,240	80,166	1,306
		Champaign	501,240	80,166	152
		Chicago	501,240	80,166	48,540
		Danville	501,240	80,166	43
		Decatur	501,240	80,166	3,169
		Des Plaines	501,240	80,166	1,493
		Elmhurst	501,240	80,166	892
		Evanston	501,240	80,166	1,754
		Frankfort	501,240	80,166	98
		Freeport	501,240	80,166	216
		Glenview	501,240	80,166	158
		Highland Park	501,240	80,166	2,035
		Naperville	501,240	80,166	1,288
		Normal	501,240	80,166	367
		Oak Park	501,240	80,166	10
		Orland Park	501,240	80,166	340
		Oswego	501,240	80,166	322
		Palatine	501,240	80,166	116
		Park Ridge	501,240	80,166	685
		Peoria	501,240	80,166	501

Figure 5-74. *The measure "Sales", placed on the view*

The final output (Shown in Fig. 5-75):

Region	State	City	Region_Sales	State_Sales	City_Sales
Central	Iowa	Dubuque	501,240	4,580	1,687
		Iowa City	501,240	4,580	10
		Marion	501,240	4,580	358
		Urbandale	501,240	4,580	149
		Waterloo	501,240	4,580	30
	Kansas	Garden City	501,240	2,914	312
		Manhattan	501,240	2,914	274
		Olathe	501,240	2,914	896
		Overland Park	501,240	2,914	607
		Wichita	501,240	2,914	825
	Michigan	Ann Arbor	501,240	76,270	889
		Canton	501,240	76,270	818
		Dearborn	501,240	76,270	1,603
		Dearborn Heights	501,240	76,270	1,052
		Detroit	501,240	76,270	42,447
		Grand Rapids	501,240	76,270	526
		Holland	501,240	76,270	138
		Jackson	501,240	76,270	15,420
		Lansing	501,240	76,270	1,610
		Lincoln Park	501,240	76,270	388
		Midland	501,240	76,270	5,292
		Mount Pleasant	501,240	76,270	17
		Oak Park	501,240	76,270	581
		Rochester Hills	501,240	76,270	133
		Roseville	501,240	76,270	638

Figure 5-75. *"LOD – Exclude" – Demo 1 – final output*

Can you answer this?

What will happen if a dimension that is not in the view is excluded in the LOD calculation? The answer is nothing will change in the view. The exclude LOD calculation returns results relative to your visualization; this implies that it does matter what dimensions are used in the view.

5.6.2 Demo 2

Objective: To demonstrate level of detail – fixed.

Fixed LOD calculations are not relative to the view. They focus only on the dimension that we use in the "Fixed LOD Calculations", regardless of what is or what is not included in the view.

Input: "Sample – Superstore.xls"

Expected Output: Shown in Fig. 5-76.

Region	State	City	Fixed_Region_S..	Fixed_State_Sales	City_Sales
Central	Illinois	Arlington Heights	501,240	80,166	14
		Aurora	501,240	80,166	7,573
		Bloomington	501,240	80,166	964
		Bolingbrook	501,240	80,166	218
		Buffalo Grove	501,240	80,166	831
		Carol Stream	501,240	80,166	1,306
		Champaign	501,240	80,166	152
		Chicago	501,240	80,166	48,540
		Danville	501,240	80,166	43
		Decatur	501,240	80,166	3,169
		Des Plaines	501,240	80,166	1,493
		Elmhurst	501,240	80,166	892
		Evanston	501,240	80,166	1,754
		Frankfort	501,240	80,166	98
		Freeport	501,240	80,166	216
		Glenview	501,240	80,166	158
		Highland Park	501,240	80,166	2,035
		Naperville	501,240	80,166	1,288
		Normal	501,240	80,166	367
		Oak Park	501,240	80,166	10
		Orland Park	501,240	80,166	340
		Oswego	501,240	80,166	322
		Palatine	501,240	80,166	116
		Park Ridge	501,240	80,166	685
		Peoria	501,240	80,166	501

Figure 5-76. *"LOD – Fixed" – Demo 2 – expected output*

Explanation of the expected output:

We were able to get the above output using LOD – exclude, but we had to be cognizant of the dimensions present in the view. We would like to get the above output without any consideration to the dimensions present in the view. We will accomplish this using LOD – fixed. LOD – fixed provides us with increased flexibility and can be used across worksheets.

5.6.2.1 Steps

Follow the steps as provided.

5.6.2.1.1 Step 1

Read in the data from "Sample – Superstore.xls" into Tableau (Shown in Fig. 5-77).

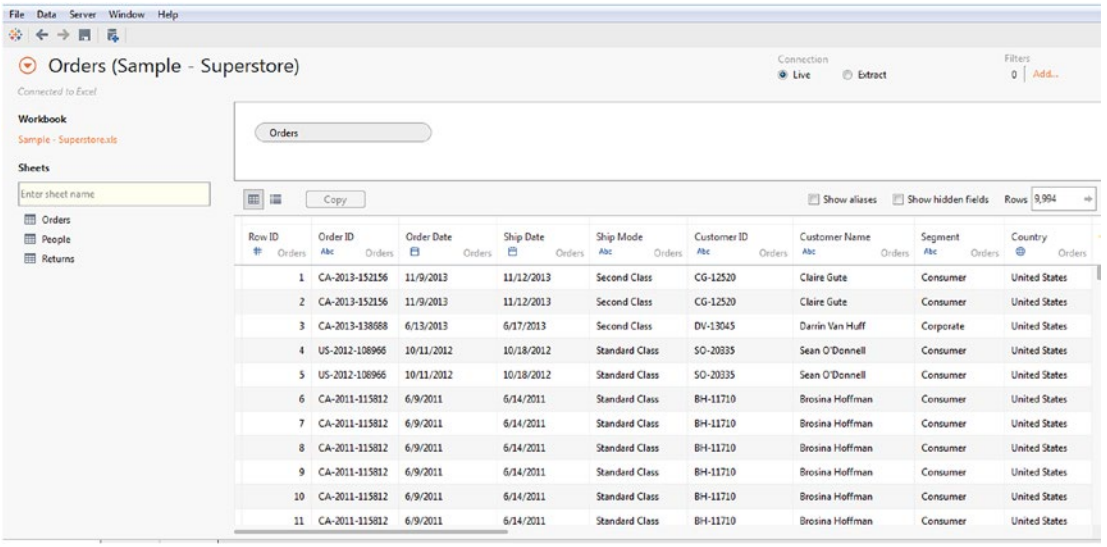


Figure 5-77. *Data from "Sample -Superstore.xls" read into Tableau*

5.6.2.1.2 Step 2

Drag the **dimension** "Region" from the dimensions area under the data pane to the rows shelf (Shown in Fig. 5-78).

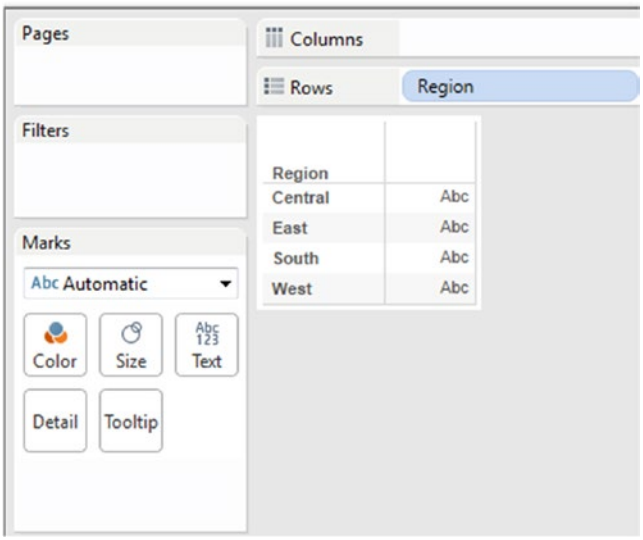


Figure 5-78. *Dimension "Region" placed on the rows shelf*

Drag the **dimension** "State" from the dimensions area under the data pane to the rows shelf (Shown in Fig. 5-79).

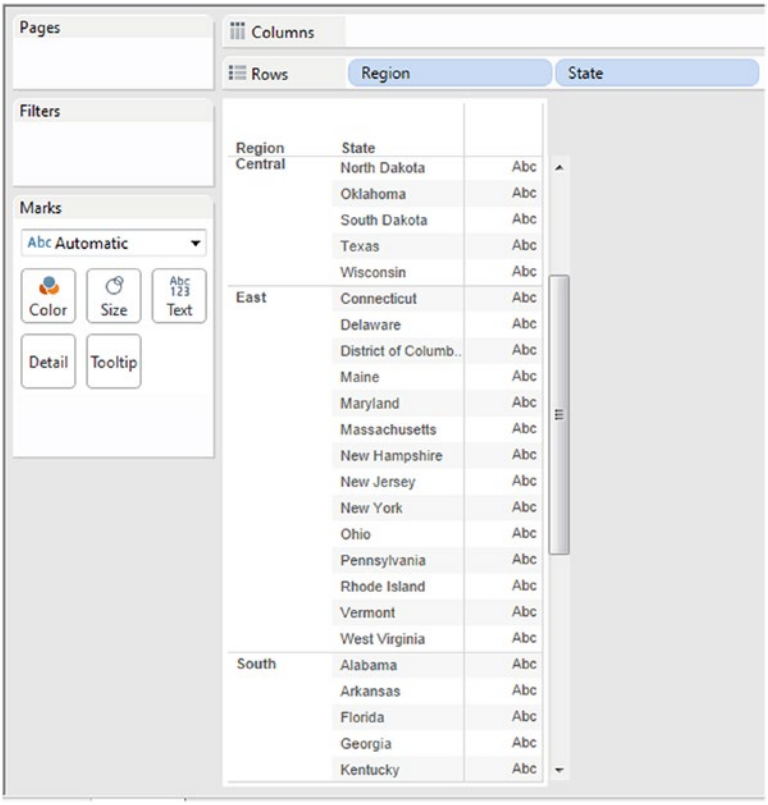


Figure 5-79. *Dimension "State" placed on the rows shelf*

Drag the **dimension** "City" from the dimensions area under the data pane to the rows shelf (Shown in Fig. 5-80).

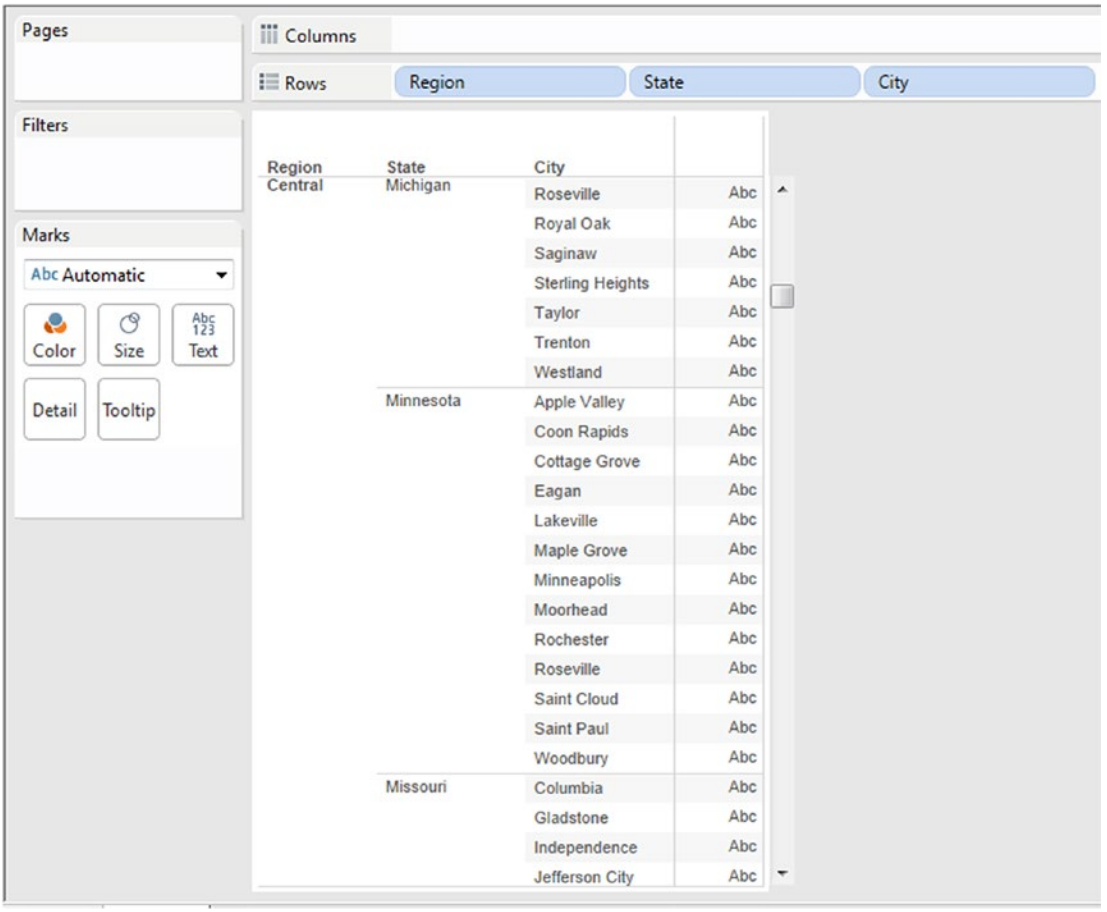


Figure 5-80. *Dimension "City" placed on the rows shelf*

5.6.2.1.3 Step 3

Create two calculated fields, "Fixed_Region_Sales" and "Fixed_State_Sales" (Shown in Fig. 5-81 & Figure 5-82).

Figure 5-81. *Calculated field field "Fixed_Region_Sales" being created*

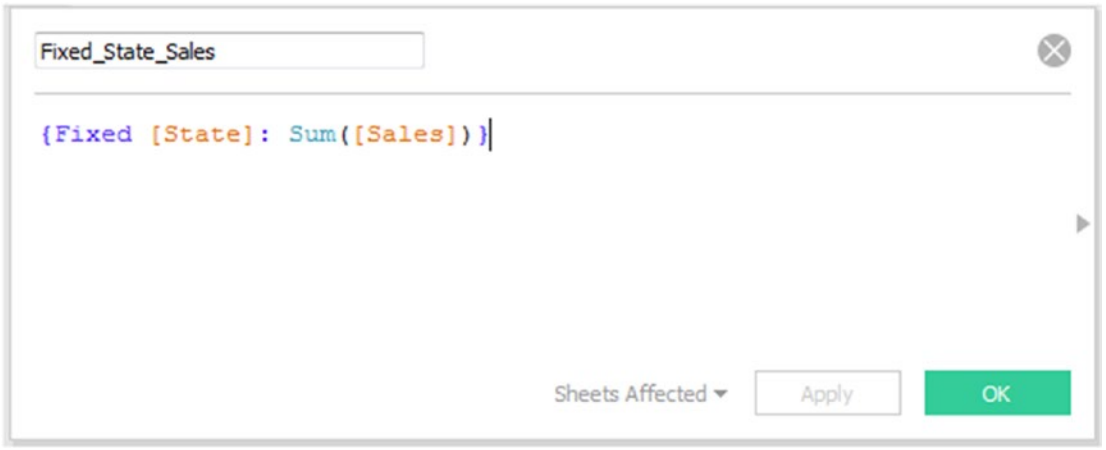


Figure 5-82. *Calculated field field "Fixed_State_Sales" being created*

5.6.2.1.4 Step 4

Drag the **measure** "Fixed_Region_Sales" from the measures area under the data pane and drop it into the view/worksheet (Shown in Fig. 5-83).

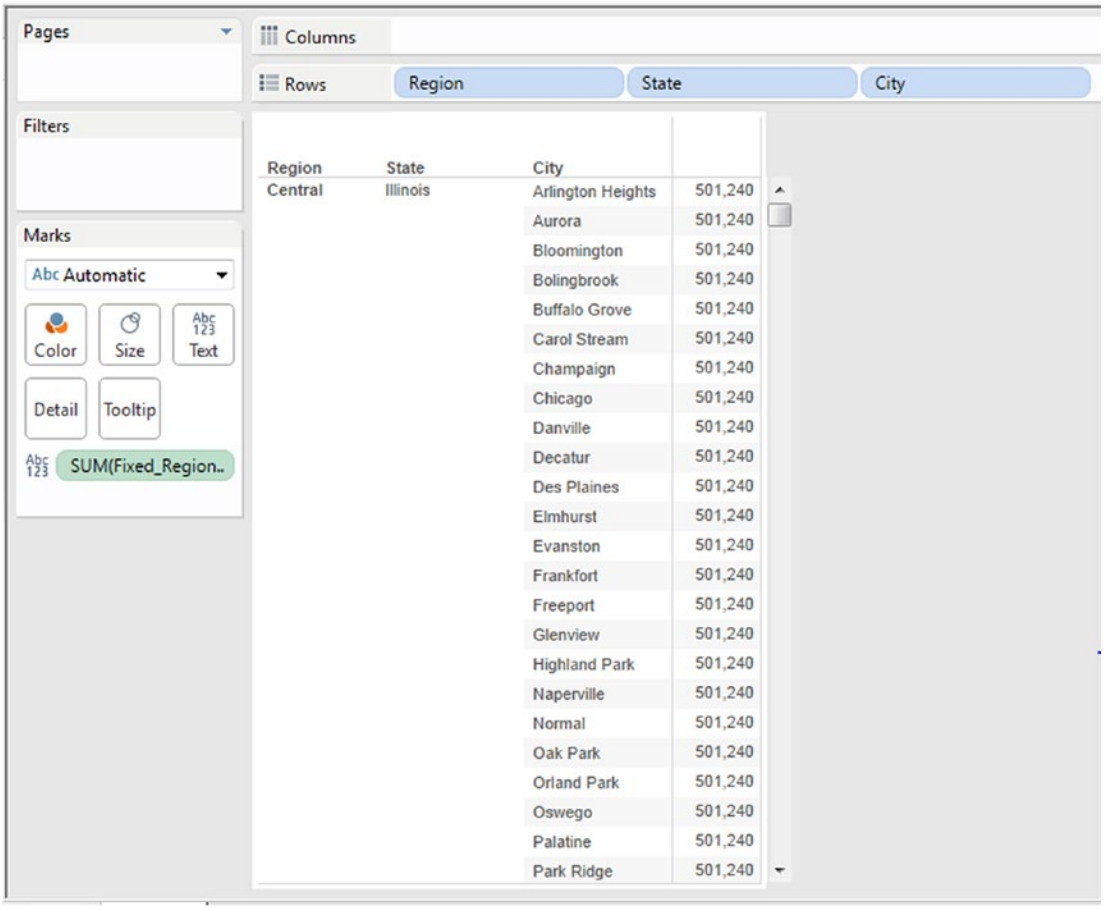
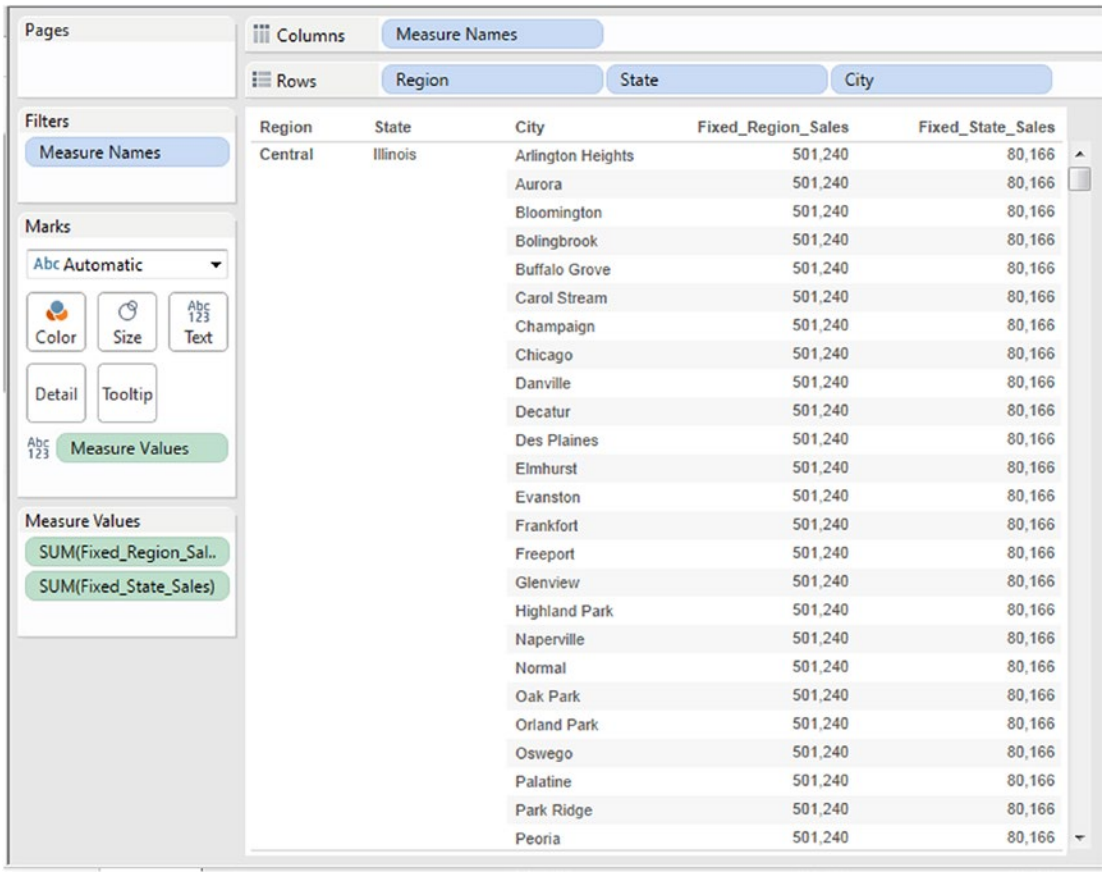


Figure 5-83. *Calculated field "Fixed_Region_Sales" placed on the view*

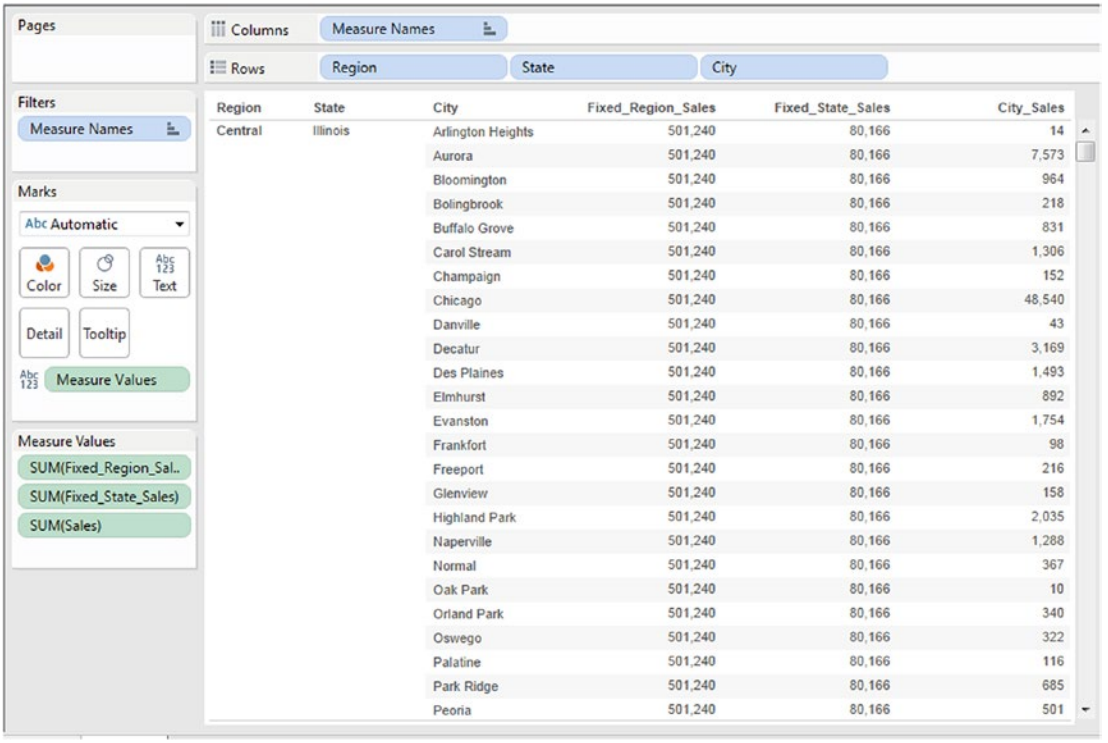
Drag the **measure** "Fixed_State_Sales" from the measures area under the data pane and drop it into the view/worksheet (Shown in Fig. 5-84).



Region	State	City	Fixed_Region_Sales	Fixed_State_Sales
Central	Illinois	Arlington Heights	501,240	80,166
		Aurora	501,240	80,166
		Bloomington	501,240	80,166
		Bolingbrook	501,240	80,166
		Buffalo Grove	501,240	80,166
		Carol Stream	501,240	80,166
		Champaign	501,240	80,166
		Chicago	501,240	80,166
		Danville	501,240	80,166
		Decatur	501,240	80,166
		Des Plaines	501,240	80,166
		Elmhurst	501,240	80,166
		Evanston	501,240	80,166
		Frankfort	501,240	80,166
		Freeport	501,240	80,166
		Glenview	501,240	80,166
		Highland Park	501,240	80,166
		Naperville	501,240	80,166
		Normal	501,240	80,166
		Oak Park	501,240	80,166
		Orland Park	501,240	80,166
		Oswego	501,240	80,166
		Palatine	501,240	80,166
		Park Ridge	501,240	80,166
		Peoria	501,240	80,166

Figure 5-84. *Calculated field field "Fixed_State_Sales" placed on the View*

Drag the **measure** "Sales" from the measures area under the data pane and drop it into the view/ worksheet (Shown in Fig. 5-85).



Region	State	City	Fixed_Region_Sales	Fixed_State_Sales	City_Sales
Central	Illinois	Arlington Heights	501,240	80,166	14
		Aurora	501,240	80,166	7,573
		Bloomington	501,240	80,166	964
		Bolingbrook	501,240	80,166	218
		Buffalo Grove	501,240	80,166	831
		Carol Stream	501,240	80,166	1,306
		Champaign	501,240	80,166	152
		Chicago	501,240	80,166	48,540
		Danville	501,240	80,166	43
		Decatur	501,240	80,166	3,169
		Des Plaines	501,240	80,166	1,493
		Elmhurst	501,240	80,166	892
		Evanston	501,240	80,166	1,754
		Frankfort	501,240	80,166	98
		Freeport	501,240	80,166	216
		Glenview	501,240	80,166	158
		Highland Park	501,240	80,166	2,035
		Naperville	501,240	80,166	1,288
		Normal	501,240	80,166	367
		Oak Park	501,240	80,166	10
		Orland Park	501,240	80,166	340
		Oswego	501,240	80,166	322
		Palatine	501,240	80,166	116
		Park Ridge	501,240	80,166	685
		Peoria	501,240	80,166	501

Figure 5-85. *Measure "Sales" placed on the view*

The final output as shown in Fig. 5-86.

Region	State	City	Fixed_Region_S..	Fixed_State_Sales	City_Sales
Central	Illinois	Arlington Heights	501,240	80,166	14
		Aurora	501,240	80,166	7,573
		Bloomington	501,240	80,166	964
		Bolingbrook	501,240	80,166	218
		Buffalo Grove	501,240	80,166	831
		Carol Stream	501,240	80,166	1,306
		Champaign	501,240	80,166	152
		Chicago	501,240	80,166	48,540
		Danville	501,240	80,166	43
		Decatur	501,240	80,166	3,169
		Des Plaines	501,240	80,166	1,493
		Elmhurst	501,240	80,166	892
		Evanston	501,240	80,166	1,754
		Frankfort	501,240	80,166	98
		Freeport	501,240	80,166	216
		Glenview	501,240	80,166	158
		Highland Park	501,240	80,166	2,035
		Naperville	501,240	80,166	1,288
		Normal	501,240	80,166	367
		Oak Park	501,240	80,166	10
		Orland Park	501,240	80,166	340
		Oswego	501,240	80,166	322
		Palatine	501,240	80,166	116
		Park Ridge	501,240	80,166	685
		Peoria	501,240	80,166	501

Figure 5-86. *"LOD – Fixed" – Demo 2 – final output*

5.6.2.2 Demo 3

Objective: To demonstrate the "level of detail – include".

Input: "LOD.xls"

The dataset as available in "LOD.xls".

	A	B	C
1	TransactionID	CustomerName	Amount
2	1	Alex Maxwell	1000
3	2	Alex Maxwell	250
4	3	Barbara Mori	1200
5	4	Barbara Mori	300
6	5	Barbara Mori	450
7	6	Ileana D'Souza	350
8	7	Ileana D'Souza	450
9	8	Esha Mathews	600
10	9	Esha Mathews	600
11	10	John Tukey	650
12	11	Kelly M	700
13	12	George T	800

Expected Output: Shown in Fig. 5-87**.**

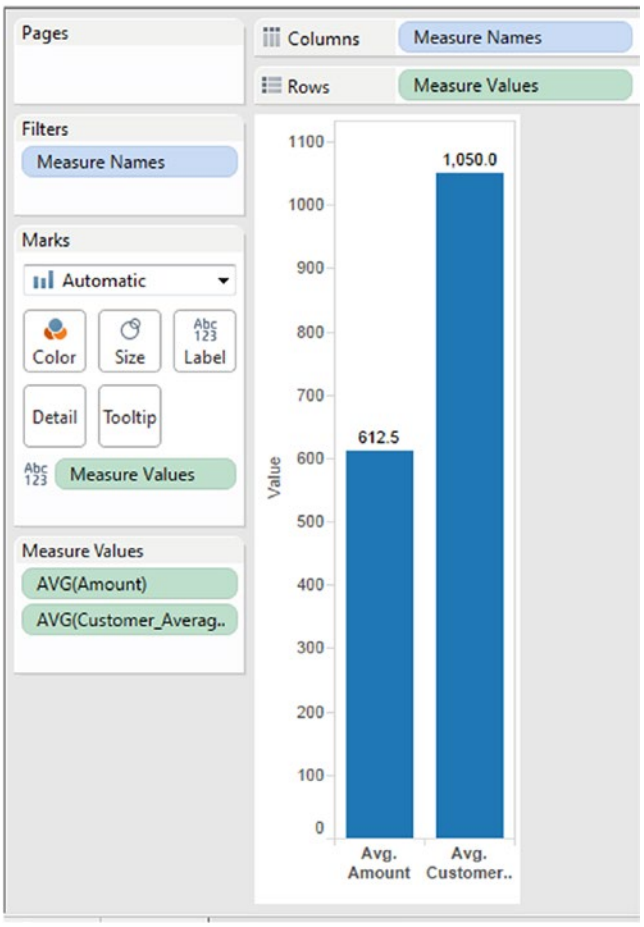


Figure 5-87. *"LOD – Include" – Demo 3 - expected output*

Explanation of the expected output:

We would like to view the "Average Transaction Amount" of all customers. Alongside this, we would like to view the "Average Transaction Amount per Customer".

To get the "Average Transaction Amount", simply sum up the transaction amounts of all customers and then divide by the number of customers (Shown in Table 5-6).

Table 5-6. *Average Transaction Amount*

	A	B	C
1	**TransactionID**	**CustomerName**	**Amount**
2	1	Alex Maxwell	1000
3	2	Alex Maxwell	250
4	3	Barbara Mori	1200
5	4	Barbara Mori	300
6	5	Barbara Mori	450
7	6	Ileana D'Souza	350
8	7	Ileana D'Souza	450
9	8	Esha Mathews	600
10	9	Esha Mathews	600
11	10	John Tukey	650
12	11	Kelly M	700
13	12	George T	800
14		**Grand Total**	**7350**
15		**Average**	**612.5**

To get the "Average Transaction Amount per Customer", aggregate the transaction amount for each customer and then sum up the aggregated amount for all customers. Finally divide the aggregated amount for all customers by the number of unique customers (Shown in Table 5-7).

Table 5-7. *Average Transaction Amount per Customer*

TransactionID	CustomerName	Amount	Transaction amount per Customer
1	Alex Maxwell	1000	1250
2	Alex Maxwell	250	
3	Barbara Mori	1200	1950
4	Barbara Mori	300	
5	Barbara Mori	450	
6	Ileana D'Souza	350	800
7	Ileana D'Souza	450	
8	Esha Mathews	600	1200
9	Esha Mathews	600	
10	John Tukey	650	650
11	Kelly M	700	700
12	George T	800	800
		Grand Total	**7350**
		Average (Grand Total divided by 7)	**1050**

Let us look at the steps to accomplish the same in Tableau.

5.6.2.3 Steps

Follow the steps as provided.

5.6.2.3.1 Step 1

Read in the data from “LOD.xls” into Tableau (Shown in Fig. 5-88).

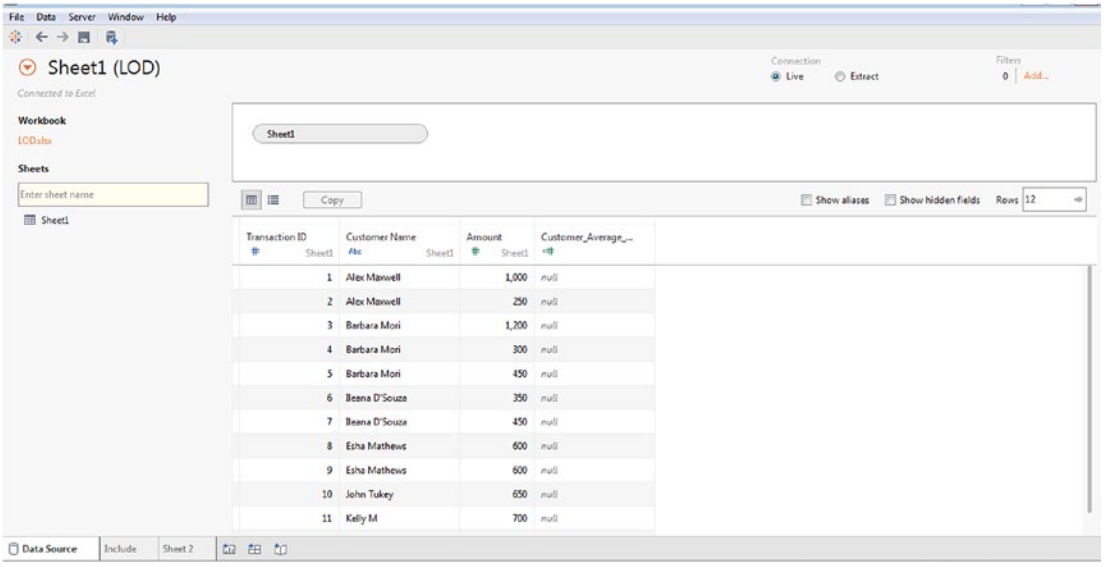


Figure 5-88. *Read in data from "LOD.xls" into Tableau*

5.6.2.3.2 Step 2

Create a calculated field, "Customer_Average_Amount" (Shown in Fig. 5-89).

Figure 5-89. *Calculated field "Customer_Average_Amount" being created*

5.6.2.3.3 Step 3

Drag the **dimension** "Measure Names" from the dimensions area under the data pane to the columns shelf (Shown in Fig. 5-90).

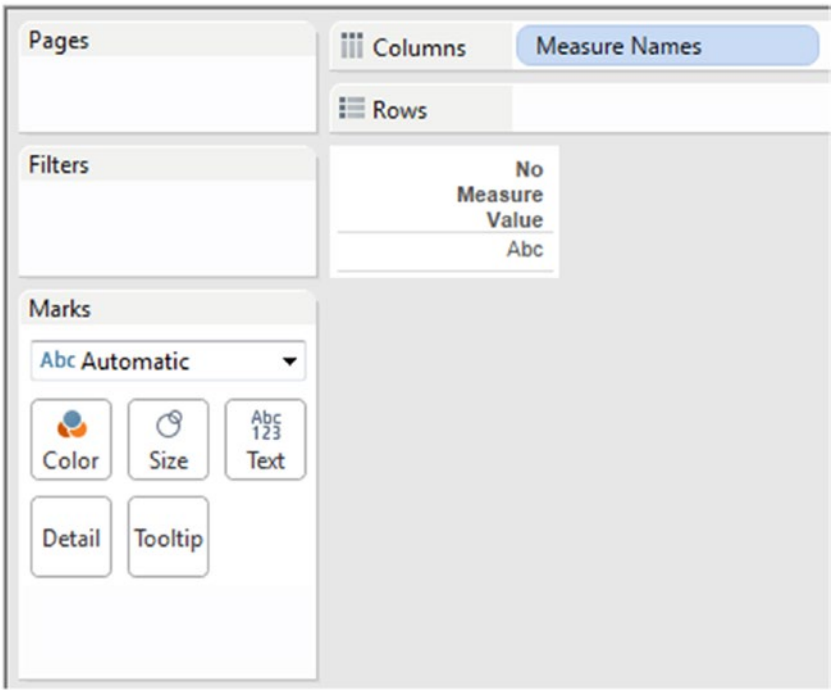


Figure 5-90. *Dimension "Measure Names" placed on the columns shelf*

5.6.2.3.4 Step 4

Drag the **dimension** "Measure Names" from the dimensions area under the data pane to Filters Shelf (Shown in Fig. 5-91).

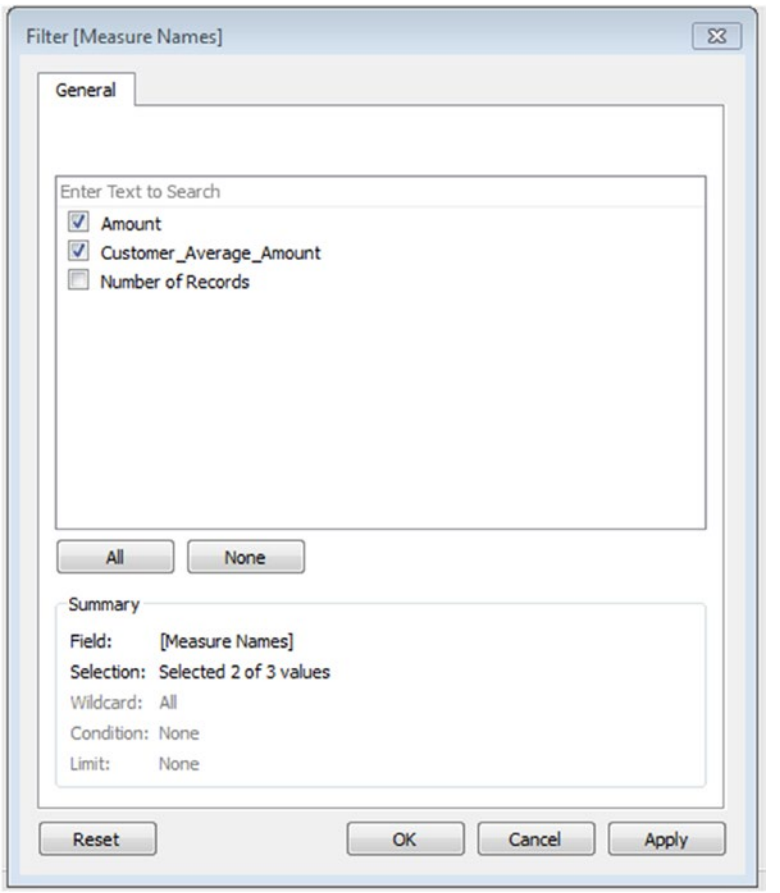


Figure 5-91. *Filter dialog box for "Measure Names"*

5.6.2.3.5 Step 5

Drag the **measure** "Measure Values" from the measures area under the data pane to the rows shelf (Shown in Fig. 5-92).

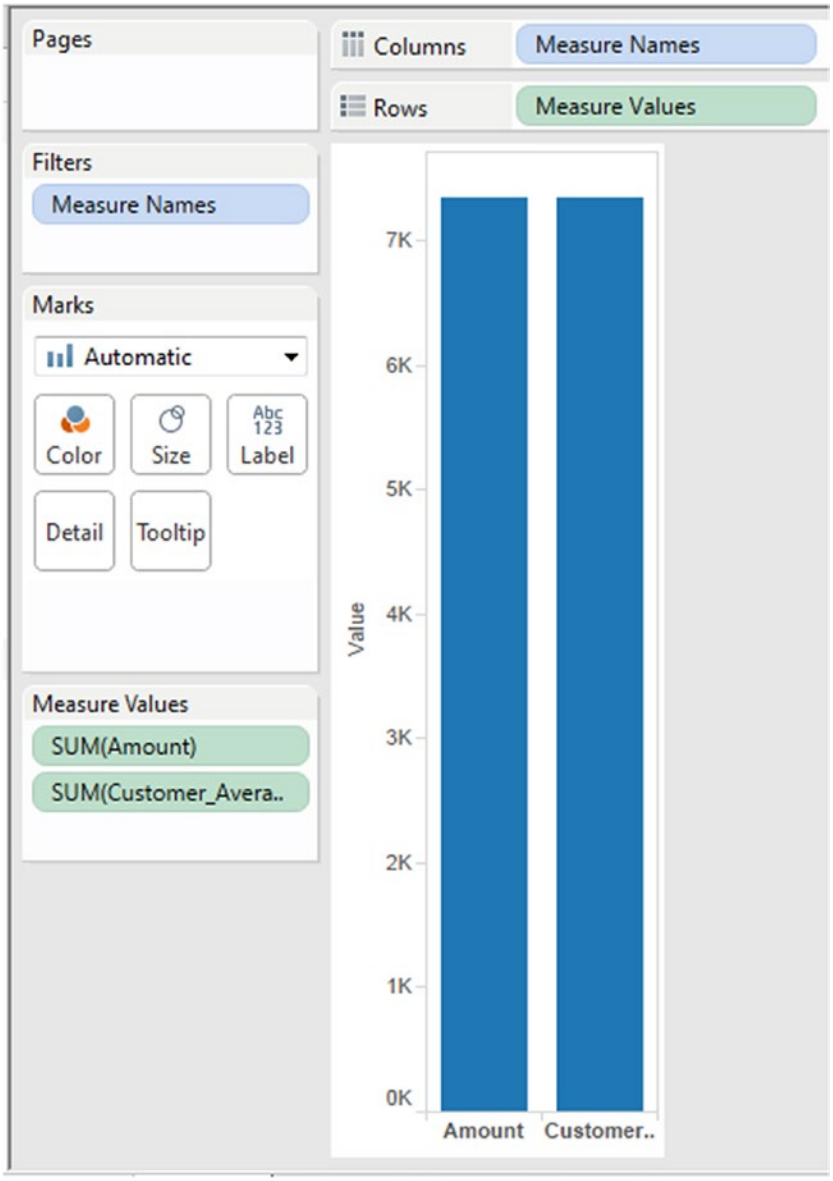


Figure 5-92. *Measure "Measure Values" placed on the rows shelf*

5.6.2.3.6 Step 6

Change the aggregation of both the measures, "Amount" and "Customer_Average_Amount" to "Average" (Shown in Fig. 5-93 & Figure 5-94).

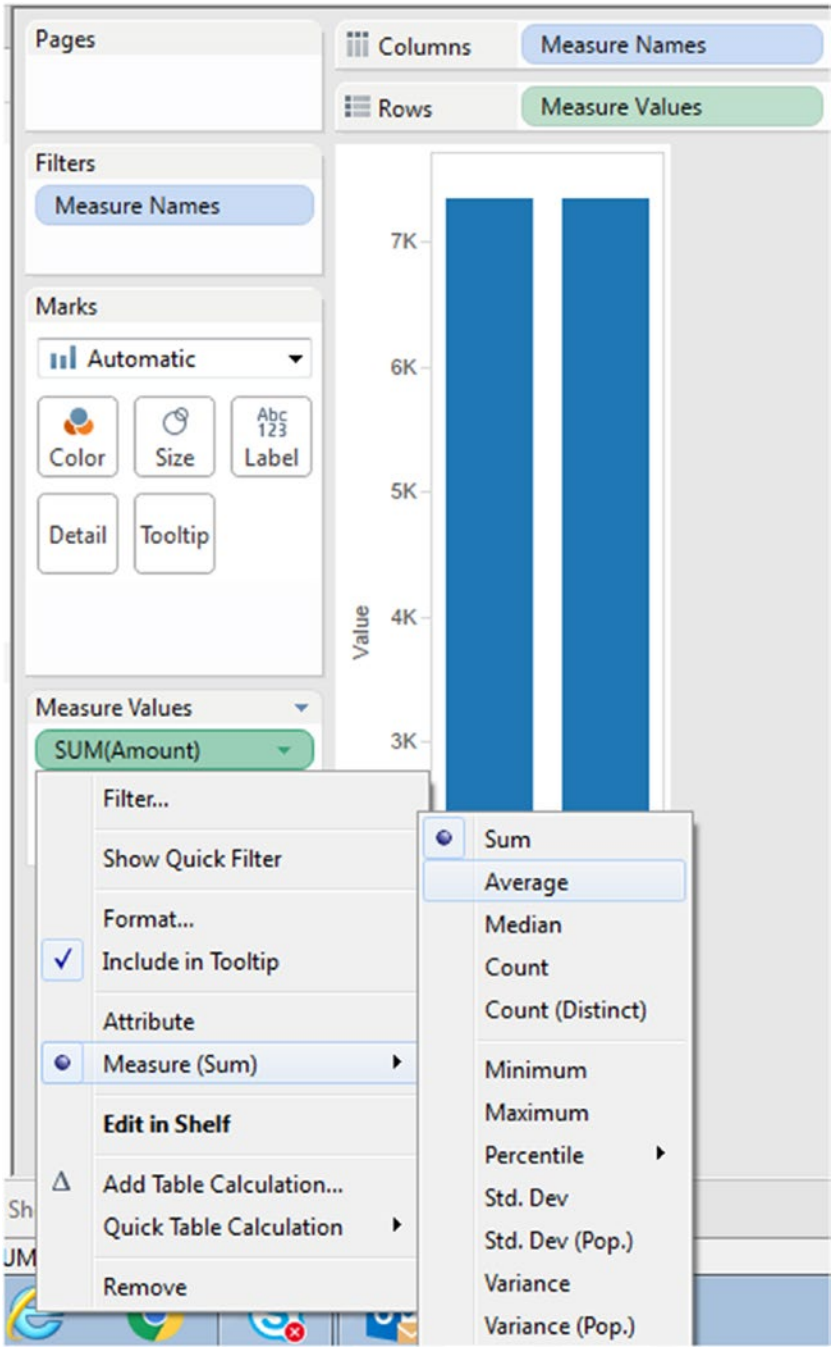


Figure 5-93. *Aggregation for measure "Amount" set to "Average"*

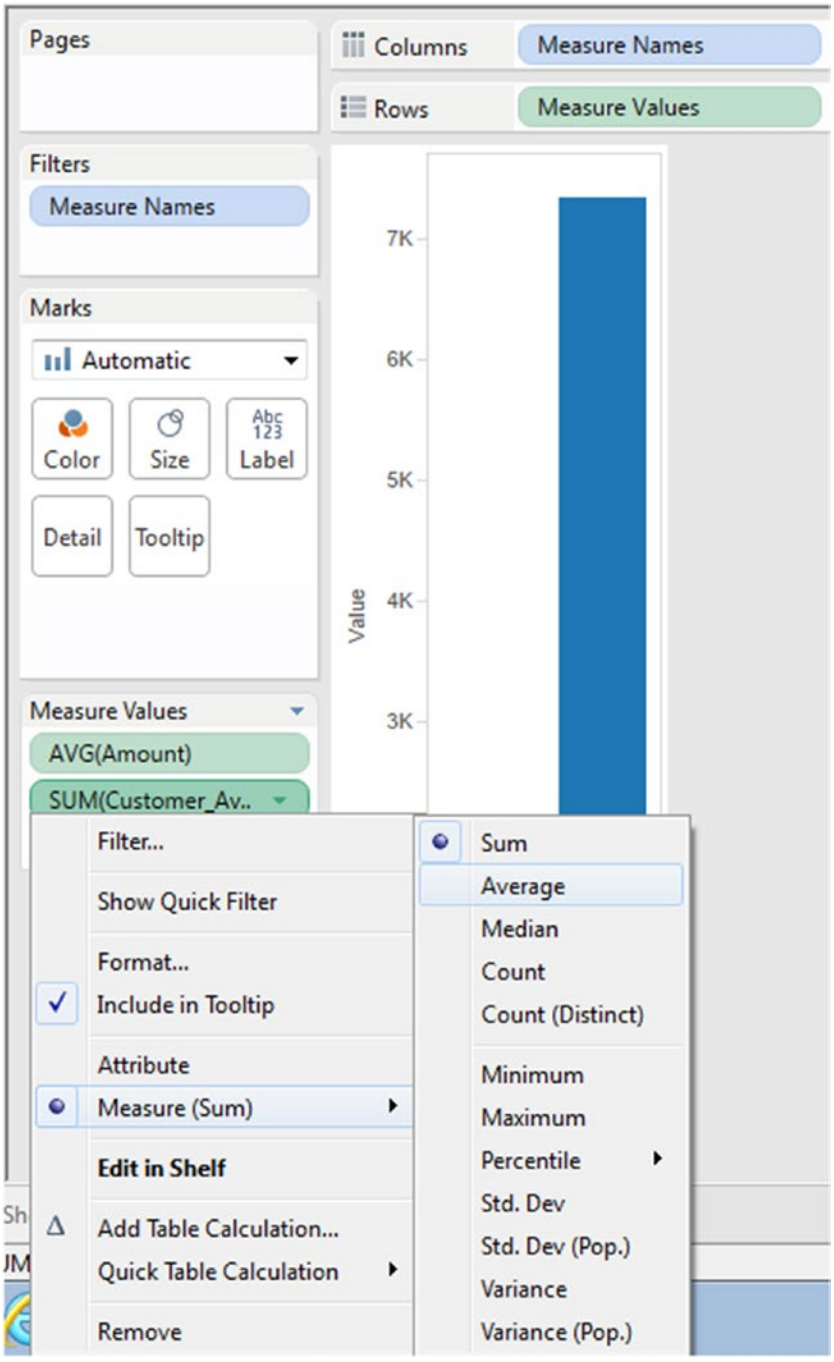


Figure 5-94. *Aggregation for measure "Customer_Average_Amount" set to "Average"*

The output after changing the aggregation of the measures to “Average” (Shown in Fig. 5-95).

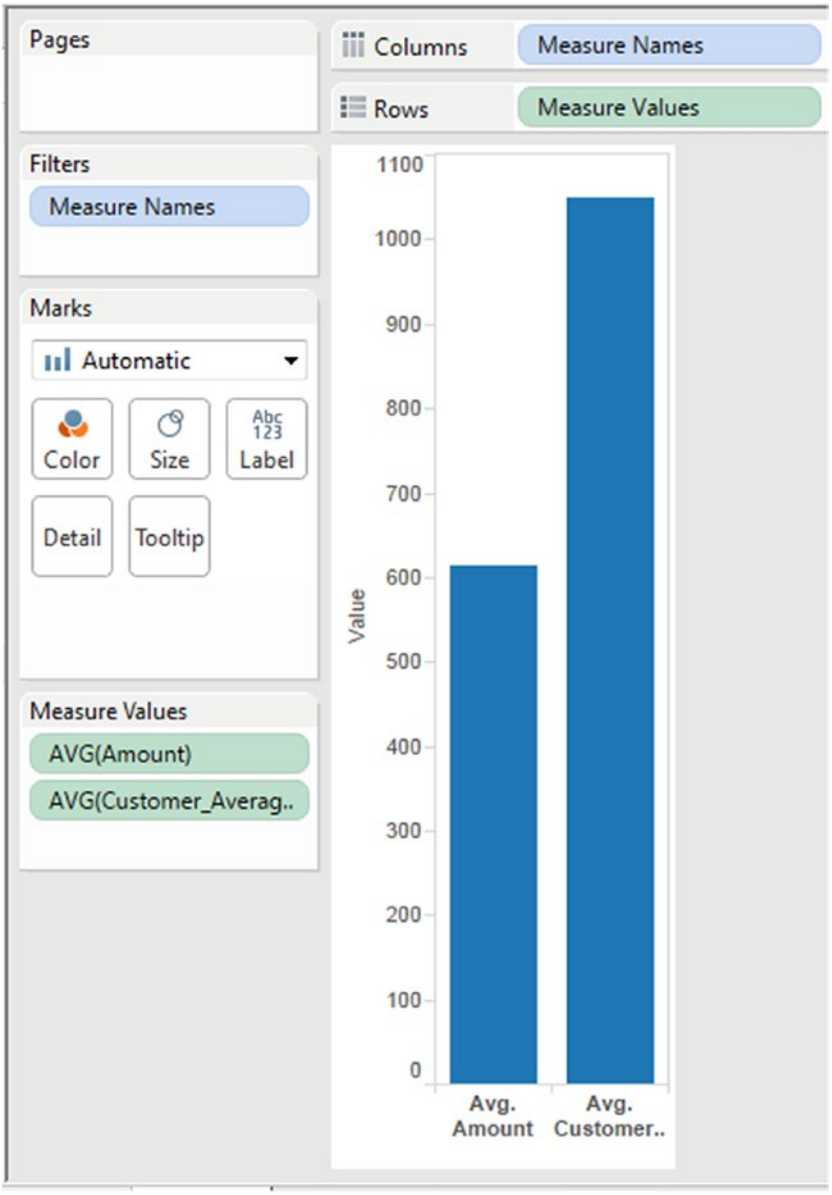


Figure 5-95. *Output after the measure's aggregation is set to “Average”*

5.6.2.3.7 Step 7

Press the CTRL key and drag the "Measure Values" from the rows shelf to "Label" on the marks card (Shown in Fig. 5-96).

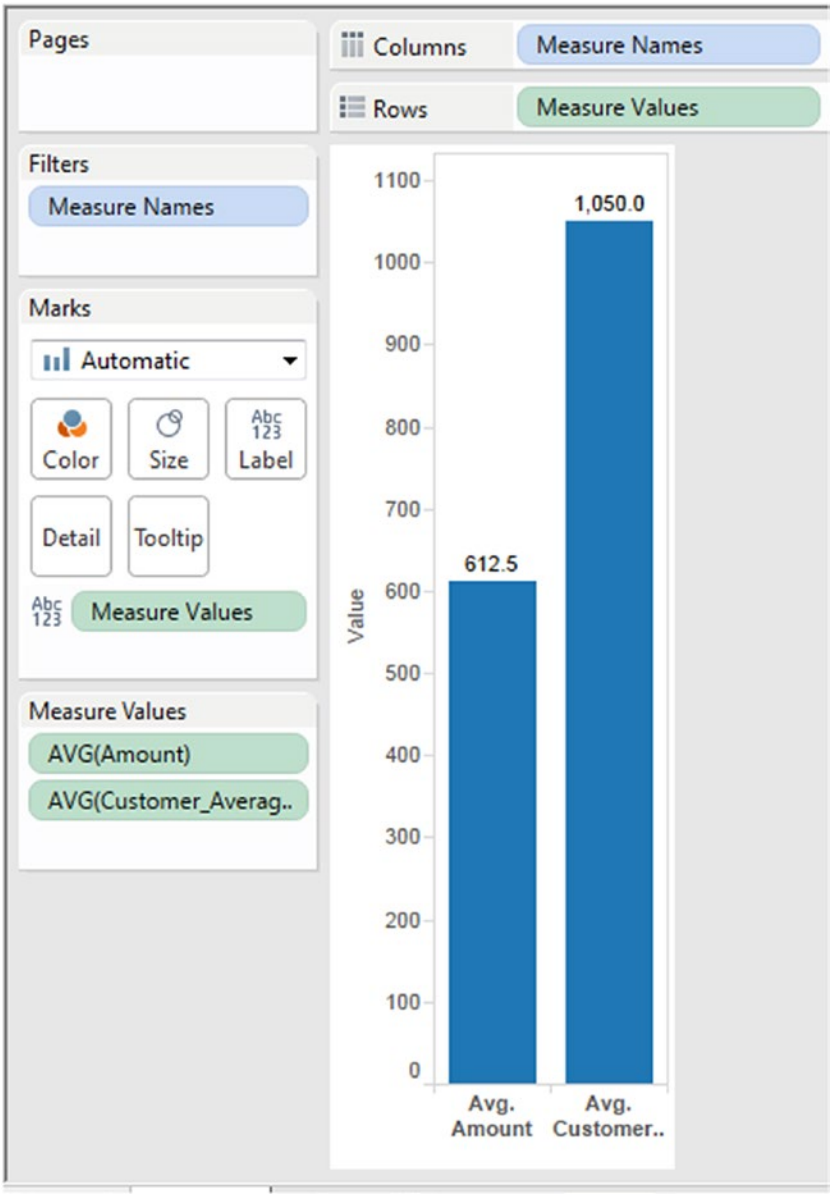


Figure 5-96. *"Measure Values" placed on "Label" on the marks card*

Just to cross-verify the "Average Sales Amount per Customer", let us perform the below steps:

Drag the **dimension** "Customer Name" from the dimensions area under the data pane and place it on the rows shelf (Shown in Fig. 5-97).

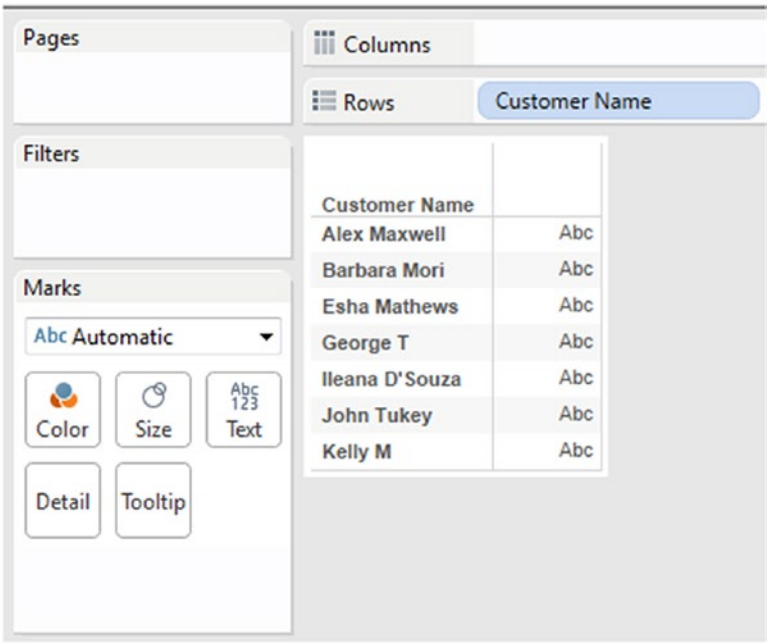


Figure 5-97. *Dimension "Customer Name" placed on the rows shelf*

Drag the **measure** "Amount" from the measures area under the data pane and place it on the **"Label"** on the marks card (Shown in Fig. 5-98).

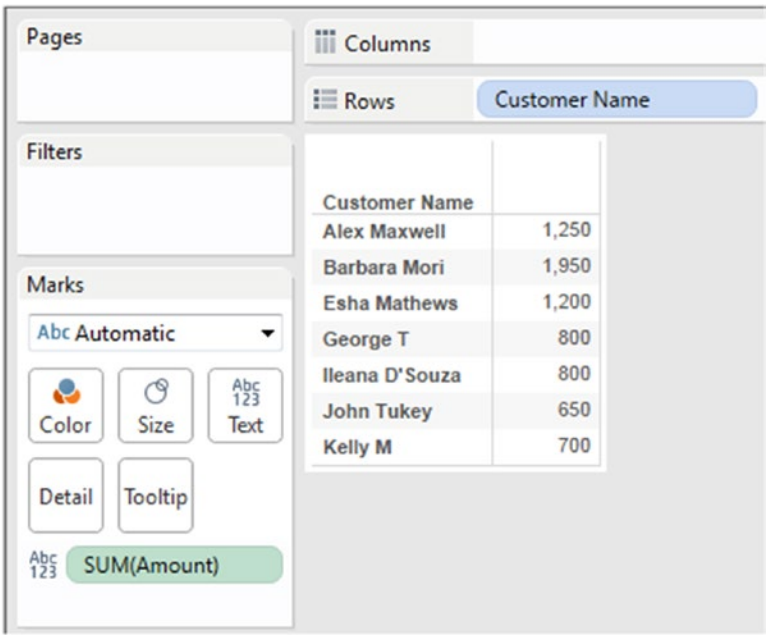


Figure 5-98. *Measure "Amount" placed on "Label" on the marks card*

Select "Analysis" on the menu bar. Select "Totals" and then select "Show Column Grand Totals".
Analysis ➤ Totals ➤ Show Column Grand Totals.
Refer to Fig. 5-99.

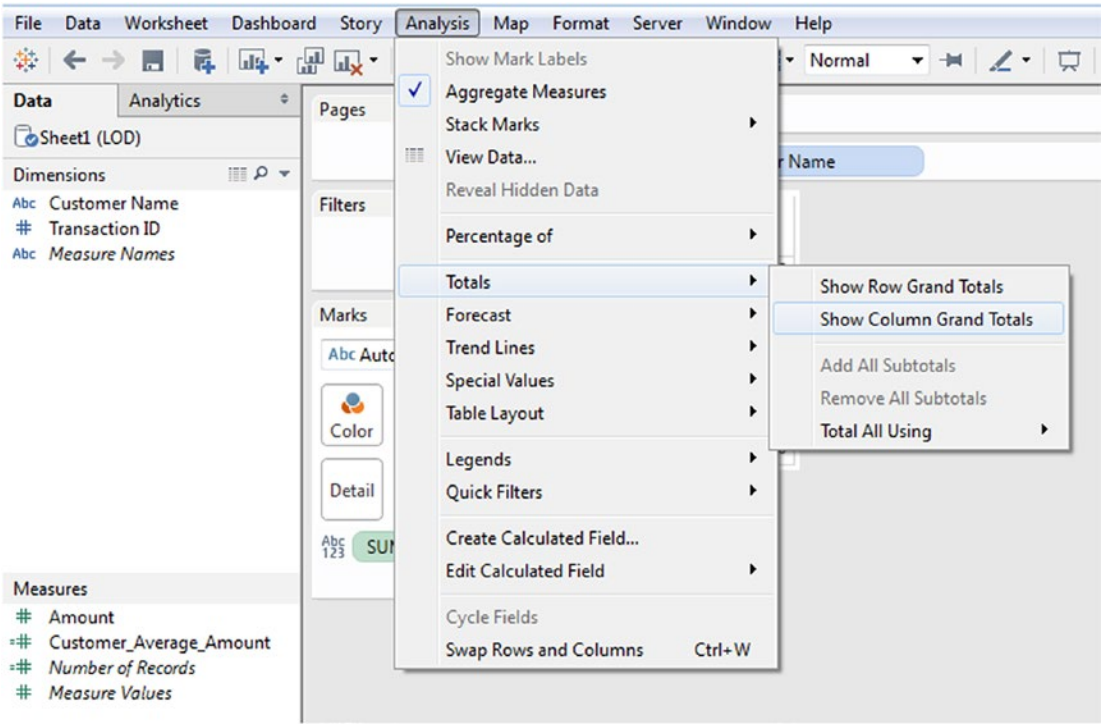


Figure 5-99. *"Show Column Grand Totals" for the measure on the view*

The output is shown in Fig. 5-100.

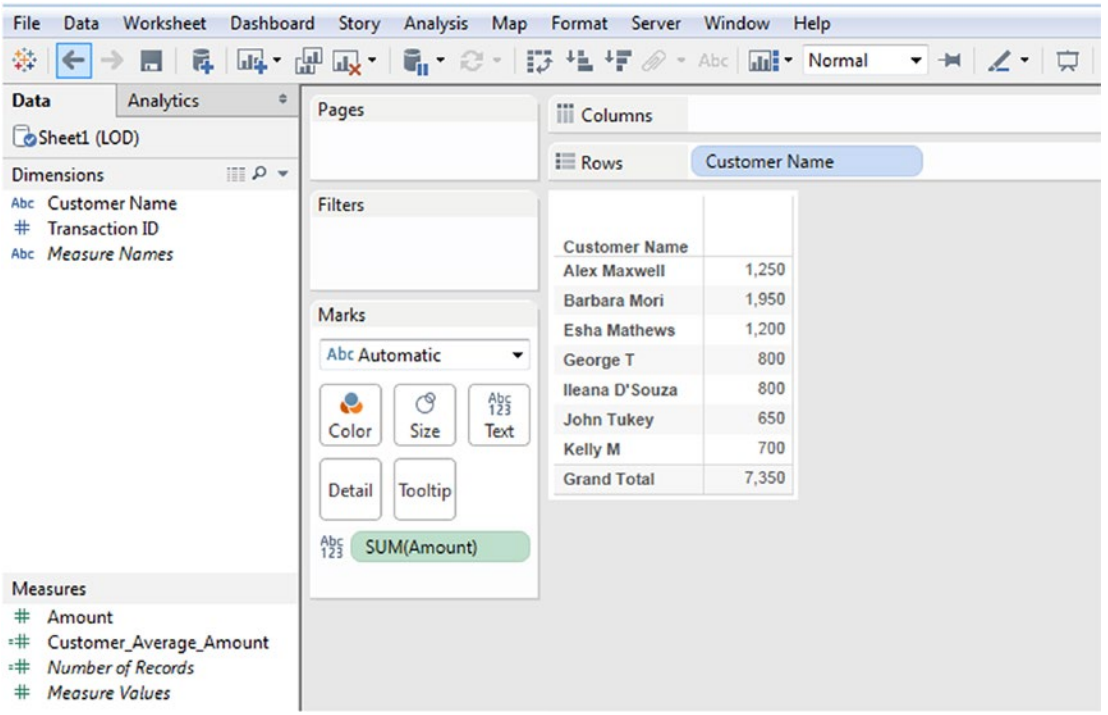


Figure 5-100. *Output with "Column Grand Totals"*

Use "Average" aggregation for grand total (Shown in Fig. 5-101).

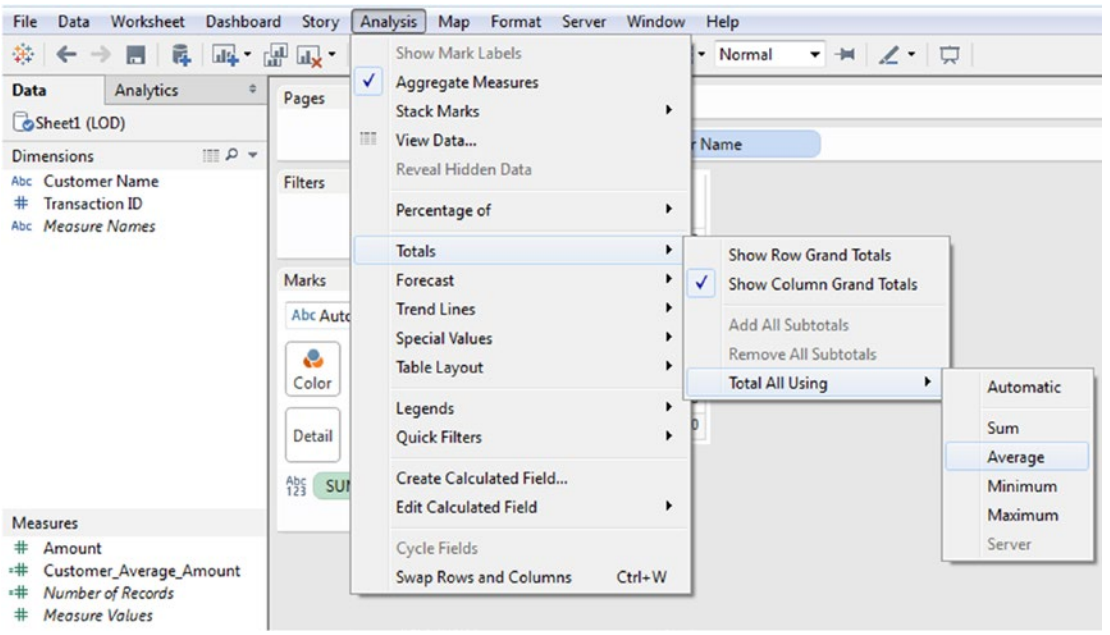


Figure 5-101. *"Total All Using - Average" for measure on the view*

The output after the aggregation for grand total was changed to “Average” (Shown in Fig. 5-102).

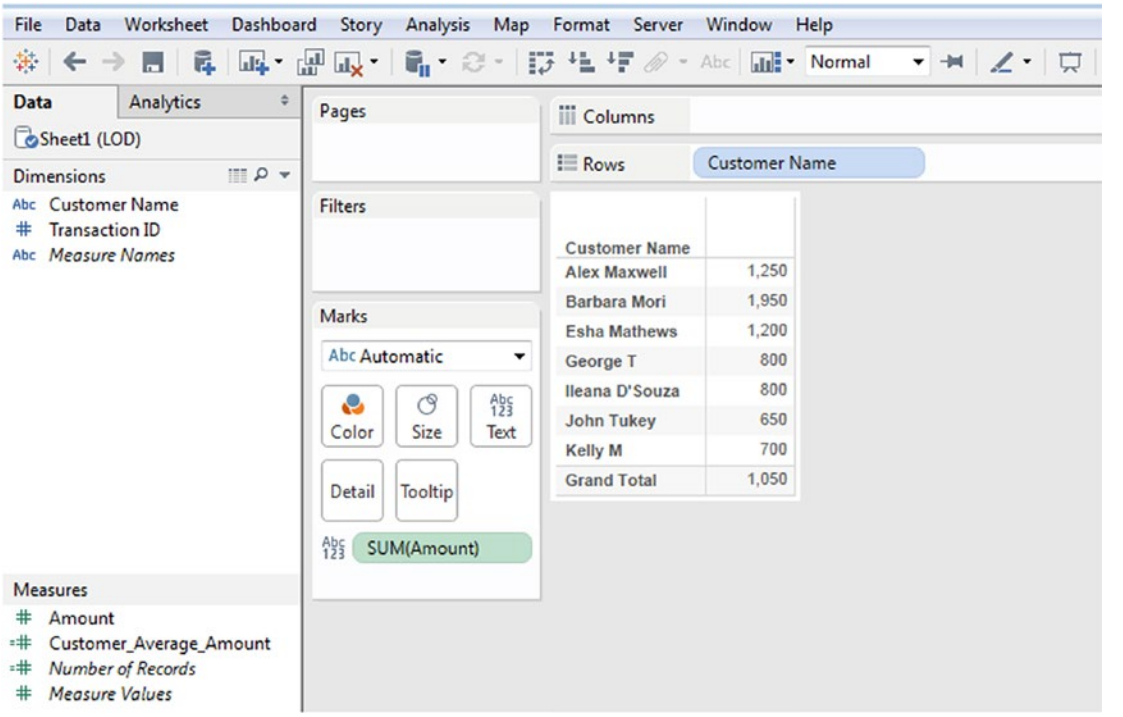


Figure 5-102. *Output with “Column Grand Total - Total All Using - Average”*

This can be achieved in another two ways:
WINDOW_AVG(Sum(Amount))
Or
AVG({Fixed [Customer Name] : Sum(Amount)})
The value of “1,050” matches with our value computed for “Average Sales Amount per Customer”.

5.7 Percentile

Percentiles divide the data set into 100 equal parts. Percentiles measure position from the bottom. They are used to determine the relative standing of an individual in a population. In other words, they provide the rank position of an individual. Where have we seen percentiles being used? It is usually used with test scores and graduation standings. Graduation standings refers to the individual’s standing at graduation relative to other graduate students.

Definition: Percentile is a measure used to determine the percentage of total frequency scored below that measure. Percentile rank is percentage of scores that fall below a given score.

Formula: To determine the percentile rank of a score, x, out of a total of n scores, the formula is
Percentile Rank = ((Number of scores below x) / n) * 100

Example: In a class of 200 students, Mason scored 25th rank. His percentile standing in the class is:
(175 / 200) * 100 = 87.5%
At 87.5%, his scores are better than 88% of the class.

5.7.1 Demo 1

Objective: To compute the percentile for students of VIII grade.

Input: "Percentile.xlsx"

The sample data set as available in "Percentile.xlsx".

	A	B
1	RollNo	CGPA
2	1	4.6
3	2	4.2
4	3	4.4
5	4	4.3
6	5	3.9
7	6	5
8	7	4.3
9	8	4.4
10	9	4.6
11	10	4.7

Expected output: Shown in Fig. 5-103.

Roll No	CGPA	Percentile
1	4.600	80.00%
2	4.200	20.00%
3	4.400	60.00%
4	4.300	40.00%
5	3.900	10.00%
6	5.000	100.00%
7	4.300	40.00%
8	4.400	60.00%
9	4.600	80.00%
10	4.700	90.00%

Figure 5-103. *Percentile – Demo 1 – expected output*

5.7.1.1 Steps

Follow the steps as provided.

5.7.1.1.1 Step 1

Read in the data from “Percentile.xlsx” into Tableau (Shown in Fig. 5-104).

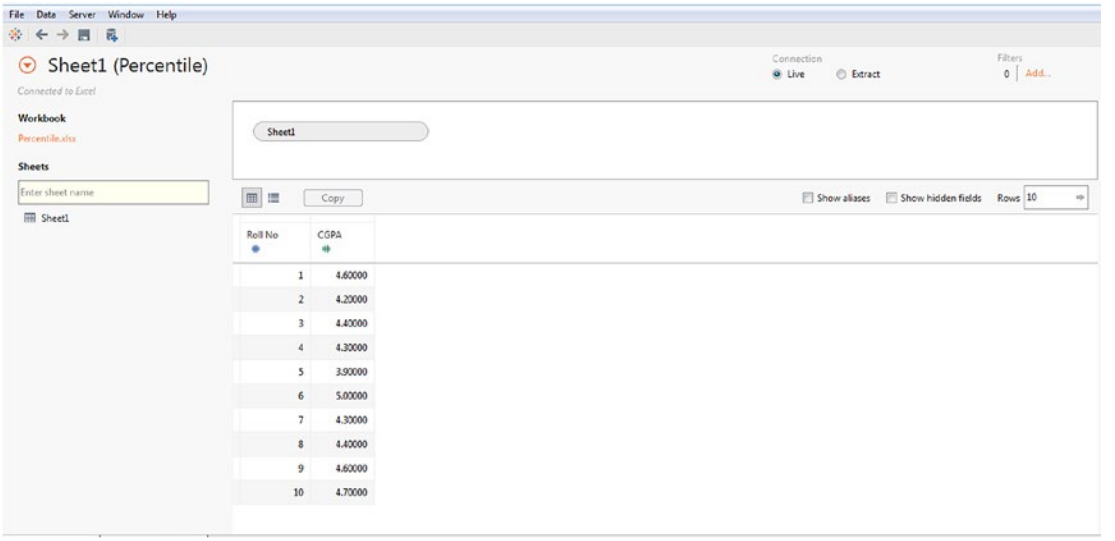


Figure 5-104. *Data from “Percentile.xls” read into Tableau*

5.7.1.1.2 Step 2

Drag the **dimension** “Roll No” from the dimensions area under the data pane and place it on the rows shelf (Shown in Fig. 5-105).

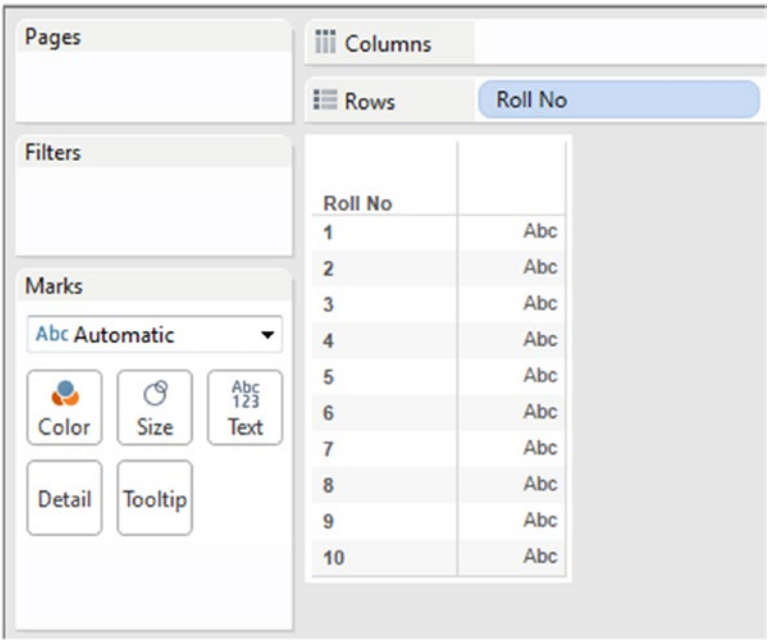


Figure 5-105. *Dimension "Roll No" placed on the rows shelf*

5.7.1.1.3 Step 3

Drag the **measure** "CGPA" from the measures area under the data pane and place it on the **"Label"** on the marks card (Shown in Fig. 5-106).

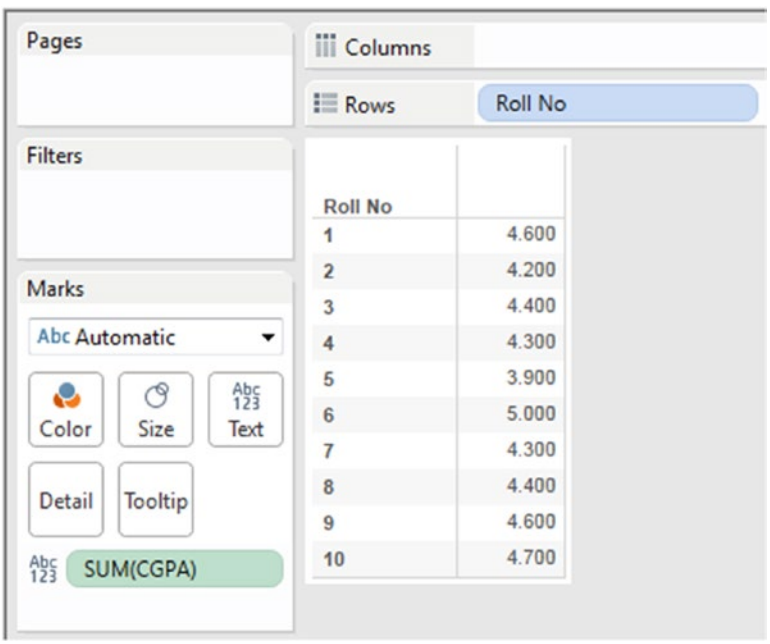


Figure 5-106. *Measure "CGPA" placed on "Label" on the marks card*

5.7.1.1.4 Step 4

Add a table calculation “Percentile” to the measure “CGPA” placed on the “Label” on the marks card (Shown in Fig. 5-107).

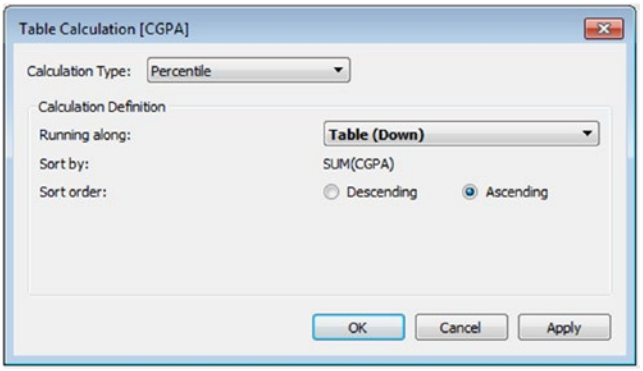


Figure 5-107. *“Table Calculation - Percentile” being applied to measure “CGPA”*

The output after adding the “Table Calculation – Percentile” to the measure “CGPA” (Shown in Fig. 5-108).

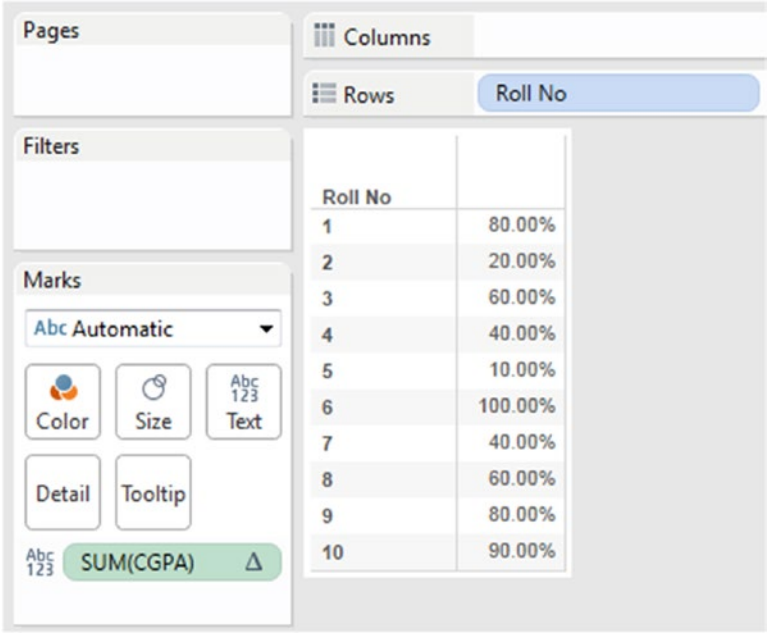


Figure 5-108. *Output after applying “Table Calculation - Percentile” to measure “CGPA”*

5.7.1.1.5 Step 5

Drag the measure "CGPA" from the measures area under the data pane and place it in the text area (Shown in Fig. 5-109).

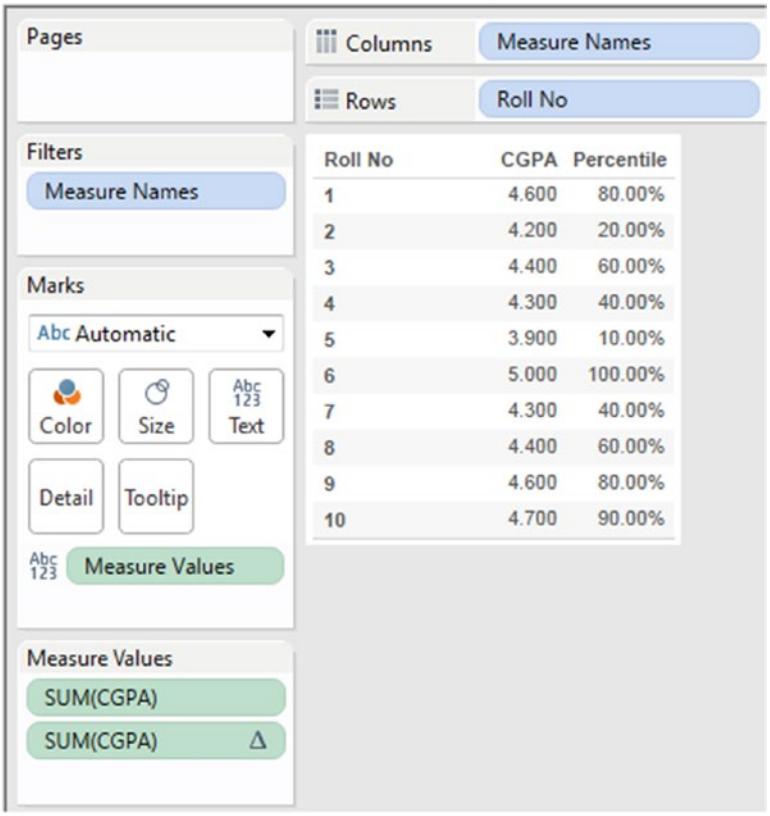


Figure 5-109. *Measure "CGPA" placed in the view*

The formula used in the calculation:

(Number of values less than or equal to the value under consideration / Total number of values) * 100

Example: Let us consider the CGPA score for student whose roll number is 1. The Student's CGPA score is 4.6. There are eight CGPA scores that are less than or equal to the CGPA score of 4.6.

(8 / 10) * 100 = 80%

Now, just to reconfirm the formula, let us consider another student's CGPA value. This time let us consider the CGPA value of 3.9.

This is the least CGPA score that a student has attained.

(1/10)*100 = 10%

■ **Note** While computing percentiles, Tableau ignores null values. Null values if present appear as blank rows in a cross-tab and do not count towards the total number of items used in the calculation (%).

5.8 Year over Year Growth

In layman's terms, YOY means the company's financial performance this year as against last year. YOY performance is used to gauge whether the performance of the company is improving or debilitating.

Formula to compute the Year over year growth:

- Subtract last year's number (sales or profit) from this year's number. This will constitute the total difference for the year (this number if positive will indicate a year-over-year gain otherwise it implies loss).
- Divide the difference by last year's number. The result is the year-over-year growth rate.

5.8.1 Demo 1

Objective:

Data is provided for 4 years (2011, 2012, 2013 and 2014). The senior executive at the firm would like a visualization that shows the Year over Year growth.

Input: "Sample – Superstore.xls"

Expected Output: Shown in Fig. 5-110**.**

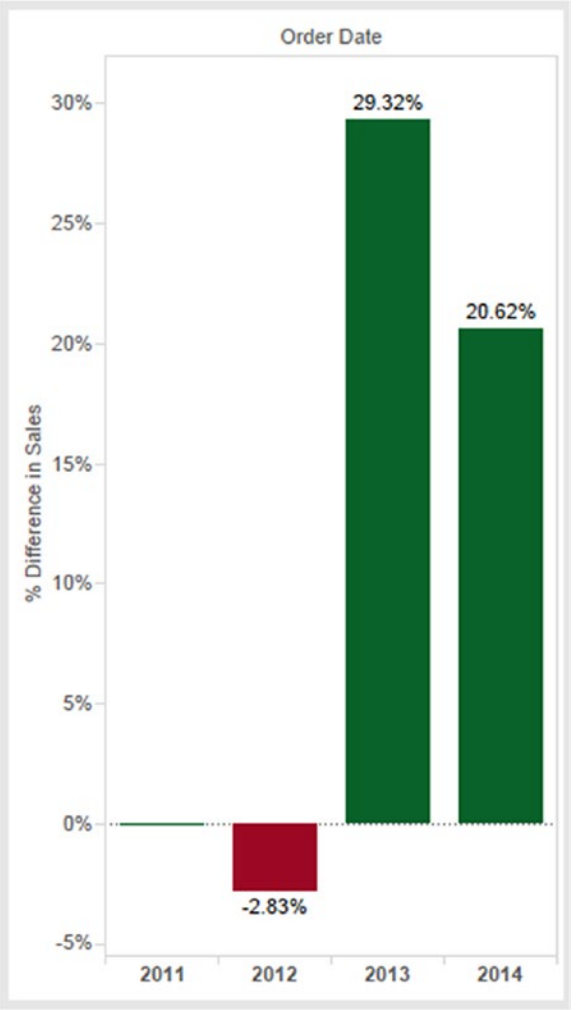


Figure 5-110. *Year Over Year Growth – Demo 1 – expected output*

5.8.1.1 Steps

Follow the steps as provided.

5.8.1.1.1 Step 1

Read in data from "Sample – Superstore.xls" into Tableau (Shown in Fig. 5-111).

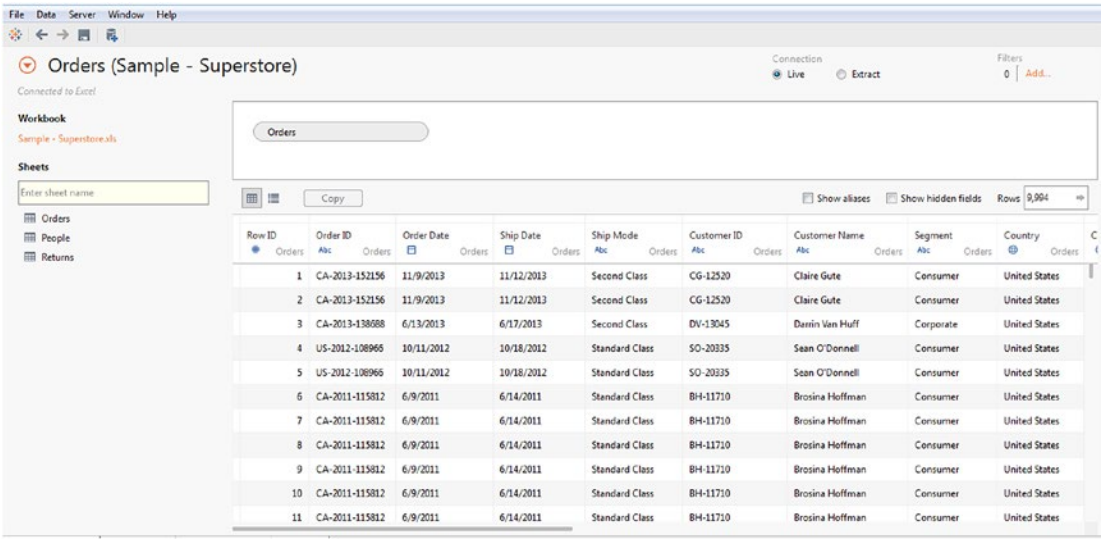


Figure 5-111. *Data from "Sample - Superstore.xls" read into Tableau*

5.8.1.1.2 Step 2

Drag the **dimension** "Order Date" from the dimensions area under the data pane to the columns shelf. Retain the date hierarchy at the default, i.e. "Year". Retain the "Order Date" at "Discrete" (Shown in Fig. 5-112).

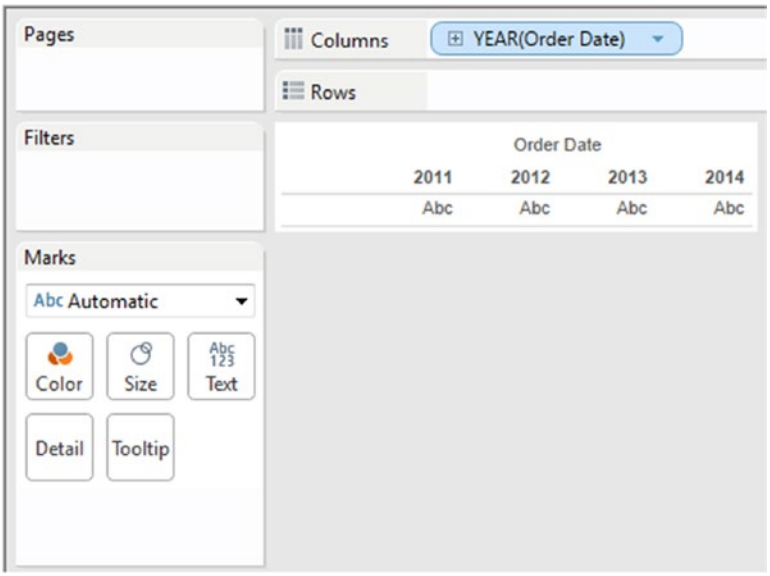


Figure 5-112. *Dimension "Order Date" placed on the columns shelf*

5.8.1.1.3 Step 3

Drag the measure “Sales” from the measures area under the data pane and place it on the rows shelf (Shown in Fig. 5-113).

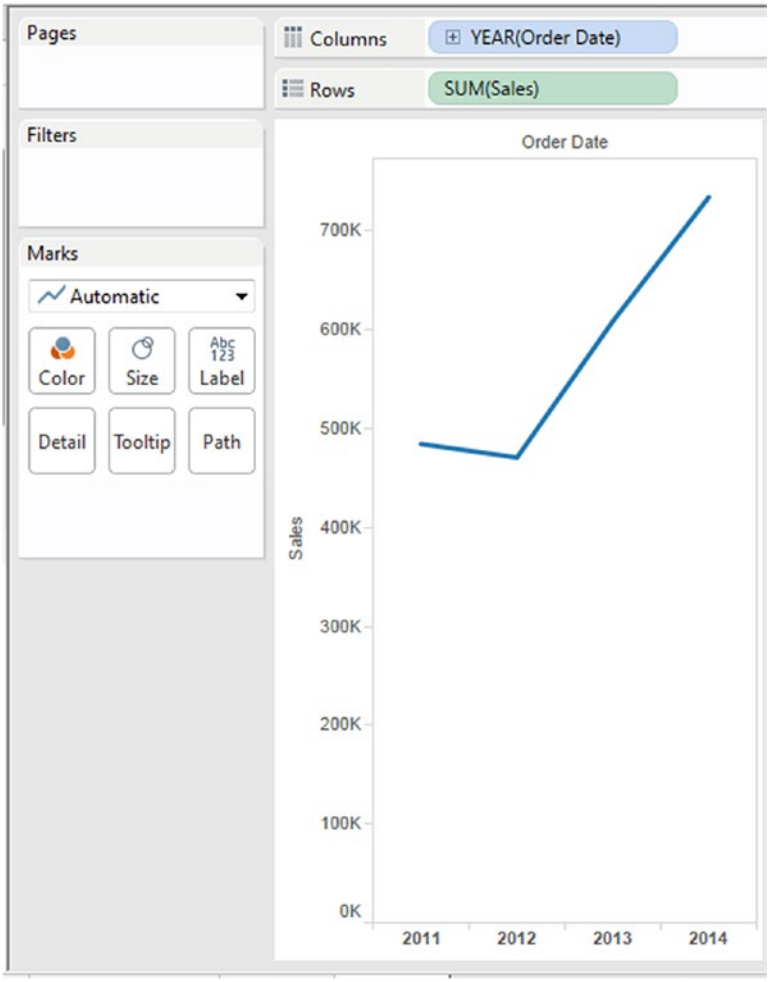


Figure 5-113. *Measure “Sales” placed on the rows shelf*

5.8.1.1.4 Step 4

Change the “Mark Type” to “Bar” (Shown in Fig. 5-114).

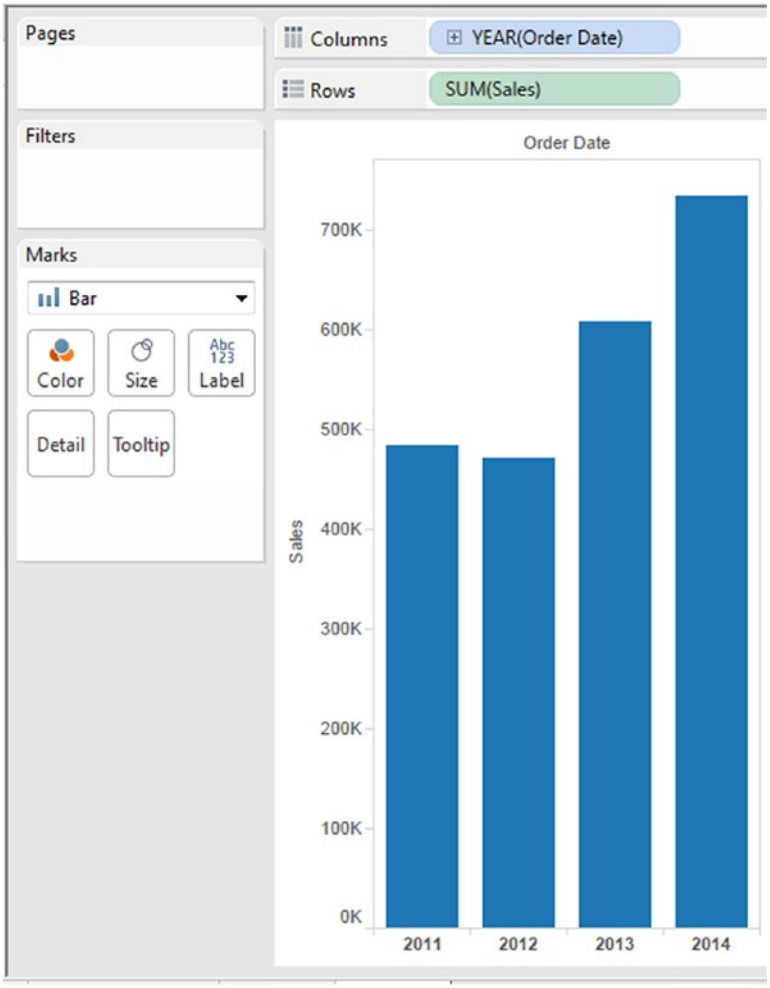


Figure 5-114. *Mark Type changed to "Bar"*

5.8.1.1.5 Step 5

Add a "Quick Table Calculation – Year over Year Growth" to the measure "Sales" on the rows shelf (Shown in Fig. 5-115 & Figure 5-116).

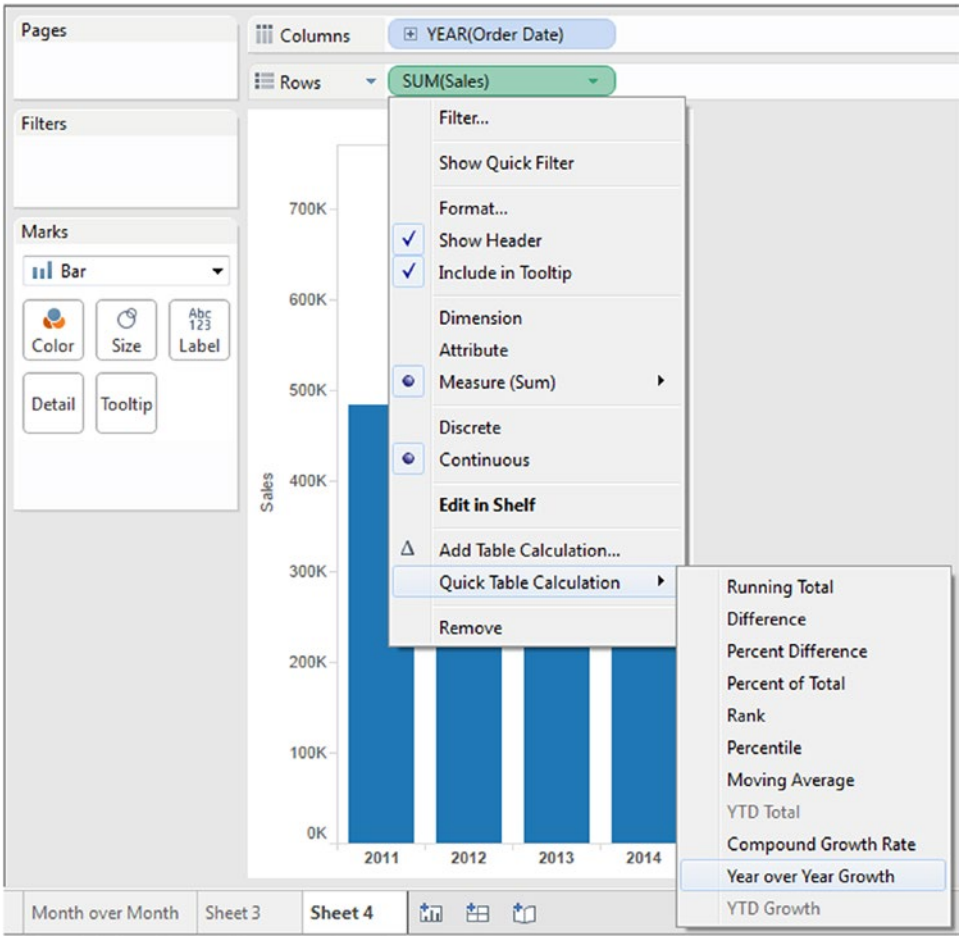


Figure 5-115. *"Table Calculation - Year Over Year Growth" being applied to measure "Sales"*

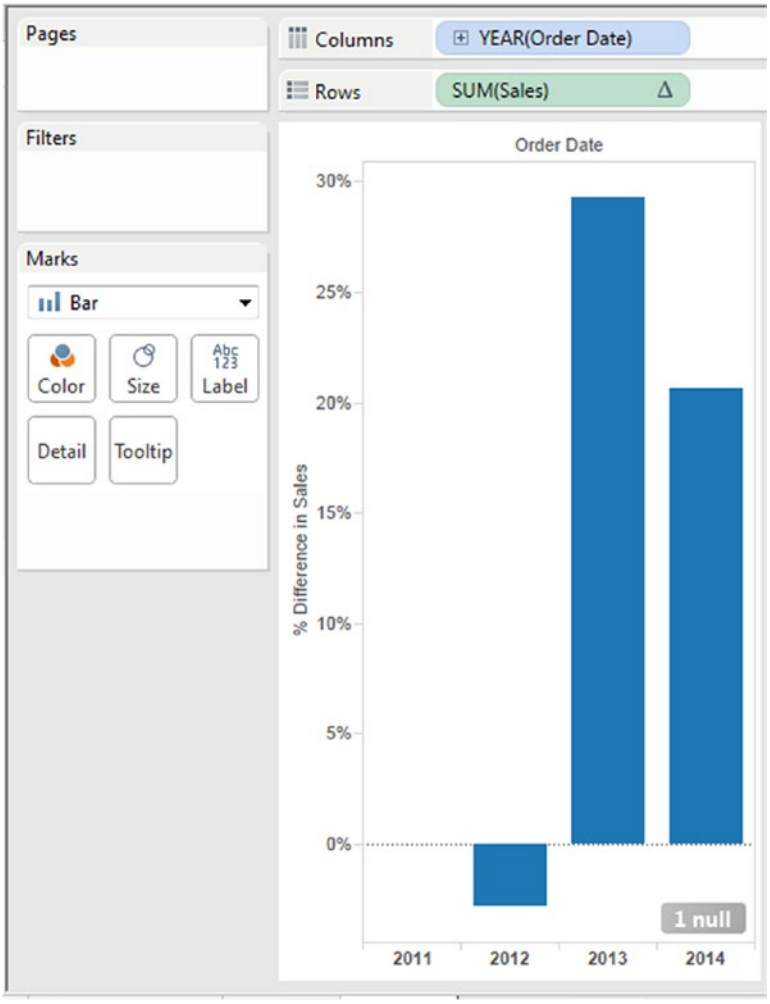


Figure 5-116. *The output after applying the "Table Calculation - Year over Year Growth"*

Click on the message, "1 null" at the bottom of the view/worksheet to bring up the "Special Values for [% Difference in Sales]" dialog box (Shown in Fig. 5-117).

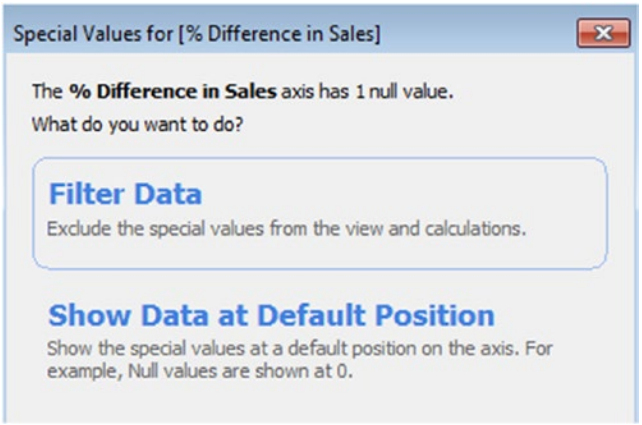


Figure 5-117. *"Special Values for [% Difference in Sales]" dialog box*

Click on "Show Data at Default Position". The output after considering the null value (Shown in Fig. 5-118).

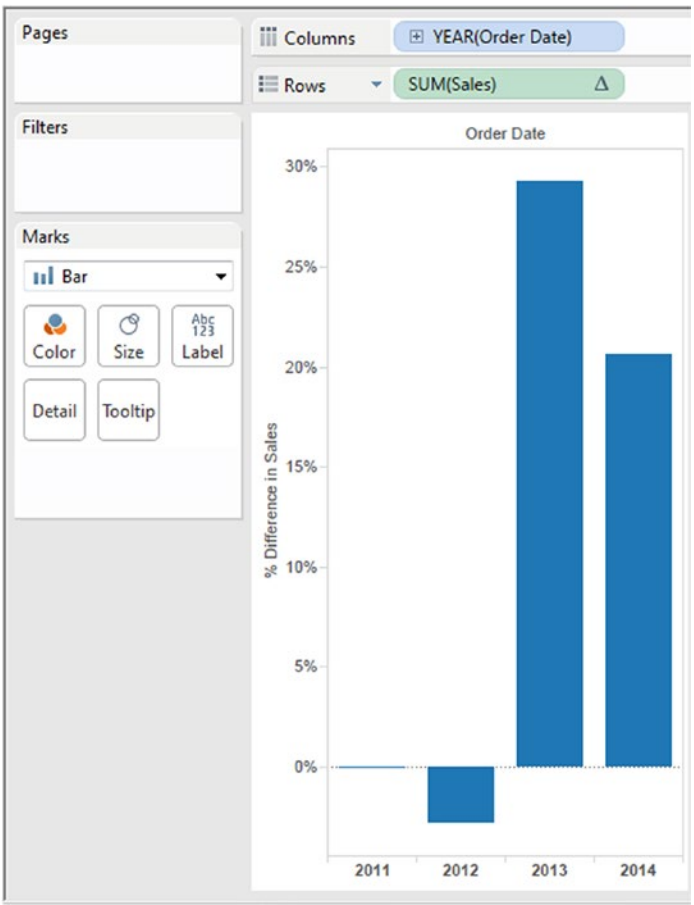


Figure 5-118. *The output after taking the null values into consideration*

A look at the “Table Calculation” dialog box. The “Year over Year Growth” is computed as a “Percent Difference” from “Previous” (Shown in Fig. 5-119).

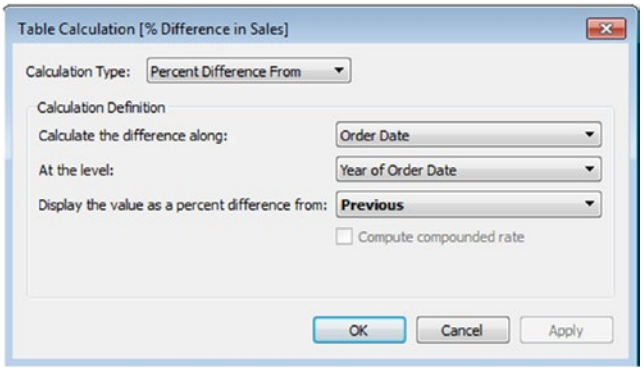


Figure 5-119. *“The “Year over Year Growth” is computed as a “Percent Difference” from “Previous”*

5.8.1.1.6 Step 6

Press “CTRL” and drag the measure “Sales” from the rows shelf and place it on “Color” on the marks card (Shown in Fig. 5-120).

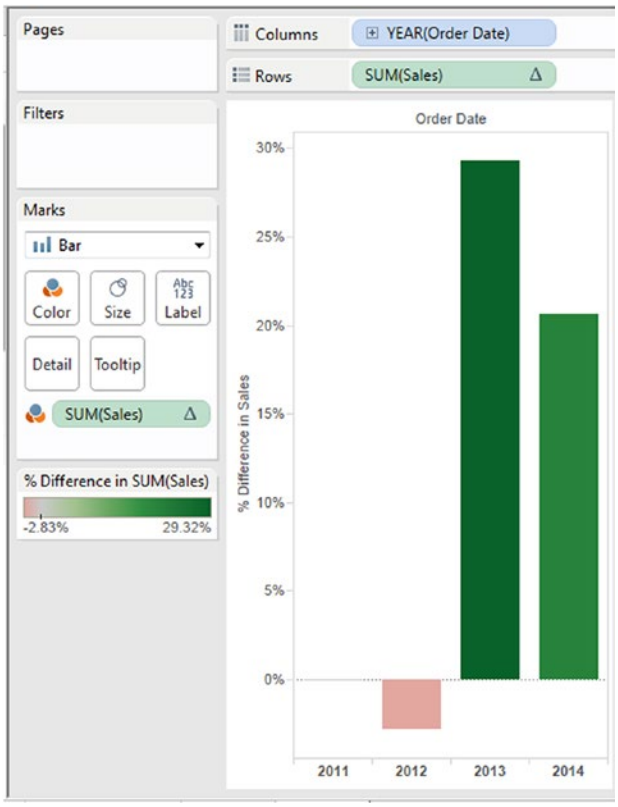


Figure 5-120. *Measure “Sales” placed on “Colors” on the marks card*

Change the stepped color to 2 (Shown in Fig. 5-121).

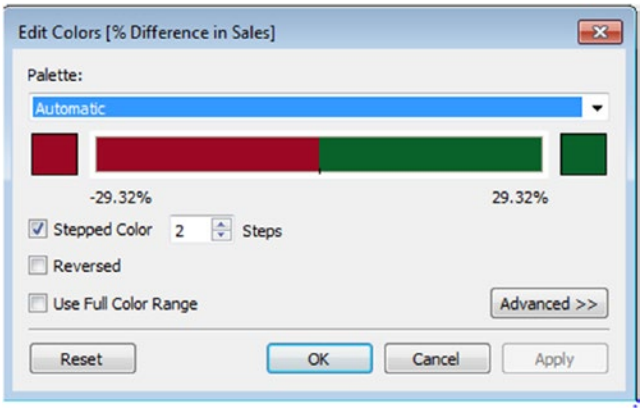


Figure 5-121. *"Stepped Color" changed to 2*

The output after changing the stepped color to 2 (Shown in Fig. 5-122).

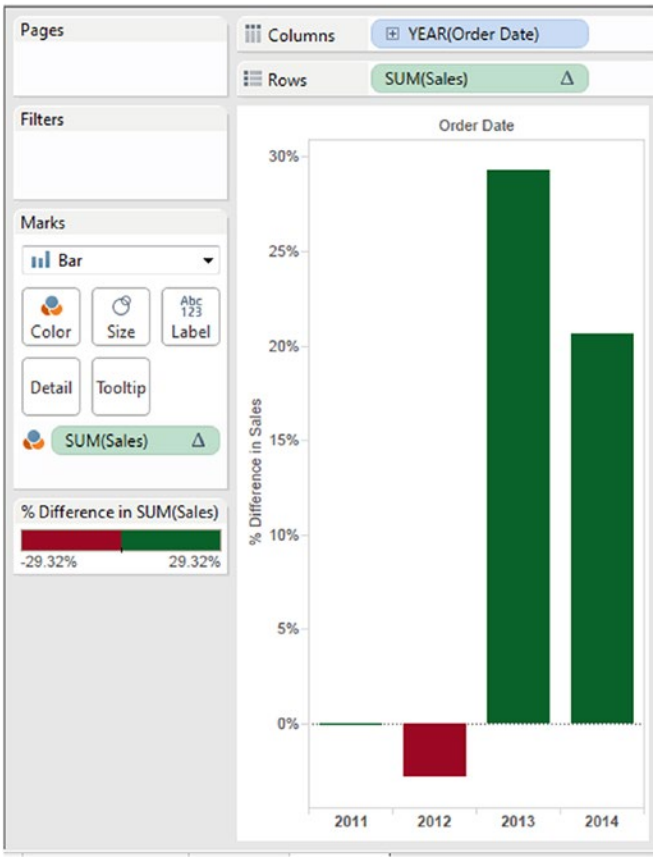


Figure 5-122. *Output after changing the "Stepped Color" to two*

5.8.1.1.7 Step 7

Press “CTRL” and drag the **measure** “Sales” from the rows shelf to “Label” on the marks card (Shown in Fig. 5-123).

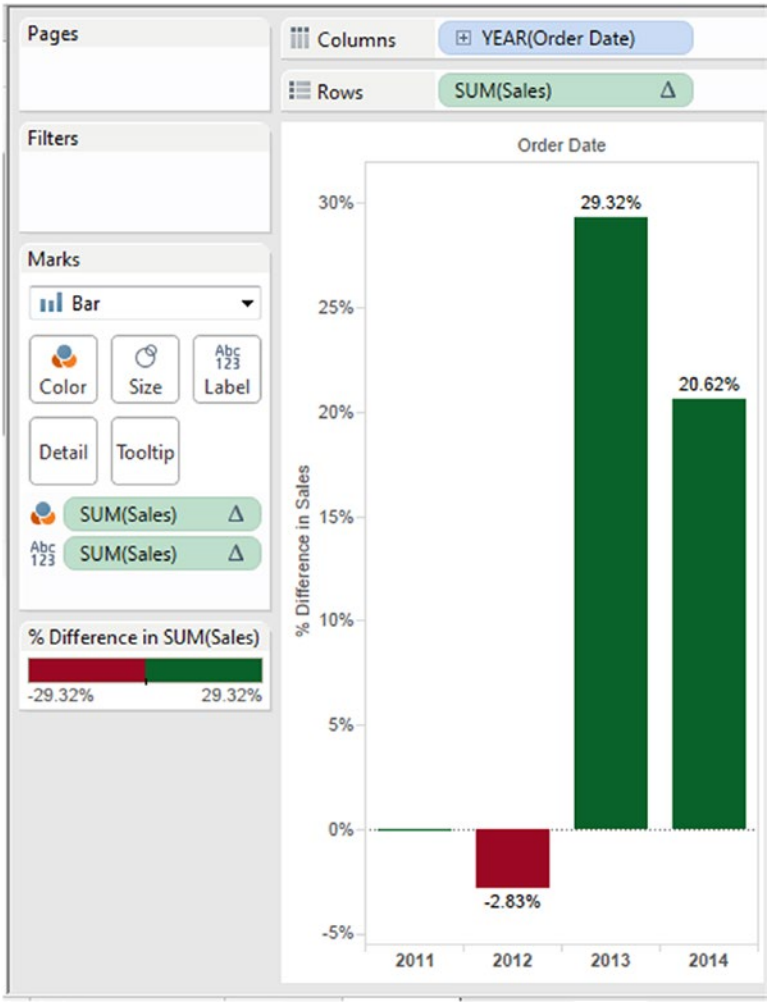


Figure 5-123. *Measure “Sales” placed on “Label” on the marks card*

The final output: Shown in Fig. 5-124.

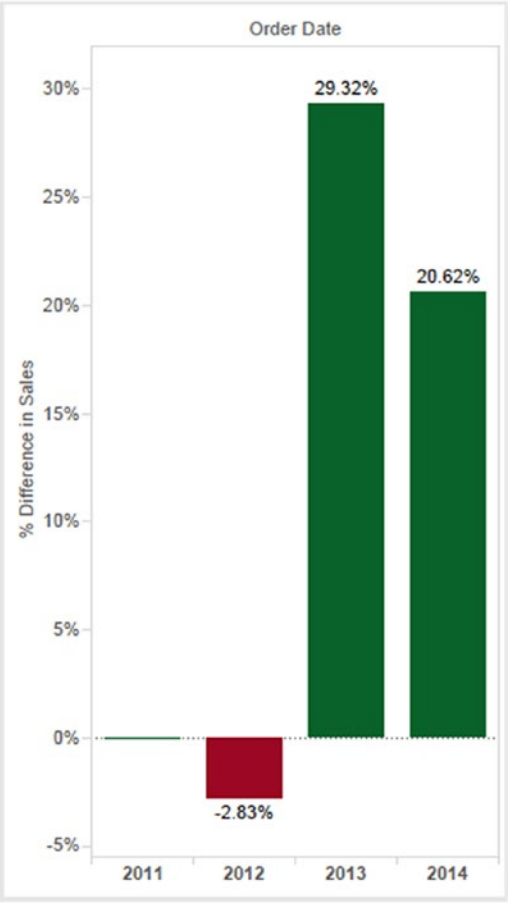


Figure 5-124. *Year Over Year Growth – Demo 1 – final output*

5.8.2 Demo 2

Objective:

The senior sales executive of "XYZ" corporation would like a visualization that presents the "Month over Month Growth" for the years 2011 and 2012.

Input: "Sample – Superstore.xls"

Expected Output: Shown in Fig. 5-125.

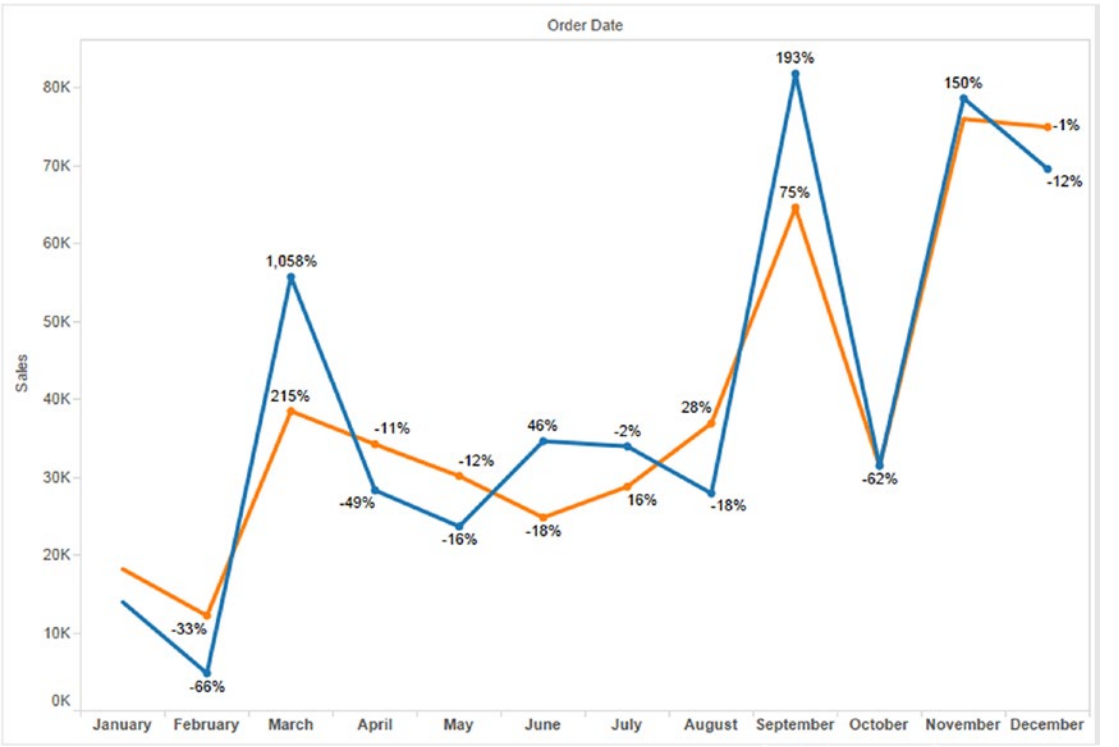


Figure 5-125. *"Year Over Year Growth" - Demo 2 – expected output*

5.8.2.1 Steps

Follow the steps as provided.

5.8.2.1.1 Step 1

Create a calculated field, "YearToDisplay" (Shown in Fig. 5-126).

Figure 5-126. *Calculated Field - "YearToDisplay" being created*

5.8.2.1.2 Step 2

Convert the calculated field, “YearToDisplay” to “Dimension” (Shown in Fig. 5-127).

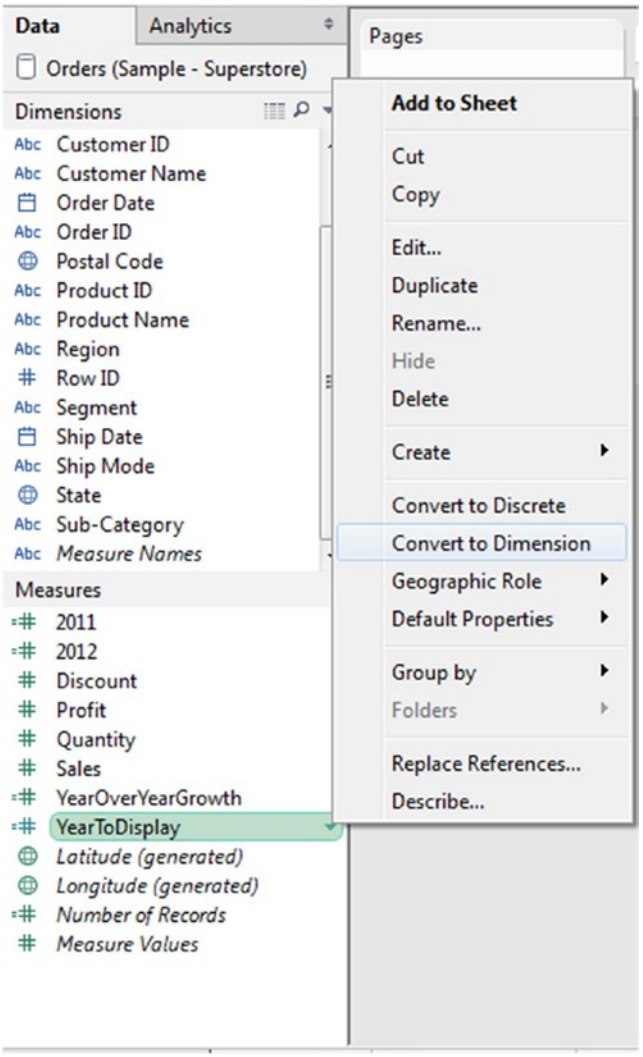


Figure 5-127. *Calculated field field “YearToDisplay” being converted to “Dimension”*

5.8.2.1.3 Step 3

Place the calculated field, “YearToDisplay” on the filters shelf.

Select the years “2011” and “2012” in the filter dialog box (Shown in Fig. 5-128).

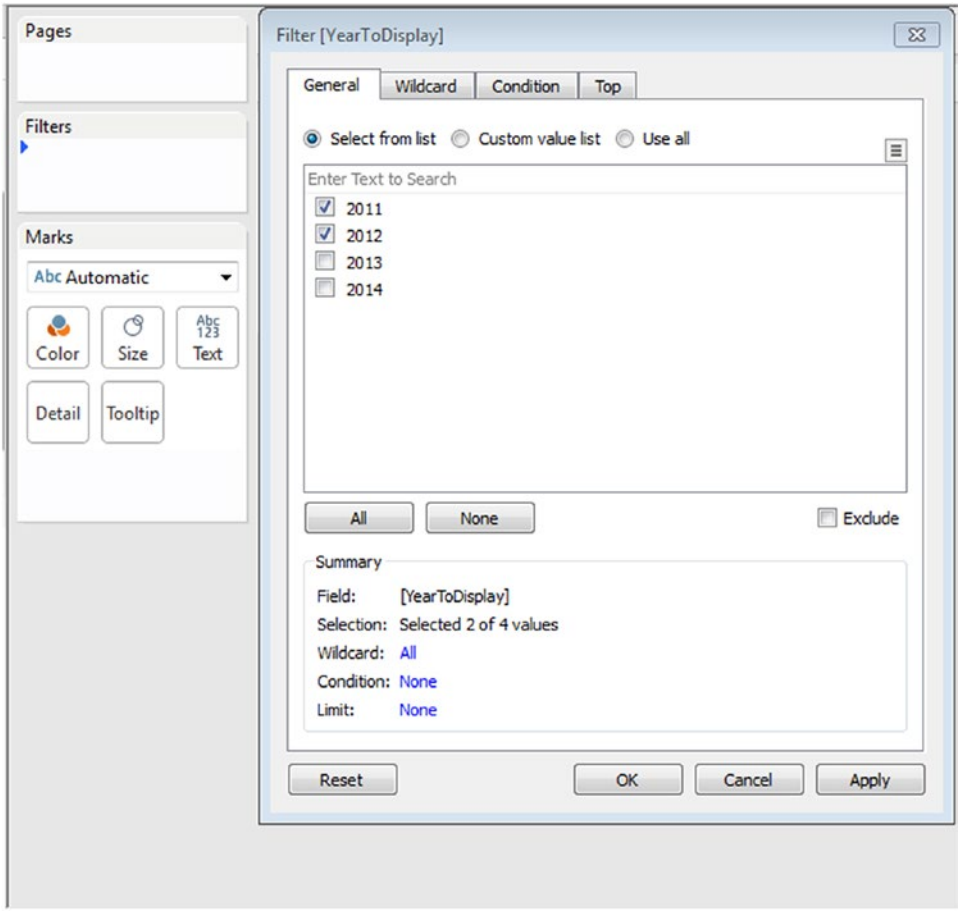


Figure 5-128. *Calculated field "YearToDisplay" placed on the filters shelf*

5.8.2.1.4 Step 4

Place the calculated field, "YearToDisplay" on "Color" on the marks card (Shown in Fig. 5-129).

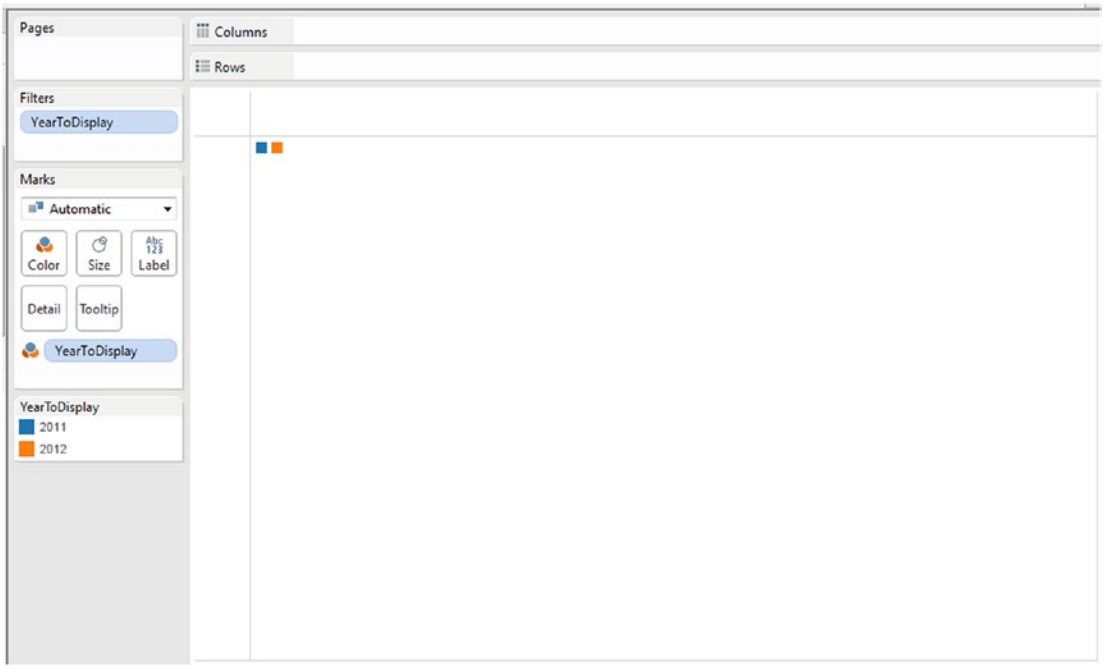


Figure 5-129. *Calculated field "YearToDisplay" placed on "Colors" on the marks card*

5.8.2.1.5 Step 5

Drag the **dimension** "Order Date" from the dimensions area under the data pane to the columns shelf. Change the date hierarchy to "Month". Retain it as "Discrete" (Shown in Fig. 5-130).

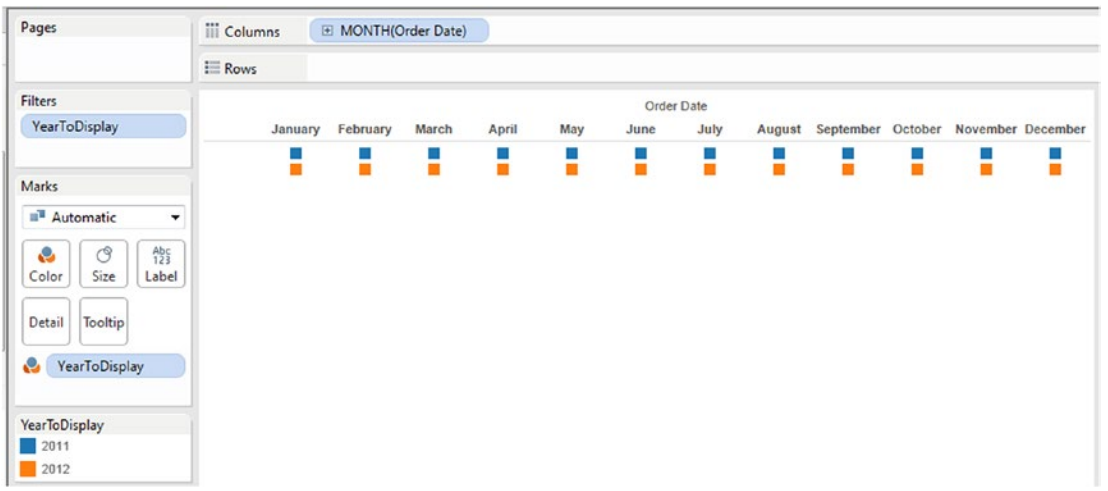


Figure 5-130. *Dimension "Order Date" placed on columns shelf*

Change the “Mark Type” to “Line” (Shown in Fig. 5-131).

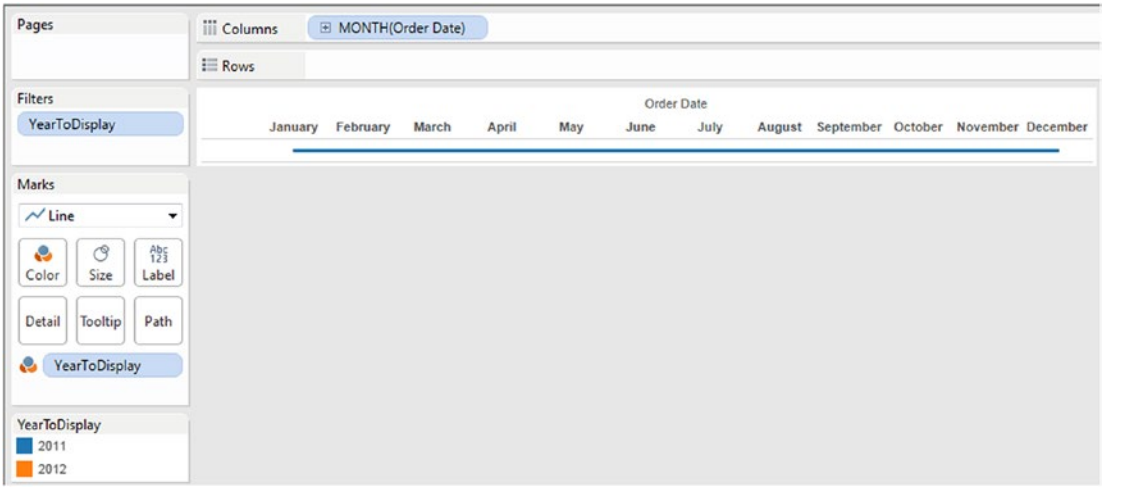


Figure 5-131. *“Mark Type” changed to “Line”*

5.8.2.1.6 Step 6

Drag the **measure** “Sales” from the measures area under the data pane and place it on the rows shelf (Shown in Fig. 5-132).

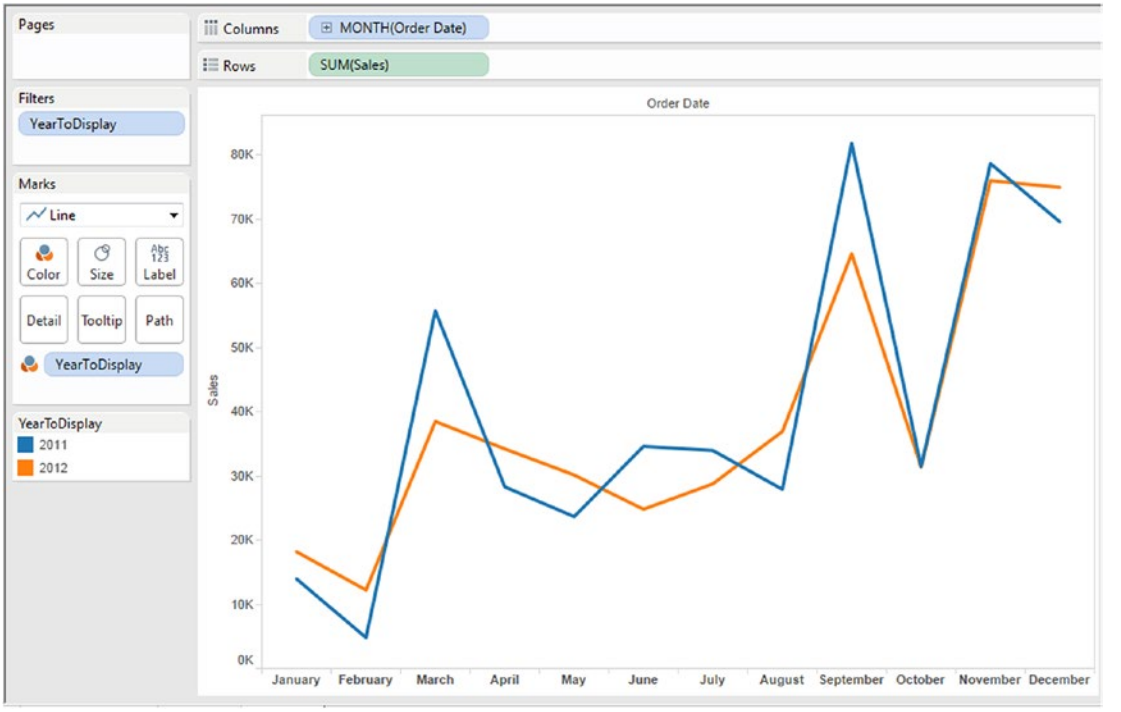


Figure 5-132. *Measure “Sales” placed on the rows shelf*

5.8.2.1.7 Step 7

Drag the **measure** "Sales" from the measures area under the data pane to "Label" on the marks card (Shown in Fig. 5-133).

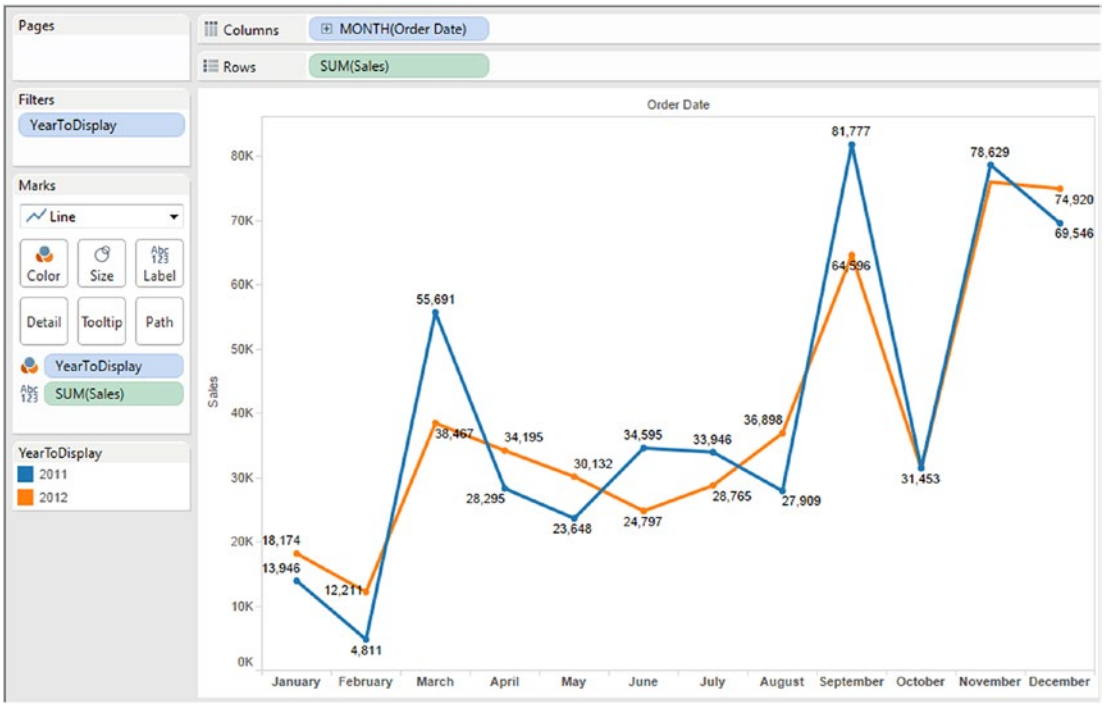


Figure 5-133. *Measure "Sales" placed on "Label" on the marks card*

5.8.2.1.8 Step 8

Add a "Quick Table Calculation – Year over Year Growth" to the **measure** "Sales" which is placed on the **"Label"** on the marks card (Shown in Fig. 5-134).

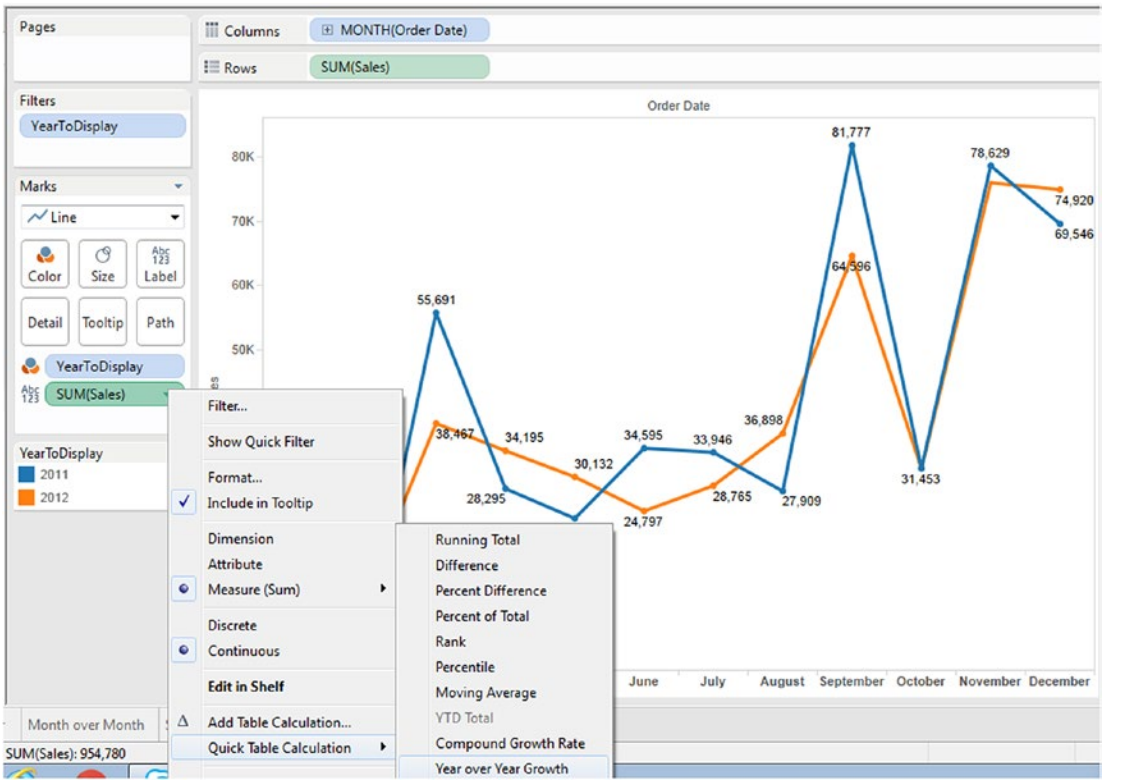


Figure 5-134. *"Add Table Calculation - Year over Year Growth" to the measure "Sales"*

Look at the "Table Calculation" dialog box (Shown in Fig. 5-135).

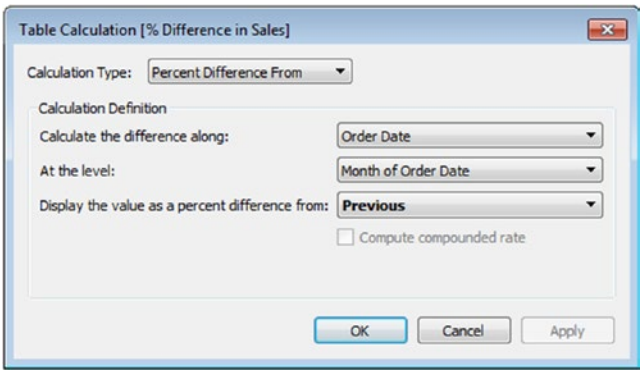


Figure 5-135. *"Table Calculation [% Difference in Sales]" dialog box*

Note that the "Percent Difference" is calculated at the level, "Month of Order Date" and displays the value as a percent difference from "Previous"

The final output: (Shown in Fig. 5-136).

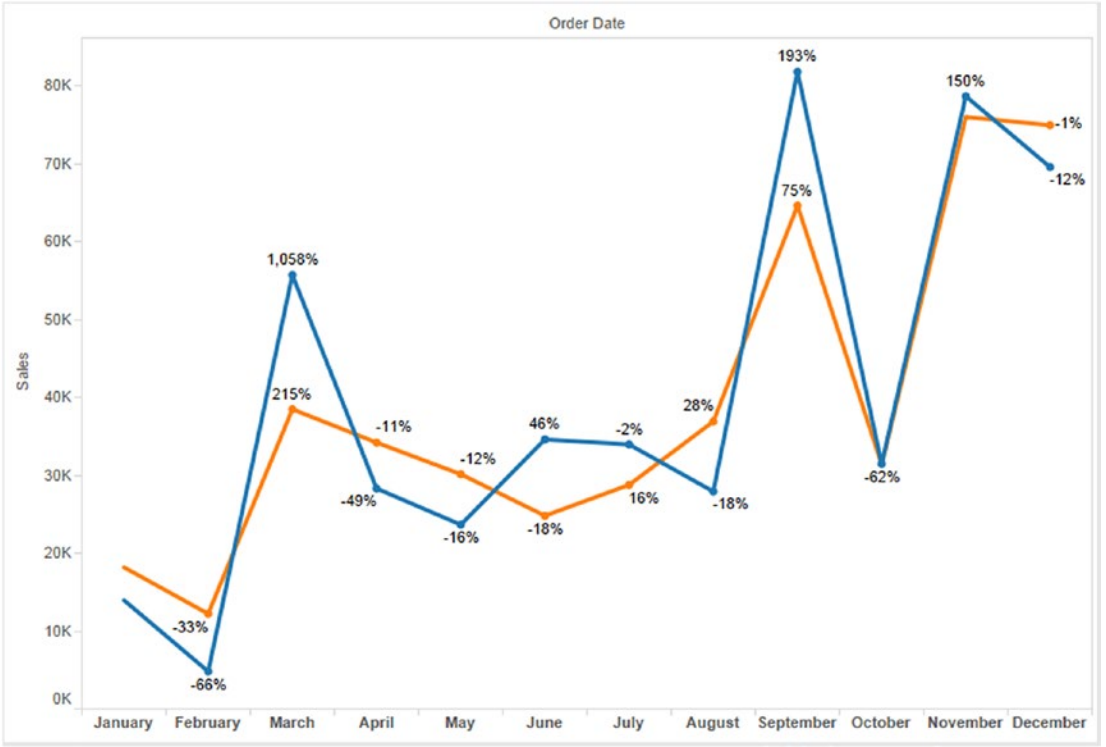


Figure 5-136. *Year over Year Growth – Demo 2 – final output*

Verify the output as follows:

Year	Month	Sales	Percent difference
2011	Jan	13,946	((4811- 13946) / 13946) * 100 = -66%
	Feb	4,811	
	Mar	55,691	((55691-4811)/4811) *100 = 1058%
	Apr	28,295	
	May	23,648	((28295 – 55691) / 55691) * 100 = -49%
	June	34,595	
			((23648 – 28295) / 28295) *100 = -16%

And so on…

Verify for the rest of the values.

5.8.3 Demo 3

Objective:

Data is provided for 4 years (2011, 2012, 2013 and 2014). The senior executive at the firm would like a visualization that shows the Year over Year Growth only for the years, 2011 and 2012.

Input: "Sample – Superstore.xls"

Expected output: Shown in Fig. 5-137.

2011	2012	YearOverYearGrowth
484,247	470,533	-2.83%

Figure 5-137. *Year over Year Growth – Demo 3 – expected output*

5.8.3.1 Steps

Follow the steps as provided.

5.8.3.1.1 Step 1

Create a calculated field, "2011" (Shown in Fig. 5-138).

Figure 5-138. *Calculated field "2011" being created*

Create a calculated field, “2012” (Shown in Fig. 5-139).

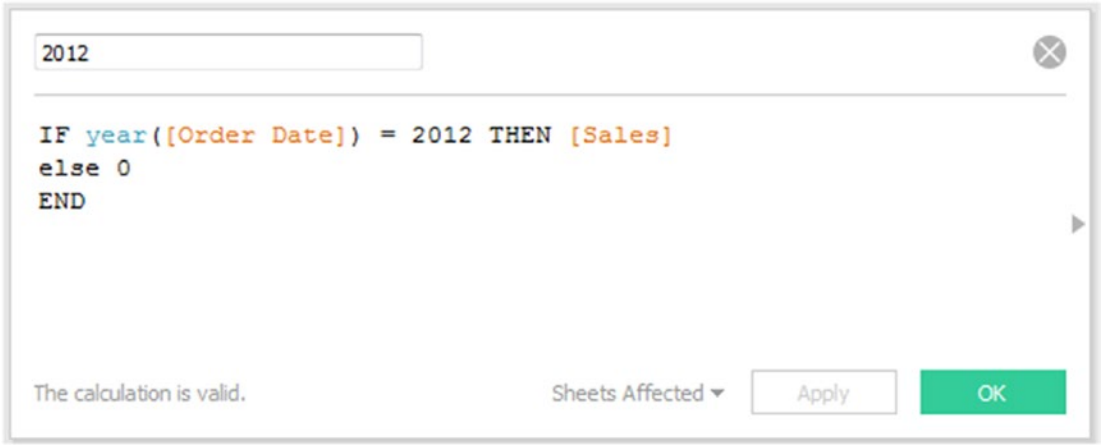


Figure 5-139. *Calculated field “2012” being created*

Create a calculated field, “YearOverYearGrowth” (Shown in Fig. 5-140).

Figure 5-140. *Calculated field “YearOverYearGrowth” being created*

5.8.3.1.2 Step 2

Drag the **dimension** "Measure Names" from the dimensions area under the data pane to the columns shelf (Shown in Fig. 5-141).

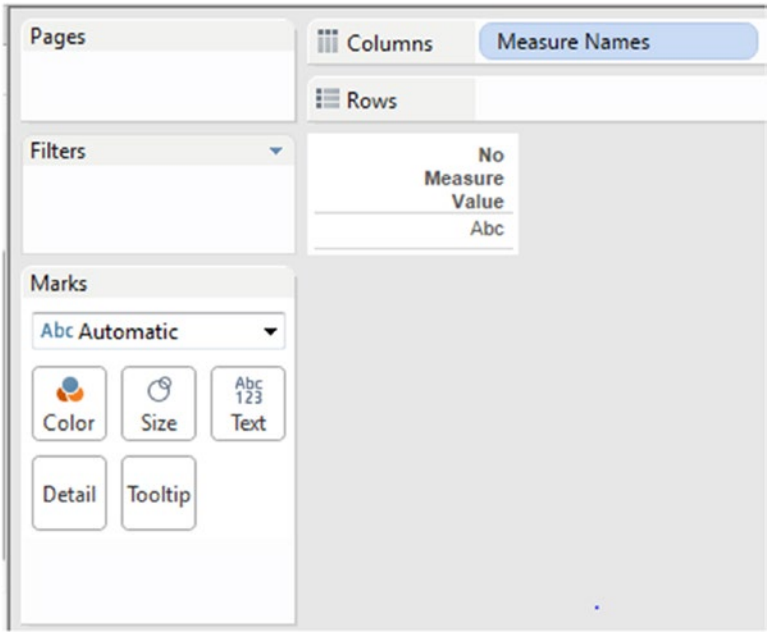


Figure 5-141. *"Measure Names" placed on the columns shelf*

5.8.3.1.3 Step 3

Drag the **dimension** "Measure Names" from the dimensions area under the data pane to the filters shelf (Shown in Fig. 5-142).

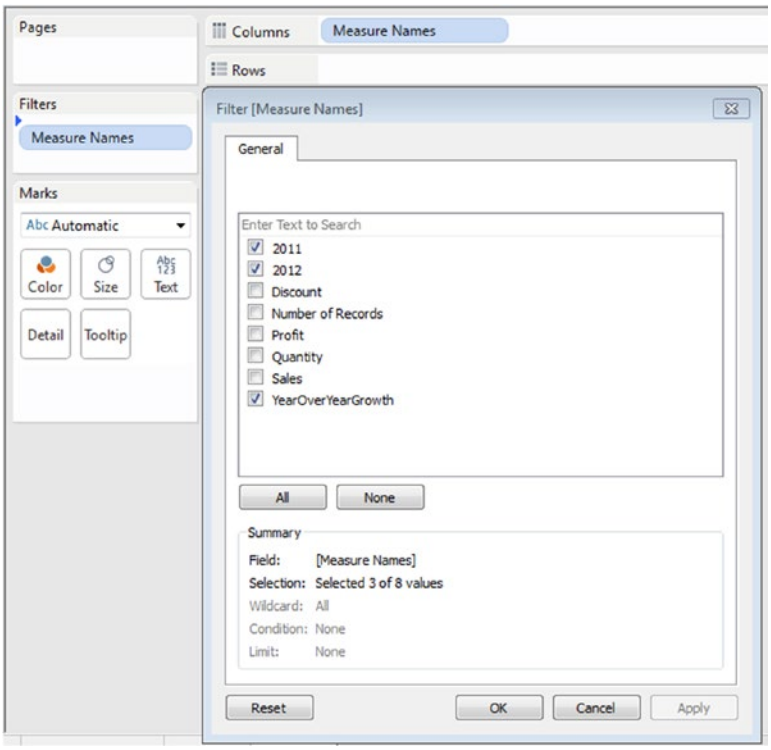


Figure 5-142. *"Measure Names" on the filters shelf*

Select the measures, "2011", "2012" and "YearOverYearGrowth".

5.8.3.1.4 Step 4

Drag the **measure** "Measure Values" from the measures area under the data pane and place it on **"Label"** on the marks card (Shown in Fig. 5-143).

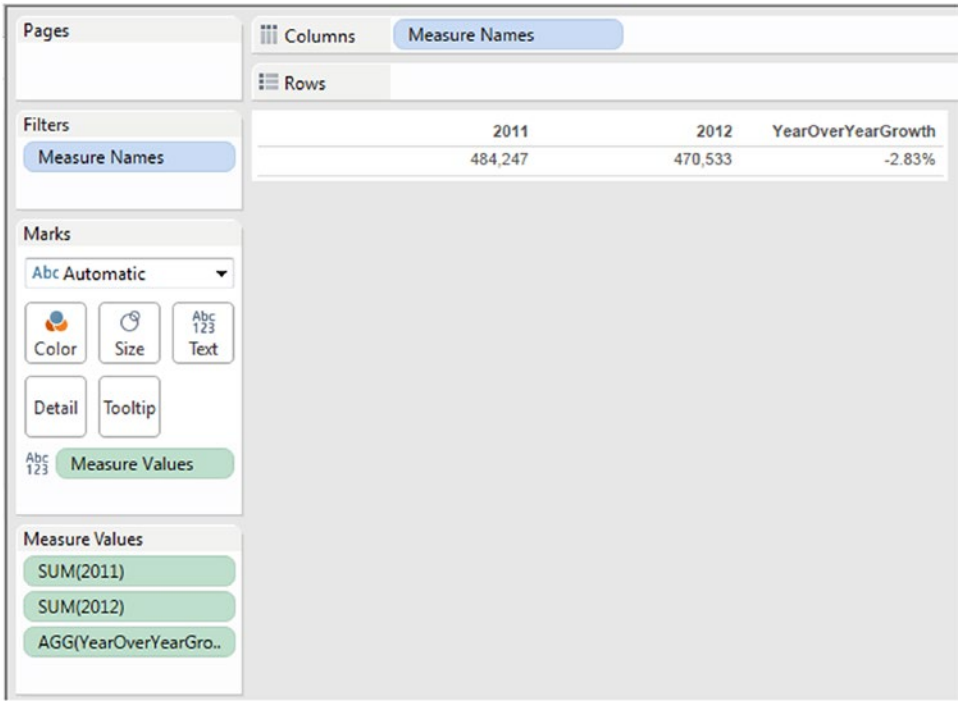


Figure 5-143. *"Measure values" placed on "Label" on the marks card*

The final output: Shown in Fig. 5-144.

2011	2012	YearOverYearGrowth
484,247	470,533	-2.83%

Figure 5-144. *Year over Year Growth – Demo 3 – final output*

5.9 Points to remember

- Table calculations aid in deriving additional insights from data. For example: (a) it helps to compare growth or differences across time periods (Year over year growth). (b) It helps to compute running total (running total) of inventory. The product list grows as products are added each day.
- There are ways in which table calculations can be customized such as by using its context menu or the calculation editor. To bring up the context menu, click on any field.
- Table calculations are generally applied to values in the entire table. For example to compute the running total or running average, a single method of calculation needs to be applied to the entire table.

5.10 Next Steps

This chapter familiarized us with table calculations. The next chapter will introduce us to string, numeric, date and logical functions.

CHAPTER 6

Customizing Data

"You can achieve simplicity in the design of effective charts, graphs and tables by remembering three fundamental principles: restrain, reduce, emphasize."

— Garr Reynolds, internationally acclaimed communications consultant and the author of best-selling books including the award-winning Presentation Zen, Presentation Zen Design

Chapter 5 introduced us to the various table calculations in Tableau. This chapter will help to explore and understand the following functions in Tableau:

- Number functions
- String functions
- Logical functions
- Date functions
- Aggregate functions
- Table calculation functions

6.1 Number functions

Tableau provides quite a few "Number" functions. Refer to Table 6-1 for Numeric functions supported by Tableau.

S. Acharya and S. Chellappan, *Pro Tableau*, DOI 10.1007/978-1-4842-2352-9_6

Table 6-1. *Numeric functions supported by Tableau*

Function Name	Description	Examples
ABS(number)	Absolute value of a given number is returned	ABS(-5) = 5
CEILING(number)	Rounds a number to the nearest integer of equal or greater value	CEILING(5.2345) = 6
DIV(integer1, integer2)	Integer part of a division operation is returned. Here integer1 is divided by integer2	DIV(13,2) = 6
FLOOR	Rounds a number to the nearest integer of equal or lesser value	FLOOR(5.3143) = 5
MIN(number, number)	Returns the minimum of the two arguments. The two arguments must be of same type. Returns Null if either argument is Null	MIN(6,5)= 5
MAX(number, number)	Returns the maximum of the two arguments. The two arguments must be of same type. Returns Null if either argument is Null	MAX(6,5)= 6
PI()	Returns a numeric constant value	3.14159
POWER(number, power)	Raises the number to the specified power.	POWER(6,2) = 36
ROUND(number,[decimals])	Rounds the number to a specified number of digits	ROUND(4.1567) = 4 ROUND(4.6567) = 5
SQRT(number)	Returns the square root of a number	SQRT(25) = 5
SQUARE(number)	Returns the square of a number	SQUARE(5) = 25

To learn more about number functions, refer to the link below.

https://onlinehelp.tableau.com/current/pro/desktop/en-us/functions_functions_number.html

Let us discuss a few number functions.

6.1.1 CEILING(number) and FLOOR(number)

Refer to Table 6-1 for description of the CEILING () and FLOOR() functions.

Let us learn to work with CEILING(number), FLOOR(number) functions.

Consider the below “Trainer Feedback” data set (Shown in Fig. 6-1).

	A	B
1	TrainerName	Feedback
2	John	3.14
3	James	4.78
4	Jack	3.35
5	Joshi	4.56
6	Joseph	4.23

Figure 6-1. *"Trainer Feedback" data set*

6.1.1.1 Steps to demonstrate the use of CEILING() and FLOOR() functions

Perform the following steps.

6.1.1.1.1 Step 1

Read in data from "Trainer Feedback" data set as shown in Fig. 6-2.

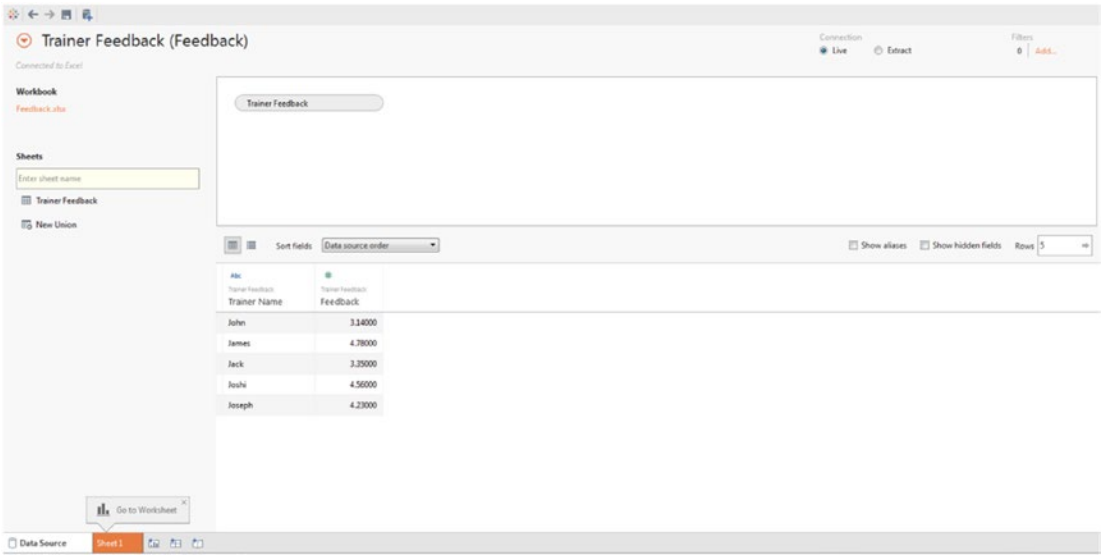


Figure 6-2. *Data source page showing the "Trainer Feedback" data set*

6.1.1.1.2 Step 2

Drag the dimension “Trainer Name” from the dimensions area under the data pane to the rows shelf. Drag the measure “Feedback” from the measures area under the data pane and place it on “Text” on the marks card (Shown in Fig. 6-3).

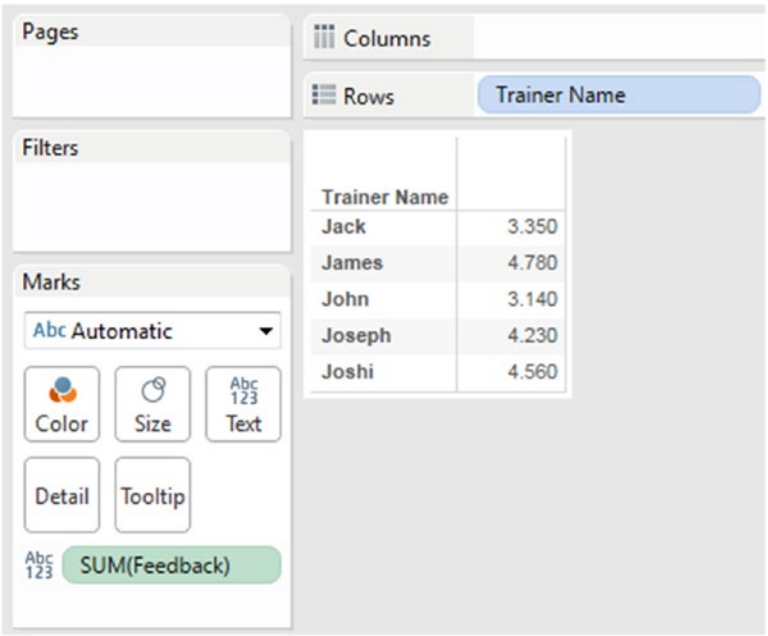


Figure 6-3. *Dimension, “Trainer Name” placed on the rows shelf, measure “Feedback” placed on “Text” on the marks card*

6.1.1.1.3 Step 3

Create a calculated field “Tableau CEILING” as shown in Fig. 6-4.

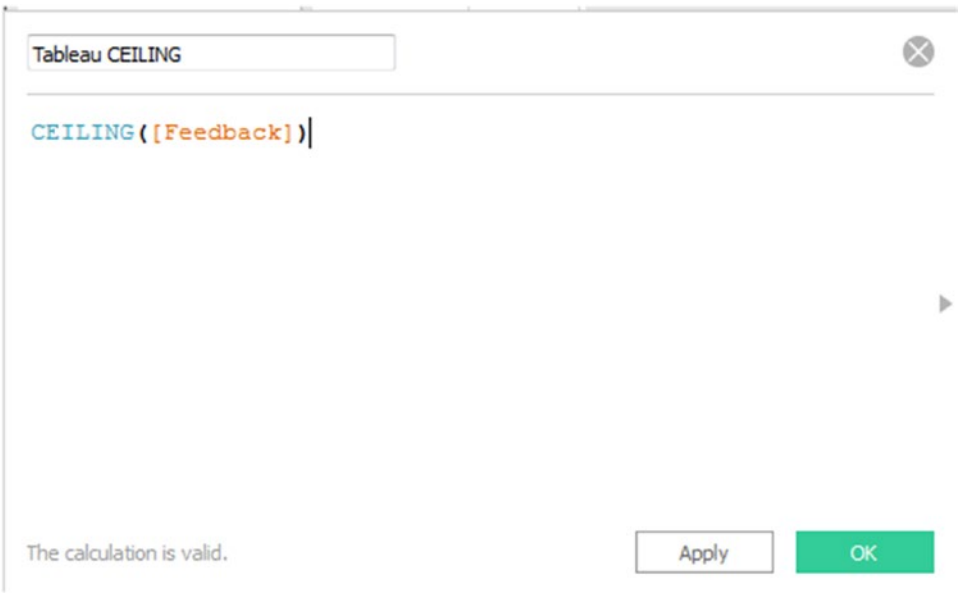


Figure 6-4. *Calculated field "Tableau CEILING" being created*

6.1.1.1.4 Step 4

Create a calculated field "Tableau FLOOR" as shown in Fig. 6-5.

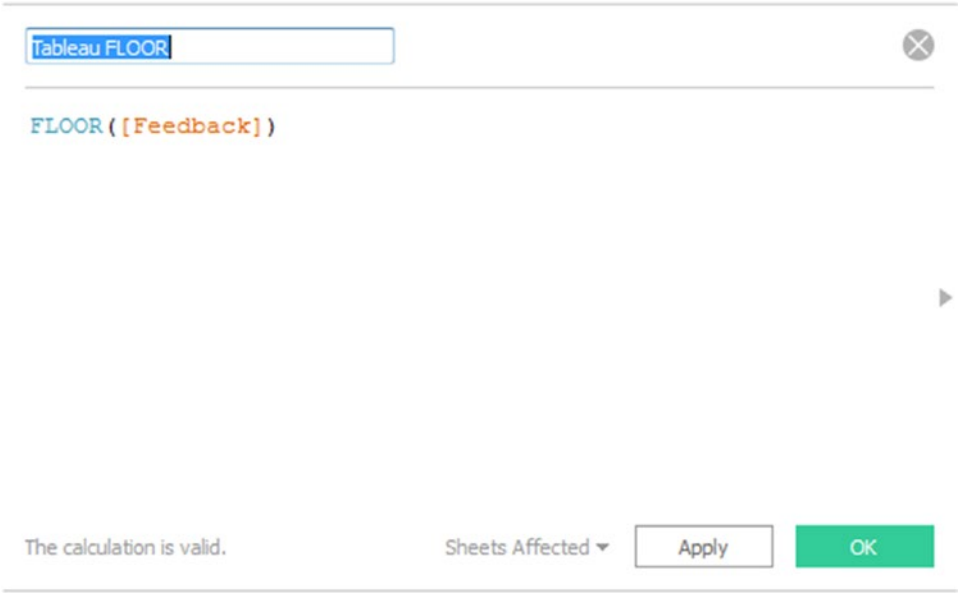


Figure 6-5. *Calculated field "Tableau FLOOR" being created*

6.1.1.1.5 Step 5

Double click on the calculated fields to place it on the view. Observe the CEILING and FLOOR value for the measure “Feedback” (Shown in Fig. 6-6).

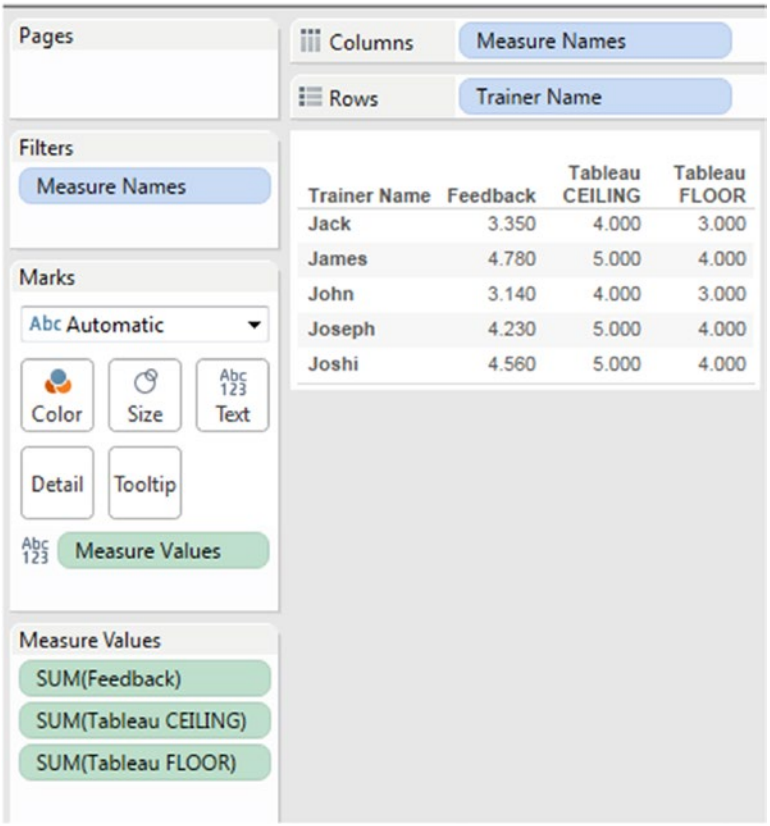


Figure 6-6. *View using the calculated fields “Tableau CEILING” and “Tableau FLOOR”*

6.1.2 MAX(number, number), MIN(number, number)

Refer to Table 6-1 for a description for the functions.

Consider the “Student” data set (Shown in Fig. 6-7).

	A	B	C
1	Stud Name	Mark 1	Mark 2
2	Smith	23	14
3	Jack	18	24
4	John	20	21
5	Scott	22	24
6	James	17	24

Figure 6-7. *"Student" data set*

6.1.2.1 Steps to demonstrate MAX() and MIN() functions

Perform the following steps:

6.1.2.1.1 Step 1

Read data from "Student" data set into Tableau (Shown in Fig. 6-8).

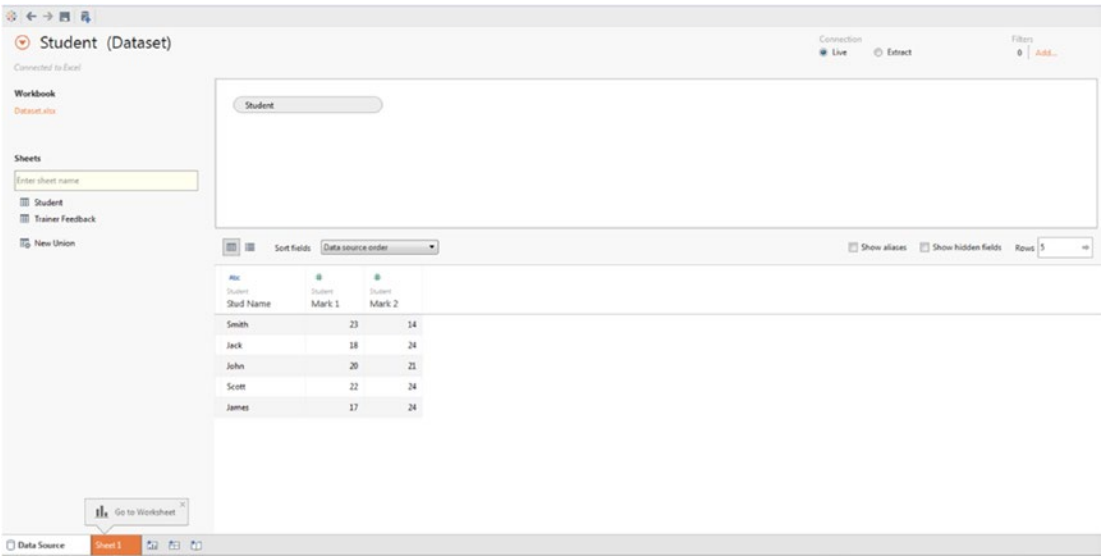


Figure 6-8. *Data source page showing the "Student" data set*

6.1.2.1.2 Step 2

Create a view as shown in Fig. 6-9.

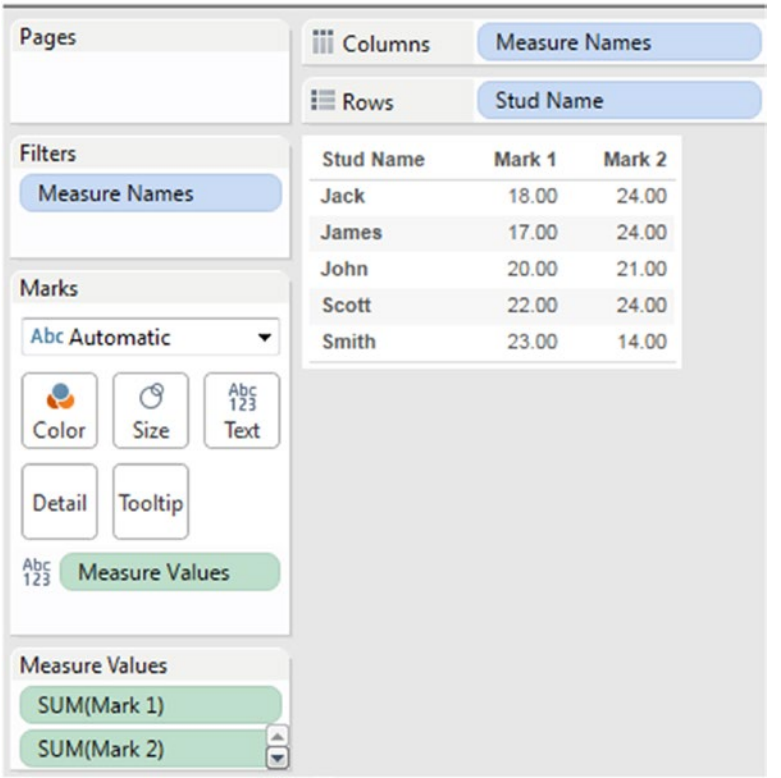


Figure 6-9. *View displaying the details of students' performance*

6.1.2.1.3 Step 3

Create a calculated field "Max Mark" (Shown in Fig. 6-10).

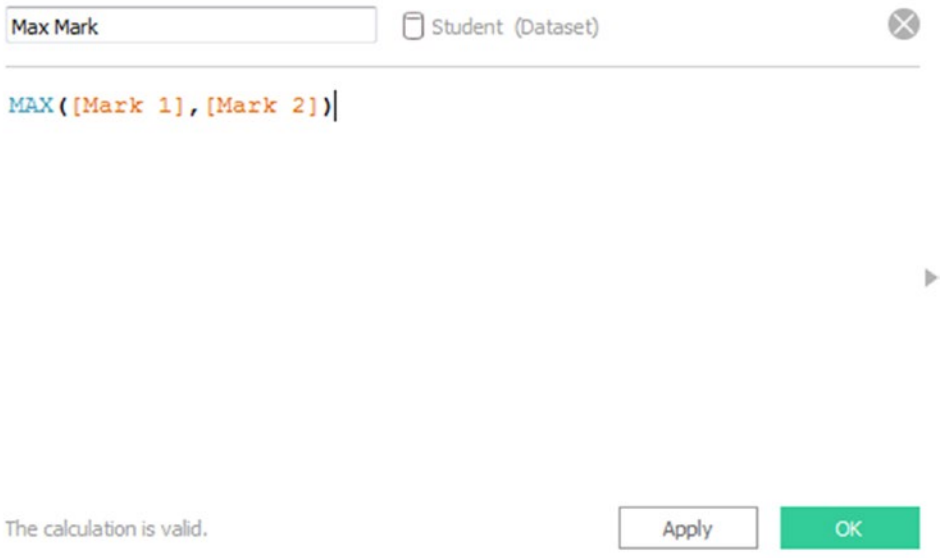


Figure 6-10. *Calculated field "Max Mark" being created*

6.1.2.1.4 Step 4

Create a calculated field "Min Mark" (Shown in Fig. 6-11).

Figure 6-11. *Calculated field "Min Mark" being created*

6.1.2.1.5 Step 5

Double click on the calculated fields to display “Max Mark” and “Min Mark” (Shown in Fig. 6-12).

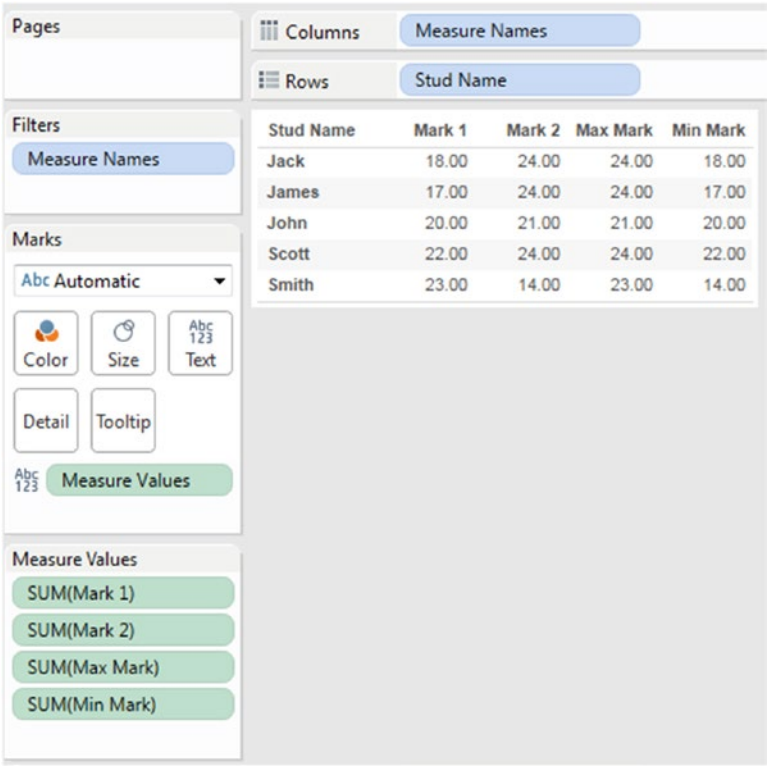


Figure 6-12. *View using the calculated fields “Max Mark” and “Min Mark”*

6.1.3 ABS(number)

Refer to Table 6-1 for description of the function.

Consider the “Items” data set (Shown in Fig. 6-13).

	A	B
1	Item Name	Profit
2	Books	-7800
3	Tables	9000
4	Chairs	-3500
5	Papers	-2400
6	Pens	7800

Figure 6-13. *"Items" data set*

6.1.3.1 Steps to demonstrate the use of ABS() function

Perform the following steps.

6.1.3.1.1 Step 1

Read data from "Items" data set into Tableau (Shown in Fig. 6-14).

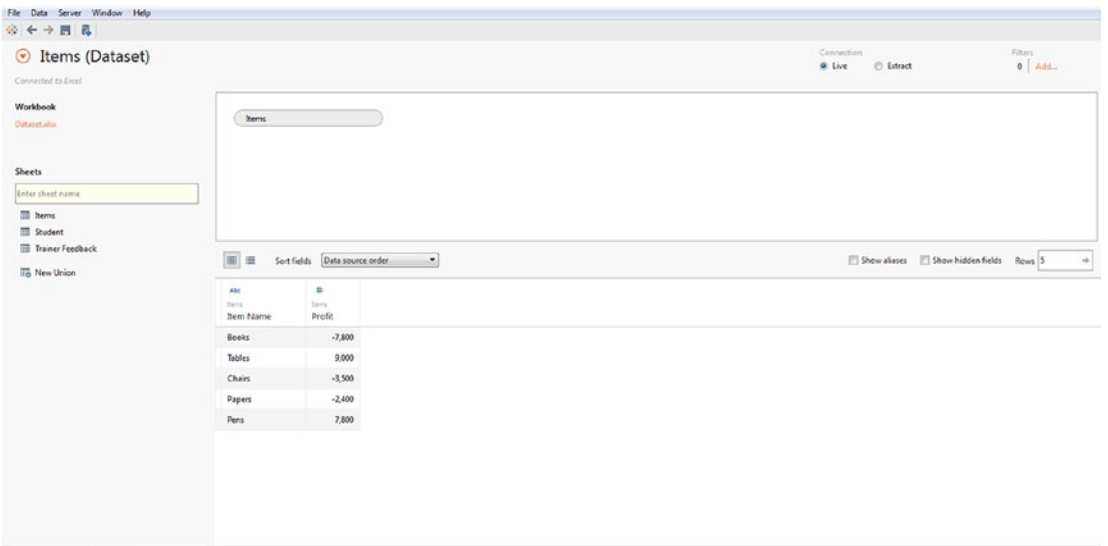


Figure 6-14. *Data source page showing the "Items" data set*

6.1.3.1.2 Step 2

Create a view as shown in Fig. 6-15.

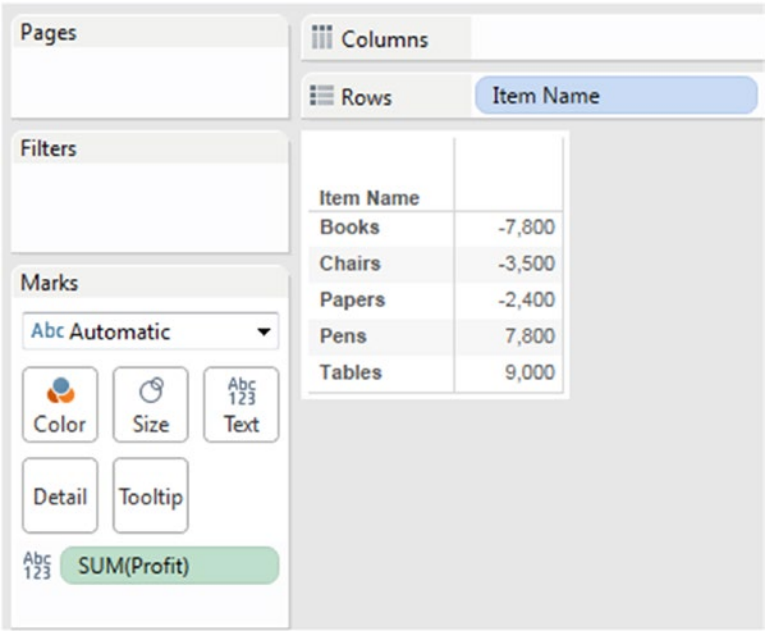


Figure 6-15. *View displaying the dimension "Item Name" placed on the rows shelf and measure, "Profit" placed on "Text" on the marks card*

6.1.3.1.3 Step 3

Create a calculated field “Tableau ABS()” as shown in Fig. 6-16.

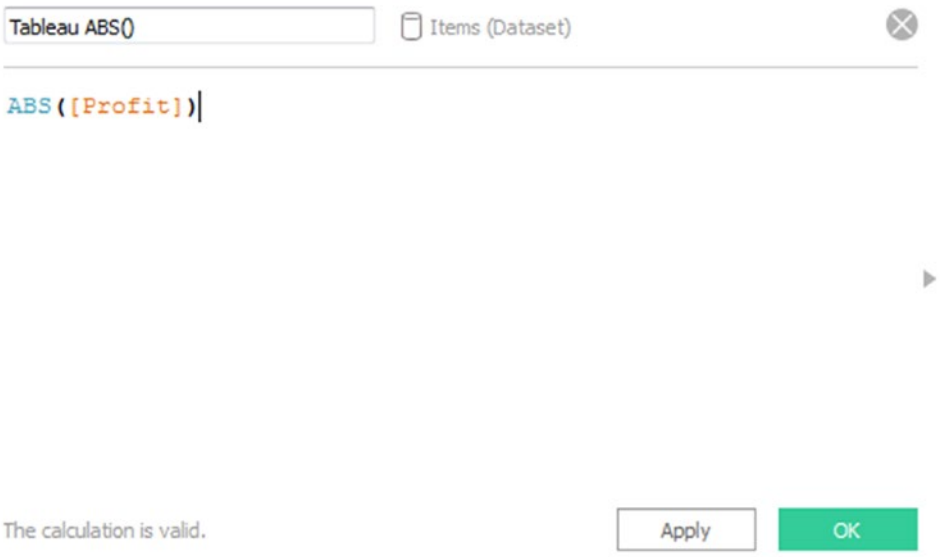


Figure 6-16. *Calculated field “Tableau ABS()” being created*

6.1.3.1.4 Step 4

Double click on the calculated field “Tableau ABS()” to place it on the view (Shown in Fig. 6-17).

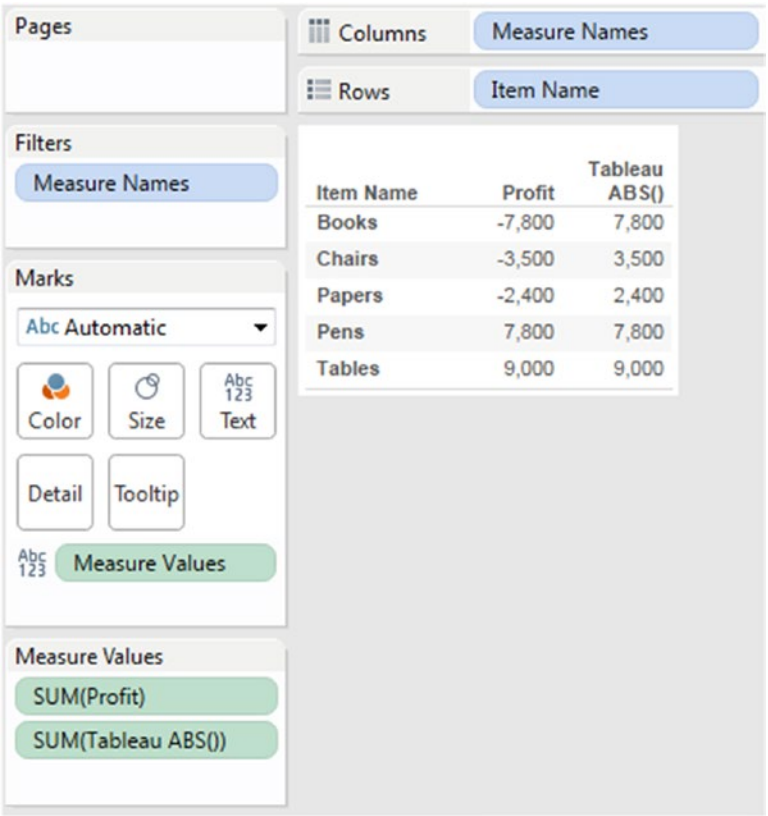


Figure 6-17. *View using the calculated field "Tableau ABS()"*

6.2 String functions

Tableau supports various string functions to enable working with strings.

Refer to Table 6-2 for string functions.

Table 6-2. *String functions in Tableau*

Function syntax	Example	
Ascii(character)	ASCII("T")	Returns 84
Char(integer)	CHAR(84)	Returns T
Len(string)	LEN("Tableau")	Returns 7
Max(a, b)	MAX("Tableau", "TABLEAU")	Returns Tableau
Min(a, b)	MIN("Tableau", "TABLEAU")	Returns TABLEAU
Replace(string, substring, replacement)	REPLACE("Visualisation", "sation","zation")	Returns Visualization
Startswith()	STARTSWITH("TABLEAU","T")	Returns TRUE
Upper(string)	UPPER("tableau")	Returns TABLEAU
Lower(string)	LOWER("TABLEAU")	Returns tableau
Left(string, num_characters)	LEFT("TABLEAU",3)	Returns TAB
Right(string, num_characters)	RIGHT("COMPASS",4)	Returns PASS
Trim()	TRIM(" Visualization ")	Returns Visualization
Rtrim()	RTRIM("Visualization ")	Returns "Visualization"
Ltrim()	LTRIM(" Visualization")	Returns "Visualization"

Let us discuss a few string functions in Tableau.

6.2.1 Concatenation

Objective: To concatenate the dimension "Customer_ID" with the dimension "Customer Name".

Formula: "Customer ID" + " : " + str([Customer ID]) + " , " + [Customer Name]

6.2.1.1 Steps to demonstrate concatenation

Perform the following steps:

6.2.1.1.1 Step 1

Read data from “Sample – Superstore” data set.

6.2.1.1.2 Step 2

Create the calculated field “Customer ID + Customer Name” (Shown in Fig. 6-18).

Figure 6-18. *Calculated field "Customer ID + Customer Name" being created*

6.2.1.1.3 Step 3

Drag the calculated field “Customer ID + Customer Name” to the rows shelf (Shown in Fig. 6-19).

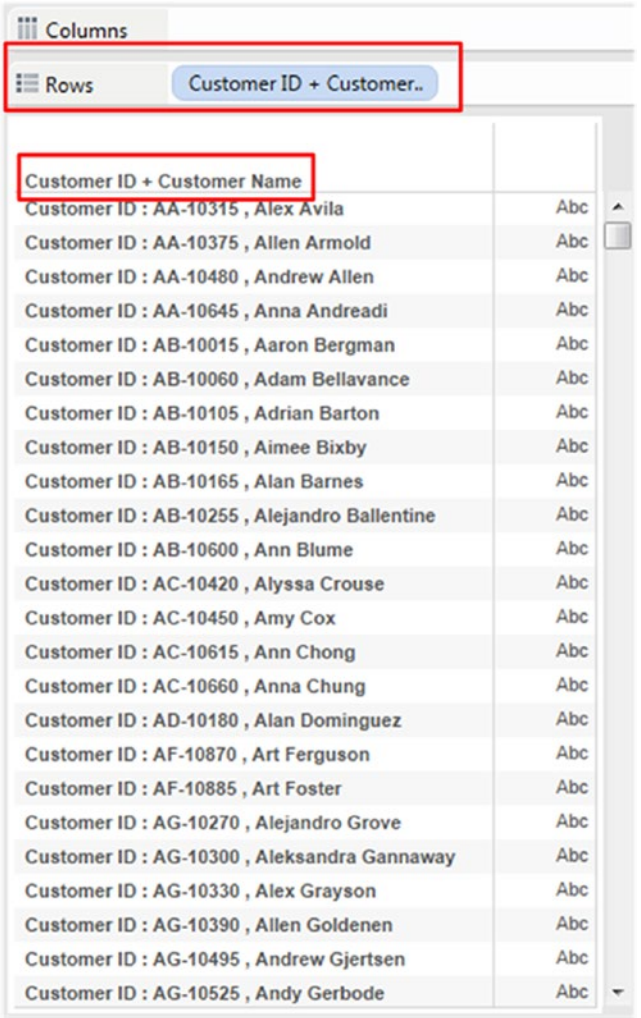


Figure 6-19. *View that shows concatenation of "Customer ID and Customer Name"*

6.2.2 Left() and Find() functions

Objective: To extract the first name from the “Customer Name” dimension.

Functions used: Left() and Find()

Syntax of Left()

Left(String, Number of characters to extract)

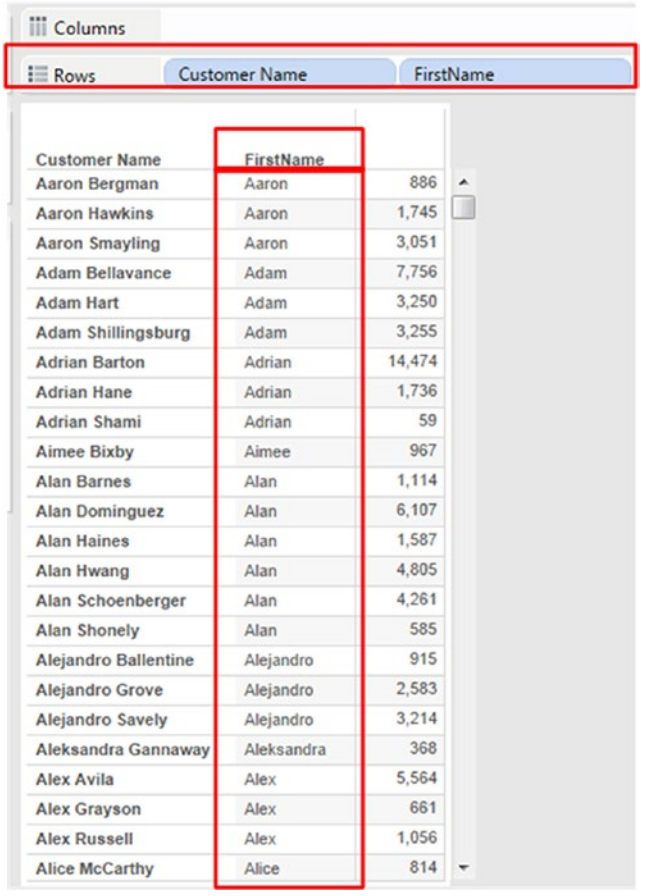
Syntax of Find()

Find(String, Substring, Start position)

Create a calculated field by the name, “FirstName” (Shown in Fig. 6-20).

Figure 6-20. *Calculated field "FirstName" being created*

Formula: Left([Customer Name], find([Customer Name]," " , 1))
Refer to Fig. 6-21 for output.



Customer Name	FirstName	
Aaron Bergman	Aaron	886
Aaron Hawkins	Aaron	1,745
Aaron Smayling	Aaron	3,051
Adam Bellavance	Adam	7,756
Adam Hart	Adam	3,250
Adam Shillingsburg	Adam	3,255
Adrian Barton	Adrian	14,474
Adrian Hane	Adrian	1,736
Adrian Shami	Adrian	59
Aimee Bixby	Aimee	967
Alan Barnes	Alan	1,114
Alan Dominguez	Alan	6,107
Alan Haines	Alan	1,587
Alan Hwang	Alan	4,805
Alan Schoenberger	Alan	4,261
Alan Shonely	Alan	585
Alejandro Ballentine	Alejandro	915
Alejandro Grove	Alejandro	2,583
Alejandro Savely	Alejandro	3,214
Aleksandra Gannaway	Aleksandra	368
Alex Avila	Alex	5,564
Alex Grayson	Alex	661
Alex Russell	Alex	1,056
Alice McCarthy	Alice	814

Figure 6-21. *View that shows "FirstName" from the dimension "Customer Name"*

6.2.3 Contains() function

Returns true if the given string contains the specified substring.

6.2.3.1 Problem statement

Given below is a list of product names. We are looking for those product names that contain the word "Wall Clock" in it. Display a list of only those product names that contains the word "Wall Clock".

Example:

Product Name	Sales
12-1/2 Diameter Round Wall Clock	$551
"6""Cubicle Wall Clock, Black"	$125

Input: Refer to Fig. 6-22**.**

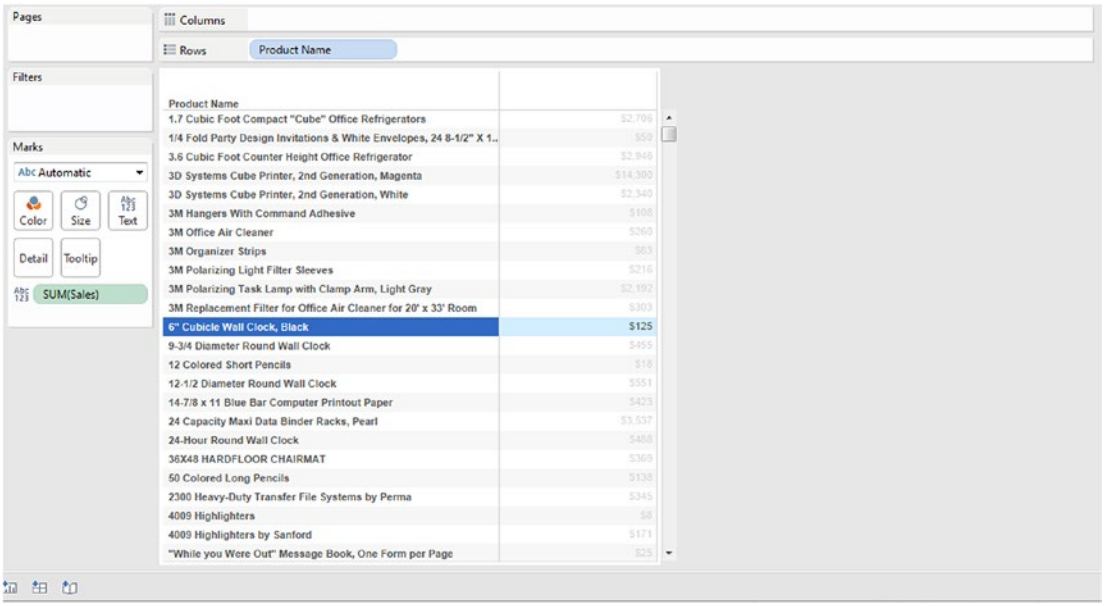


Figure 6-22. *Dimension "Product Name" placed on the rows shelf*

Create a calculated field "Product-Wall Clock" (Shown in Fig. 6-23).

Figure 6-23. *Calculated field "Product-Wall Clock" being created*

Output:
To display the list of only those products that contain the string "Wall Clock".
Drag the dimension "Product-Wall clock" into the filters shelf and select only "True" (Shown in Fig. 6-24).

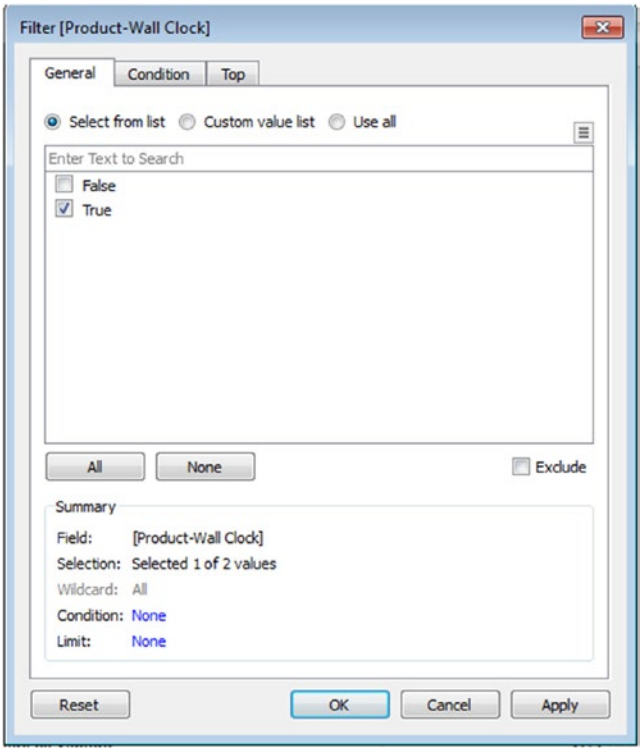


Figure 6-24. *"Filter[Product-Wall clock]" dialog box*

Notice the list now shows only those "Product Name" that contains the string "Wall Clock". (Shown in Fig. 6-25).

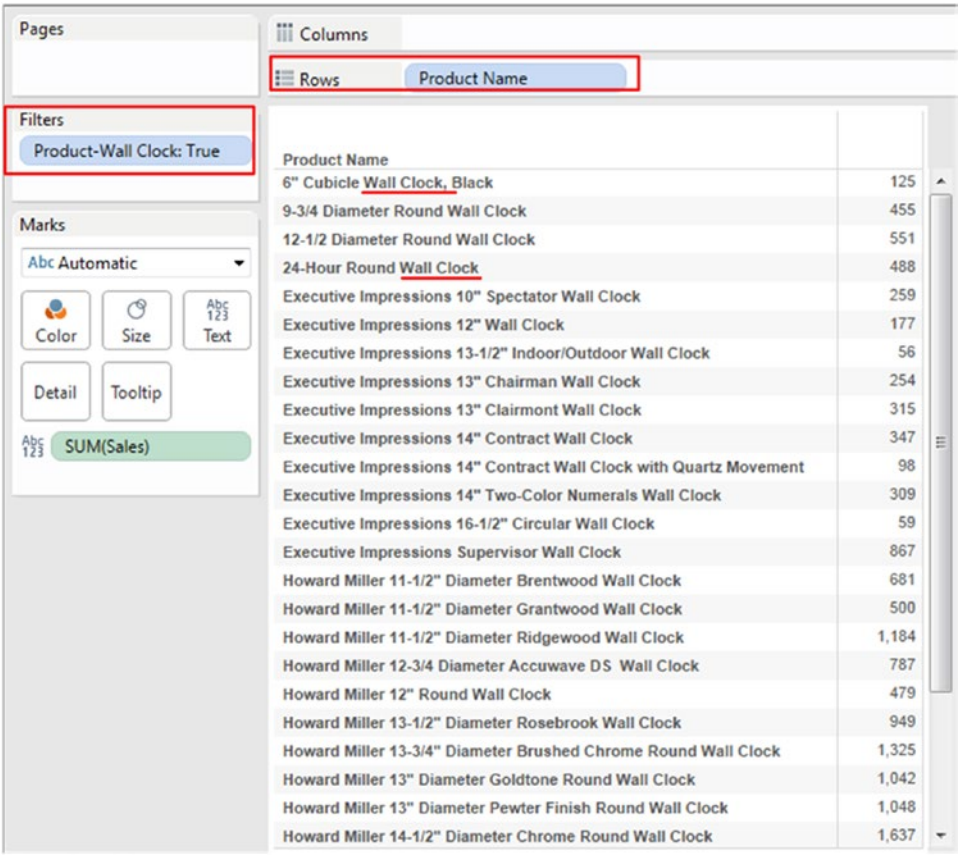


Figure 6-25. *View displays only those "Product Name" that contains the string "Wall Clock"*

6.2.4 Len() function

Returns the length of the string.

Objective: To count the number of words in a sentence.
Hint: Use len() function.
For example, count the number of words in the sentence, "Tableau is a good data visualization tool."
Output expected is 7.

6.2.4.1 Steps to demonstrate the use of Len() function

Perform the following steps.

6.2.4.1.1 Step 1

Create a calculated field "NumberOfWordsinSentence" (Shown in Fig. 6-26).

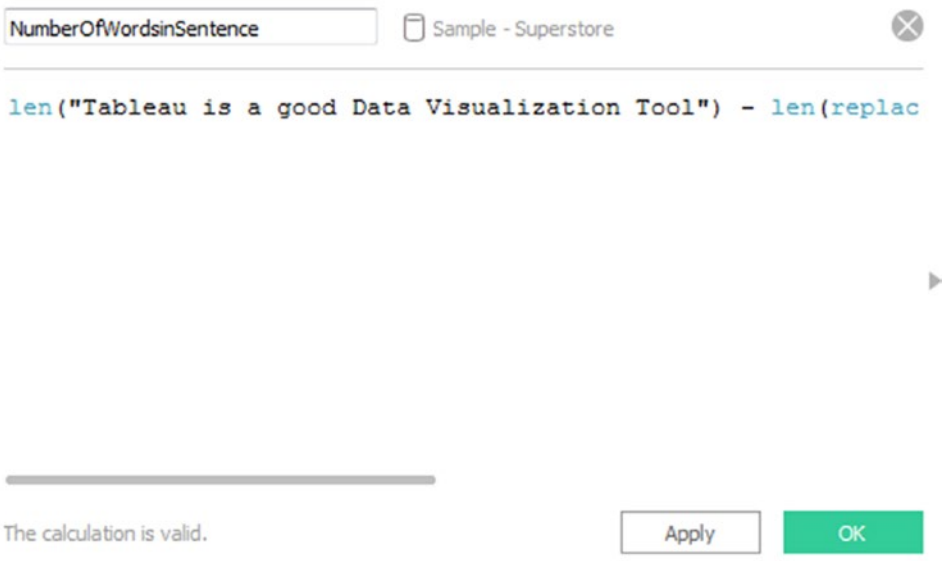


Figure 6-26. *Calculated field "NumberOfWordsinSentence" being created*

Type in the below formula in the calculated field dialog box:

len("Tableau is a good data visualization tool") - len(replace("Tableau is a good data visualization tool"," ","")) + 1

Let us look at what the formula does:

We use a replace function to replace the space between words with an empty string.

Replace("Tableau is a good data visualization tool", " ", "")

The output of the replace function is "Tableauisagooddatavisualizationtool"

We count the length of the string "Tableau is a good data visualization tool" using the below function:

len("Tableau is a good data visualization tool")

The output is 41. This means that there are 41 characters in the string, "Tableau is a good data visualization tool"

Next, we determine the length of the string "Tableauisagooddatavisualizationtool"

The output of the above is 35.

Let us take a look at the formula again:

len("Tableau is a good data visualization tool") - len(replace("Tableau is a good data visualization tool", " ", "")) + 1

Substituting the values returned by the functions:

41 - len(replace("Tableau is a good data visualization tool", " ", "")) + 1

41 – len("Tableauisagooddatavisualizationtool") + 1

41 – 35 + 1

Why are we adding one at the end? That is because the last word does not have a space after it.

41 – 35 + 1 returns 7.

This is the number of words in the sentence "Tableau is a good data visualization tool".

6.2.4.1.2 Step 2

Because the calculated field "NumberOfWordsinSentence" will return a numeric value, by default it is placed under measures in the measure area under the Data Pane. Convert it to a "Dimension" as it is not required to run aggregation on it (Shown in Fig. 6-27).

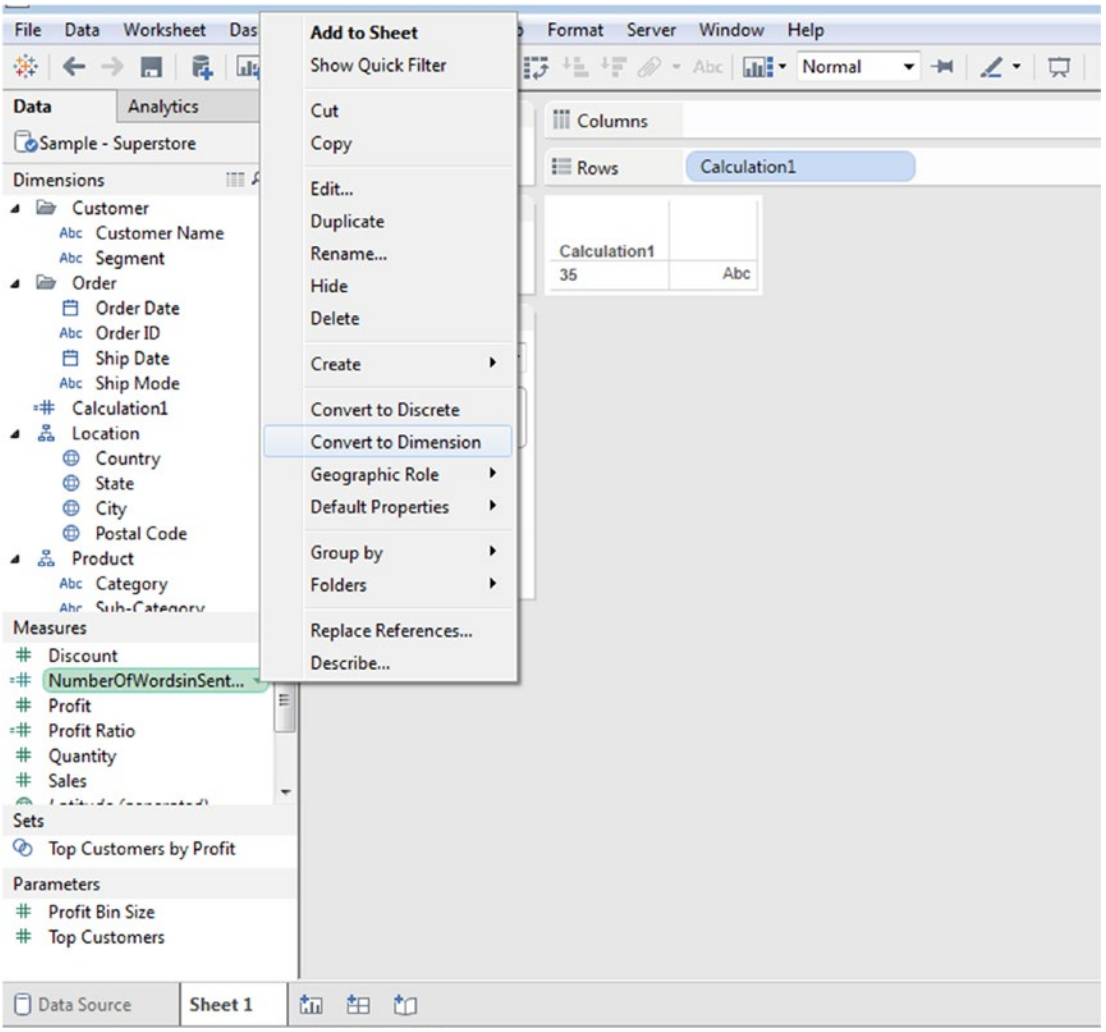


Figure 6-27. *Converting the measure "NumberOfWordsinSentence" to a dimension*

6.2.4.1.3 Step 3

Then drag the dimension "NumberOfWordsinSentence" to the rows shelf (Shown in Fig. 6-28).

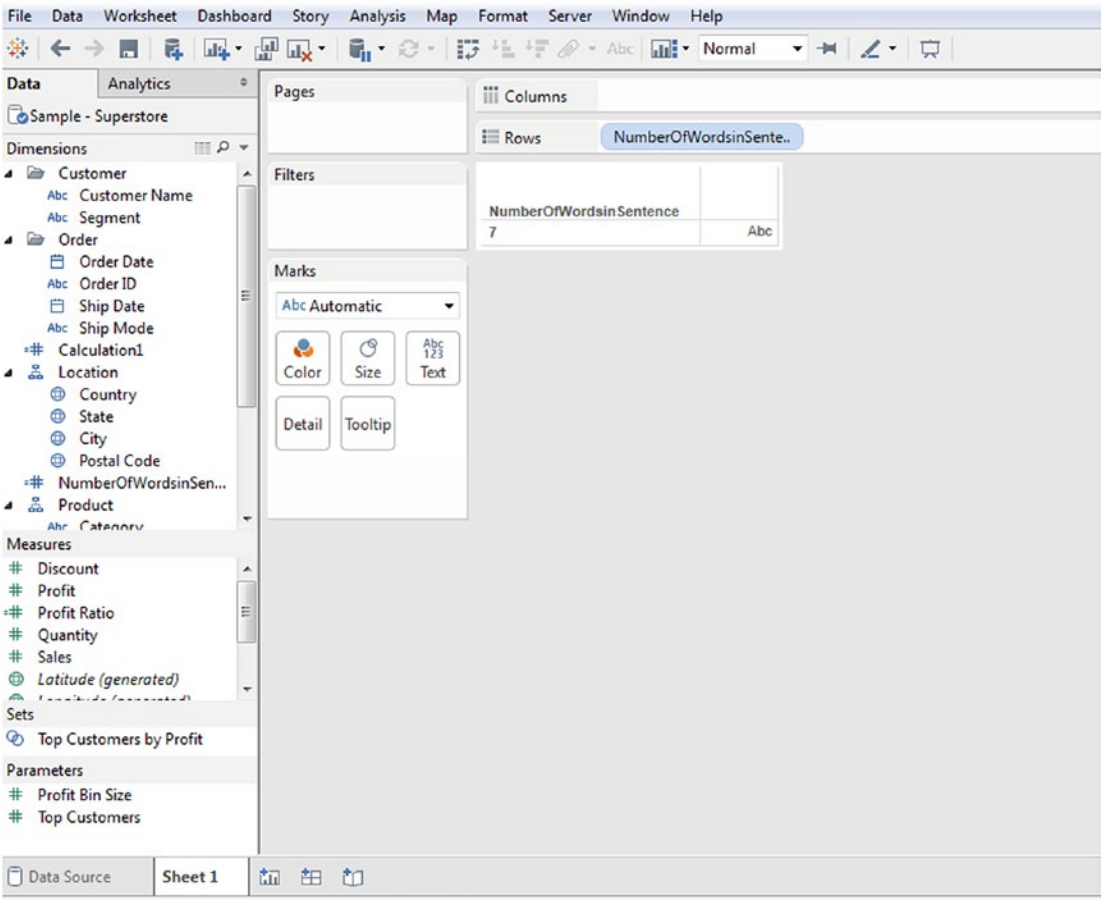


Figure 6-28. *Dimension, "NumberOfWordsinSentence" placed on the rows shelf*

6.3 Logical Functions

Let us explore few logical functions.

6.3.1 CASE

This function evaluates the expression and compares the sequence of values, value1, value2, etc. If there is a match, CASE returns the corresponding return value. Otherwise, it returns the default value. If there is no default value, it returns Null.

6.3.1.1 Steps to demonstrate CASE

Perform the following steps.

6.3.1.1.1 Step 1

Read the data from "Sample-Superstore" data set (Shown in Fig. 6-29).

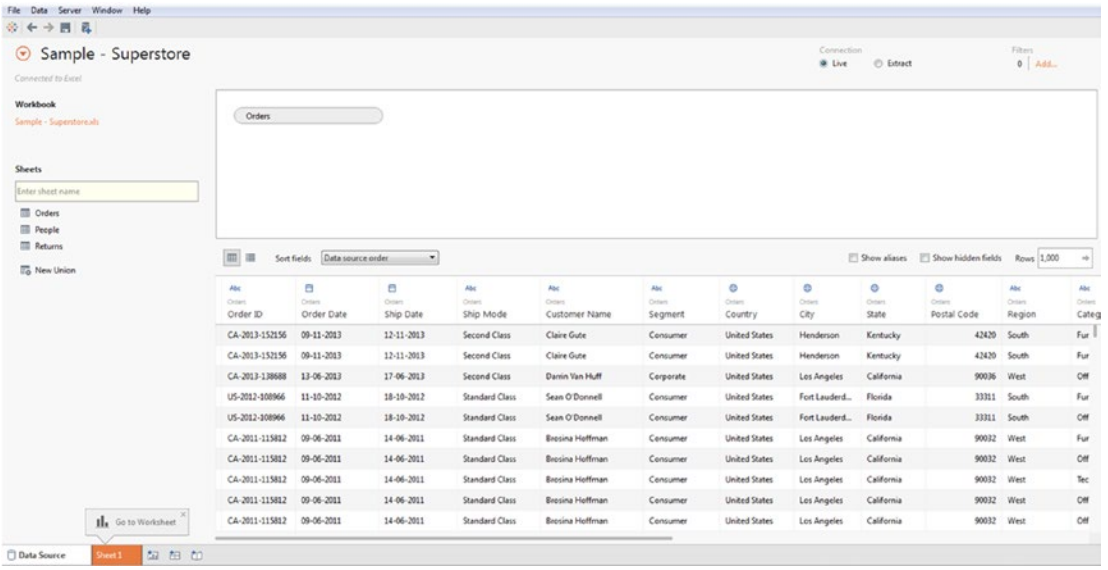


Figure 6-29. *Data source page showing "Orders" data set from "Sample – Superstore.xls"*

6.3.1.1.2 Step 2

Create a view as shown in Fig. 6-30.

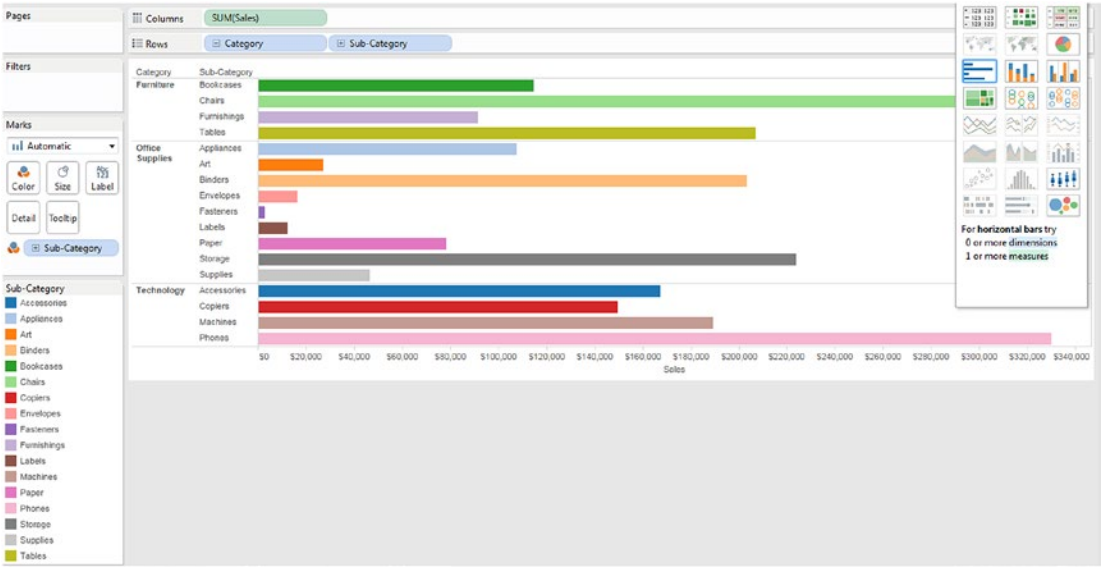


Figure 6-30. *View displaying "Sales" by "Category" and "Sub-Category"*

6.3.1.1.3 Step 3

Create a calculated field “Shipping Expense” (Shown in Fig. 6-31).

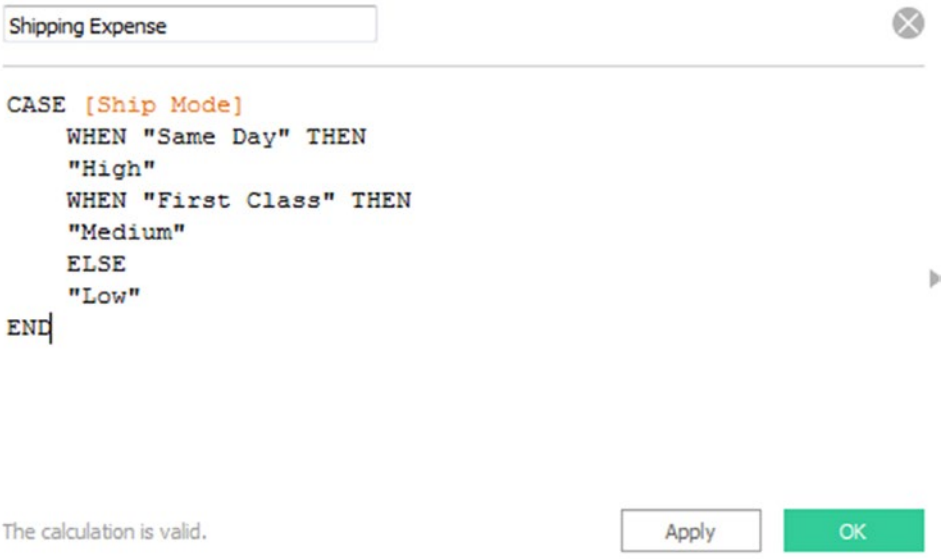


Figure 6-31. *Calculated field “Shipping Expense” being created*

6.3.1.1.4 Step 4

Drag the calculated field “Shipping Expense” to the rows shelf (Shown in Fig. 6-32).

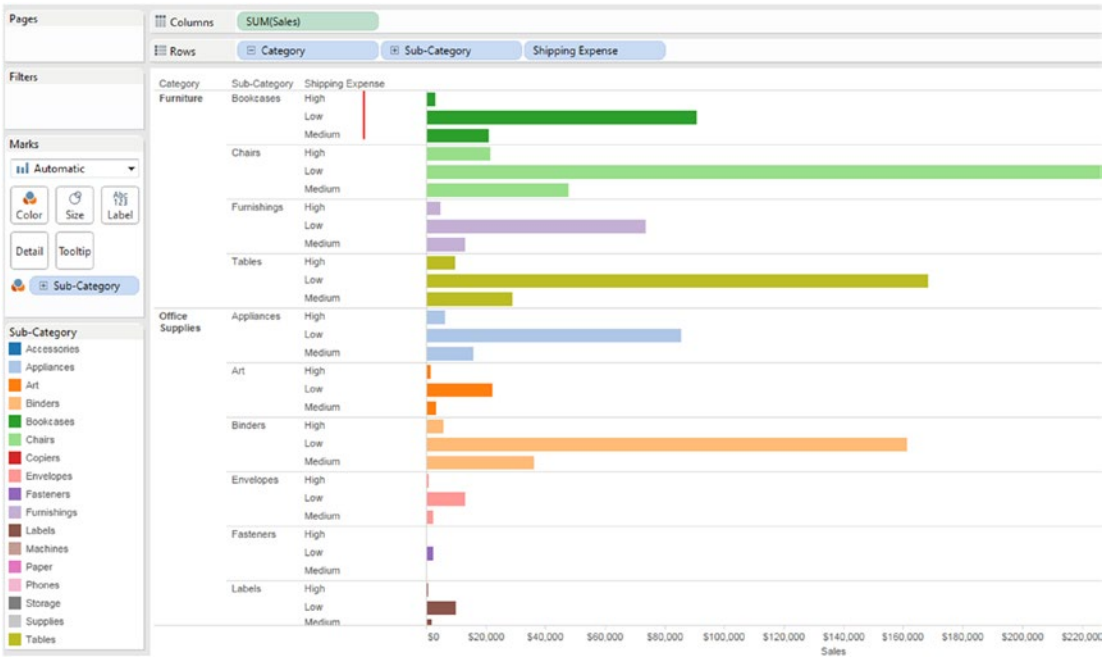


Figure 6-32. *View shows "Shipping Expense" by "Category" and "Sub-Category"*

CASE statement is useful when you need to test a single value. However, it is not suitable for comparison.

6.3.2 IIF() function

Formula: IIF (test, then, else, [unknown])

If the test evaluates to TRUE, then IIF returns the "then" value. If the test evaluates to FALSE, then IIF returns the "else" value.

6.3.2.1 Steps to demonstrate IIF() function

Perform the following steps.

6.3.2.1.1 Step 1

Read the data from "Sample-Superstore" data set.

6.3.2.1.2 Step 2

Construct the view as shown in Fig. 6-33.

Figure 6-33. *Measure "Profit" placed on the rows shelf and Dimension placed on the columns shelf*

6.3.2.1.3 Step 3

Create a calculated field "Profit Category" (Shown in Fig. 6-34).

Figure 6-34. *Calculated field "Profit Category" being created*

6.3.2.1.4 Step 4

Drag the calculated field "Profit Category" to the columns shelf (Shown in Fig. 6-35).

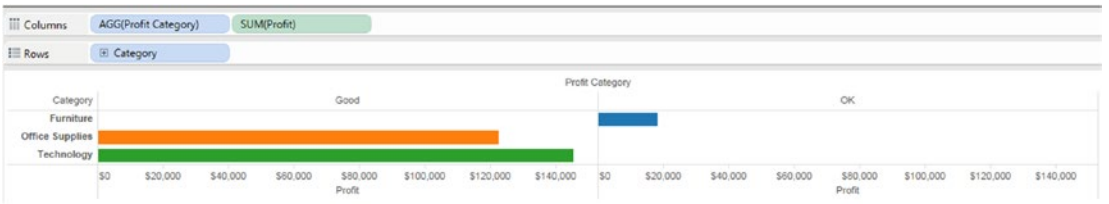


Figure 6-35. *Calculated field "Profit Category" placed on the columns shelf*

6.3.3 IF ELSE

Formula: IF test THEN value ELSE value END

This function evaluates a test condition and returns the THEN value for the condition that evaluates to "True". If no condition evaluates to True, the ELSE value is returned.

6.3.3.1 Steps to demonstrate IF THEN ELSE END

Perform the following steps.

6.3.3.1.1 Step 1

Consider the "Population" data set (Shown in Fig. 6-36).

	A	B
1	State	Population
2	Michigan	98,95,622
3	New Jersey	88,99,339
4	New York	1,96,51,127
5	Ohio	1,15,70,808
6	Illinois	1,28,82,135

Figure 6-36. *"Population" data set*

6.3.3.1.2 Step 2

Read the data from "Population" data set (Shown in Fig. 6-37).

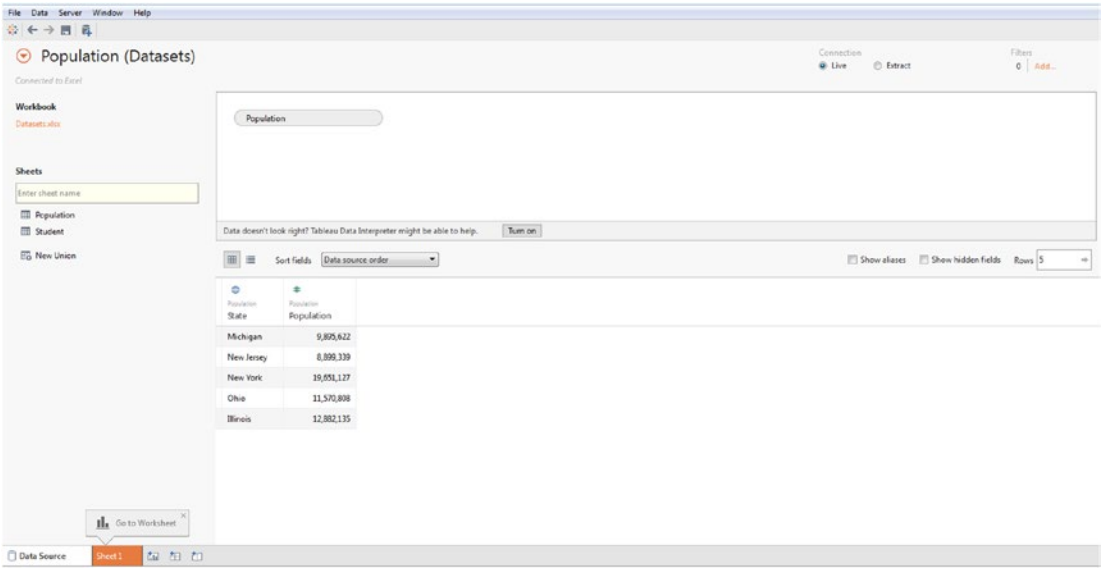


Figure 6-37. *Data source page showing the "Population" data set*

6.3.3.1.3 Step 3

Create a calculated field "Population Category" (Shown in Fig. 6-38).

Figure 6-38. *Calculated field "Population Category" being created*

6.3.3.1.4 Step 4

Drag the calculated field "Population Category" to the rows shelf (Shown in Fig. 6-39).

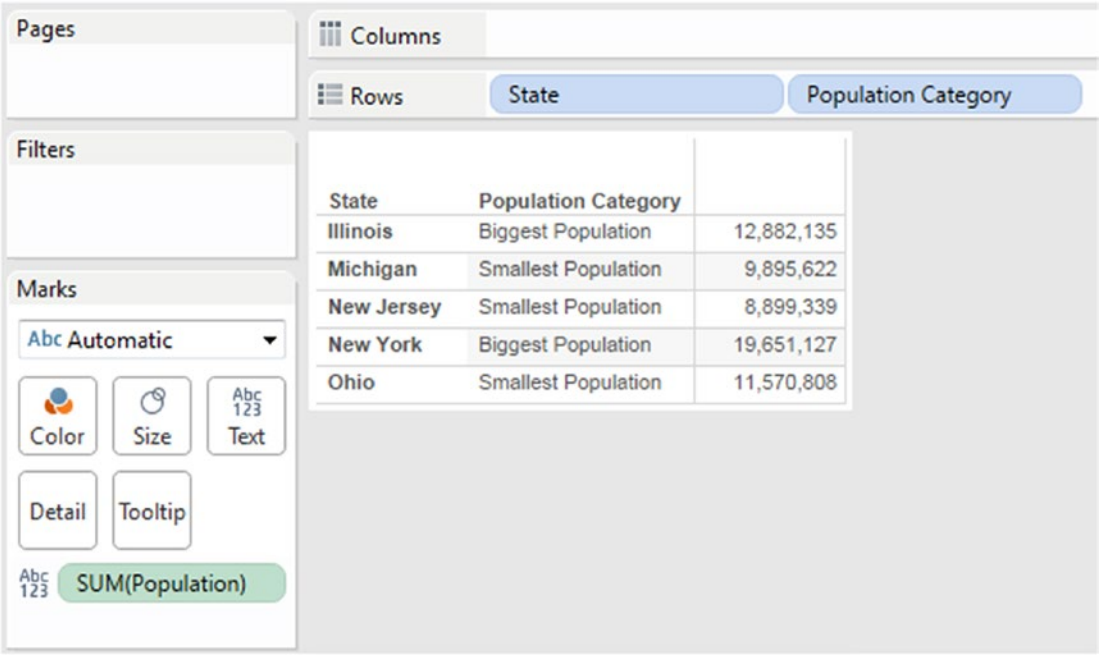


Figure 6-39. *Calculated field "Population Category" placed on the rows shelf*

6.3.4 IF ELSEIF

Formula: IF test THEN value1 ELSEIF test2 THEN value2 ELSE value3 END.

You can use this version of IF function, when you need to perform logical tests recursively.

6.3.4.1 Steps to demonstrate IF ELSEIF

Perform the following steps.

Consider the "Student" data set (Shown in Fig. 6-40).

	A	B	C
1	Stud Name	Mark 1	Mark 2
2	Smith	23	14
3	Jack	18	24
4	John	20	21
5	Scott	22	24
6	James	17	24

Figure 6-40. *"Student" data set*

6.3.4.1.1 Step 1

Read data from “Student” data set (Shown in Fig. 6-41).

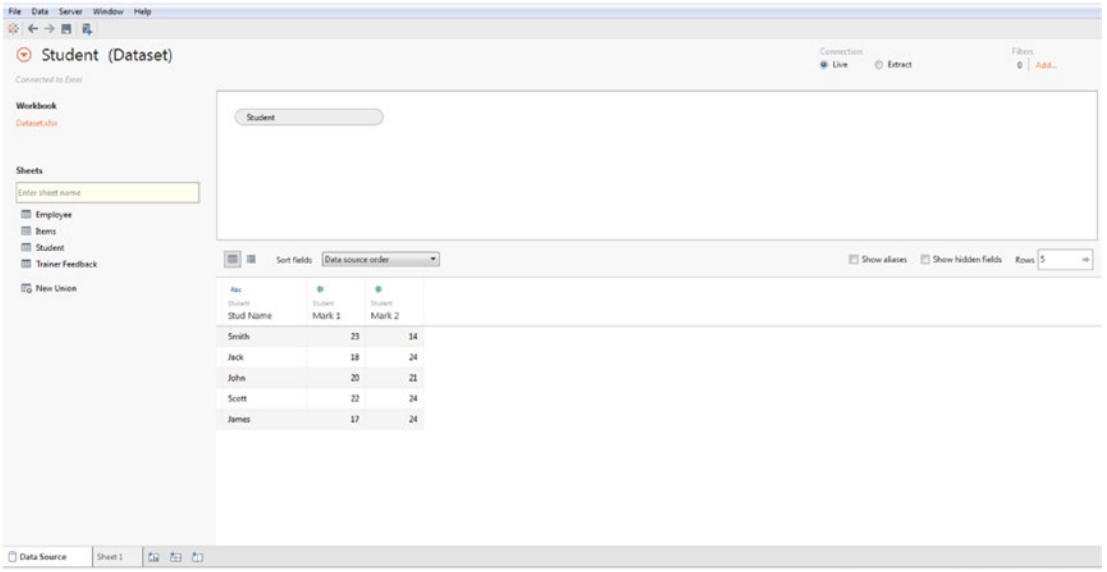


Figure 6-41. *Data source page showing “Student” data set*

6.3.4.1.2 Step 2

Create a view as shown in Fig. 6-42.

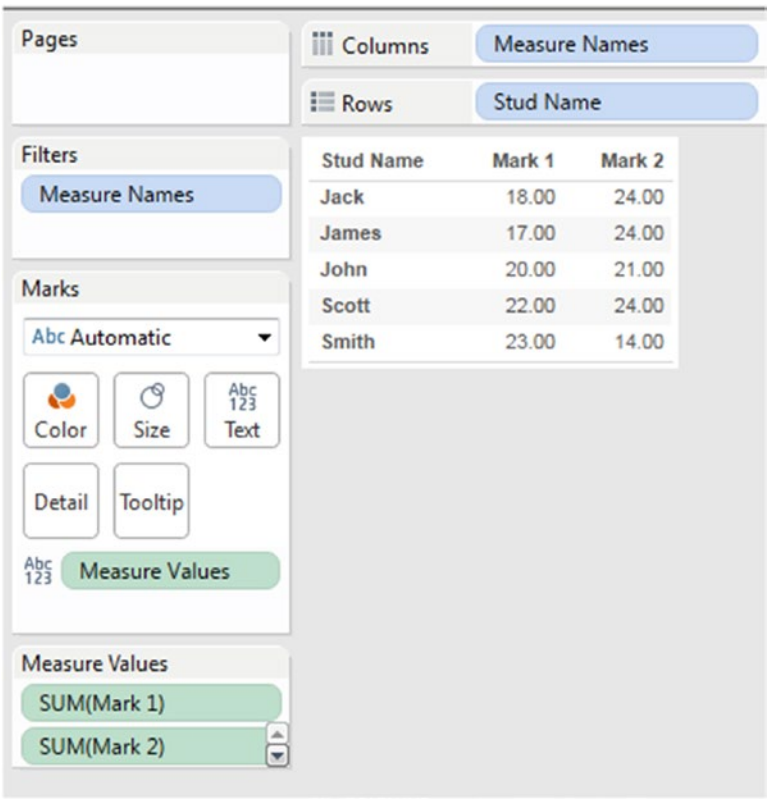


Figure 6-42. *View shows the details of students*

6.3.4.1.3 Step 3

Create a calculated field "Comments" as shown in Fig. 6-43.

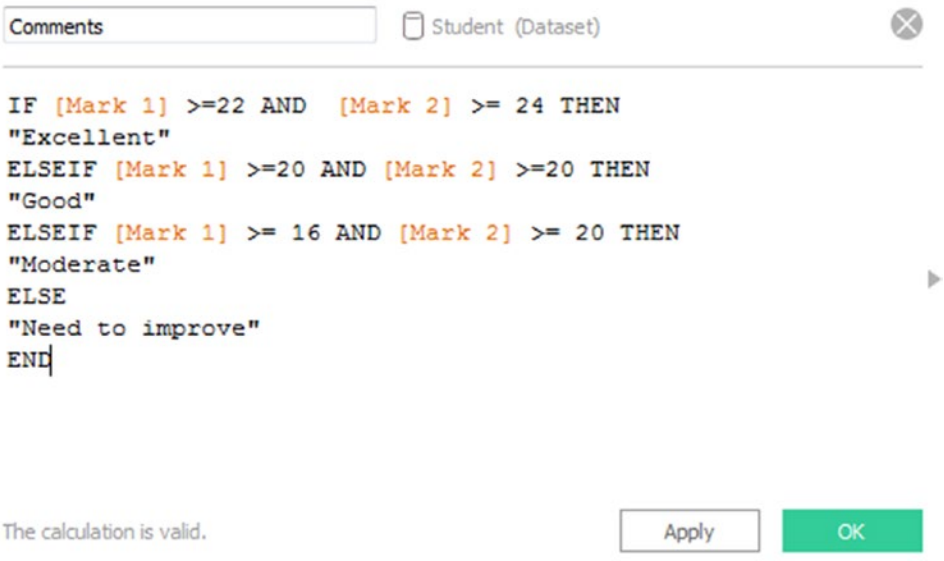


Figure 6-43. *Calculated field "Comments" being created*

6.3.4.1.4 Step 4

Drag the calculated field "Comments" to the rows shelf (Shown in Fig. 6-44).

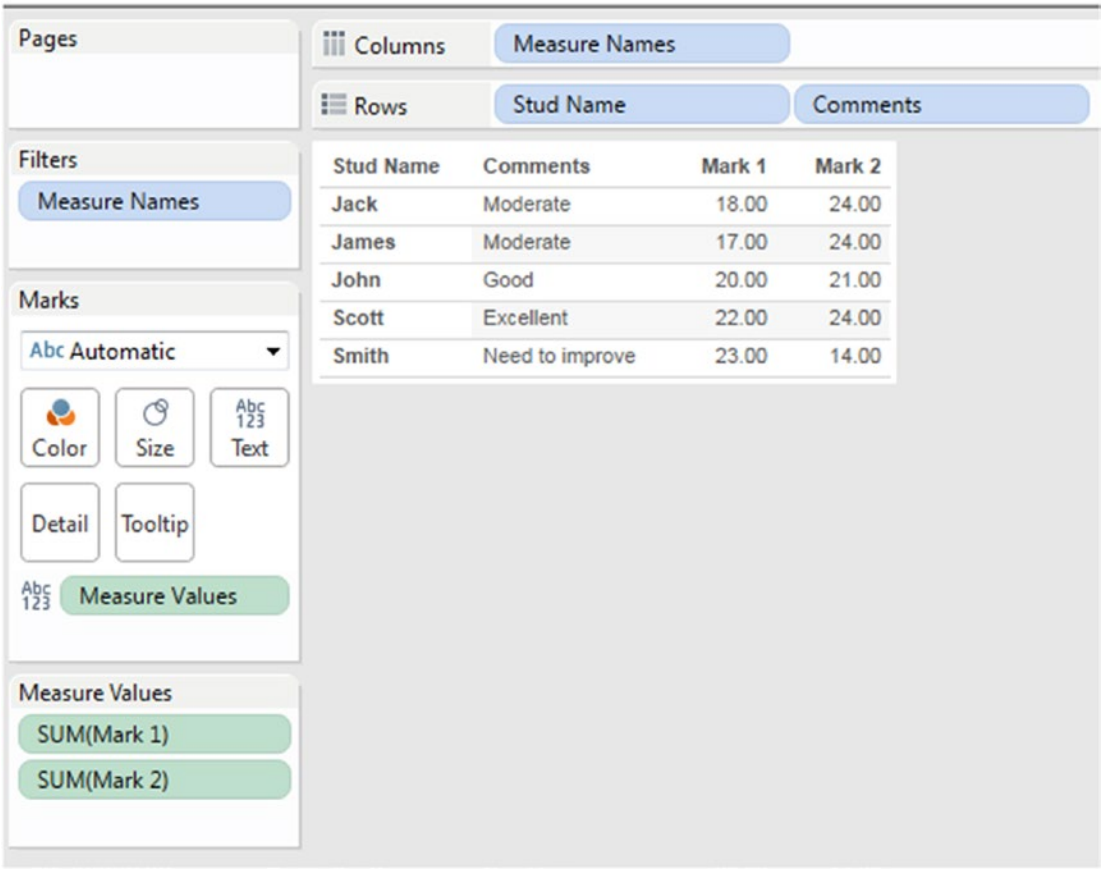


Figure 6-44. *Calculated field "Comments" placed on the rows shelf*

6.4 Date functions

Tableau provides a variety of date functions. Many date functions, use date_part, which is a constant string argument.

Refer to Table 6-3 for date_part and its value.

Table 6-3. *date_part and its value*

date_part	Values
`'year'`	Four-digit year
`'quarter'`	1-4
`'month'`	1-12 or "January", "February", and so on
`'dayofyear'`	Day of the year; Jan 1 is 1, Feb 1 is 2, and so on
`'day'`	1-31
`'weekday'`	1-7 or "Sunday", "Monday", and so on
`'week'`	1-52
`'hour'`	0-23
`'minute'`	0-59
`'second'`	0-59

6.4.1 DATEDIFF()

Formula: DATEDIFF (date_part, date1, date2, [start_of_week])
Returns the difference between `date1` and `date2` , expressed in units of `date_part`.

6.4.1.1 Steps to demonstrate DATEDIFF function

Perform the following steps.

6.4.1.1.1 Step 1

Read data from “Sample-Superstore.xls” data set.

6.4.1.1.2 Step 2

Create a view as shown in Fig. 6-45.

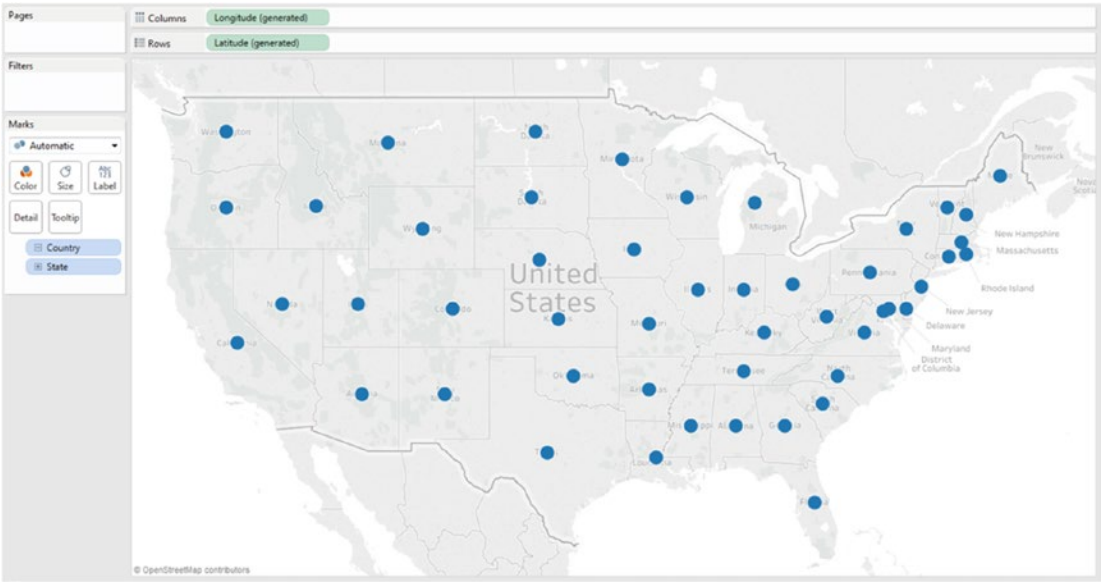


Figure 6-45. *View shows the States of United States*

6.4.1.1.3 Step 3

Create a calculated field "Time to ship" as shown in Fig. 6-46.

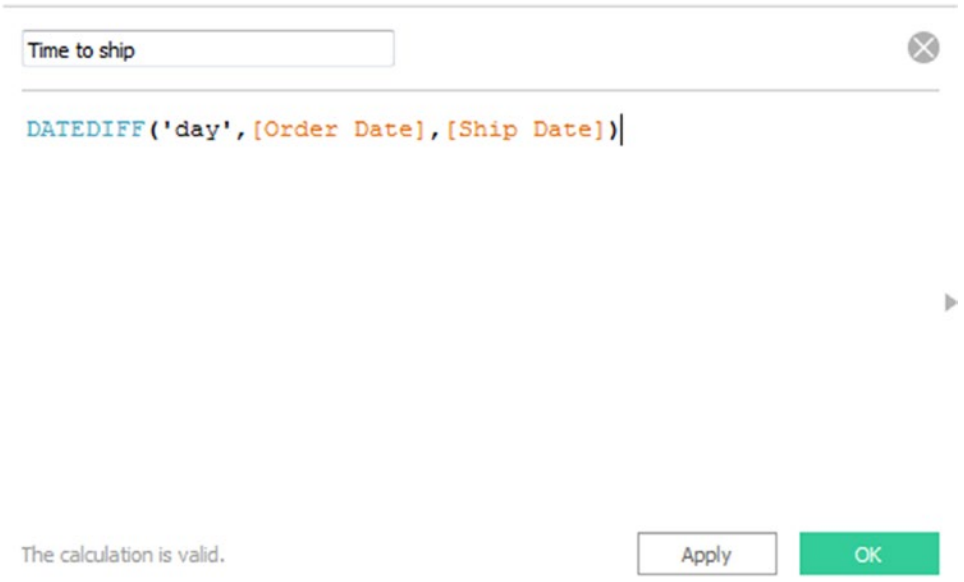


Figure 6-46. *Calculated field "Time to ship"*

6.4.1.1.4 Step 4

Drag the calculated field “Time to ship” to “Color” on the marks card (Shown in Fig. 6-47).

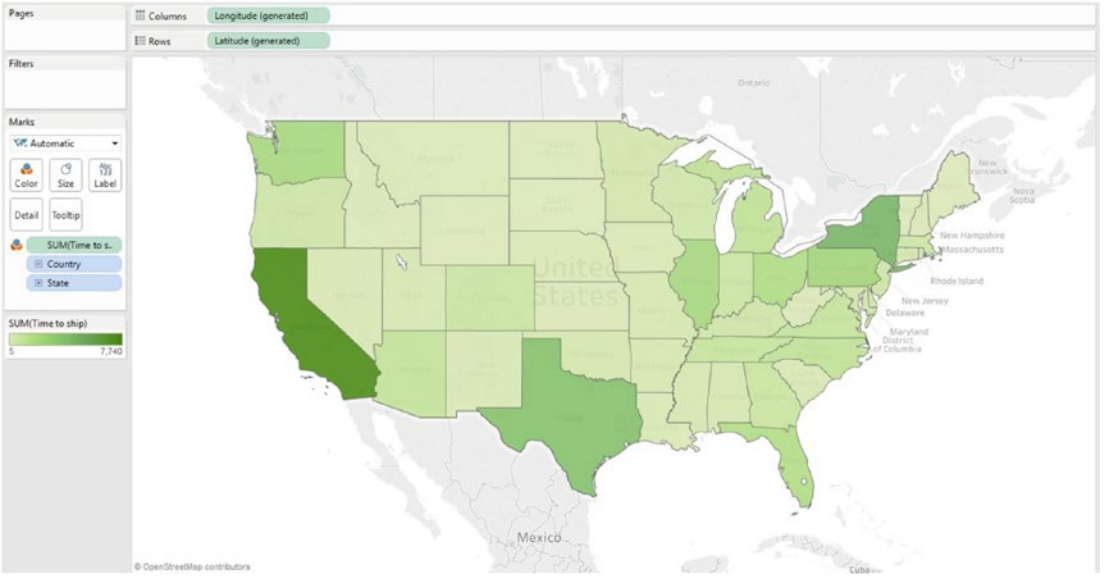


Figure 6-47. *Calculated field “Time to ship” placed on “Color” on the marks card*

6.4.1.1.5 Step 5

Convert the aggregation for the calculated field “Time to ship” from “SUM” to “AVERAGE” (Shown in Fig. 6-48 and Fig. 6-49).

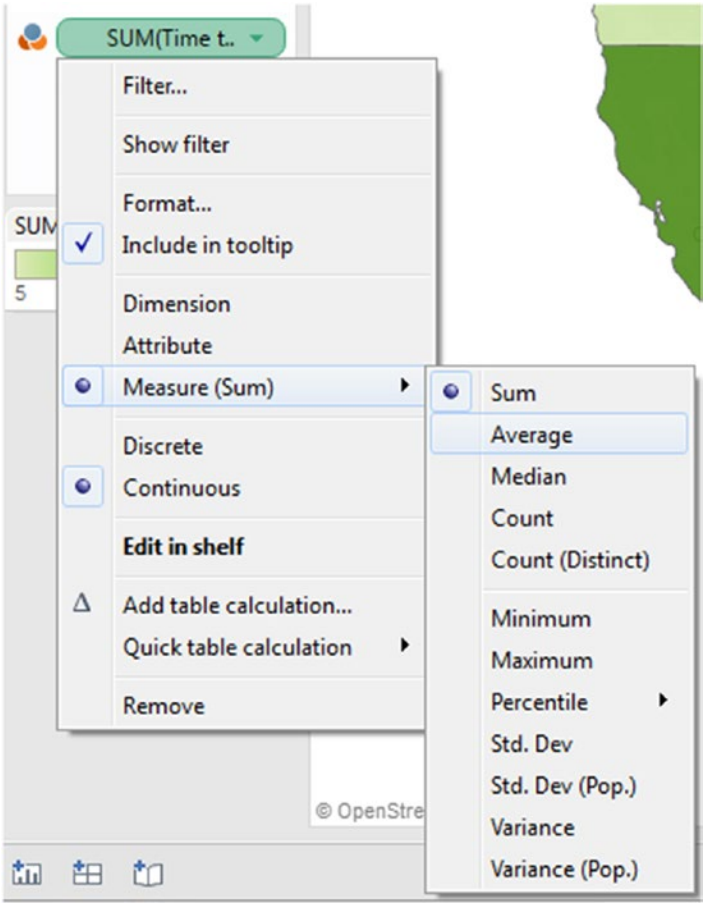


Figure 6-48. *Converting the aggregation for measure, "Time to ship" from "SUM" to "AVERAGE"*

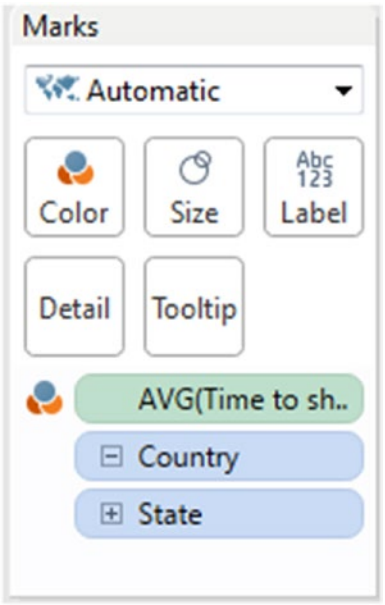


Figure 6-49. *Measure, "Time to ship" placed on "Color" on the marks card*

6.4.1.1.6 Step 6

Right click on "AVG(Time to ship)", select "Edit Colors[Avg. Time to ship] to edit the color (Shown in Fig. 6-50).

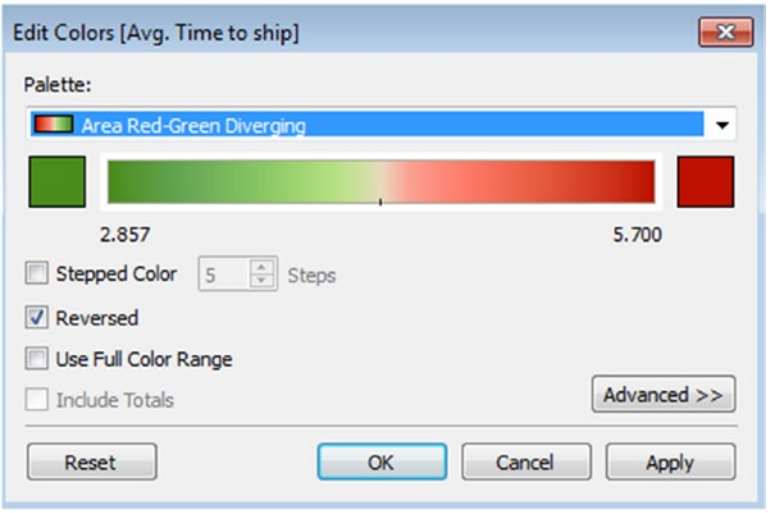


Figure 6-50. *"Edit Colors[Avg. Time to ship]" dialog box*

6.4.1.1.7 Step 7

Place the calculated field “Time to ship” to “Label” on the marks card (Shown in Fig. 6-51).

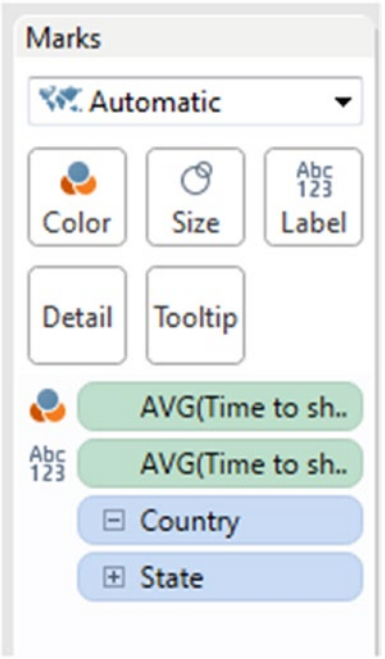


Figure 6-51. *Calculated field “Time to ship” to “Label” on the marks card*

6.4.1.1.8 Step 8

You can see the calculated field “Time to ship” in the visualization (Shown in Fig. 6-52).

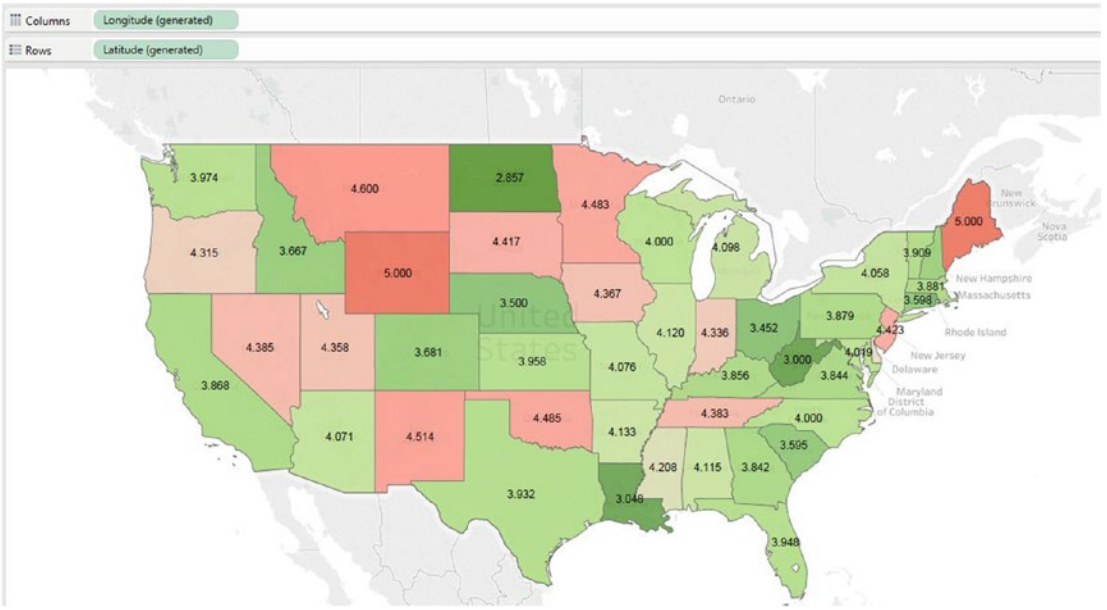


Figure 6-52. *Visualization that shows the calculated field "Time to ship"*

6.4.2 DATEADD() function

Formula: DATEADD (date_part, interval, date)

Returns the specified date with the specified number interval added to the specified date_part of that date.

6.4.2.1 Steps to demonstrate DATEADD() function

Perform the following steps.

6.4.2.1.1 Step 1

Read data from "Sample-Superstore" data set.

6.4.2.1.2 Step 2

Drag the dimension "Order Date" from the dimensions area under the data pane to the rows shelf. Set the hierarchy as "YEAR". Right click on "Order Date", select "Exact Date" as shown in Fig. 6-53. Refer to Fig. 6-54 for output.

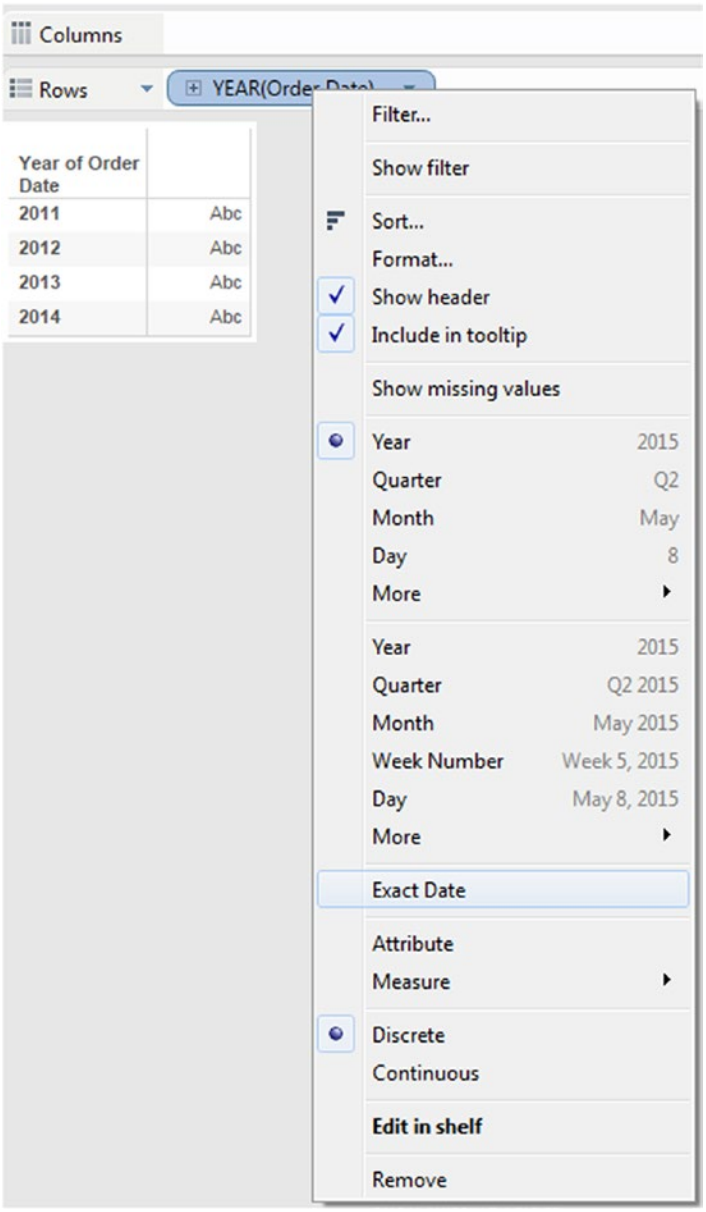


Figure 6-53. *"Exact Date" option*

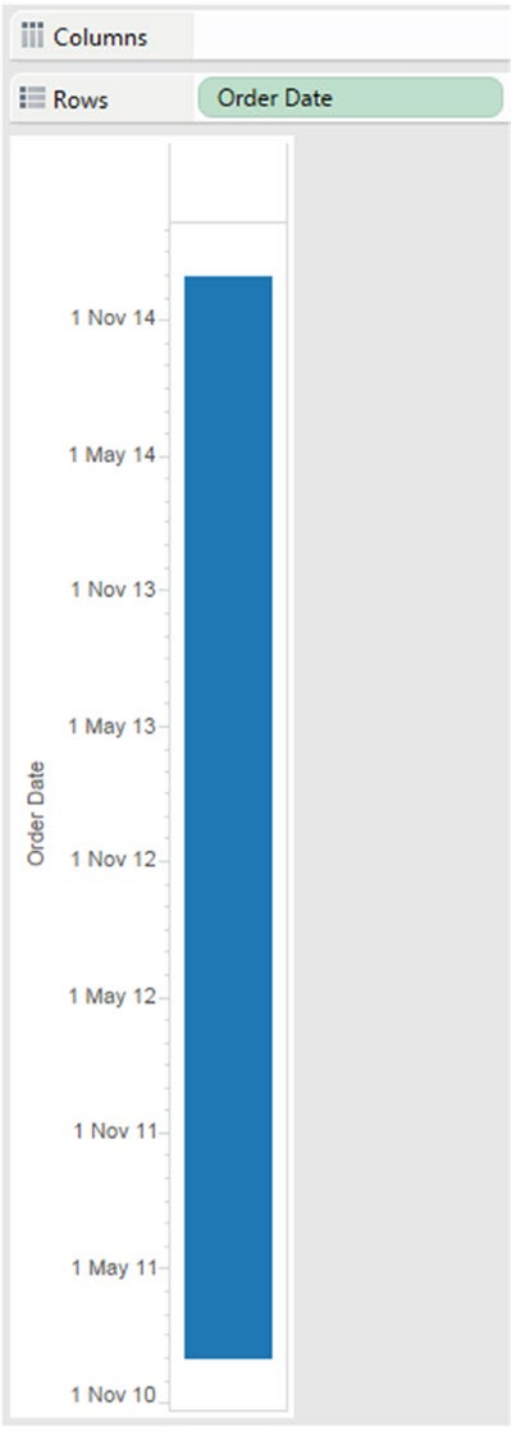


Figure 6-54. *Dimension "Order Date" is set to "Exact Order" date*

6.4.2.1.3 Step 3

Right click on “Order Date” select “Discrete” (Shown in Fig. 6-55). The output is shown in Fig. 6-56.

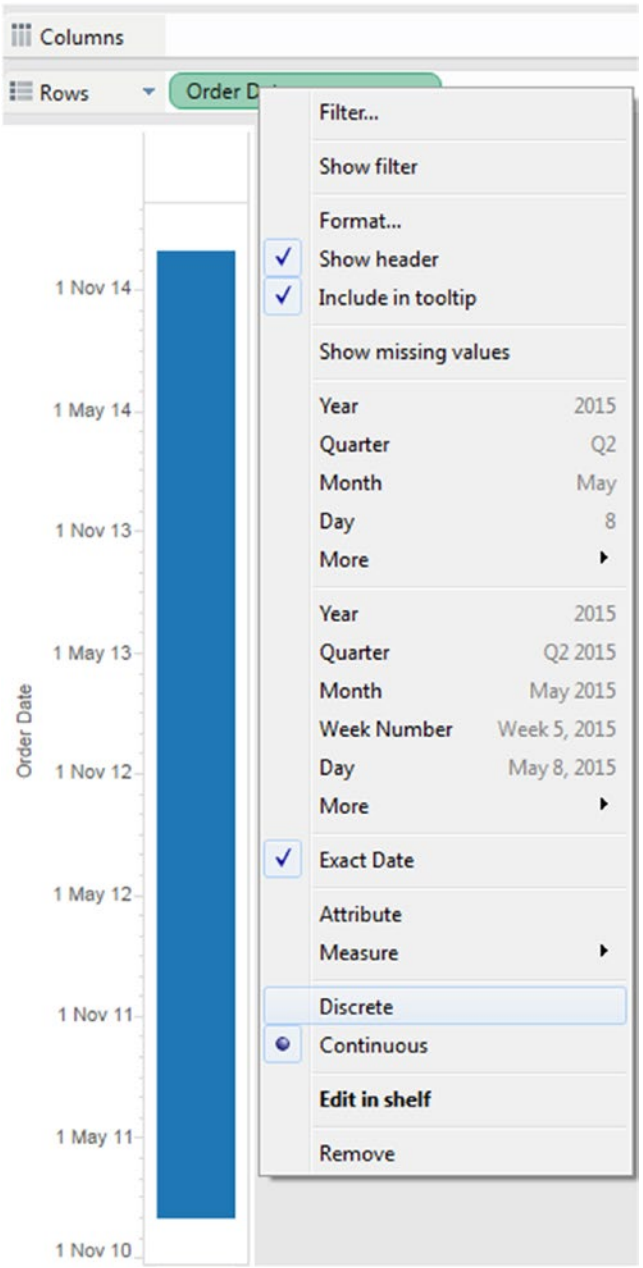


Figure 6-55. *Converting the dimension “Order Date” from “Continuous” to “Discrete”*

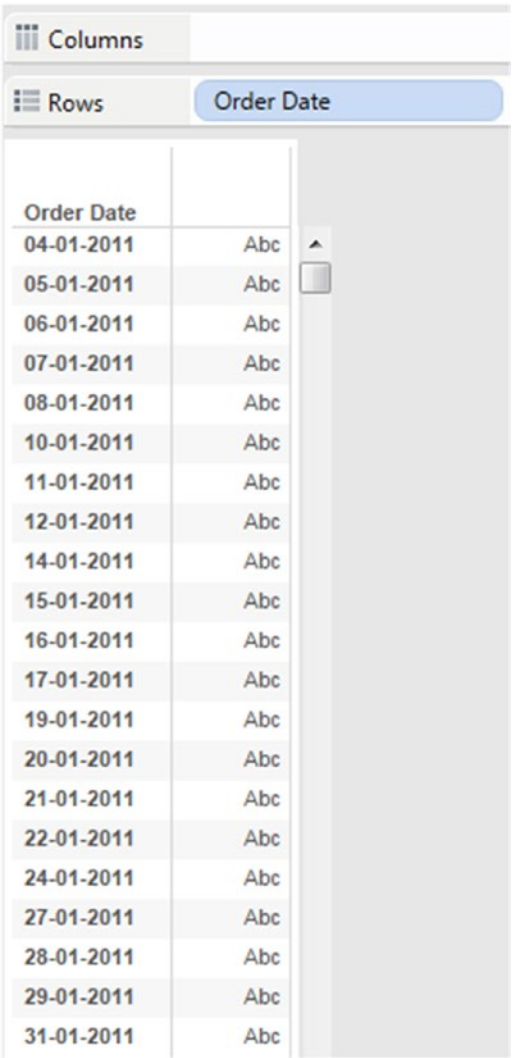


Figure 6-56. *View after setting the "Order Date" to "Discrete"*

6.4.2.1.4 Step 4

Create the calculated field "DATEADD" to add one day to the dimension "Order Date" (Shown in Fig. 6-57).

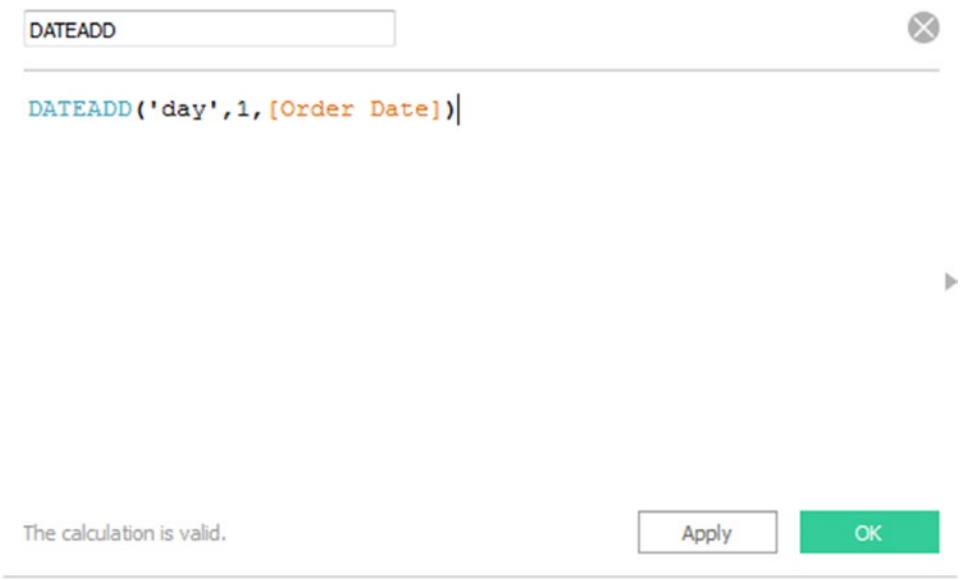
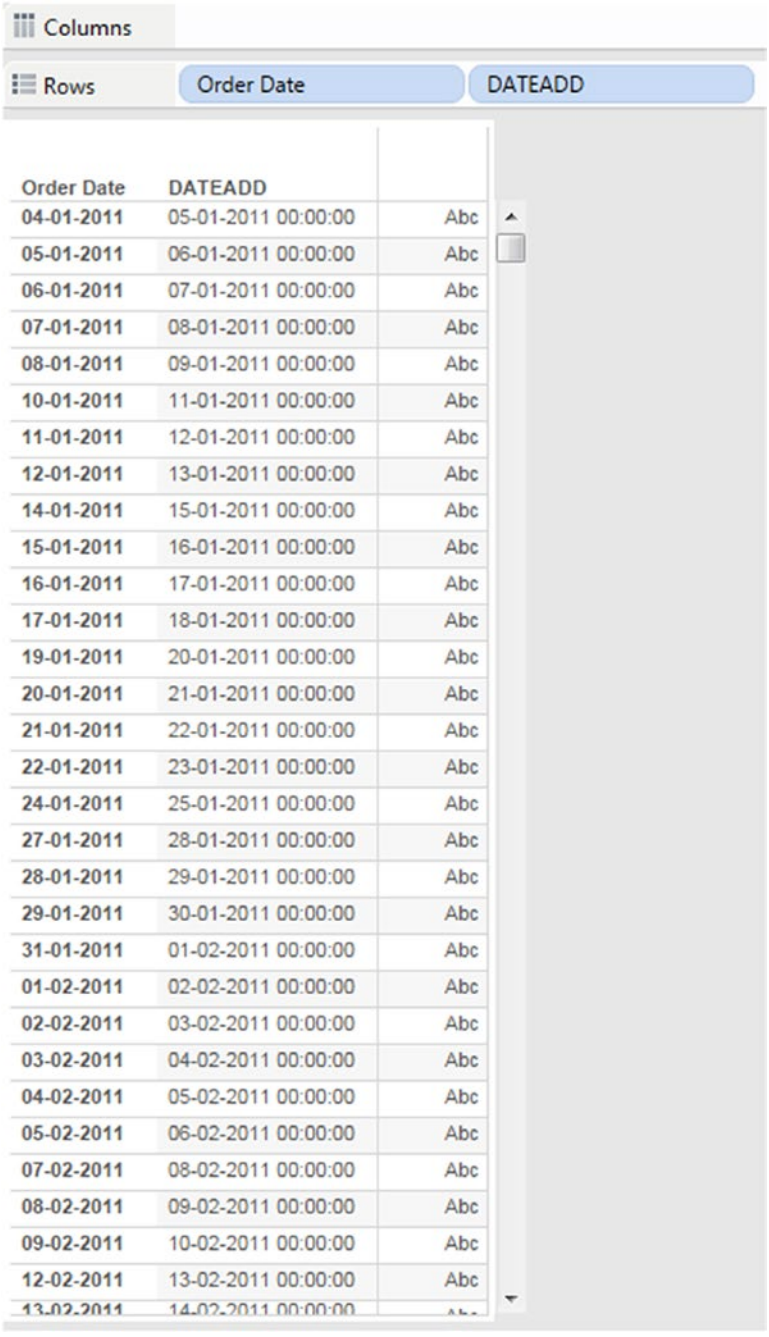


Figure 6-57. *Calculated field "DATEADD" being created*

6.4.2.1.5 Step 5

Drag the calculated field "DATEADD" to the rows shelf (Shown in Fig. 6-58).



Order Date	DATEADD	
04-01-2011	05-01-2011 00:00:00	Abc
05-01-2011	06-01-2011 00:00:00	Abc
06-01-2011	07-01-2011 00:00:00	Abc
07-01-2011	08-01-2011 00:00:00	Abc
08-01-2011	09-01-2011 00:00:00	Abc
10-01-2011	11-01-2011 00:00:00	Abc
11-01-2011	12-01-2011 00:00:00	Abc
12-01-2011	13-01-2011 00:00:00	Abc
14-01-2011	15-01-2011 00:00:00	Abc
15-01-2011	16-01-2011 00:00:00	Abc
16-01-2011	17-01-2011 00:00:00	Abc
17-01-2011	18-01-2011 00:00:00	Abc
19-01-2011	20-01-2011 00:00:00	Abc
20-01-2011	21-01-2011 00:00:00	Abc
21-01-2011	22-01-2011 00:00:00	Abc
22-01-2011	23-01-2011 00:00:00	Abc
24-01-2011	25-01-2011 00:00:00	Abc
27-01-2011	28-01-2011 00:00:00	Abc
28-01-2011	29-01-2011 00:00:00	Abc
29-01-2011	30-01-2011 00:00:00	Abc
31-01-2011	01-02-2011 00:00:00	Abc
01-02-2011	02-02-2011 00:00:00	Abc
02-02-2011	03-02-2011 00:00:00	Abc
03-02-2011	04-02-2011 00:00:00	Abc
04-02-2011	05-02-2011 00:00:00	Abc
05-02-2011	06-02-2011 00:00:00	Abc
07-02-2011	08-02-2011 00:00:00	Abc
08-02-2011	09-02-2011 00:00:00	Abc
09-02-2011	10-02-2011 00:00:00	Abc
12-02-2011	13-02-2011 00:00:00	Abc
13-02-2011	14-02-2011 00:00:00	Abc

Figure 6-58. *Calculated field "DATEADD" placed on the rows shelf*

6.4.2.1.6 Step 6

Right click on the calculated field "DATEADD", select "Format..." to format the date format (Shown in Fig. 6-59).

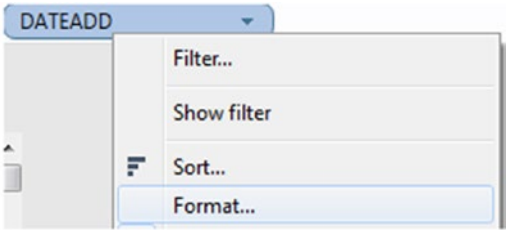


Figure 6-59. *"Format..." option*

6.4.2.1.7 Step 7

"Format Order Date" window opens. Select the format as "Standard Long Date" (Shown in Fig. 6-60).

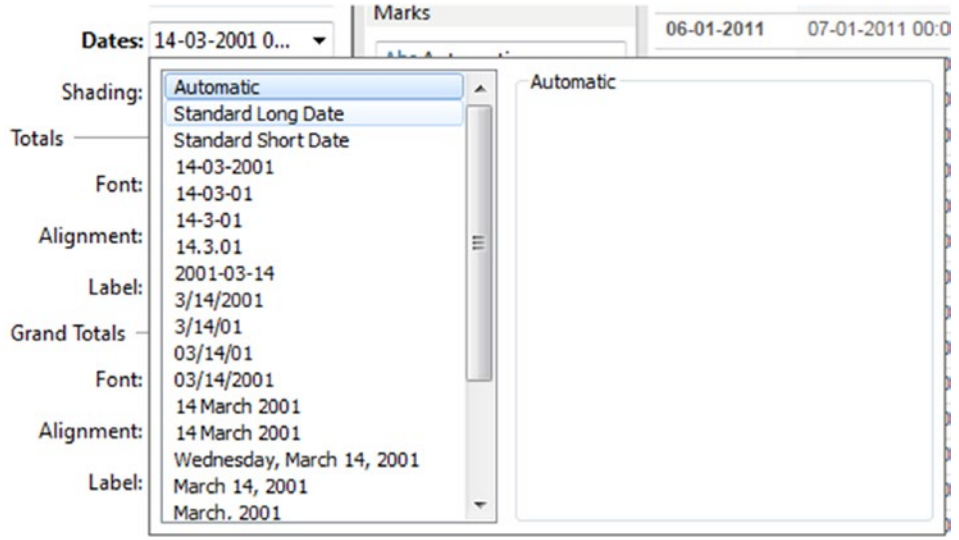


Figure 6-60. *"Format Order Date" Window*

6.4.2.1.8 Step 8

The final output is shown in Fig. 6-61.

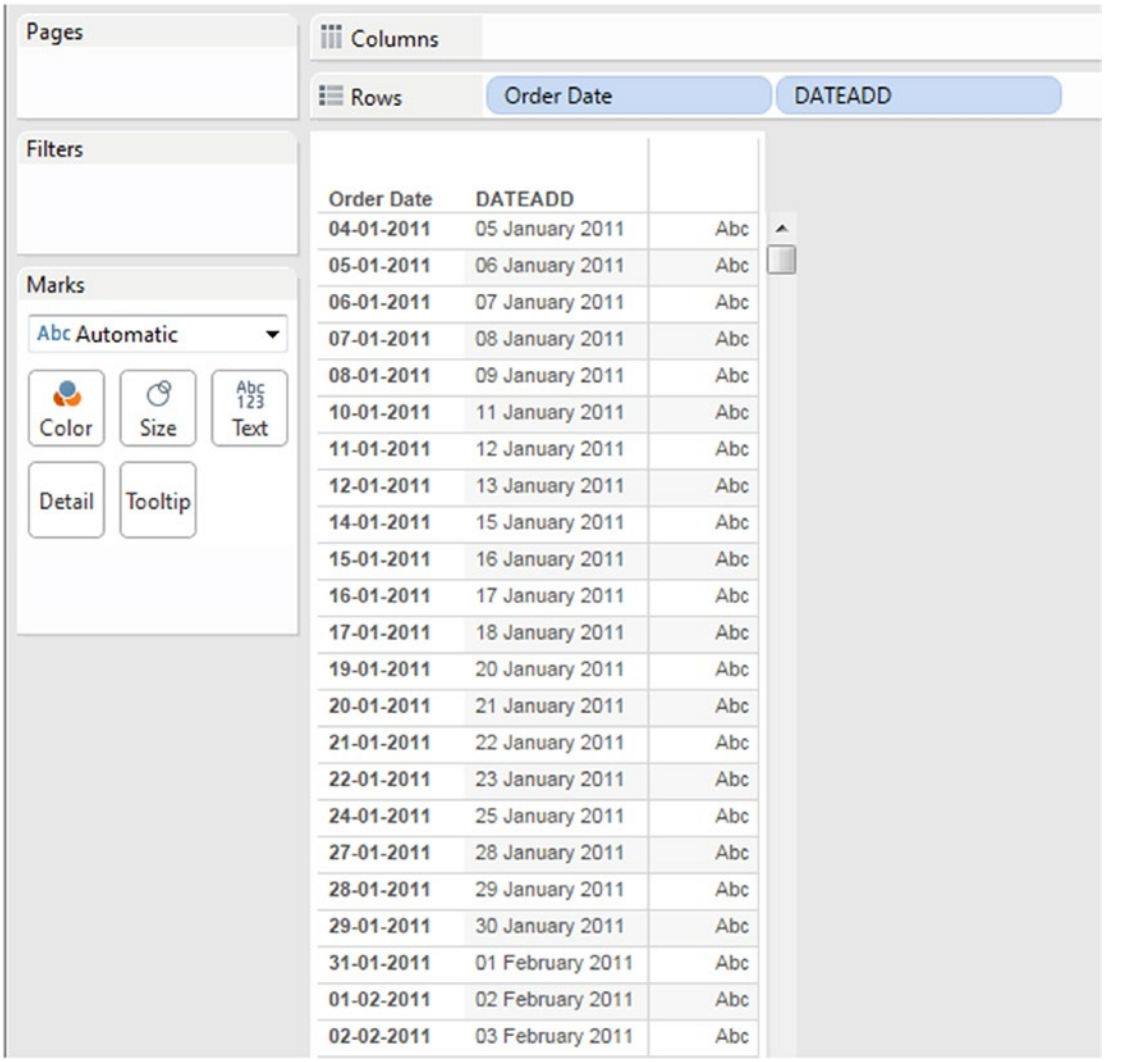


Figure 6-61. *"DATEADD" demonstration - final output*

6.4.3 DATENAME

DATENAME function returns date_part of date as a string.

Formula: DATENAME (date_part, date, [start_of_week])
Returns `date_part` of `date` as a string.

6.4.3.1 Steps to demonstrate DATENAME function

Perform the following steps.

6.4.3.1.1 Step 1

Read data from “Sample-Superstore” data set.

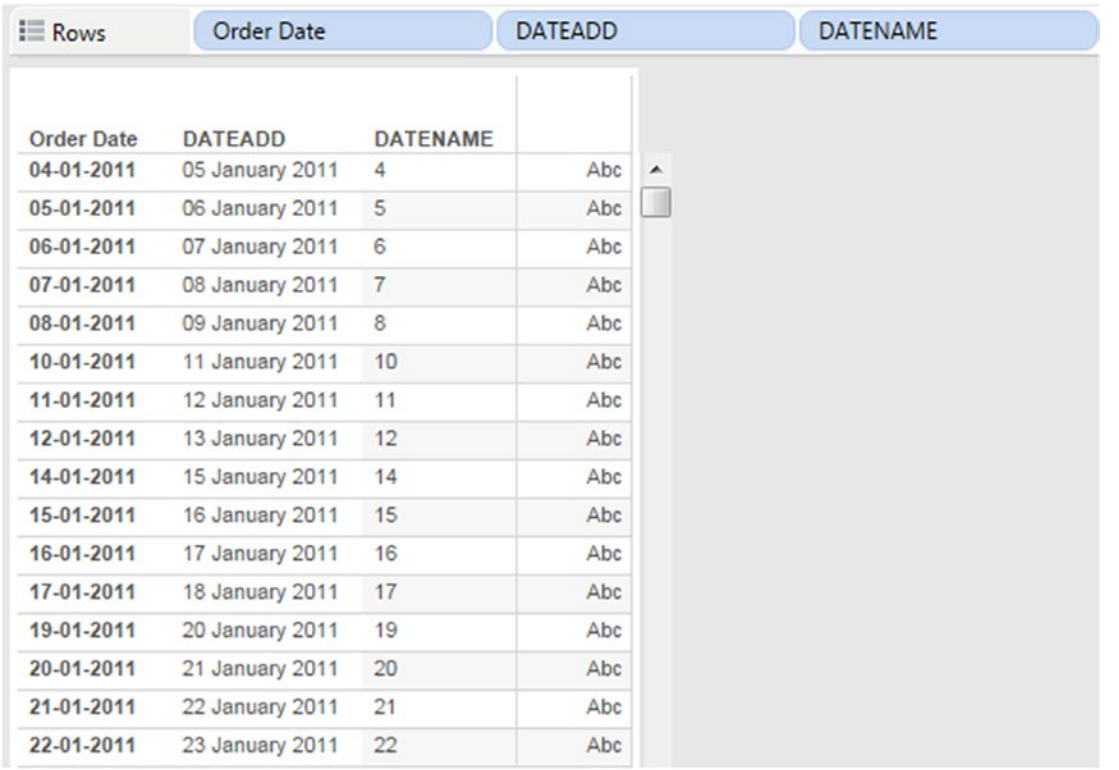
6.4.3.1.2 Step 2

Create a calculated field “DATENAME” as shown in Fig. 6-62.

Figure 6-62. *Calculated field “DATENAME” being created*

6.4.3.1.3 Step 3

Drag the calculated field “DATENAME” to the rows shelf (Shown in Fig. 6-63).



Order Date	DATEADD	DATENAME	
04-01-2011	05 January 2011	4	Abc
05-01-2011	06 January 2011	5	Abc
06-01-2011	07 January 2011	6	Abc
07-01-2011	08 January 2011	7	Abc
08-01-2011	09 January 2011	8	Abc
10-01-2011	11 January 2011	10	Abc
11-01-2011	12 January 2011	11	Abc
12-01-2011	13 January 2011	12	Abc
14-01-2011	15 January 2011	14	Abc
15-01-2011	16 January 2011	15	Abc
16-01-2011	17 January 2011	16	Abc
17-01-2011	18 January 2011	17	Abc
19-01-2011	20 January 2011	19	Abc
20-01-2011	21 January 2011	20	Abc
21-01-2011	22 January 2011	21	Abc
22-01-2011	23 January 2011	22	Abc

Figure 6-63. *Calculated field "DATENAME" placed on the rows shelf*

To learn more about date refer to the link below.

http://onlinehelp.tableau.com/current/pro/desktop/en-us/functions_functions_date.html

---

6.5 Aggregate functions

Tableau has support for various aggregate functions.

Let us discuss ATTR function.

6.5.1 ATTR(expression)

The ATTR() function returns the value of the expression if it has a single value for all rows. Otherwise returns an asterisk. Null values are ignored.

6.5.1.1 Steps to demonstrate the use of ATTR() function

Perform the following steps.

6.5.1.1.1 Step 1

Specify the Sum([Profit])/Sum[Sales]) calculation on the columns shelf (Shown in Fig. 6-64).

Figure 6-64. *Calculation placed on the columns shelf*

6.5.1.1.2 Step 2

The view shows that aggregation is part of calculation (Shown in Fig. 6-65).

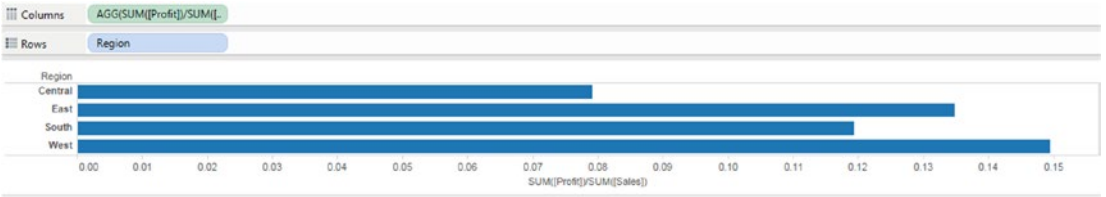


Figure 6-65. *Aggregation is part of calculation*

6.5.1.1.3 Step 3

Create a calculated field “Region Sales” (Shown in Fig. 6-66). This calculation is to find “AVERAGE” sales for “South” and “West” Region and “MEDIAN” sales for other regions.

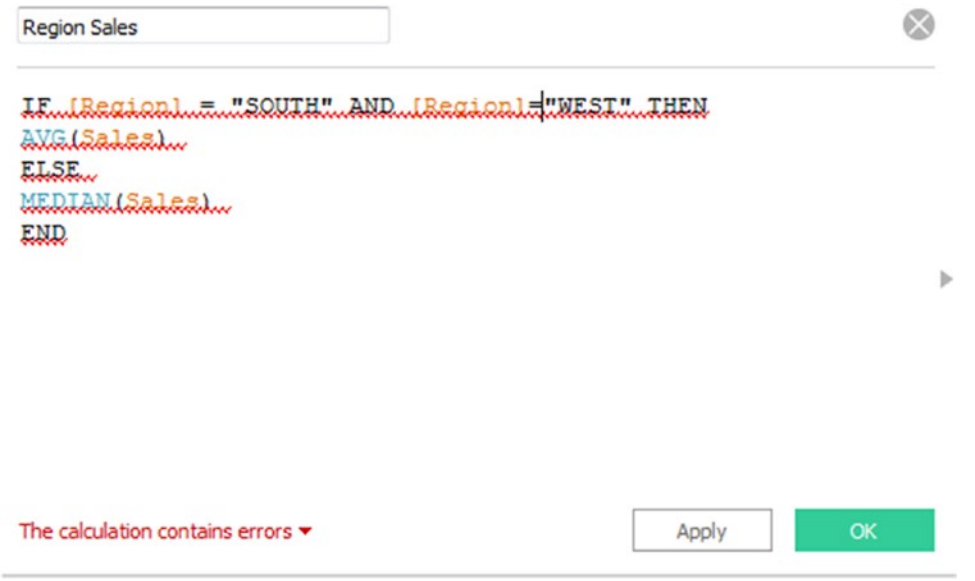


Figure 6-66. *Calculated field "Region Sales" being created. Observe that the calculation contains errors*

We cannot mix aggregate and non-aggregate functions in an expression. In such a situation, we can use ATTR function (Shown in Fig. 6-67).

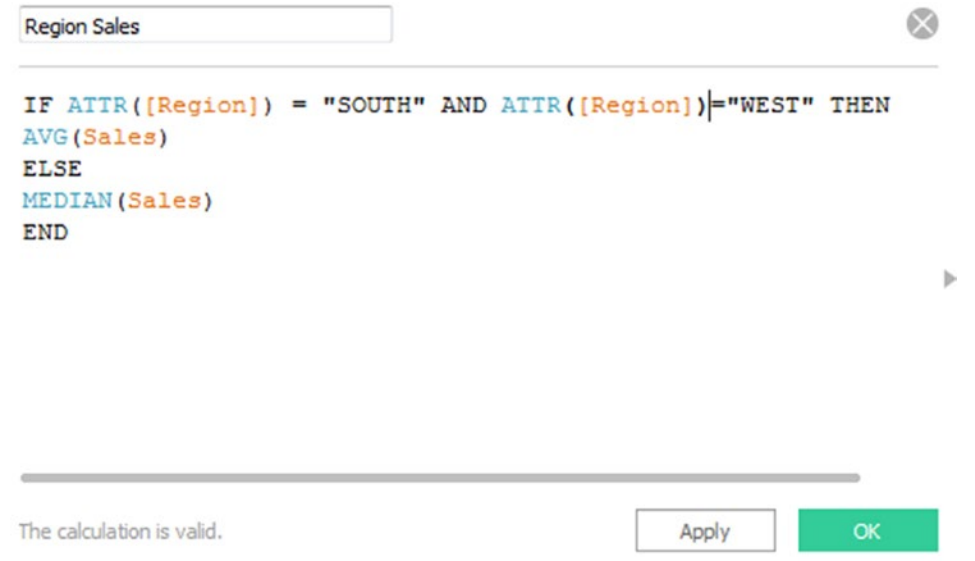


Figure 6-67. *ATTR() function being used with calculated field "Region Sales"*

6.5.1.1.4 Step 4

The final output is shown in Fig. 6-68.

Figure 6-68. *ATTR() function - final output*

Attribute checks whether there is only one value for a given field for all rows in the result set. If there is only one value for the data selected, ATTR() function returns that value. If there is more than one value for that subset of data, it returns an asterisk.

Tableau computes attributes using the following formula:

IF MIN([dimension]) = MAX([dimension]) THEN MIN([dimension]) ELSE "*" END

6.6 Table calculation functions

Table calculations are computations that are applied to the entire table. Basically, table calculations are applied to values that come back from the database at some aggregation level.

Let us explore few table calculation functions.

6.6.1 First(), Index()

First()

This function returns the number of rows from the current row to the first row in the partition.

Index()

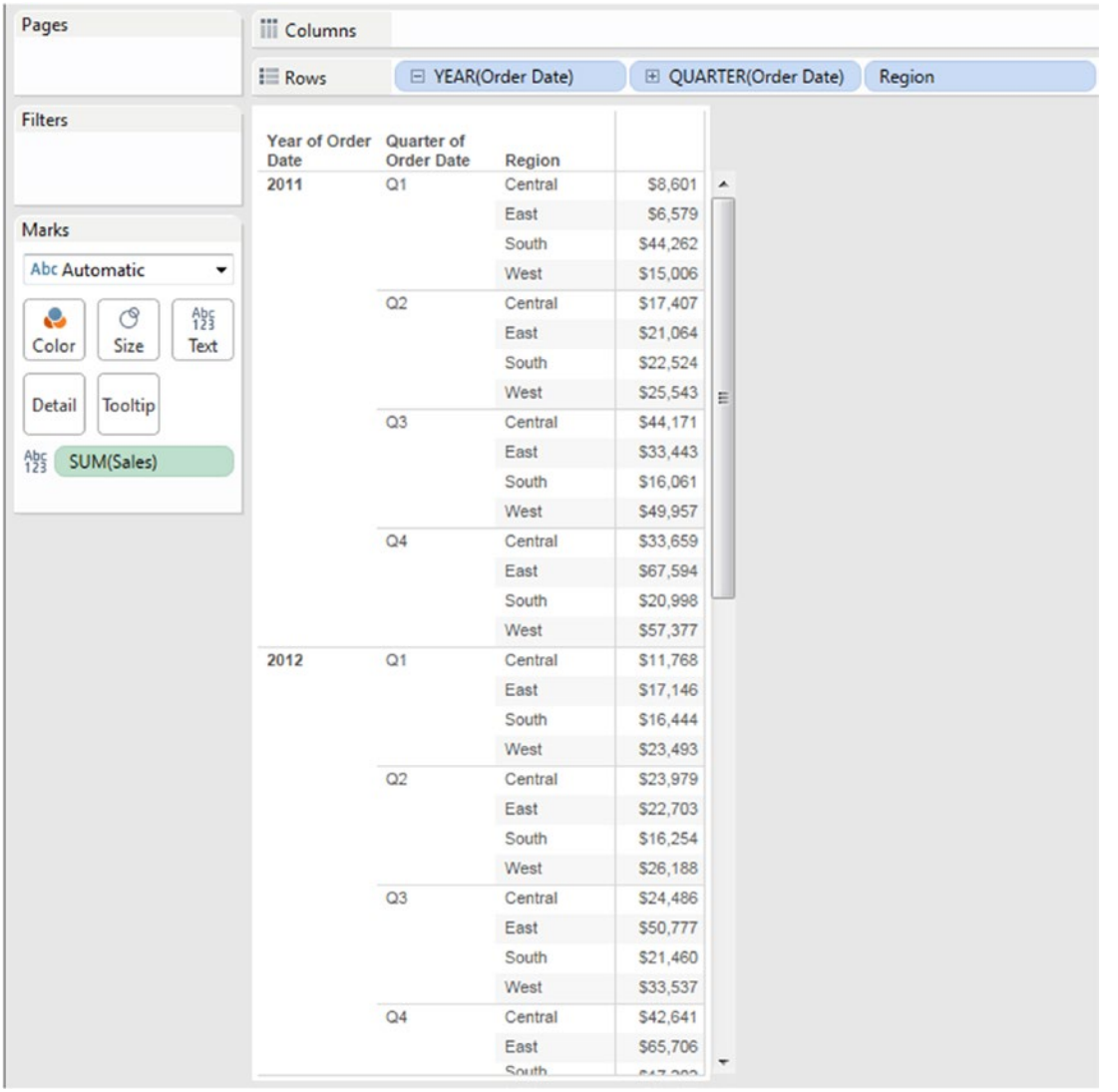
This function returns the index of the current row in the partition, without any sorting concerning the value. The index value starts at 1.

6.6.1.1 Steps to demonstrate the table calculations

Perform the following steps.

6.6.1.1.1 Step 1

Construct a view as shown in Fig. 6-69.



Year of Order Date	Quarter of Order Date	Region	
2011	Q1	Central	$8,601
		East	$6,579
		South	$44,262
		West	$15,006
	Q2	Central	$17,407
		East	$21,064
		South	$22,524
		West	$25,543
	Q3	Central	$44,171
		East	$33,443
		South	$16,061
		West	$49,957
	Q4	Central	$33,659
		East	$67,594
		South	$20,998
		West	$57,377
2012	Q1	Central	$11,768
		East	$17,146
		South	$16,444
		West	$23,493
	Q2	Central	$23,979
		East	$22,703
		South	$16,254
		West	$26,188
	Q3	Central	$24,486
		East	$50,777
		South	$21,460
		West	$33,537
	Q4	Central	$42,641
		East	$65,706
		South	

Figure 6-69. *View shows "Sales" by "Quarter" for each "Region"*

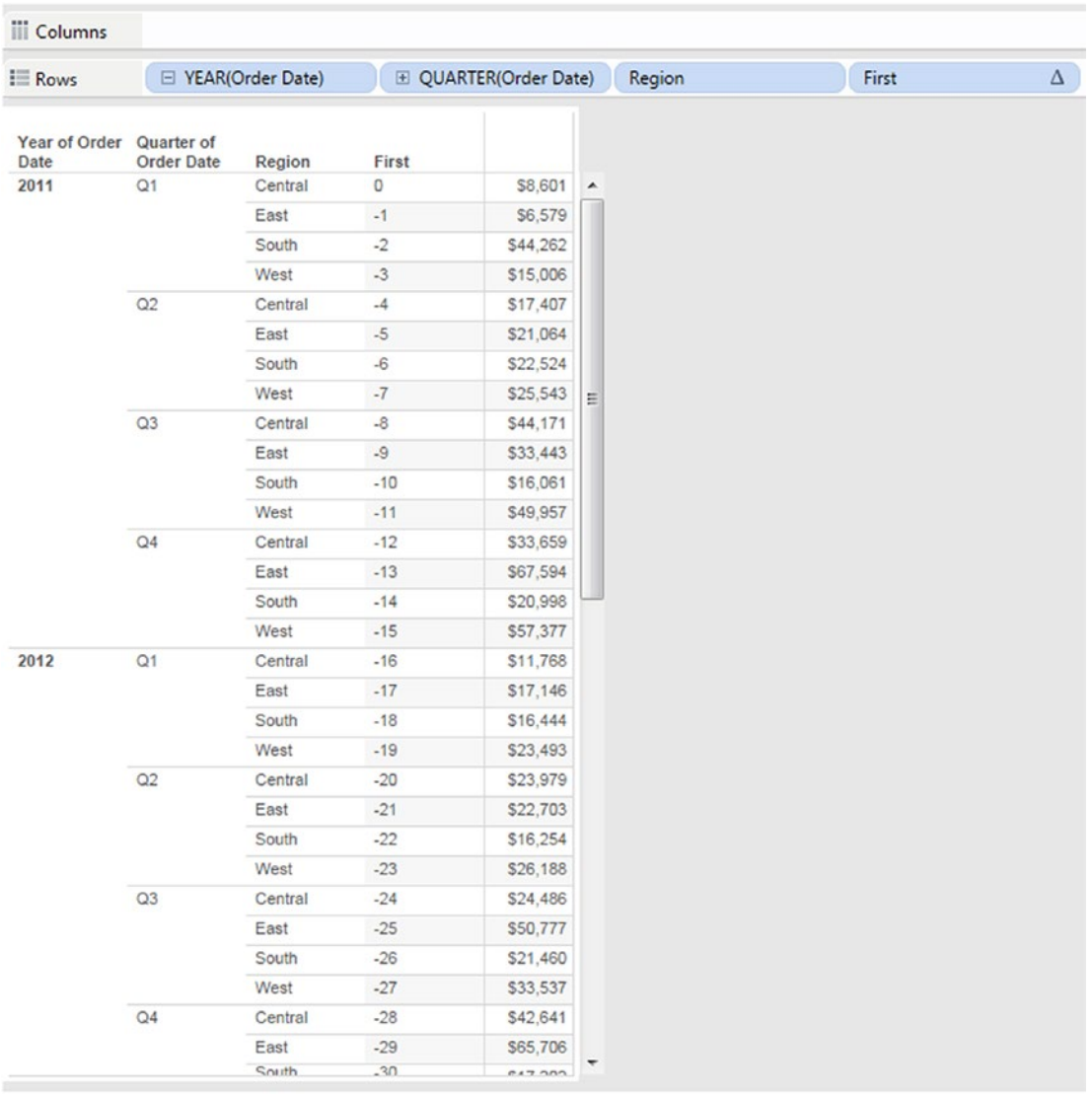
6.6.1.1.2 Step 2

Create a calculated field "First" as shown in Fig. 6-70.

Figure 6-70. *Calculated field "First" being created*

6.6.1.1.3 Step 3

Drag the calculated field "First" to the rows shelf (Shown in Fig. 6-71).



Year of Order Date	Quarter of Order Date	Region	First	
2011	Q1	Central	0	$8,601
		East	-1	$6,579
		South	-2	$44,262
		West	-3	$15,006
	Q2	Central	-4	$17,407
		East	-5	$21,064
		South	-6	$22,524
		West	-7	$25,543
	Q3	Central	-8	$44,171
		East	-9	$33,443
		South	-10	$16,061
		West	-11	$49,957
	Q4	Central	-12	$33,659
		East	-13	$67,594
		South	-14	$20,998
		West	-15	$57,377
2012	Q1	Central	-16	$11,768
		East	-17	$17,146
		South	-18	$16,444
		West	-19	$23,493
	Q2	Central	-20	$23,979
		East	-21	$22,703
		South	-22	$16,254
		West	-23	$26,188
	Q3	Central	-24	$24,486
		East	-25	$50,777
		South	-26	$21,460
		West	-27	$33,537
	Q4	Central	-28	$42,641
		East	-29	$65,706
		South	-30	

Figure 6-71. *Calculated field "First" placed on the rows shelf*

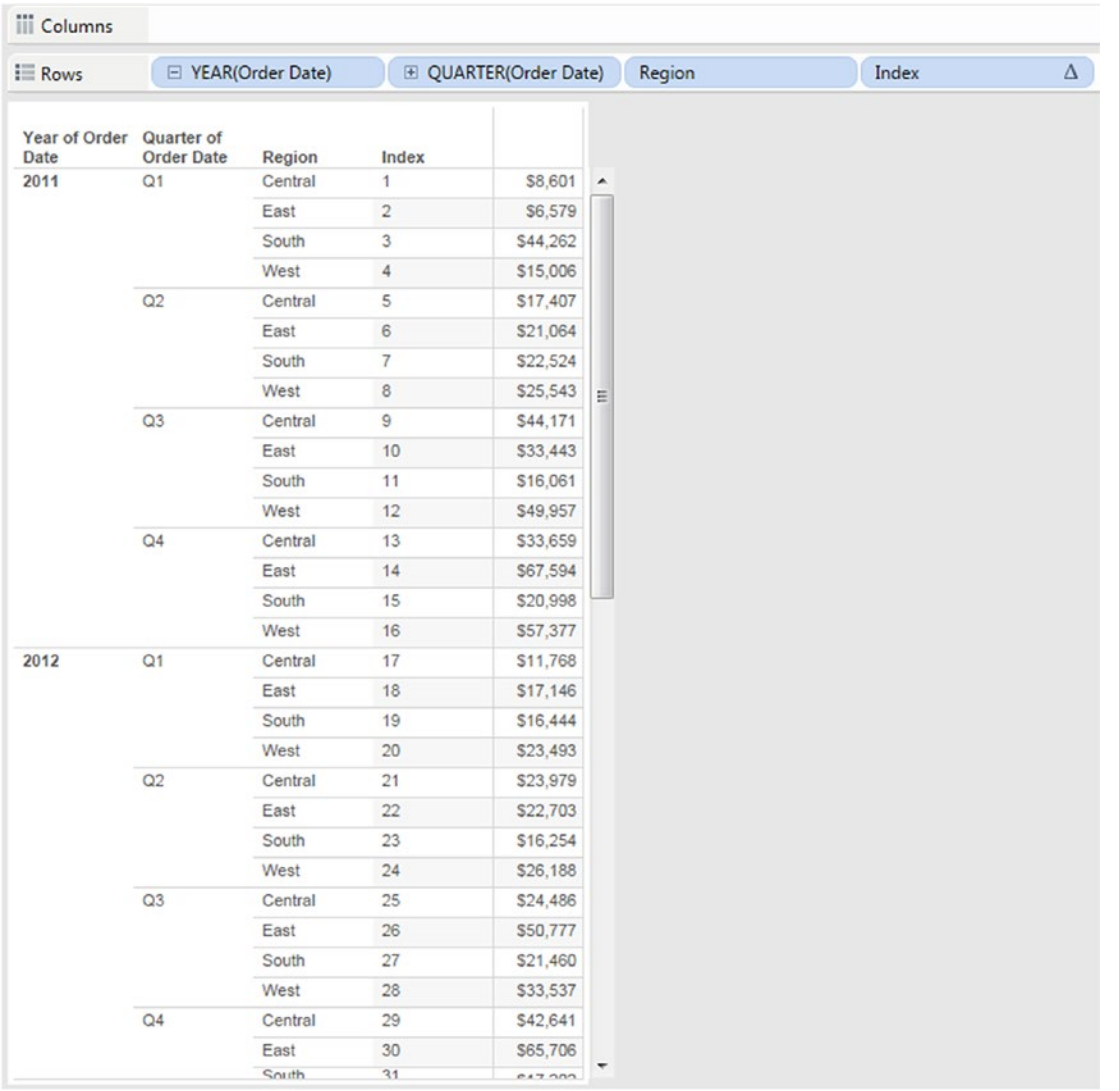
6.6.1.1.4 Step 4

Create a calculated field "Index" as shown in Fig. 6-72.

Figure 6-72. *Calculated field "Index" being created*

6.6.1.1.5 Step 5

Drag calculated field "Index" to the rows shelf (Shown in Fig. 6-73).



Year of Order Date	Quarter of Order Date	Region	Index	
2011	Q1	Central	1	$8,601
		East	2	$6,579
		South	3	$44,262
		West	4	$15,006
	Q2	Central	5	$17,407
		East	6	$21,064
		South	7	$22,524
		West	8	$25,543
	Q3	Central	9	$44,171
		East	10	$33,443
		South	11	$16,061
		West	12	$49,957
	Q4	Central	13	$33,659
		East	14	$67,594
		South	15	$20,998
		West	16	$57,377
2012	Q1	Central	17	$11,768
		East	18	$17,146
		South	19	$16,444
		West	20	$23,493
	Q2	Central	21	$23,979
		East	22	$22,703
		South	23	$16,254
		West	24	$26,188
	Q3	Central	25	$24,486
		East	26	$50,777
		South	27	$21,460
		West	28	$33,537
	Q4	Central	29	$42,641
		East	30	$65,706
		South	31	

Figure 6-73. *Calculated field "Index" placed on the rows shelf*

6.7 Points to remember

- Tableau supports a variety of functions such as number, string, logical and date functions, etc.
- Tableau also has support for user functions, type conversions, etc.
- Tableau computes attribute using the following formula:

IF MIN([dimension]) = MAX([dimension]) THEN MIN([dimension]) ELSE "*" END.

6.8 Next steps

In the next chapter, we will learn statistics. We will introduce the following:

- Basics of statistics
- Five magic number summary
- Box plot
- Statistics tools in Tableau
- Forecasting