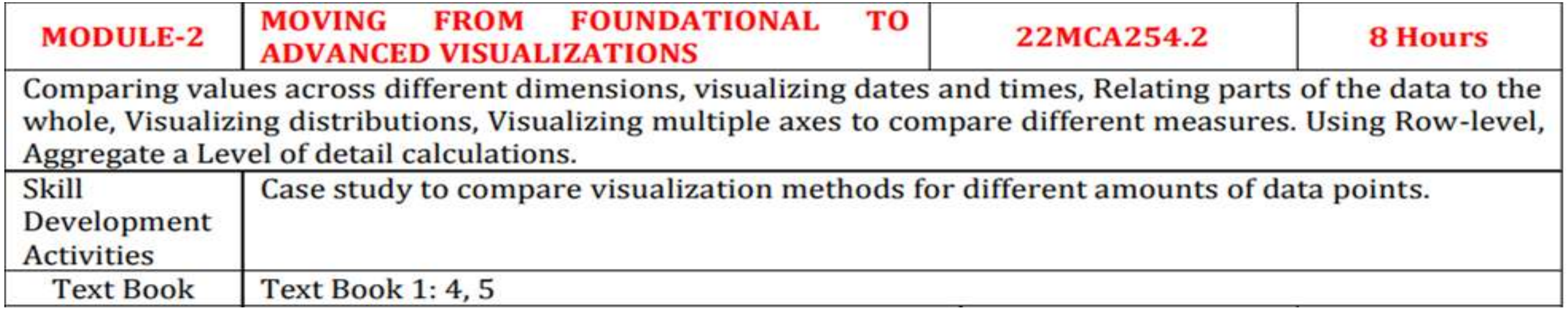
BUSINESS INTELLIGENCE AND DATA ANALYTICS

22MCA254

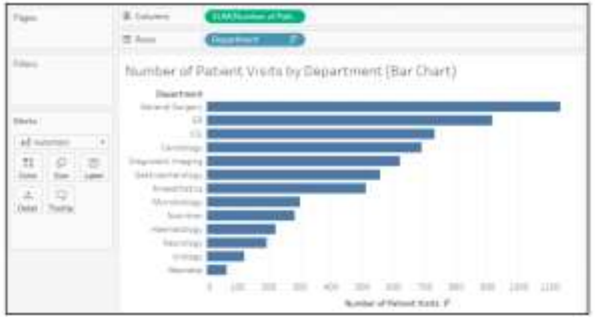
MODULE 2



MODULE-2	MOVING FROM FOUNDATIONAL TO ADVANCED VISUALIZATIONS	22MCA254.2	8 Hours
Comparing values across different dimensions, visualizing dates and times, Relating parts of the data to the whole, Visualizing distributions, Visualizing multiple axes to compare different measures. Using Row-level, Aggregate a Level of detail calculations.			
Skill Development Activities	Case study to compare visualization methods for different amounts of data points.		
Text Book	Text Book 1: 4, 5		

Comparing values

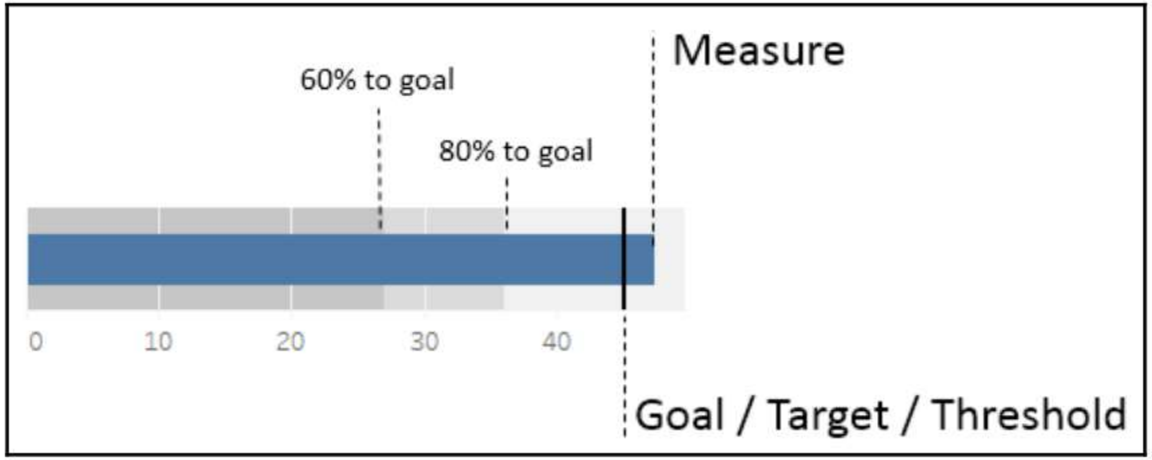
- compare the differences between measured values across different categories.
- Simple Bar Chart



- A basic bar chart can be extended in many ways to accomplish various objectives.
  - Bullet chart to show progress toward a goal, target, or threshold
  - Bar-in-bar chart to show progress toward a target or compare two specific values within a category
  - Highlighting categories of interest

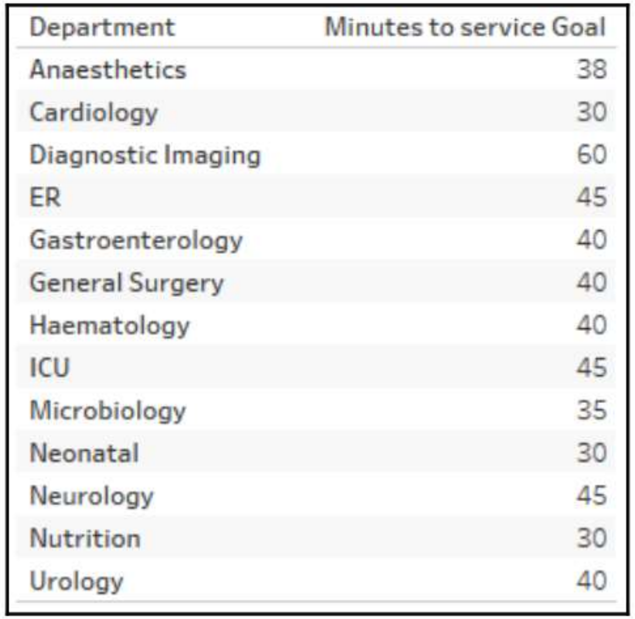
Bullet chart – comparing to a goal, target, or threshold

- A bullet graph (sometimes also called a bullet chart) is a great way to visually compare a measure with a goal, target, or threshold.
- The bar indicates the measured value, while the line indicates the target.
- Tableau also defaults to shading to indicate 60% and 80% of the distance to the goal or threshold.



Example

- Let's say that hospital administration has set some goals regarding the time to service, that is, the number of minutes between the time a patient arrives at the hospital and the time they start receiving care. The administration knows that each department has unique capabilities and requirements, so they have defined different goals for each department, as follows:

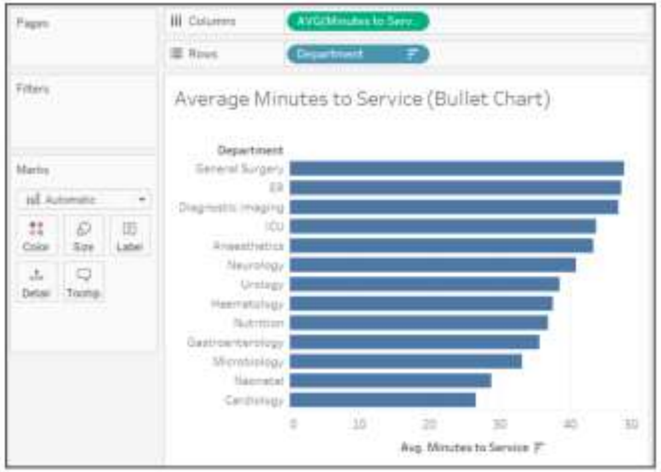


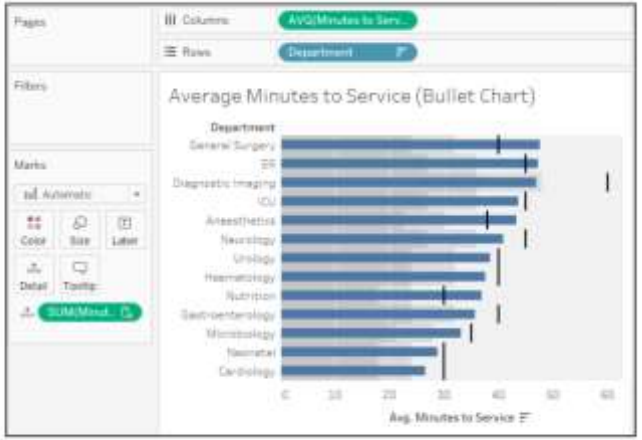
Department	Minutes to service Goal
Anaesthetics	38
Cardiology	30
Diagnostic Imaging	60
ER	45
Gastroenterology	40
General Surgery	40
Haematology	40
ICU	45
Microbiology	35
Neonatal	30
Neurology	45
Nutrition	30
Urology	40

Use Hospital Visits and Hospital Goals data sources

Steps

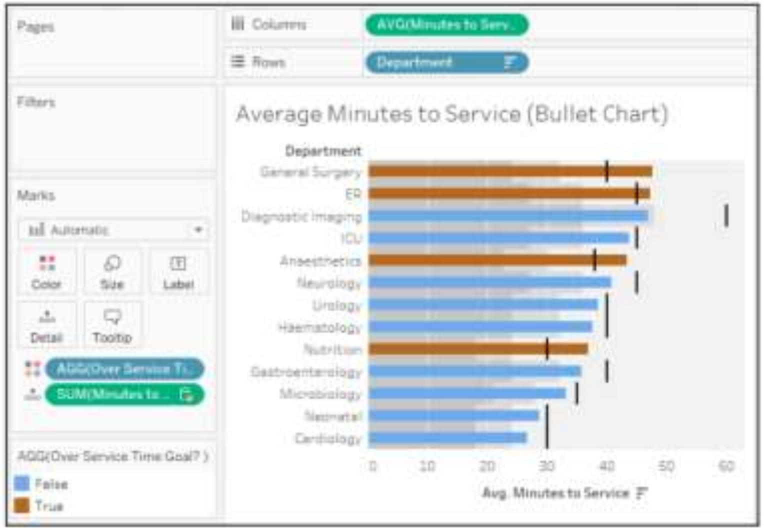
1. Open a new sheet.
2. Using the Hospital Visits data source, create a basic bar chart of the average Minutes to Service per Department.
3. Sort Department from highest to lowest. At this point, your view should look like this:
4. In the left-hand data pane, select the Hospital Goals data source and then click to select the Minutes to Service Goal field in the data pane under Measures





1.Click the Hospital Visits data source to select it.
2. Right-click an empty spot in the data pane under Dimensions or Measures and select Create Calculated Field.
3. Name the calculated field Over Service Time Goal? with the following code: AVG([Minutes to Service]) > SUM([Hospital Goals].[Minutes to Service Goal])
4. Click OK and drag the new Over Service Time Goal? field from the data pane and drop it on Color.

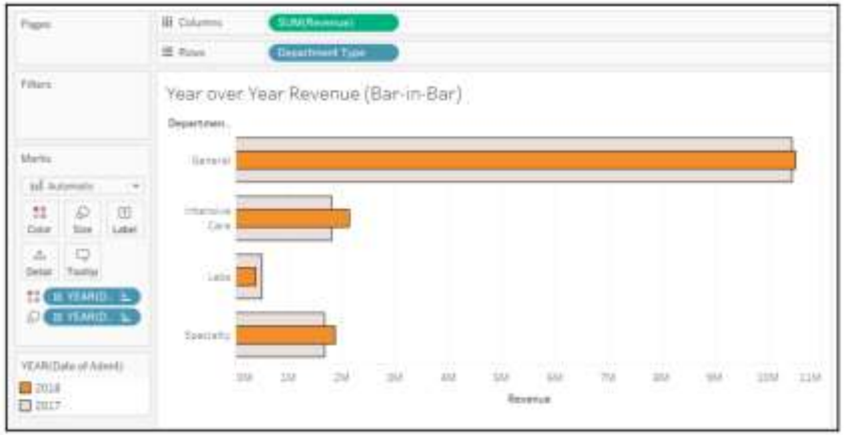
- The calculation returns true when the Average Minutes to Service values is greater than the goal value, and false otherwise.
- With the calculated field on Color, it becomes very easy to see which departments are over the threshold.
- The bullet chart makes it easy to see which departments have gone over the threshold that's been set by administration



Bar in Bar Chart

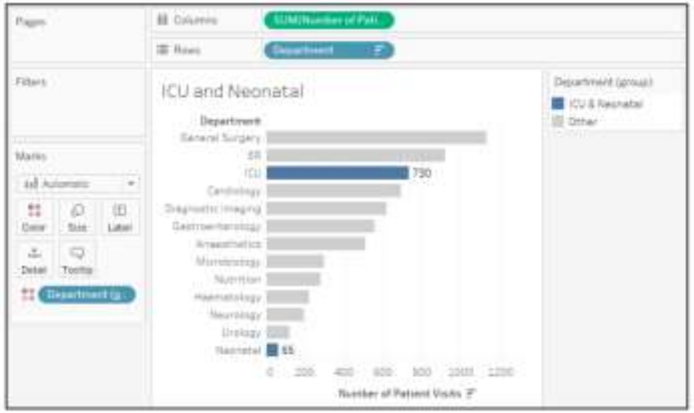
Like the bullet chart, the bar-in-bar chart can show progress toward a goal, but it can also be used to compare any two values.
For example, you might compare revenue to a target, or you might compare the revenue for the current year to the previous year in hospital visits and hospital goals.

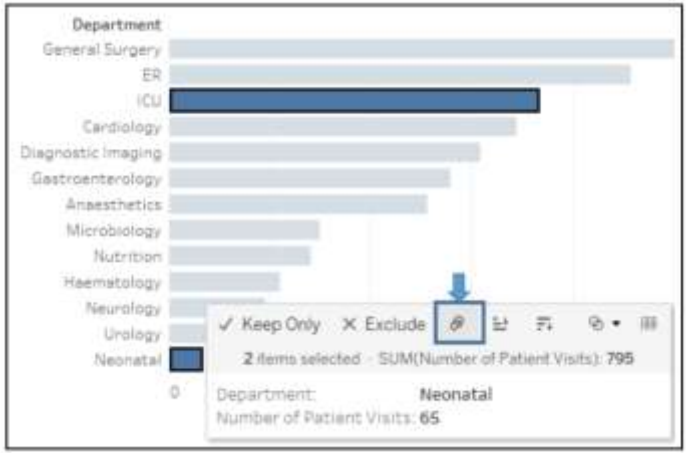
1. Drag and drop Revenue from the Hospital Visits data source onto the horizontal axis in the view (which gives the same results as dropping it onto the Columns shelf).
2. Drag and drop Department Type onto Rows.
3. Drag and drop the Date of Admit onto Color.
4. For a bar-in-bar chart, we do not want the marks to be stacked. To turn off stacking, use the top menu to select Analysis | Stack Marks | Off.
5. All of the bar segments now begin at 0, but some may be completely overlapped. To see each bar, we'll need to adjust another visual element.
6. In this case, hold down the Ctrl key while dragging the YEAR(Date of Admit) field that is currently on Color in the Marks card to Size.
7. Double-click the Color legend to edit the colors so that 2018 is emphasized.



Highlighting categories of interest

1. Place Department on Rows and Number of Patient Visits on Columns. Sort the bar chart in descending order.
2. Click on the bar in the view for ICU and, while holding down the Ctrl key, click the bar for Neonatal.
3. Hover the cursor over one of the selected bars for a few seconds and, from the menu that appears, click the Create Group button (which looks like a paperclip):
4. This will create a group, which results in a new dimension, named Department (group), in the left-hand data pane. Tableau automatically assigns this field to Color.
5. To add a label only to the bars for those two departments, right-click each bar and select Mark label | Always show. The label for the mark will always be shown, even if other labels are turned off for the view or the label overlaps marks or other labels.

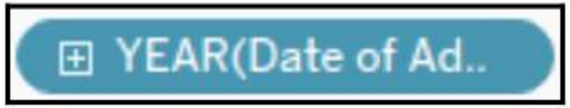




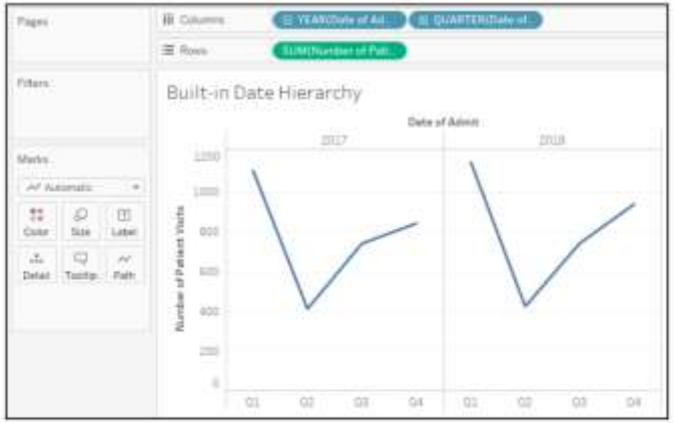
Visualizing dates and times

- When you are connected to a flat file, relational, or extracted data source, Tableau provides a robust built-in date hierarchy for any date field.
- In Tableau Desktop, cube (multidimensional) data sources are supported only in Windows.
- For cube data sources, date dimensions are usually organized into hierarchies that contain levels such as year, quarter, and month.
- In addition, some multidimensional data sources have time intelligence enabled, which makes it possible to look at data levels in different ways, such as Months by Year, Months by Quarter, Weekends, etc.
- These levels are represented as attributes of the hierarchy.

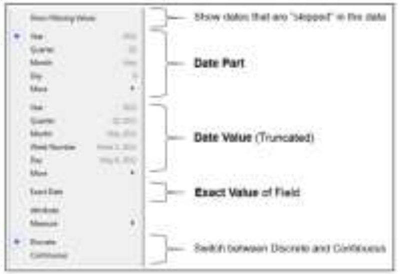
The YEAR(Date of Admit) field on Columns will have a plus sign indicator, like this



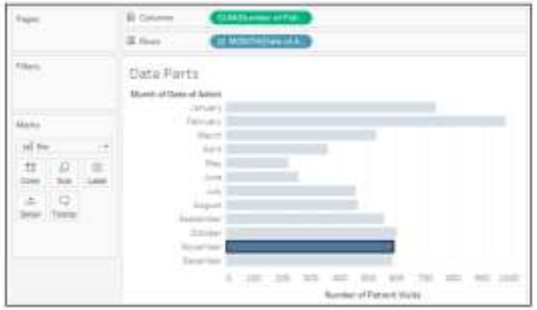
When you click it, the hierarchy expands by adding QUARTER(Date of Admit) to the right of the YEAR(Date of Admit) on Columns, and the view is expanded to the new level of the hierarchy:



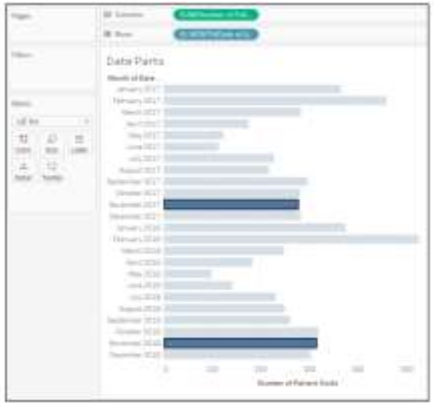
The YEAR(Date of Admit) field now has a minus sign indicator that allows you to collapse the hierarchy back to the year level. The QUARTER field also has a plus sign, indicating that you can expand the hierarchy further. Starting with Year, the hierarchy flows as follows: Year | Quarter | Month | Day. When the field is a date and time, you can further drill down into Hour | Minute | Second. Any of the parts of the hierarchy can be moved within the view or removed from the view completely.


Show dates that are "skipped" in the data
Date Part
Date Value (Truncated)
Exact Value of Field
Switch between Discrete and Continuous

- The hierarchy is made up of Date Parts, which is one of the three ways a date field can be used.
- When you right-click the date field in the view or by using the drop-down menu, you'll see multiple date options, as follows:



- The three major date types are evident, though not explicitly labeled, in the menu: Date part: This field will represent a specific part of the date, such as the quarter or month.
- The part of the date is used by itself and without reference to any other part of the date. This means that a date of November 8, 1980, when used as a month date part, is simply November.
- The November that's selected in the view here represents all of the Novembers in the dataset, while the number of patient visits is the total for both 2017 and 2018:



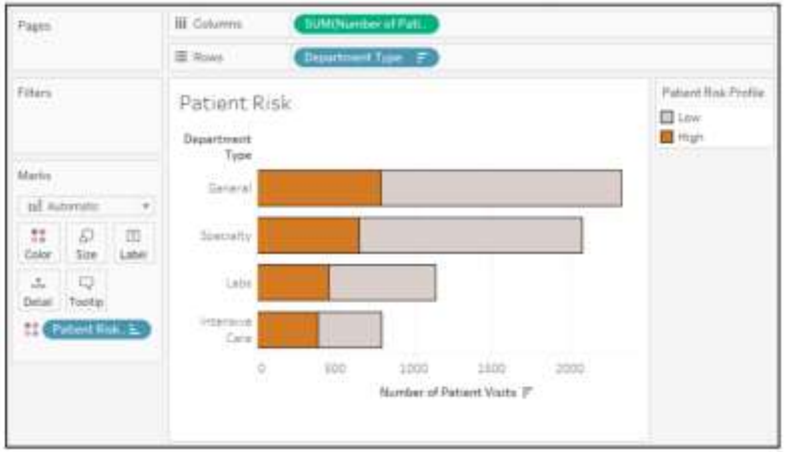
Date value: This field will represent a date value, but rolled up or truncated to the level you select. For example, if you select a date value of month, then November 8, 2018 gets truncated to the month and year, and is November 2018. You'll notice that November 2017 and November 2018 each have a separate value in the header and a distinct bar

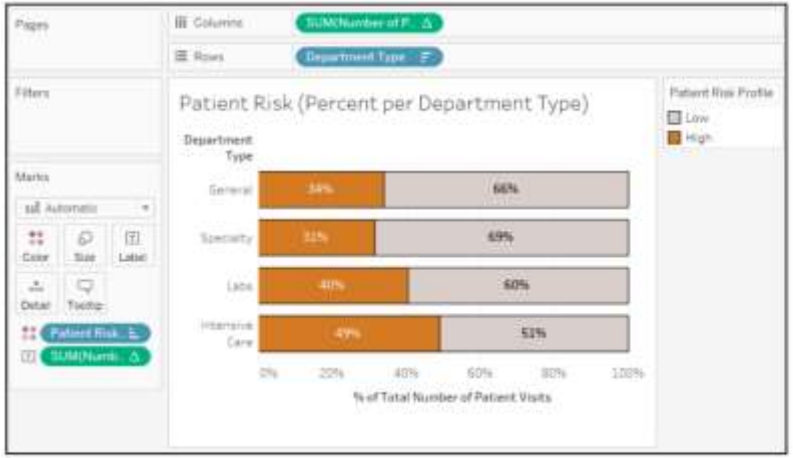
Relating parts of the data to the whole

- Stacked Charts
- Tree maps
- Area Charts
- Pie Charts

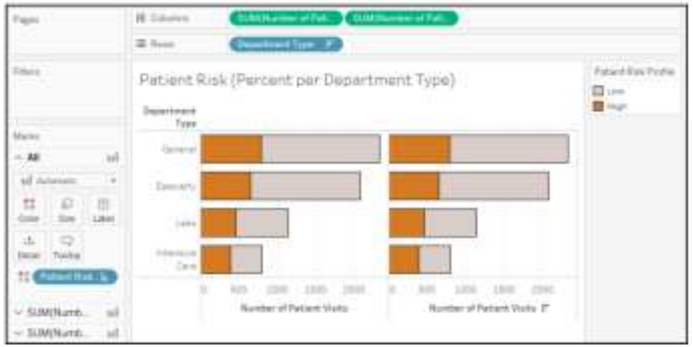
Stacked Charts

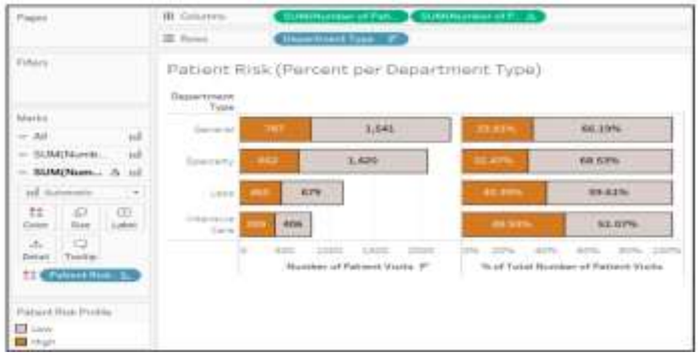
- A stacked bar in Tableau is a type of bar chart that represents values in the form of segmented bars. Here, each bar is divided into different segments or sections, providing further details about the field and regions.
- Stacked bars are used to visualize the makeup of the whole.





1. Create a stacked bar chart by placing Department Type on Rows, Number of Patient Visits on Columns, and Patient Risk Profile on Color. You'll now have a single stacked bar chart.
2. Sort the bar chart in descending order.
3. Duplicate the Number of Patient Visits field on Columns by holding down Ctrl while dragging the Number of Patient Visits field in the view to a spot on Columns, immediately to the right of its current location. Alternatively, you can drag and drop the field from the data pane to Columns. At this point, you have two Number of Patient Visits axes which, in effect, duplicate the stacked bar chart
4. Using the drop-down menu of the second Number of Patient Visits field, select Quick Table Calculation | Percent of Total. This table calculation runs a secondary calculation on the values that were returned from the data source to compute a percent of the total. Here, you will need to further specify how that total should be computed.
5. Using the same drop-down menu, select Compute Using | Patient Risk Profile. This tells Tableau to calculate the percent for each Patient Risk Profile within a given department. This means that the values will add up to 100% for each department
6. Turn on labels by clicking the T button on the top toolbar. This turns on default labels for each mark:

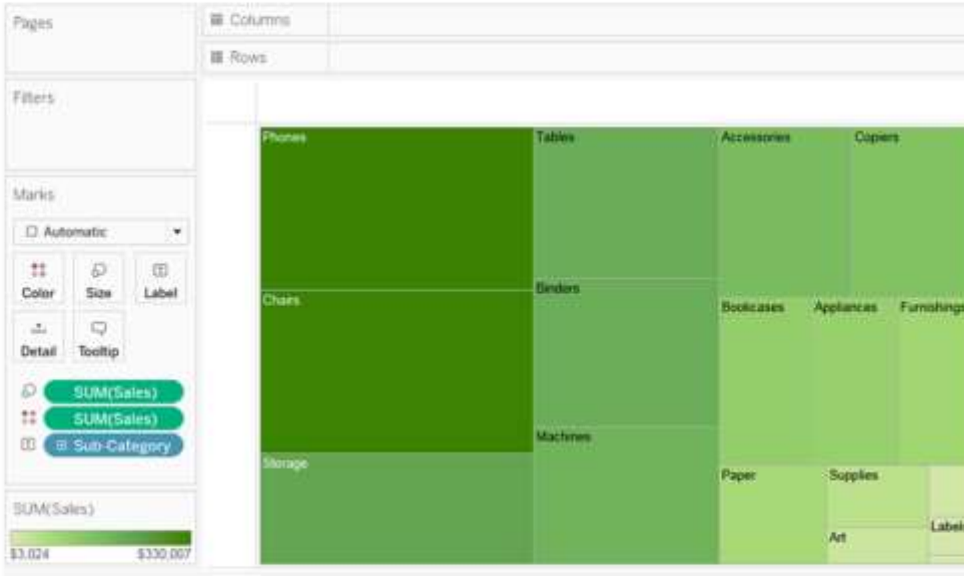



Patient Risk (Percent per Department Type)
Department Type
1,541
1,420
679
406
Number of Patient Visits
% of Total Number of Patient Visits

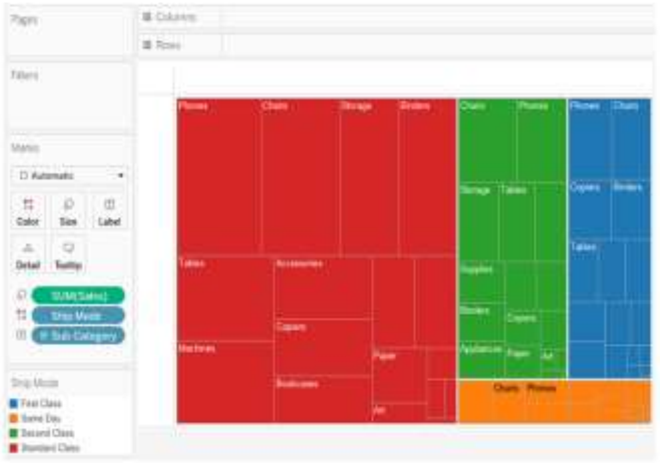
Tree Maps

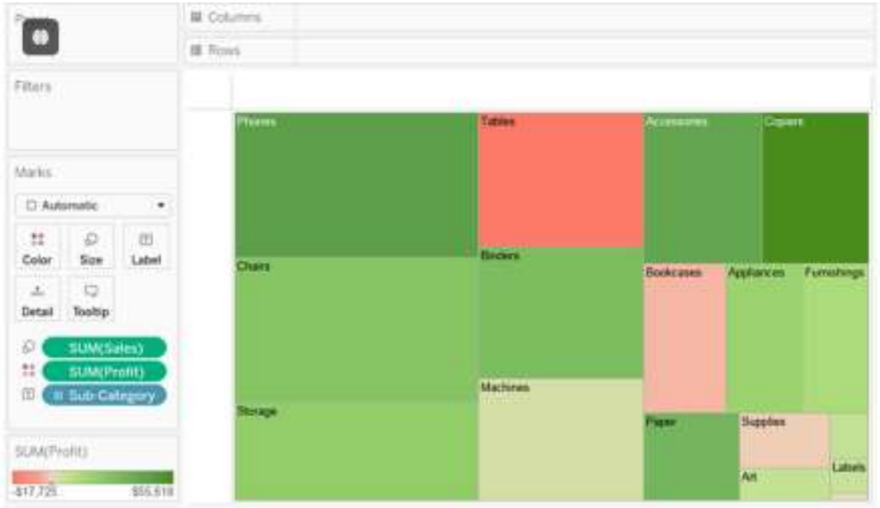
- Treemaps use a series of nested rectangles to display parts of the whole, especially within hierarchical relationships.
- Treemaps are particularly useful when you have hierarchies and dimensions with high cardinality (a high number of distinct values).
  - To create a treemap that shows aggregated sales totals across a range of product categories, follow the steps below.
  - Connect to the **Sample - Superstore** data source.
  - Drag the **Sub-Category** dimension to **Columns**.
  - A horizontal axis appears, which shows product categories.
  - Drag the **Sales** measure to **Rows**.
  - Tableau aggregates the measure as a sum and creates a vertical axis.
  - Tableau displays a bar chart—the default chart type when there is a dimension on the **Columns** shelf and a measure on the **Rows** shelf.
  - Click **Show Me** on the toolbar, then select the treemap chart type.

In this treemap, both the size of the rectangles and their color are determined by the value of **Sales**—the greater the sum of sales for each category, the darker and larger its box.



5. Drag the **Ship Mode** dimension to **Color** on the **Marks** card. In the resulting view, **Ship Mode** determines the color of the rectangles—and sorts them into four separate areas accordingly. **Sales** determines the size of the rectangles:
6. Try another option to modify the treemap: click the **Undo** button to remove **Ship Mode** from view.
7. Drag the **Profit** measure to **Color** on the **Marks** card. Now **Profit** determines the color of the rectangles, and **Sales** determines their size:



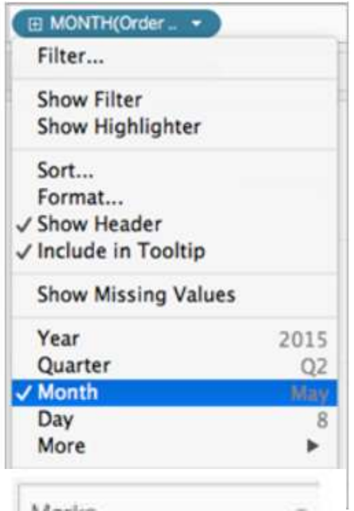


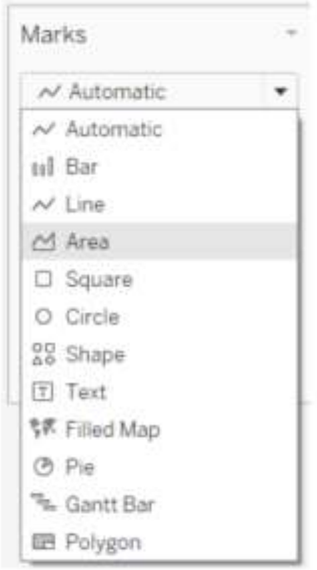
Area Charts

- Area charts in Tableau show relationships between different aspects or dimensions in a data set. This relationship is shown as the proportion of totals or percentage of certain data values. The comparison between different dimensions and values can be made by analyzing the area under each line and how does it varies with time.
- There are two types of area charts in Tableau; *Continuous Area Chart* and *Discrete Area Chart*. A continuous area chart does not have individual values and hence is always measured. On the other hand, in a discrete area chart, individual values of a category are taken and hence are always counted.

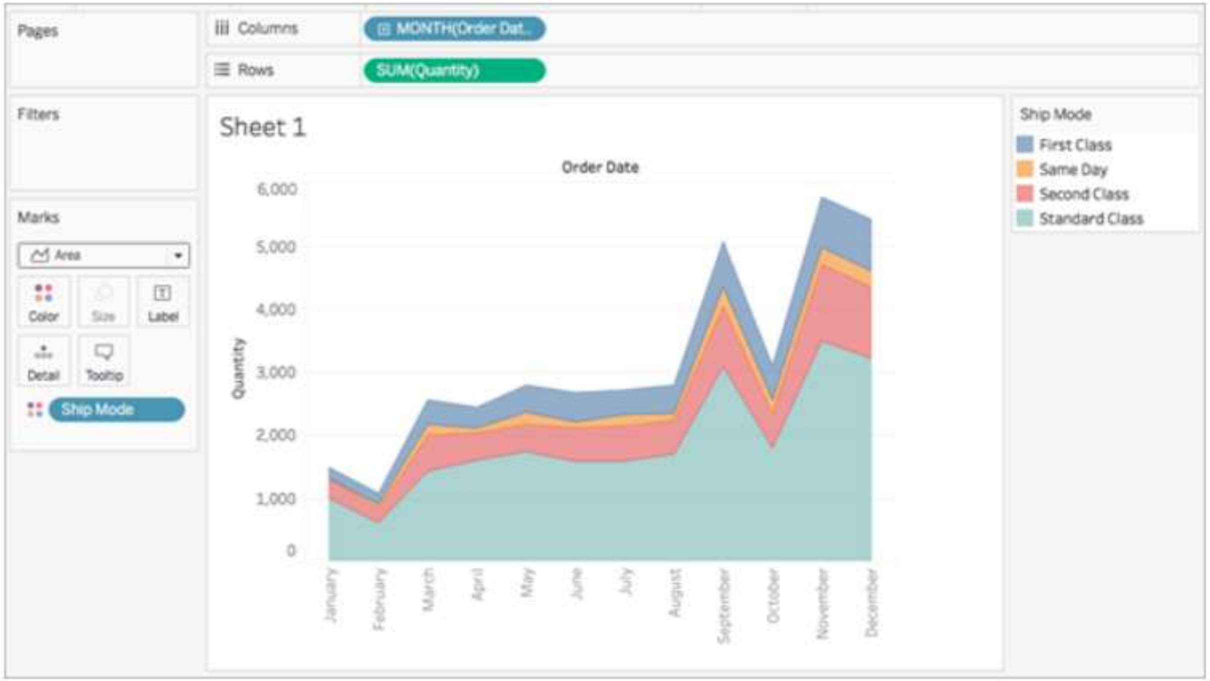
- To create an area chart, follow the steps below:

1. Open Tableau Desktop and connect to the **Sample - Superstore** data source.
2. Navigate to a new worksheet.
3. From the **Data** pane, drag **Order Date** to the **Columns** shelf.
4. On the Columns shelf, right-click **YEAR(Order Date)** and select **Month**.
5. From the **Data** pane, drag **Quantity** to the **Rows** shelf.
6. From the **Date** pane, drag **Ship Mode** to **Color** on the Marks card.
7. On the Marks card, click the Mark Type drop-down and select **Area**.



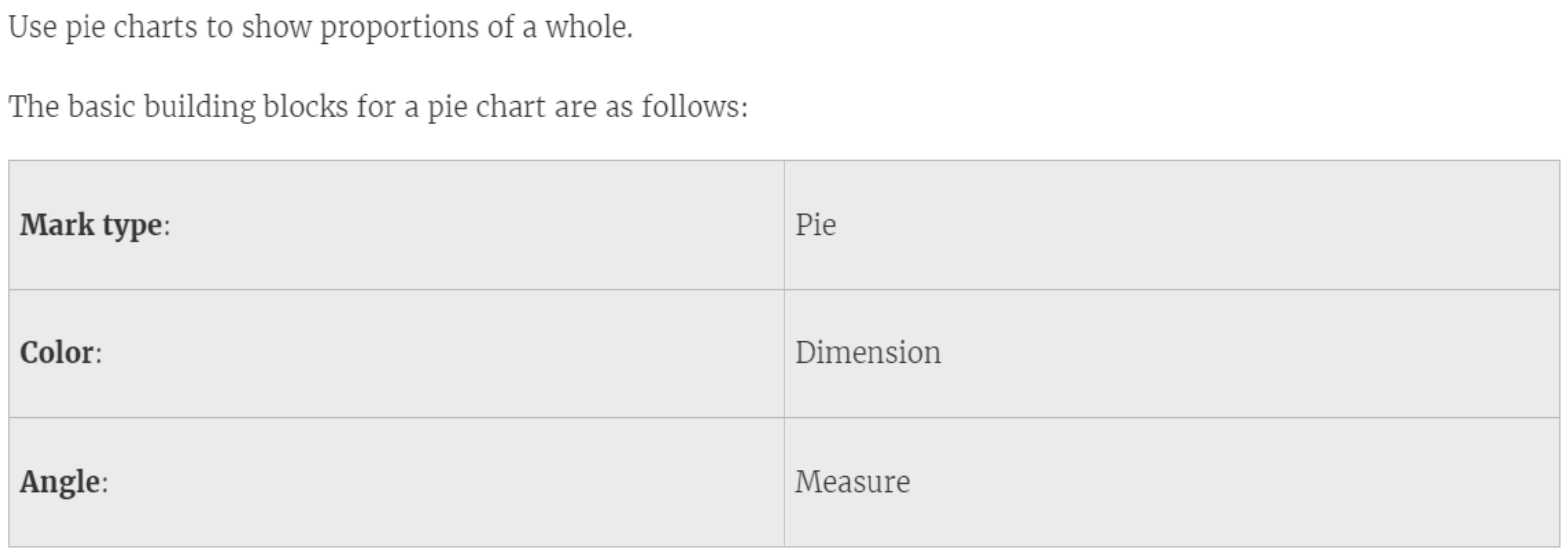


The visualization updates to the following:



Pie Charts

- Use pie charts to show proportions of a whole.



Use pie charts to show proportions of a whole.

The basic building blocks for a pie chart are as follows:

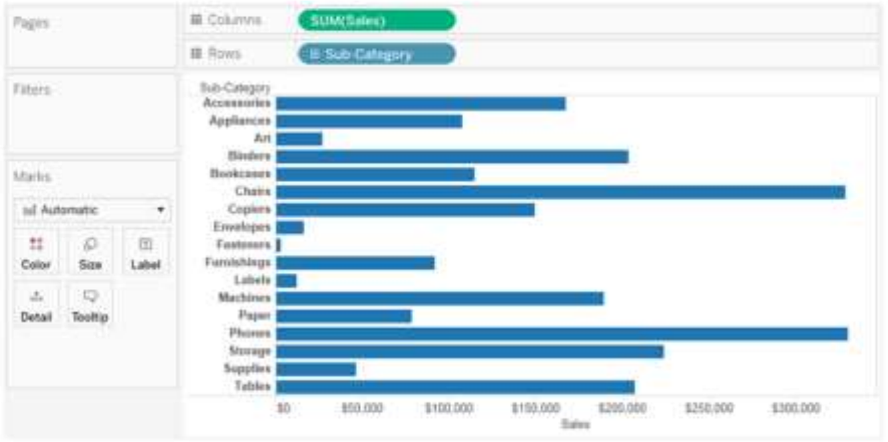
Mark type:	Pie
Color:	Dimension
Angle:	Measure

Example

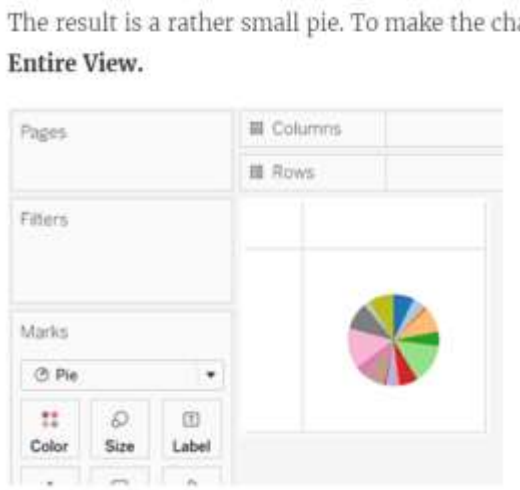
- To create a pie chart view that shows how different product categories contribute to total sales, follow these steps:

1. Connect to the **Sample - Superstore** data source.
2. Drag the **Sales** measure to **Columns** and drag the **Sub-Category** dimension to **Rows**.

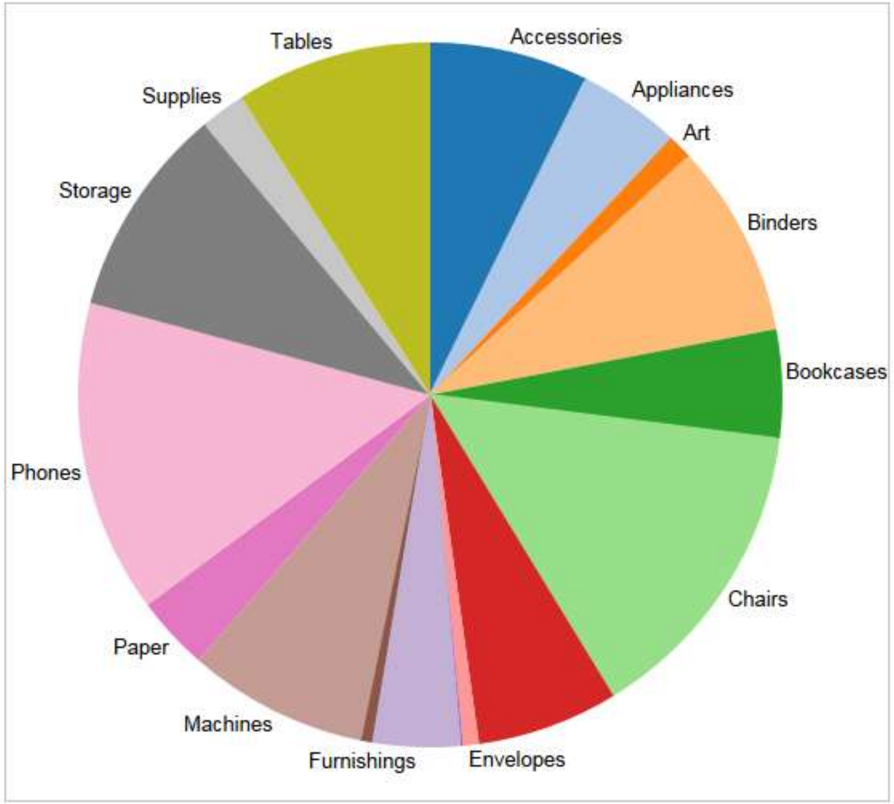
- Tableau aggregates the **Sales** measure as a sum. By default, Tableau displays a bar chart.



- Click **Show Me** on the toolbar, then select the pie chart type. Pie charts require at least one or more dimensions and one or two measures.
- The result is a rather small pie. To make the chart bigger, navigate to the Fit menu in the toolbar and select **Entire View.**


The result is a rather small pie. To make the chart bigger, navigate to the Fit menu in the toolbar and select **Entire View.**

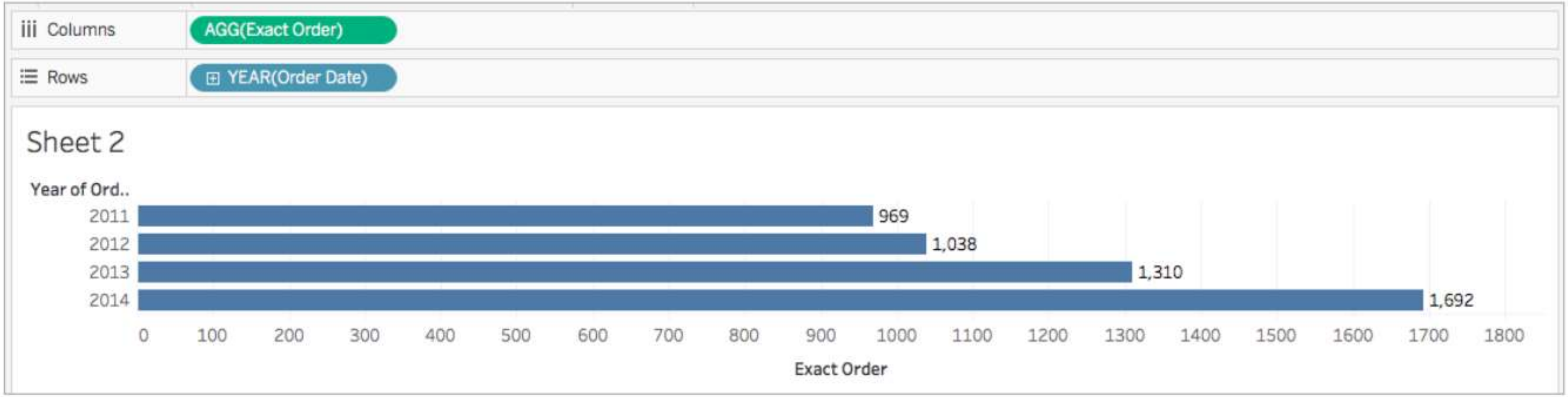
4. Add labels by dragging the **Sub-Category** dimension from the **Data** pane to **Label** on the **Marks** card.



Aggregate Functions

- Aggregate functions allow you to summarize or change the granularity of your data.
- For example, you might want to know exactly how many orders your store had for a particular year. You can use the COUNTD function to summarize the exact number of orders your company had, and then break the visualization down by year.
- The calculation might look something like this:

The visualization might look something like this:



AVG

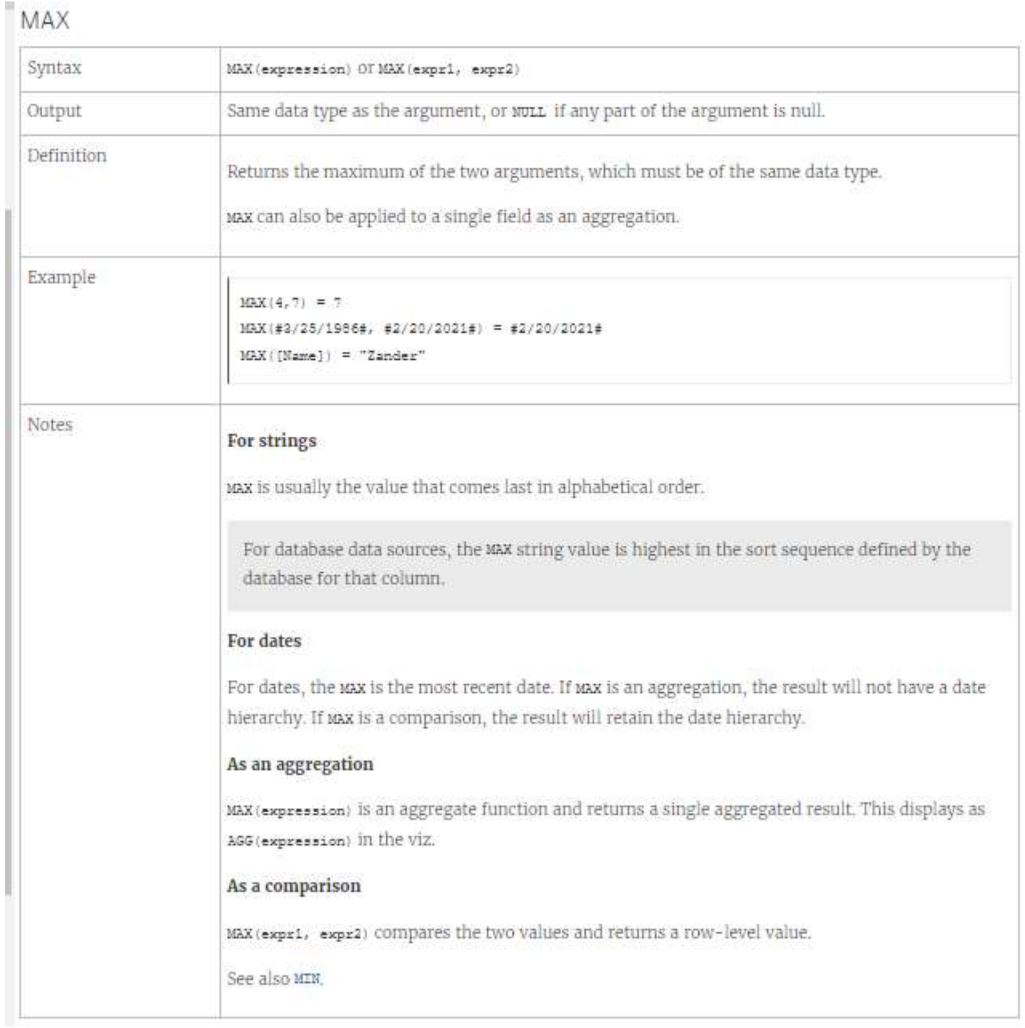
Syntax	`AVG(expression)`
Definition	Returns the average of all the values in the expression. Null values are ignored.
Notes	`AVG` can only be used with numeric fields.

COUNT

Syntax	`COUNT(expression)`
Definition	Returns the number of items. Null values are not counted.

COUNTD

Syntax	`COUNTD(expression)`
Definition	Returns the number of distinct items in a group. Null values are not counted.

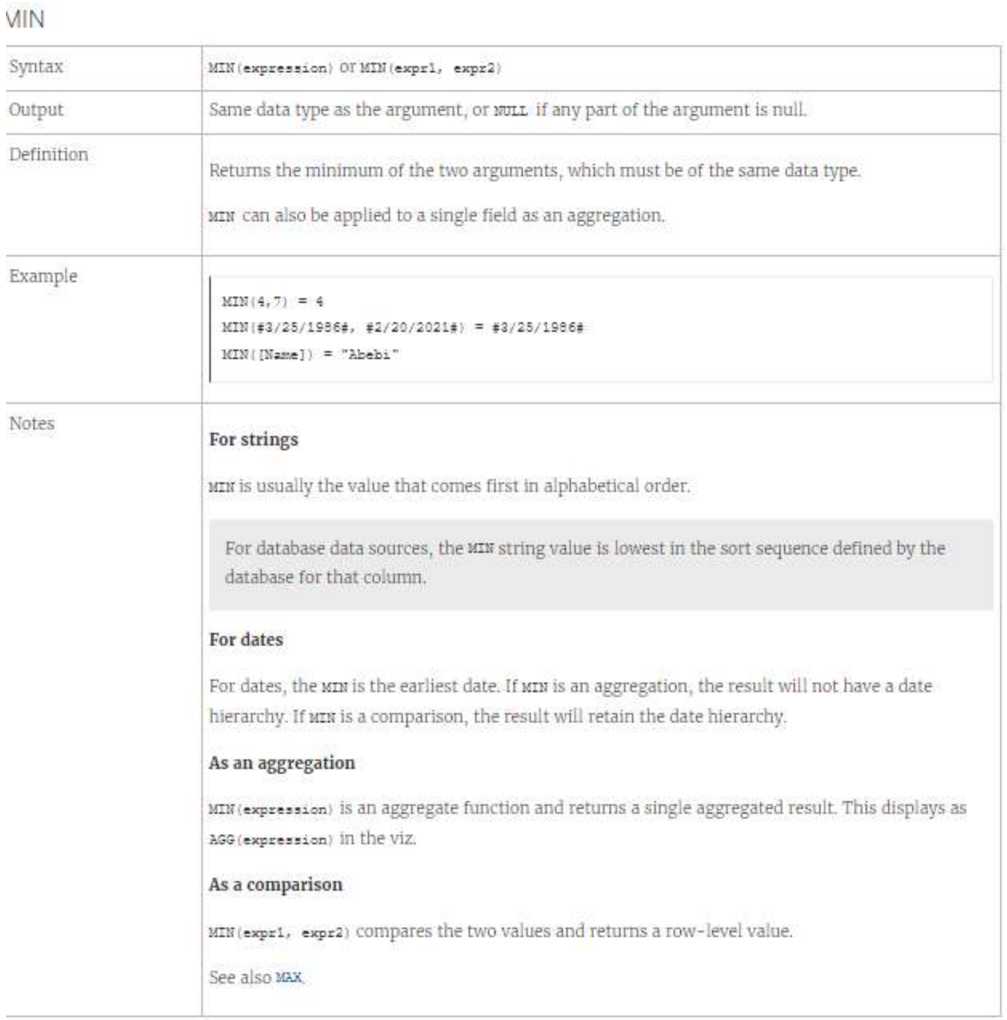


MAX

Syntax	`MAX(expression)` or `MAX(expr1, expr2)`
Output	Same data type as the argument, or `NULL` if any part of the argument is null.
Definition	Returns the maximum of the two arguments, which must be of the same data type. `MAX` can also be applied to a single field as an aggregation.

Example	`MAX(4,7) = 7`
	`MAX(#3/25/1986#, #2/20/2021#) = #2/20/2021#`
	`MAX([Name]) = "Zander"`

Notes	**For strings** `MAX` is usually the value that comes last in alphabetical order. For database data sources, the `MAX` string value is highest in the sort sequence defined by the database for that column. **For dates** For dates, the `MAX` is the most recent date. If `MAX` is an aggregation, the result will not have a date hierarchy. If `MAX` is a comparison, the result will retain the date hierarchy. **As an aggregation** `MAX(expression)` is an aggregate function and returns a single aggregated result. This displays as `AGG(expression)` in the viz. **As a comparison** `MAX(expr1, expr2)` compares the two values and returns a row-level value. See also `MIN`.



MIN

Syntax	`MIN(expression)` or `MIN(expr1, expr2)`
Output	Same data type as the argument, or `NULL` if any part of the argument is null.
Definition	Returns the minimum of the two arguments, which must be of the same data type. `MIN` can also be applied to a single field as an aggregation.

Example	`MIN(4,7) = 4`
	`MIN(#3/25/1986#, #2/20/2021#) = #3/25/1986#`
	`MIN([Name]) = "Abebi"`

Notes	**For strings** `MIN` is usually the value that comes first in alphabetical order. For database data sources, the `MIN` string value is lowest in the sort sequence defined by the database for that column. **For dates** For dates, the `MIN` is the earliest date. If `MIN` is an aggregation, the result will not have a date hierarchy. If `MIN` is a comparison, the result will retain the date hierarchy. **As an aggregation** `MIN(expression)` is an aggregate function and returns a single aggregated result. This displays as `AGG(expression)` in the viz. **As a comparison** `MIN(expr1, expr2)` compares the two values and returns a row-level value. See also `MAX`.

SUM

Syntax	`SUM(expression)`
Definition	Returns the sum of all values in the expression. Null values are ignored.
Notes	`SUM` can only be used with numeric fields.

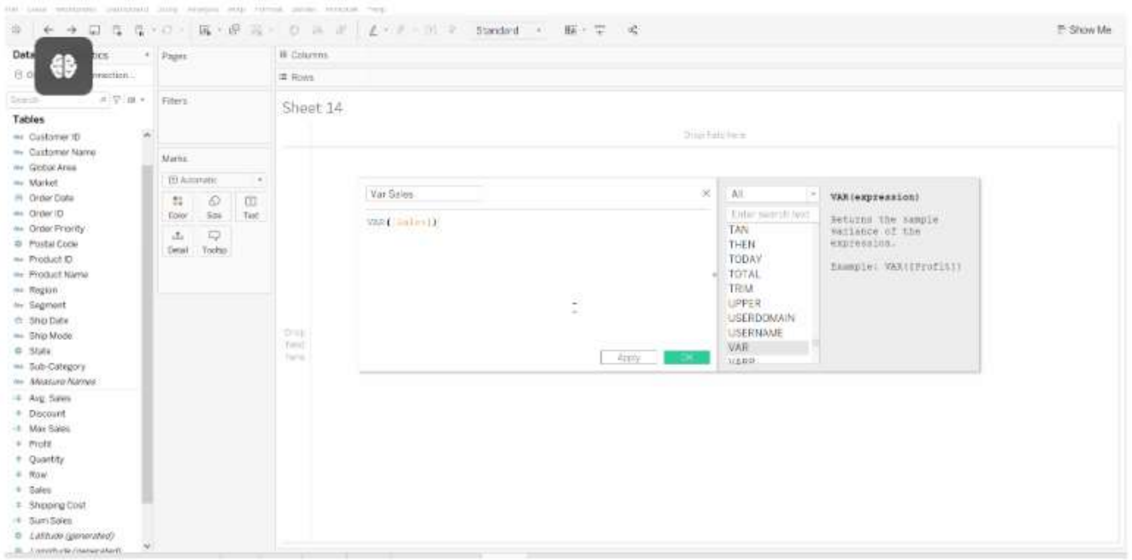
Tableau VAR Function

The Tableau VAR function is an aggregate function in tableau, which is used to find the Variance of the sample population.

Syntax:

```
VAR(Expression)
```

Example:



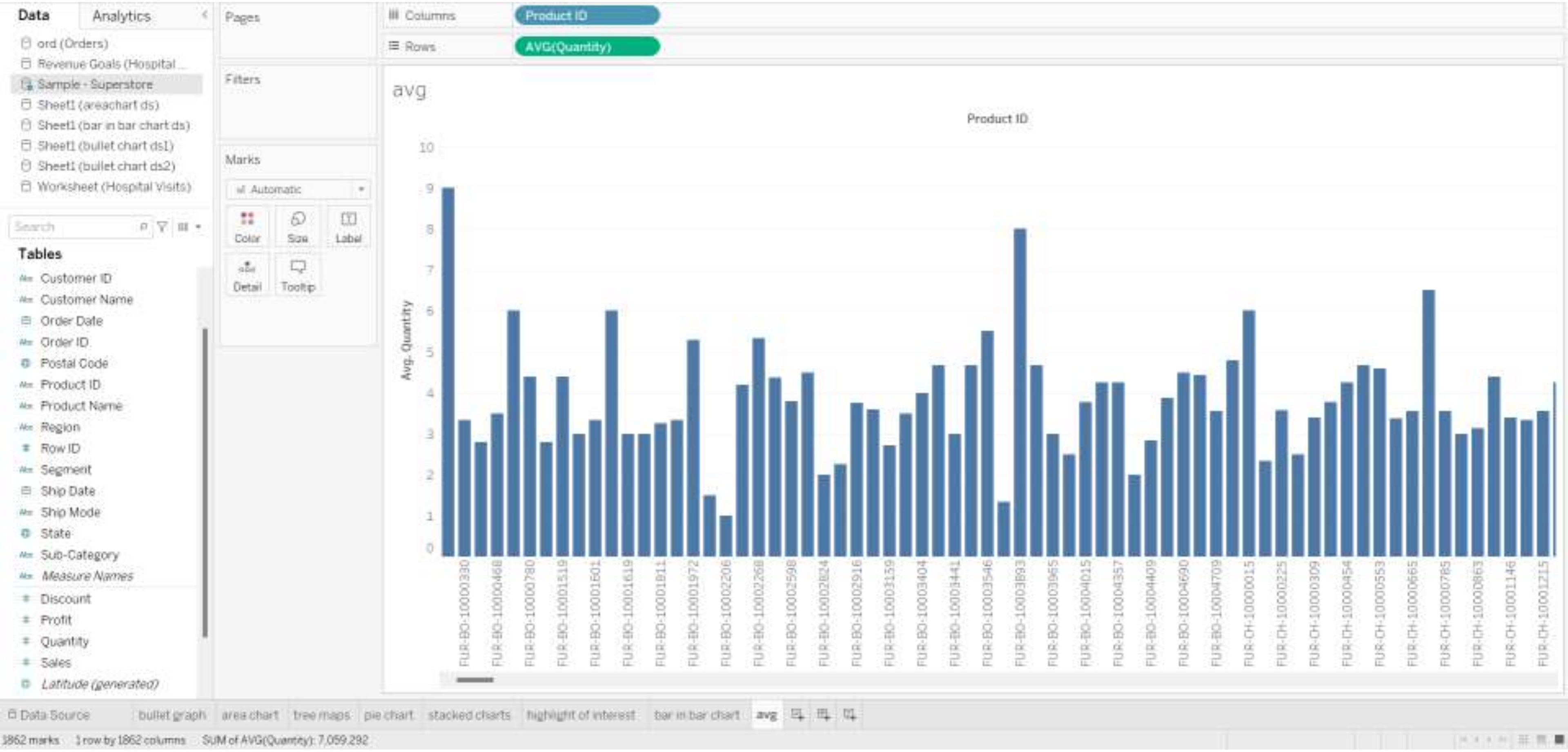
Var Sales

```
VAR([Sales])
```

VAR(expression)

Returns the sample variance of the expression.

Example: VAR([Profit])



Columns: Product ID

Rows: AVG(Quantity)

1862 marks   1 row by 1862 columns   SUM of AVG(Quantity): 7,059.292

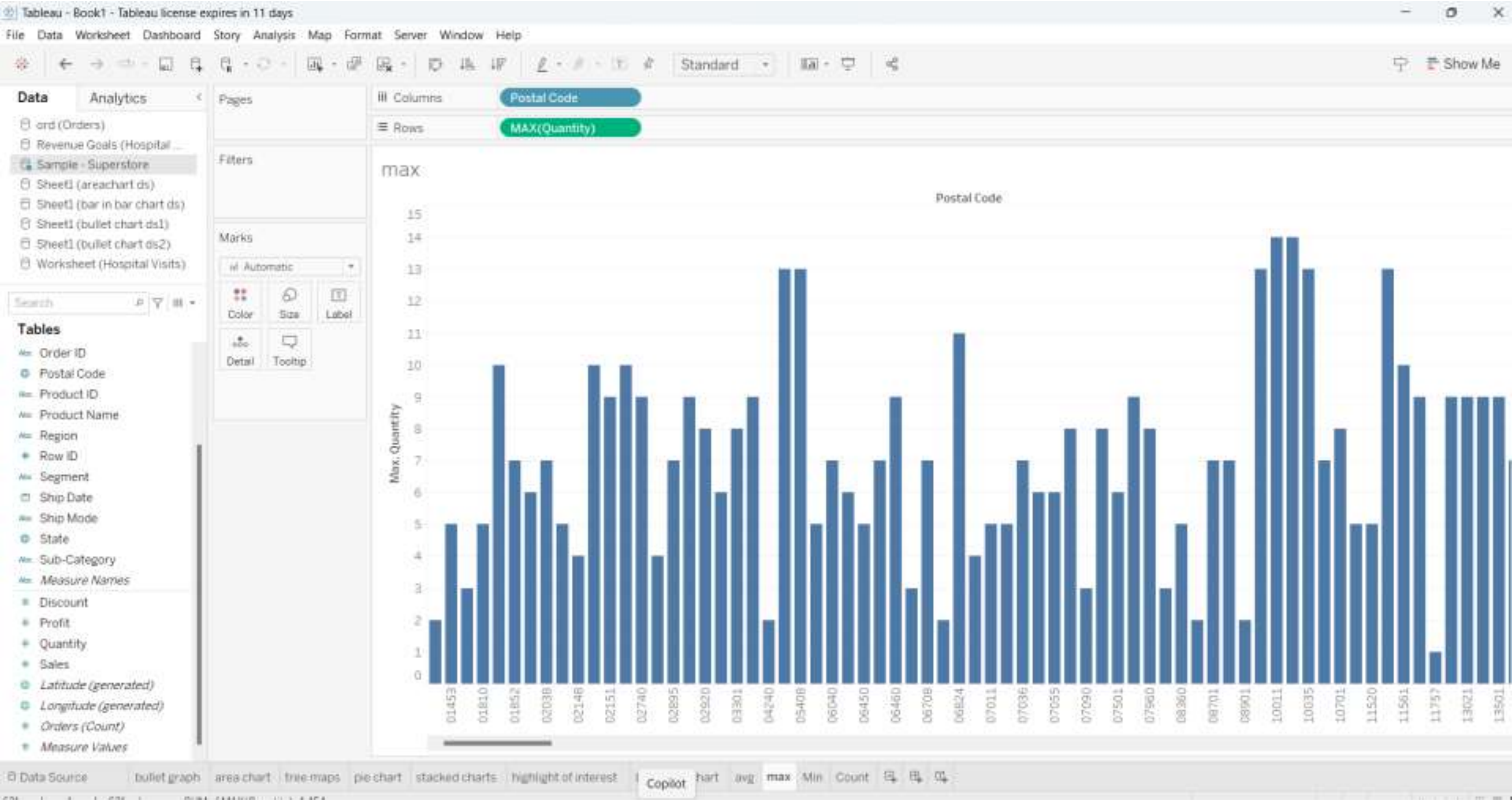
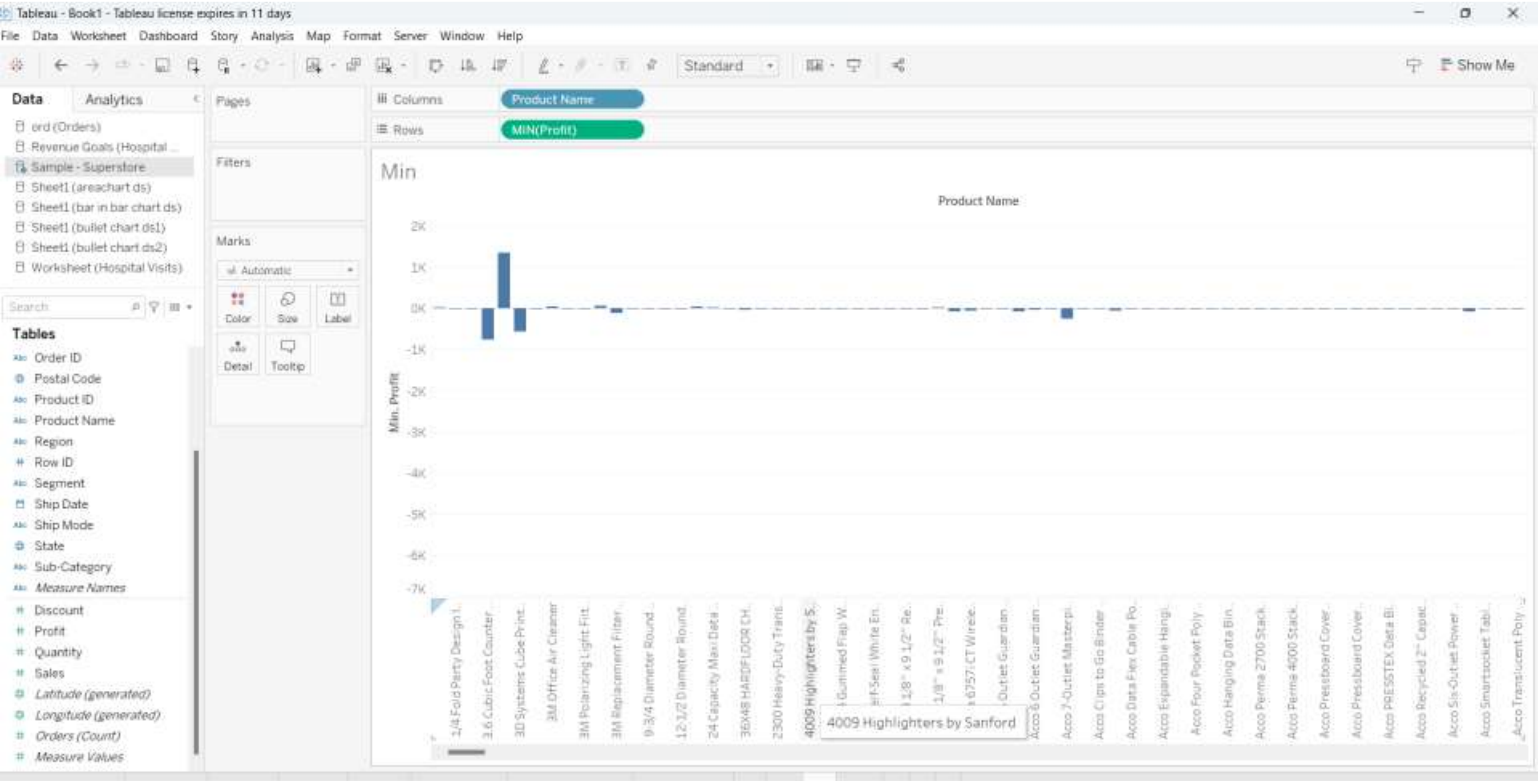


Tableau - Book1 - Tableau license expires in 11 days
File Data Worksheet Dashboard Story Analysis Map Format Server Window Help
Standard
Show Me
Data Analytics
ord (Orders)
Revenue Goals (Hospital ...
Sample - Superstore
Sheet1 (areachart ds)
Sheet1 (bar in bar chart ds)
Sheet1 (bullet chart ds1)
Sheet1 (bullet chart ds2)
Worksheet (Hospital Visits)
Tables
Order ID
Postal Code
Product ID
Product Name
Region
Row ID
Segment
Ship Date
Ship Mode
State
Sub-Category
Measure Names
Discount
Profit
Quantity
Sales
Latitude (generated)
Longitude (generated)
Orders (Count)
Measure Values
Pages
Filters
Marks
Automatic
Color
Size
Label
Detail
Tooltip
Columns
Postal Code
Rows
MAX(Quantity)
max
Postal Code
Max. Quantity
01453
01810
01852
02038
02148
02151
02740
02895
02920
03301
04240
05408
06040
06450
06460
06708
06824
07011
07036
07055
07090
07501
07960
08360
08701
08901
10011
10035
10701
11520
11561
11757
13021
13501
Data Source
bullet graph
area chart
tree maps
pie chart
stacked charts
highlight of interest
Copilot
hart
avg
max
Min
Count



Columns: Product Name

Rows: MIN(Profit)

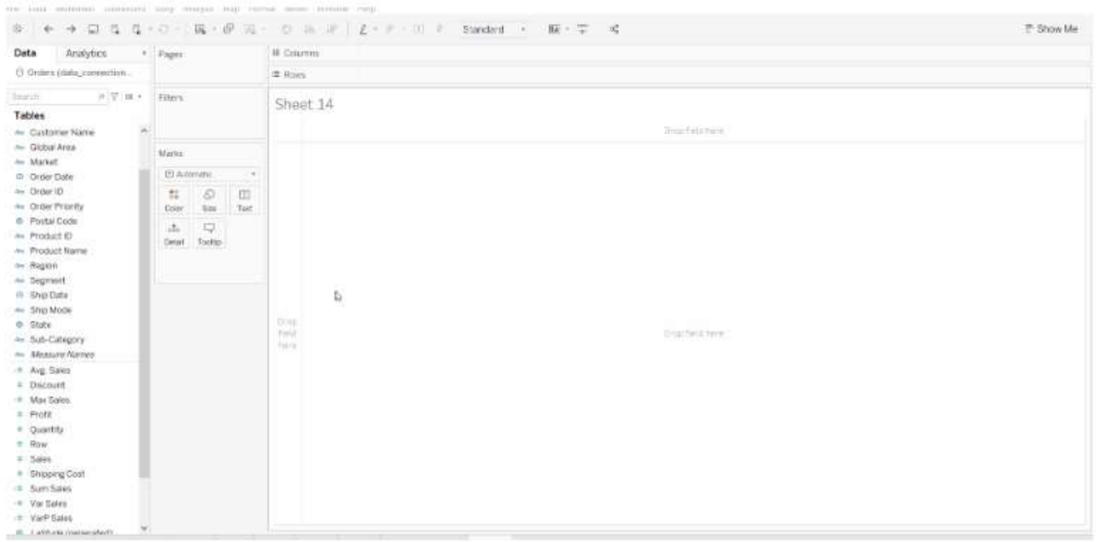
Tableau STDEV Function

The Tableau STDEV function is one of the aggregate functions in tableau, which is used to find the standard deviation of the sample population.

Syntax:

```
STDEV(Expression)
```

Example:

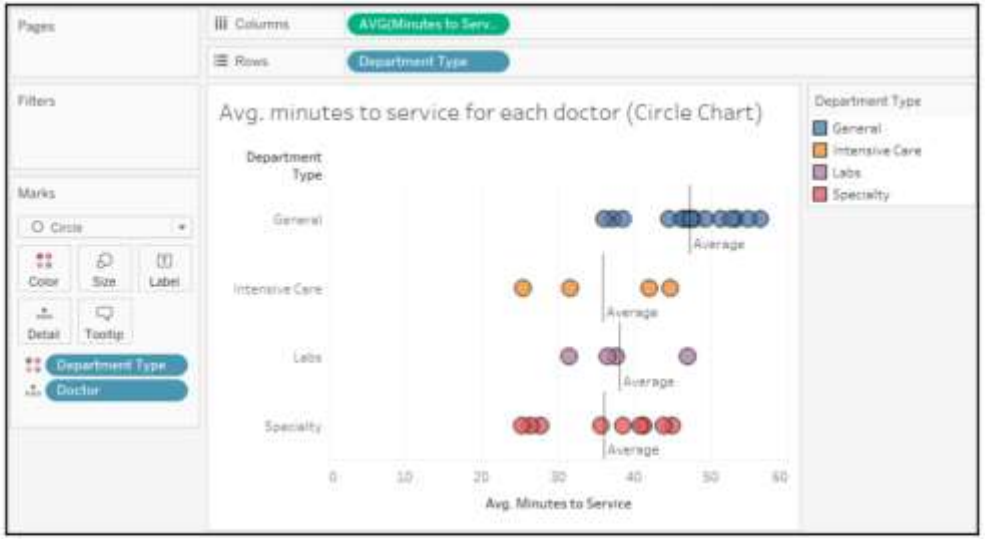


Visualising Distributions

- Data distribution is a way to organize trends and patterns observed in sets of data so that they are easier to understand.
- There are a variety of graphical ways to represent data distribution, such as
- Circle Plots
- Box Plots
- Histograms

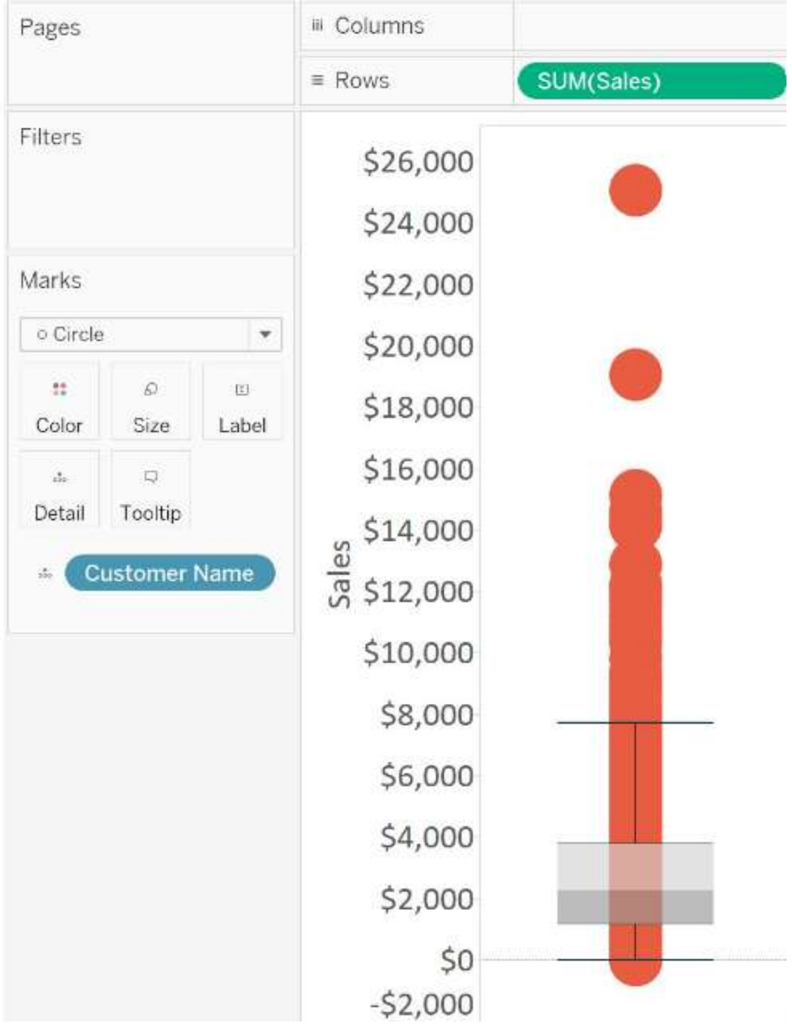
Circle Charts

- Circle charts are one way to visualize a distribution.
- The circle view is another powerful visualization for comparative analysis. It is just like the scatter plot with the circle marker.
- Here, every mark is of shape circle and can be used for further operations.
- For this we have to import the dataset first.
- Open Tableau tool and connect a dataset into it.
- Drag and drop the one sheet of the connected dataset.
- Click on sheet1 to open the tableau worksheet.
- On clicking Sheet1 you will get whole dataset attributes on the left side and a worksheet for work.
- To draw circle view you have to select minimum two attributes(one in row and one in column) by dragging and dropping then select the chart option as circle views.

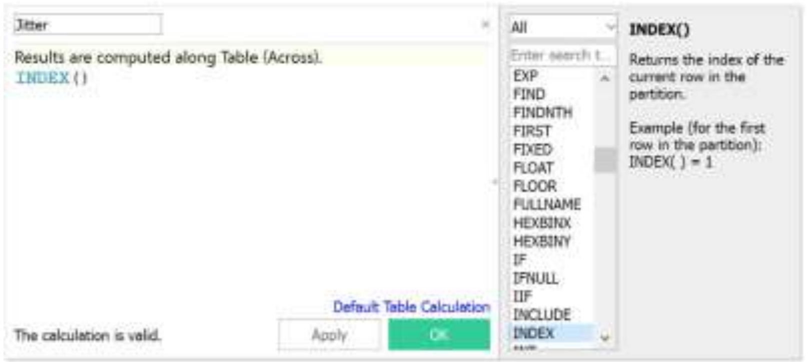


Jittering

- Jittering” is a technique for separating overlapping marks on a view.
- By giving marks some extra room by separating them into different columns, hidden data is often revealed and it is easier to visualize how the data points are distributed.



- Jitter plots are created using the INDEX() function and changing the addressing to the most granular level of detail – which in our case is Customer Name. To get started with jitter plots, create a calculated field named Jitter that includes only the INDEX() function:
- When this newly created calculated field is placed on the Columns Shelf, a new axis will appear, but all of the marks will still be lined up in one column. To apply the jitter, click on the calculated field, hover over “Compute using” and choose the most granular level of detail: Customer Name:



Jitter

Results are computed along Table (Across).

```
INDEX()
```

Default Table Calculation

The calculation is valid.

Apply OK

All

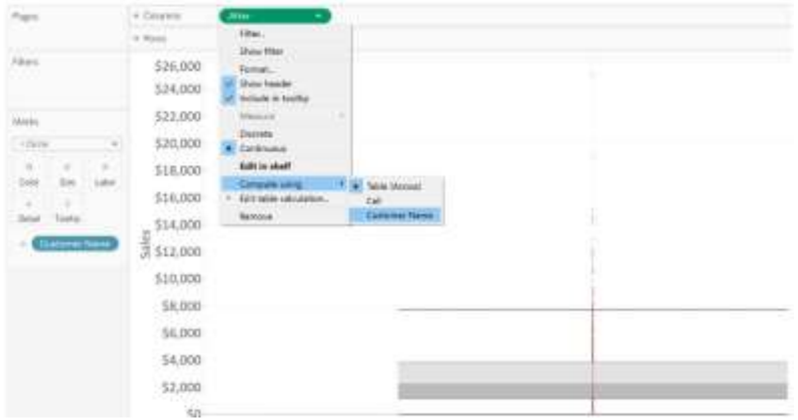
Enter search t...

EXP
FIND
FINDNTH
FIRST
FIXED
FLOAT
FLOOR
FULLNAME
HEXBINX
HEXBINY
IF
IFNULL
IIF
INCLUDE
INDEX

INDEX()

Returns the index of the current row in the partition.

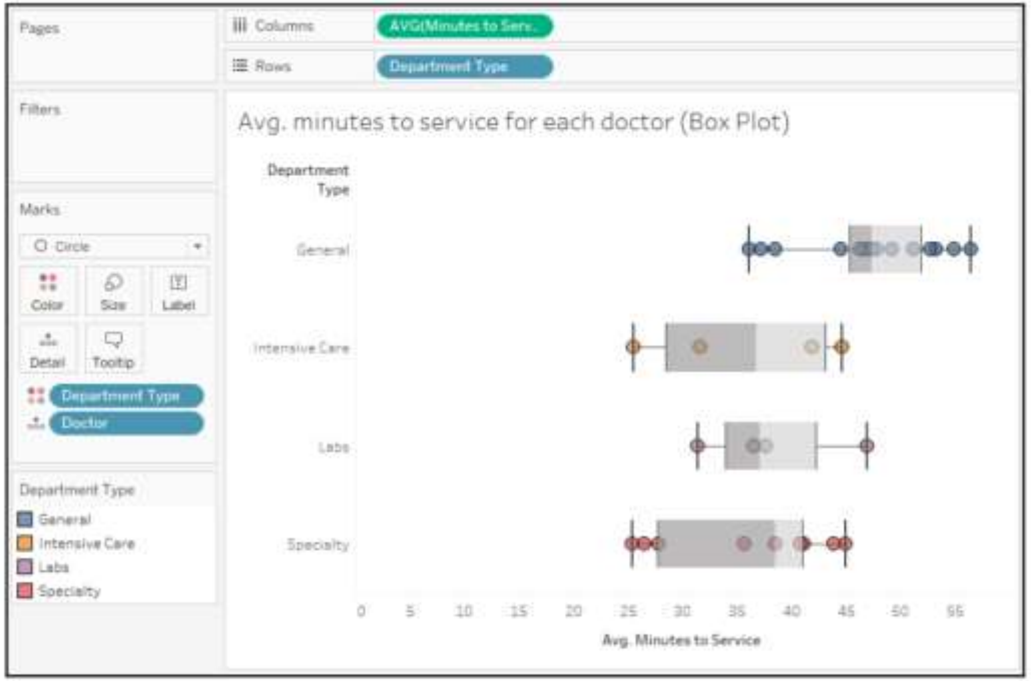
Example (for the first row in the partition):
INDEX( ) = 1



Box and Whisker Plots

- Box and whisker plots, sometimes known as box plots, are a great chart to use when showing the distribution of data points across a selected measure.
- These charts display ranges within variables measured. This includes the outliers, the median, the mode, and where the majority of the data points lie in the “box”.
- These visuals are helpful to compare the distribution of many variables against each other.
- Box and whisker plots were first drawn by John Wilder Tukey. An American mathematician, he came up with the formula as part of his toolkit for exploratory data analysis in 1970.
- Box and whisker plots portray the distribution of your data, outliers, and the median. The box within the chart displays where around 50 percent of the data points fall.
- It summarizes a data set in five marks. The mark with the greatest value is called the maximum. It will likely fall far outside the box. The mark with the lowest value is called the minimum. It will likely fall outside the box on the opposite side as the maximum.
- The box itself contains the lower quartile, the upper quartile, and the median in the center.
- The median is the value separating the higher half from the lower half of a data sample, a population, or a probability distribution
- The whiskers (the lines extending from the box on both sides) typically extend to 1.5* the Interquartile Range (the box) to set a boundary beyond which would be considered outliers. Hence the name, box, and whisker plot.

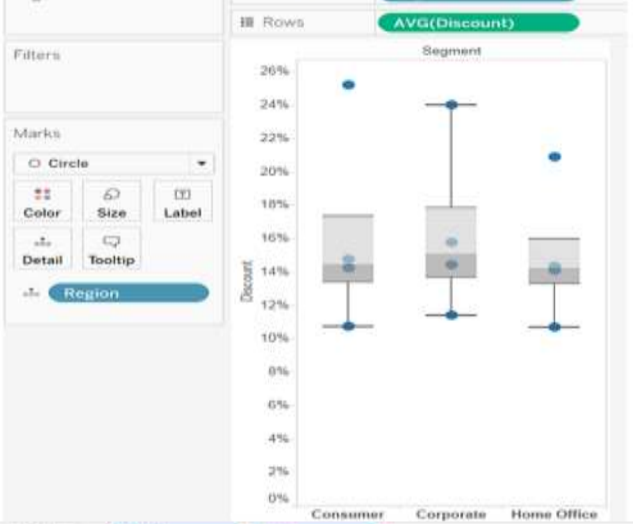
- To add box and whisker plots, use the Analytics tab on the left sidebar and drag Box Plot to the view. Doing this to the circle chart we considered previously yields the following chart:

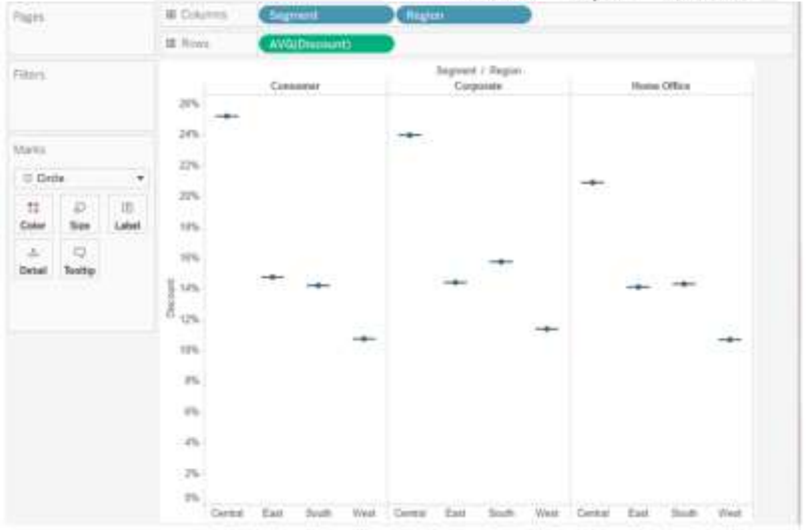


To create a box plot that shows discounts by region and customer segment, follow these steps:

1. Connect to the **Sample - Superstore** data source.
2. Drag the **Segment** dimension to **Columns**.
3. Drag the **Discount** measure to **Rows**.
4. Tableau creates a vertical axis and displays a bar chart—the default chart type when there is a dimension on the **Columns** shelf and a measure on the **Rows** shelf.
4. Drag the **Region** dimension to **Columns**, and drop it to the right of **Segment**.
5. Now you have a two-level hierarchy of dimensions from left to right in the view, with regions (listed along the bottom) nested within segments (listed across the top).

6. Click Show Me in the toolbar, then select the box-and-whisker plot chart type.





- Notice that there are only a few marks in each box plot. Also, Tableau reassigned Region from the Columns shelf to the Marks card.
- When you changed the chart type to a box plot, Tableau determined what the individual marks in the plot should represent.
- It determined that the marks should represent regions. We'll change that.
- Drag **Region** from the **Marks** card back to **Columns**, to the right of **Segment**.
- The horizontal lines are flattened box plots, which is what happens when box plots are based on a single mark.
- Box plots are intended to show a distribution of data, and that can be difficult when data is aggregated, as in the current view.
- To disaggregate data, select **Analysis** > **Aggregate Measures**.
- This command turns aggregation on or off, and because data is aggregated by default in Tableau, the first time you select this command, it disaggregates the data.

Histograms

- A histogram is a chart that displays the shape of a distribution. A histogram looks like a bar chart but groups values for a continuous measure into ranges or bins.
- The basic building blocks for a histogram are as follows:

Mark type:	Automatic
Rows shelf:	Continuous measure (aggregated by Count or Count Distinct)
Columns shelf:	Bin (continuous or discrete).

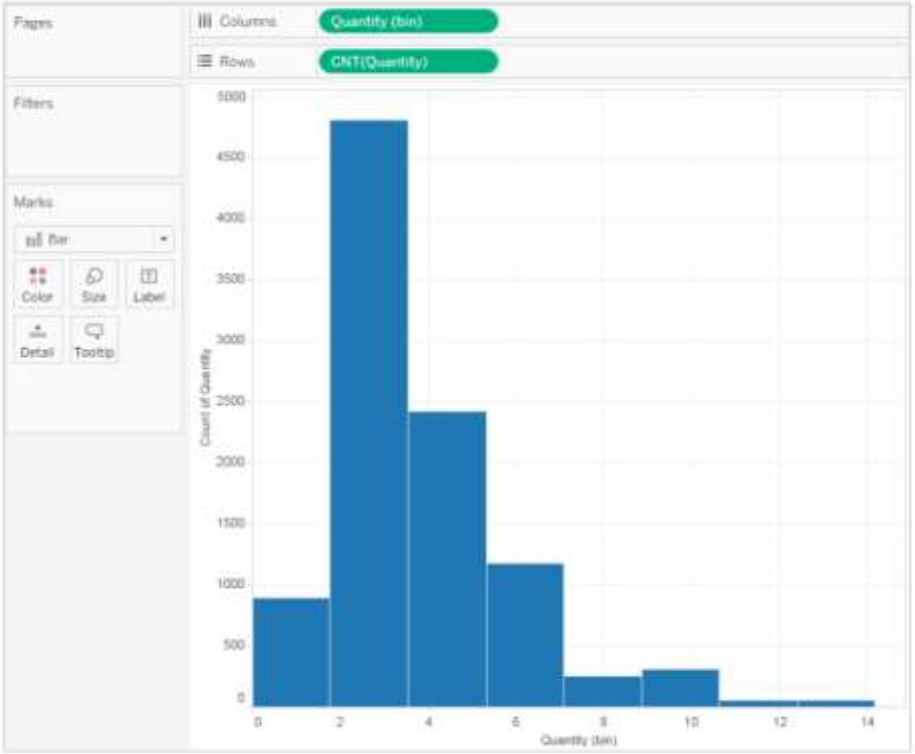
Steps to create Histogram

1. Connect to the Sample - Superstore data source.
2. Drag Quantity to Columns.
3. Click Show Me on the toolbar, then select the histogram chart type.

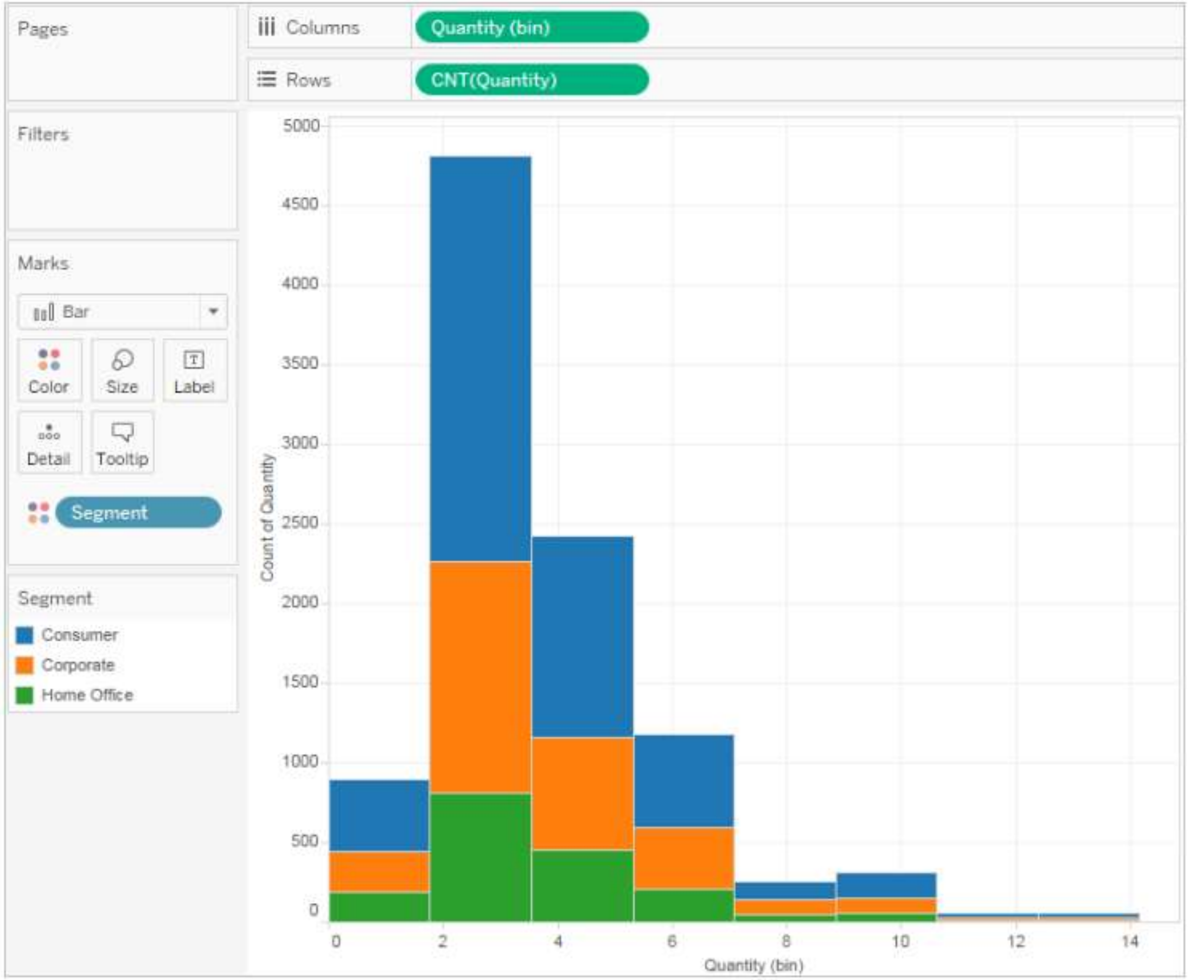
The histogram chart type is available in Show Me when the view contains a single measure and no dimensions.

Three things happen after you click the histogram icon in Show Me:

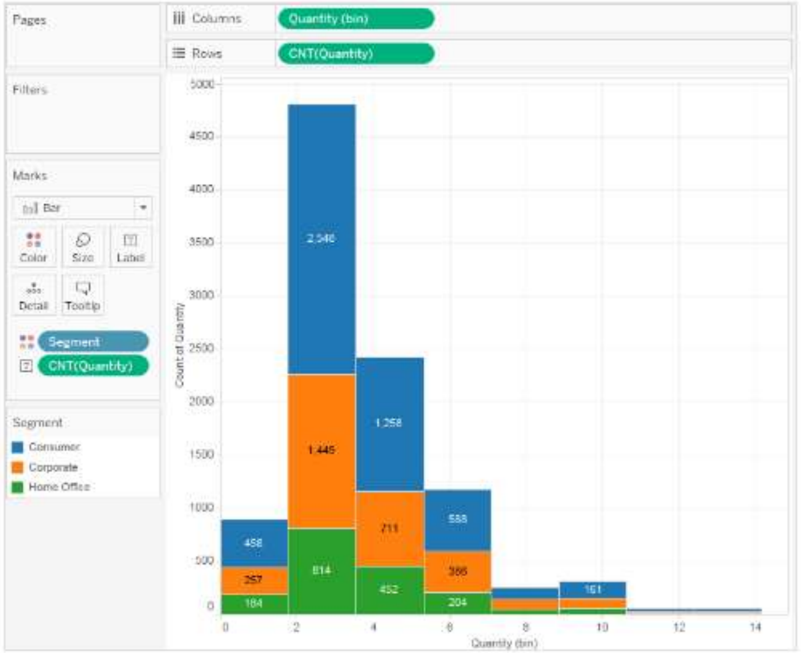
- The view changes to show vertical bars, with a continuous x-axis (1 – 14) and a continuous y-axis (0 – 5,000).
- The Quantity measure you placed on the Columns shelf, which had been aggregated as SUM, is replaced by a continuous Quantity (bin) dimension. (The green color of the field on the Columns shelf indicates that the field is continuous.)
- To edit this bin: In the Data pane, right-click the bin and select Edit in Shelf.
- The Quantity measure moves to the Rows shelf and the aggregation changes from SUM to CNT (Count).



4. Drag **Segment** to **Color**.



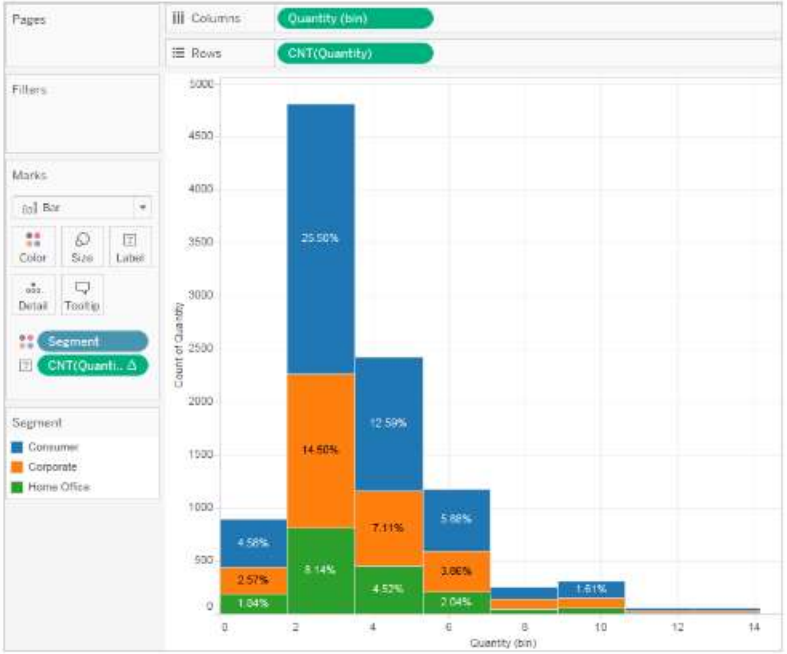
5. Hold down the Ctrl key and drag the **CNT(Quantity)** field from the **Rows** shelf to **Label**.



Holding down the Ctrl key copies the field to the new location without removing it from the original location.

6. Right-click (Control-click on a Mac) the **CNT(Quantity)** field on the **Marks** card and select **Quick Table Calculation** > **Percent of Total**.

   Now each colored section of each bar shows its respective percentage of the total quantity:



   But we want the percentages to be on a per-bar basis.

Visualizing multiple axes to compare different measures

- More than one axis may be needed to compare different measures, understand correlation, or analyze the same measure at different levels of detail.
- Scatter Plot
- Dual Axes and Combinational Chat

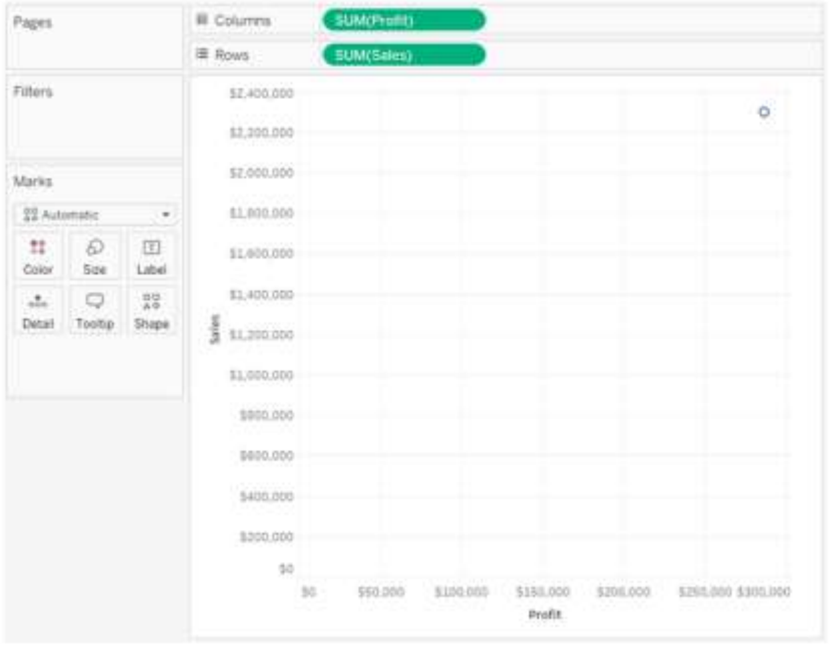
Scatter Plot

A scatterplot is an essential visualization type for understanding the relationship between two measures.

A scatter plot can use several mark types. By default, Tableau uses the shape mark type. Depending on your data, you might want to use another mark type, such as a circle or a square.

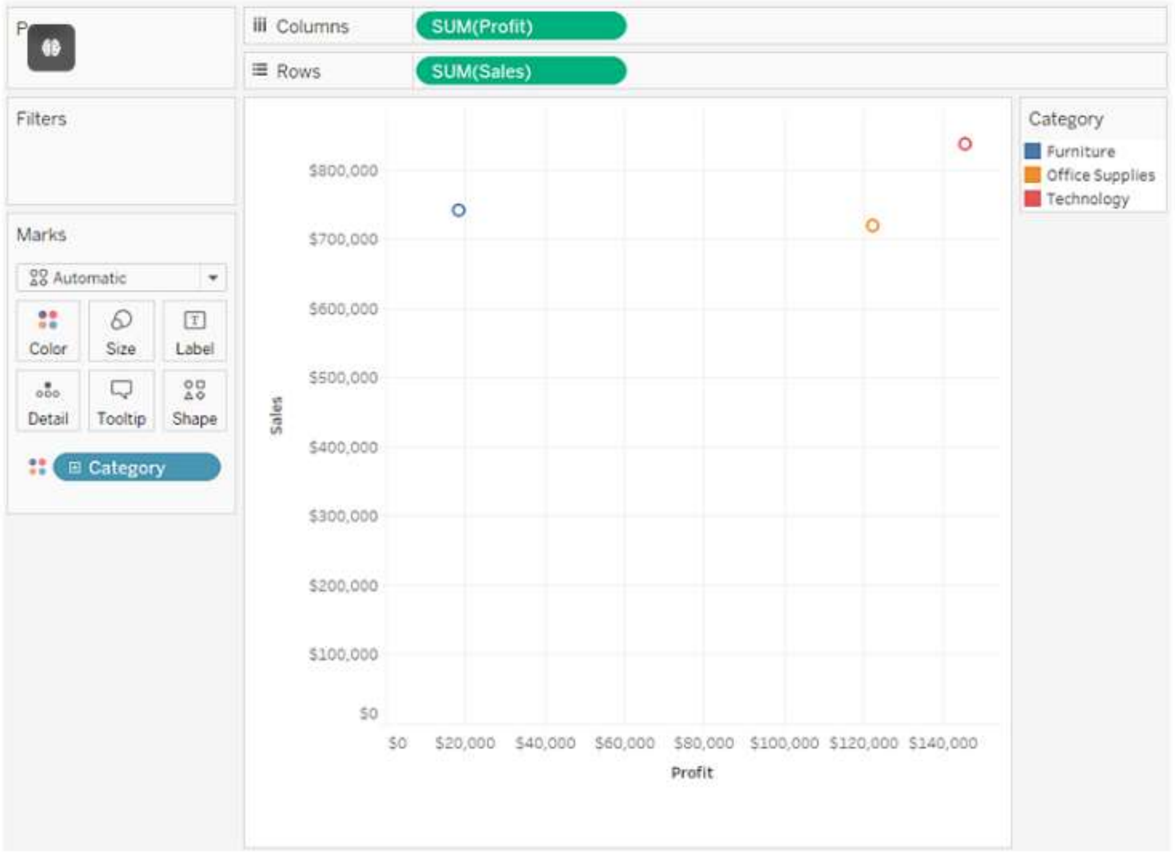
Steps to create scatter plot

1. Open the **Sample – Superstore** data source.
2. Drag the **Profit** measure to **Columns**.
3. Tableau aggregates the measure as a sum and creates a horizontal axis.
3. Drag the **Sales** measure to **Rows**.



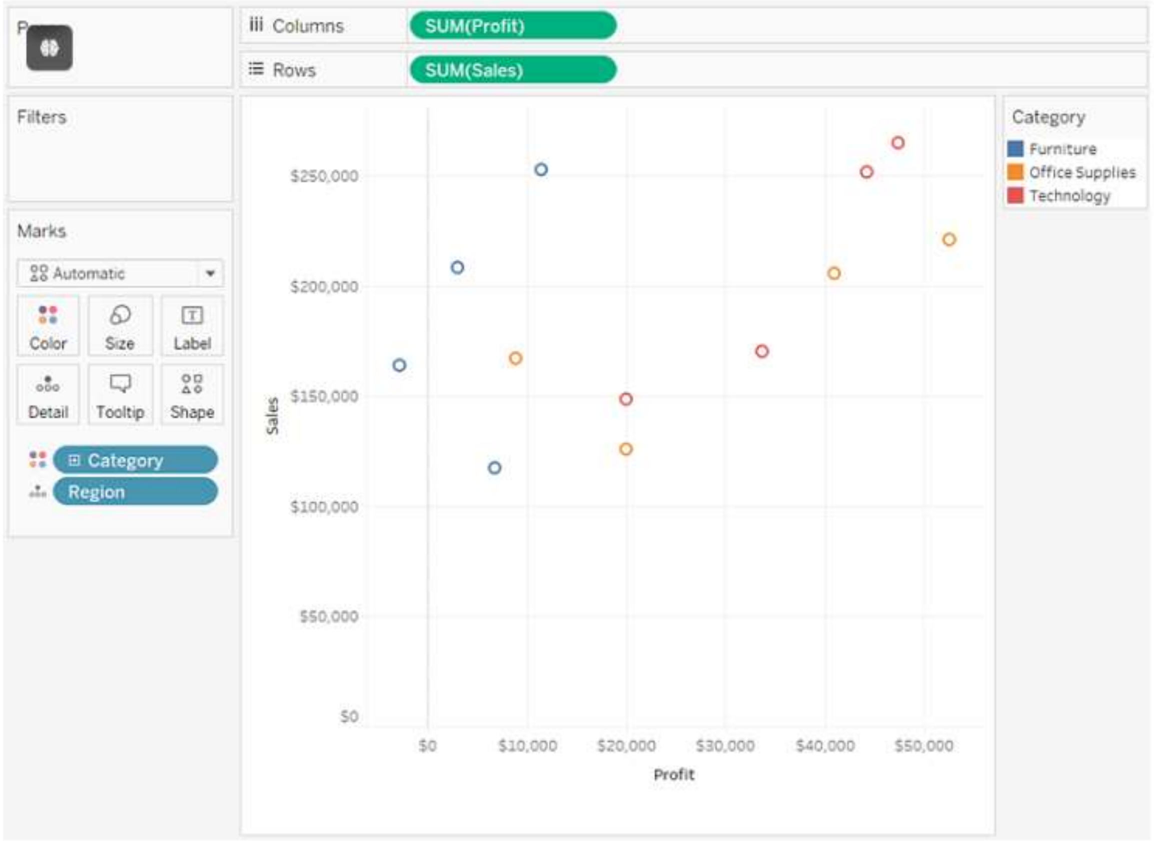
4. Drag the **Category** dimension to **Color** on the Marks card.

This separates the data into three marks—one for each dimension member—and encodes the marks using color.

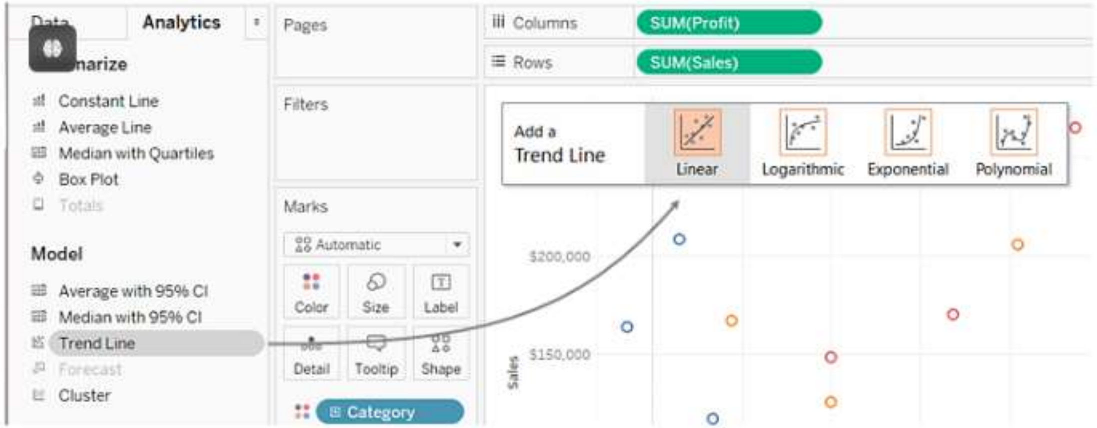


5. Drag the **Region** dimension to **Detail** on the **Marks** card.

Now there are many more marks in the view. The number of marks is equal to the number of distinct regions in the data source multiplied by the number of departments. (If you're curious, use the **Undo** button on the toolbar to see what would have happened if you'd dropped the **Region** dimension on **Shape** instead of **Detail**.)

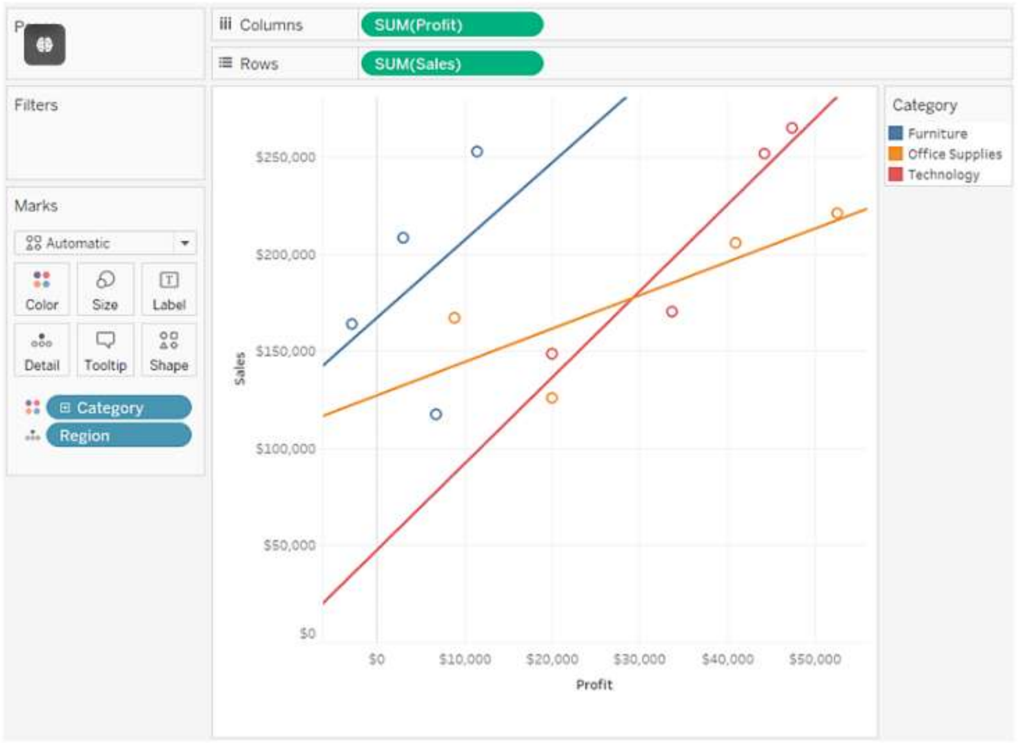


6. To add trend lines, from the **Analytics** pane, drag the **Trend Line** model to the view, and then drop it on the model type.

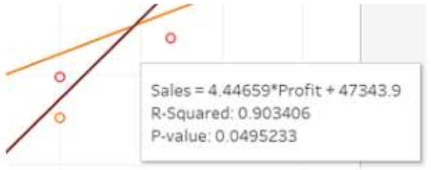


A trend line can provide a statistical definition of the relationship between two numerical values. To add trend lines to a view, both axes must contain a field that can be interpreted as a number—by definition, that is always the case with a scatter plot.

Tableau adds three linear trend lines—one for each color that you are using to distinguish the three categories.



7. Hover the cursor over the trend lines to see statistical information about the model that was used to create the line:

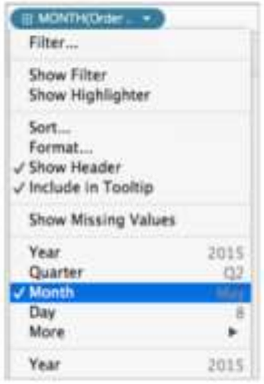


Sales = 4.44659*Profit + 47343.9
R-Squared: 0.903406
P-value: 0.0495233

Dual Axis and Combination Chart

- Combination charts are views that use multiple mark types in the same visualization. For example, you may show sum of profit as bars with a line across the bars showing sum of sales.
- Steps to create a combination chart

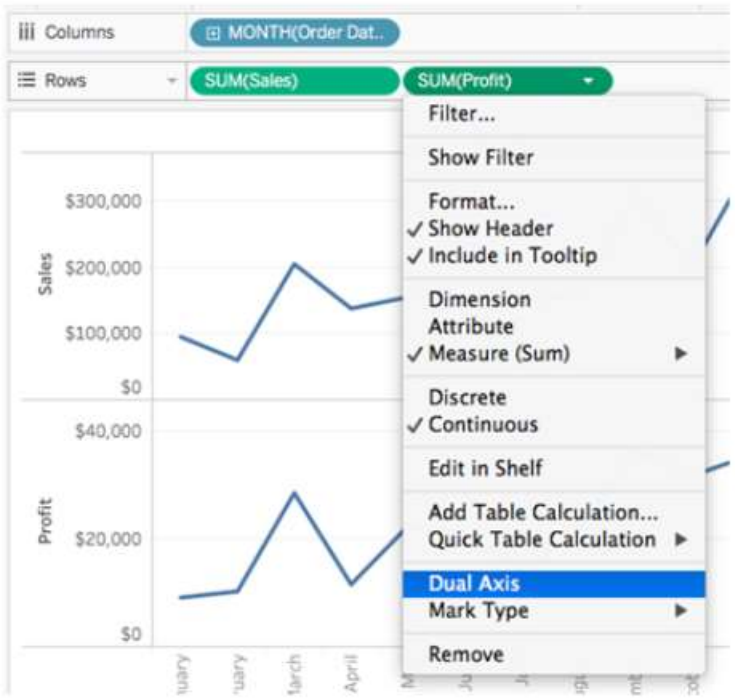
1. Open Tableau Desktop and connect to the **Sample - Superstore** data source.
2. Navigate to a new worksheet.
3. From the **Data** pane, drag **Order Date** to the **Columns** shelf.
4. On the Columns shelf, right-click **YEAR(Order Date)** and select **Month**.



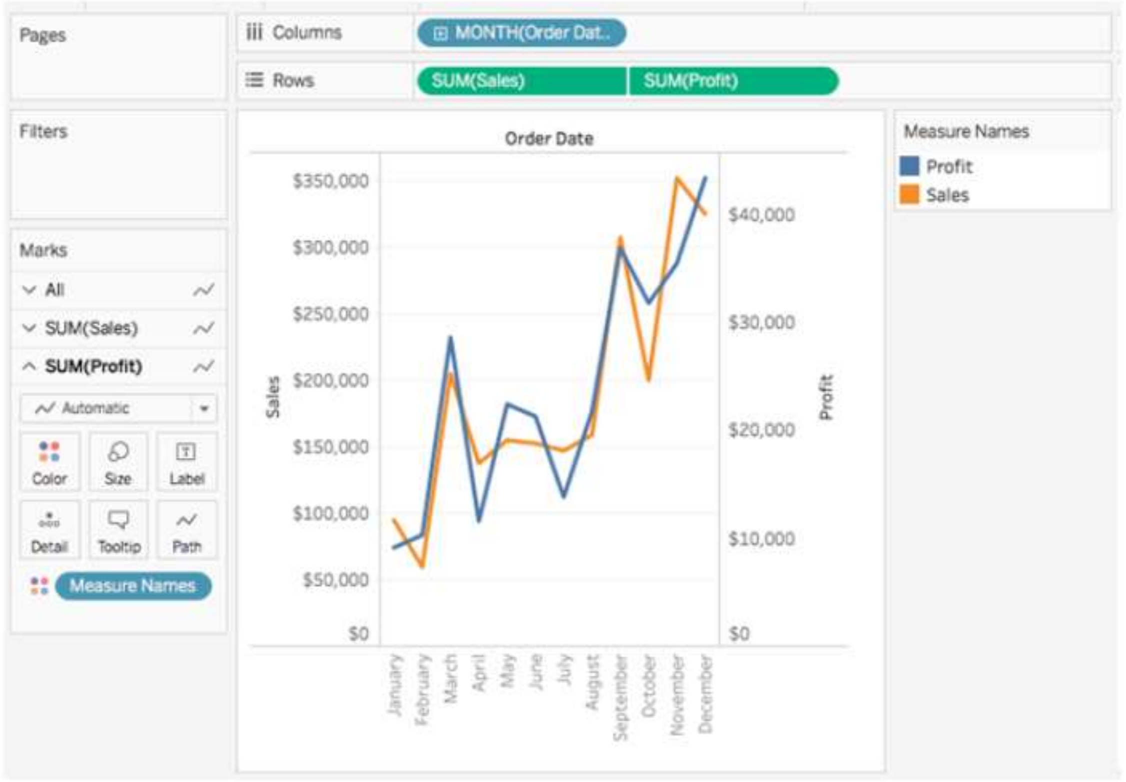
5. From the **Data** pane, drag **Sales** to the **Rows** shelf.

6. From the **Data** pane, drag **Profit** to the **Rows** shelf and place it to the right of SUM(Sales).

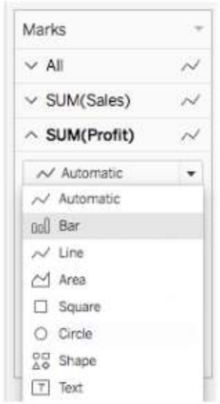
7. On the Rows shelf, right-click **SUM(Profit)** and select **Dual-Axis**.



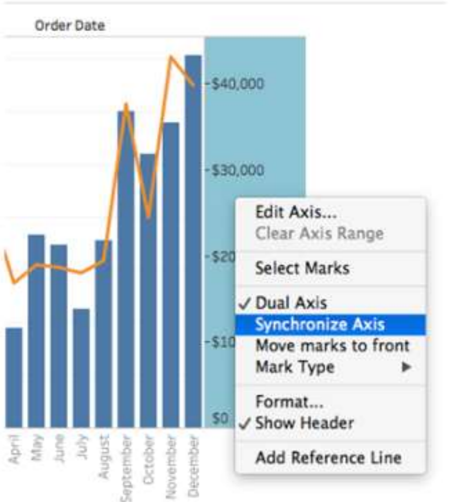
The view updates. Measure Names is added to Color on the Marks card to differentiate the lines.



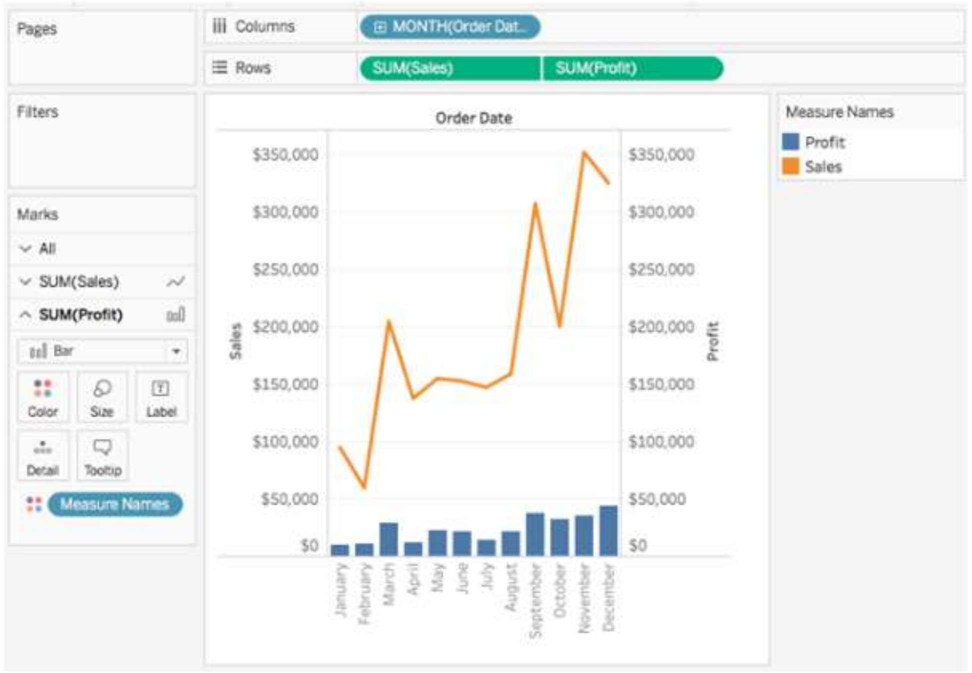
8. On the SUM(Profit) Marks card, click the Mark Type drop-down and select **Bar**.



9. In the visualization, right-click the **Profit** axis and select **Synchronize Axis**.



The view updates to look like this:

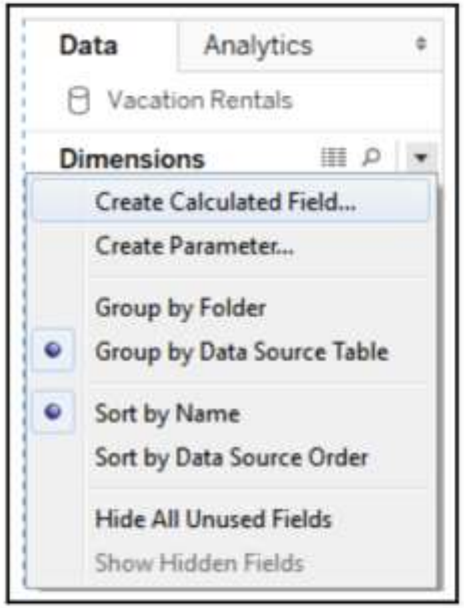


Introduction to calculations

- A calculation is often referred to as a Calculated Field in Tableau because, in most cases, when you create a calculation, it will show up as either a new measure or dimension in the data pane.
- Calculations consist of code that's made up of functions, operations, and references to other fields, parameters, constants, groups, or sets.
- This code returns a value. Sometimes, this result is per row of data, and sometimes it is done at an aggregate level.

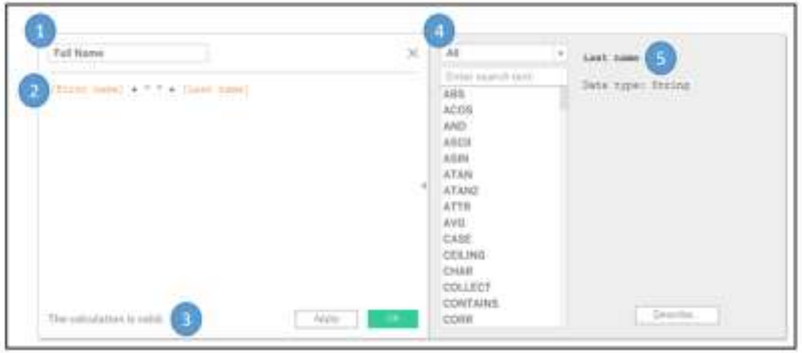
Creating and editing calculations

- There are multiple ways to create a calculated field in Tableau
- Select Analysis | Create Calculated Field... from the menu. Use the drop-down menu next to Dimensions in the data pane:
- Right-click an empty area in the data pane and select Create Calculated Field…
- Use the drop-down menu on a field, set, or parameter in the data pane and select Create | Calculated Field…
- Double-click an empty area on the Rows, Columns, or Measure Values shelves, or in the empty area on the Marks card to create an ad hoc calculation.



The interface for creating and editing calculations looks like this:

- Calculated field name: Enter the name of the calculated field here. Once created, the calculated field will show up as a field in the data pane with the name you entered in this text box.
- Code editor: Enter code in this text area to perform the calculation. The editor includes autocomplete for recognized fields and functions. Additionally, you may drag fields, sets, and parameters from the data pane or view into the code editor to insert them into your code.
- An indicator at the bottom of the editor will alert you to errors in your code.
- The functions list contains all of the functions that you can use in your code.
- Selecting a function in the list or clicking a field, parameter, or function in the code will reveal details about the selection on the right. This is helpful when nesting other calculated fields in your code when you want to see the code for that particular calculated field, or when you want to understand the syntax for a function



```
[First Name] + " " + [Last Name]
```

The calculation is valid.

Last Name

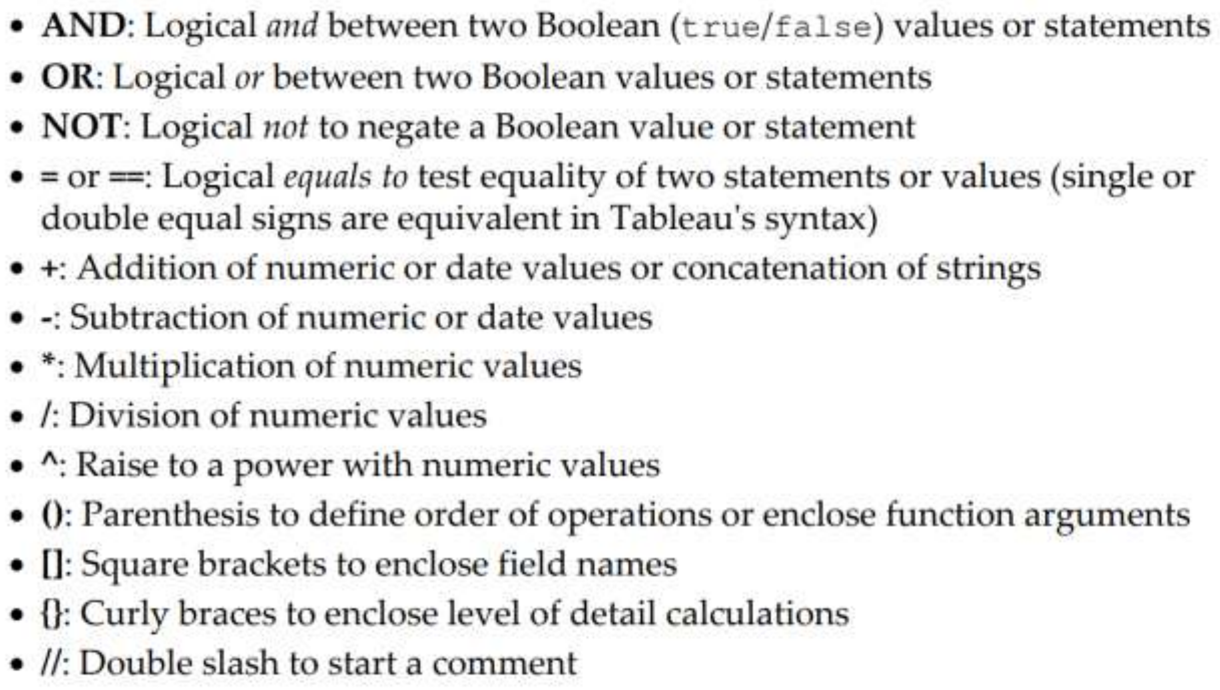
Data type: String

List of Functions used Calculated Field

- **Number:** Mathematical functions, such as rounding, absolute value, trig functions, square roots, and exponents.
- **String:** Functions that are useful for string manipulation, such as getting a substring, finding a match within a string, replacing parts of a string, and converting a string value to uppercase or lowercase.
- **Date:** Functions that are useful for working with dates, such as finding the difference between two dates, adding an interval to a date, getting the current date, and transforming strings with nonstandard formats into dates.
- **Type Conversion:** Functions that are useful for converting one type of field to another, such as converting integers into strings, floating point decimals into integers, or strings into dates.
- **Logical:** Decision-making functions, such as if then else logic or case statements.
- **Aggregate:** Functions that are used for aggregating such as summing, getting the minimum or maximum values, or calculating standard deviations or variances.
- **Pass Through:** (only available when connected live to certain databases, such as SQL Server): These functions allow you to pass through raw SQL code to the underlying database and retrieve a returned value at either a row level or aggregate level.
- **User:** Functions that are used to obtain usernames and check whether the current user is a member of a group. These functions are often used in combination with logical functions to customize the user's experience or to implement user-based security when publishing to Tableau Server or Tableau Online.
- **Table calculation:** These functions are different from the others. They operate on the aggregated data after it is returned from the underlying data source and just prior to the rendering of the view. These are some of the most powerful functions in Tableau.

Additional functions and operators

- Tableau supports numerous functions and operators. In addition to the functions that are listed on the calculation screen, Tableau supports the following operators, keywords, and syntax conventions:

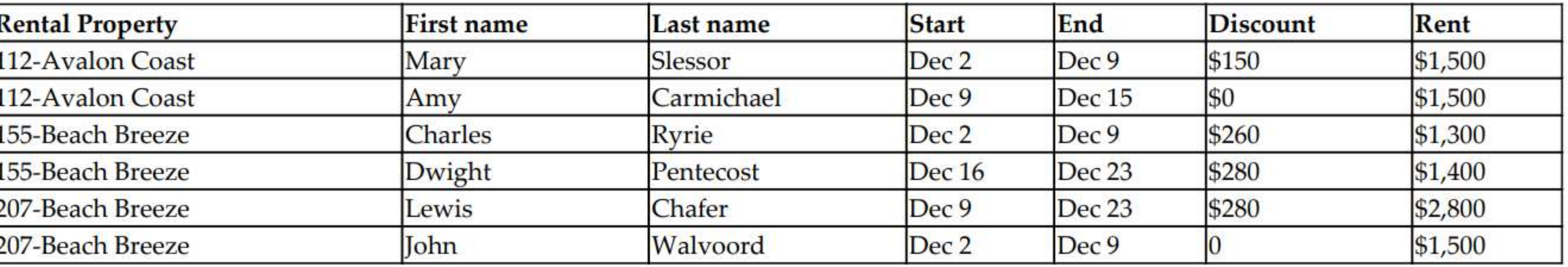


- **AND**: Logical *and* between two Boolean (`true/false`) values or statements
- **OR**: Logical *or* between two Boolean values or statements
- **NOT**: Logical *not* to negate a Boolean value or statement
- = or ==: Logical *equals to* test equality of two statements or values (single or double equal signs are equivalent in Tableau's syntax)
- +: Addition of numeric or date values or concatenation of strings
- -: Subtraction of numeric or date values
- *: Multiplication of numeric values
- /: Division of numeric values
- ^: Raise to a power with numeric values
- (): Parenthesis to define order of operations or enclose function arguments
- []: Square brackets to enclose field names
- {}: Curly braces to enclose level of detail calculations
- //: Double slash to start a comment

Four main types of calculations

- Row-level calculations: These calculations are performed for every row of underlying data.
- Aggregate-level calculations: These calculations are performed at an aggregate level, which is usually defined by the dimensions used in the view.
- Level of detail calculations: These special calculations are aggregations that are performed at a specified level of detail, with the results available at the row level.
- Table calculations: These calculations are performed at an aggregate level on the table of aggregate data that has been returned by the data source to Tableau.

Example Data



Rental Property	First name	Last name	Start	End	Discount	Rent
112-Avalon Coast	Mary	Slessor	Dec 2	Dec 9	$150	$1,500
112-Avalon Coast	Amy	Carmichael	Dec 9	Dec 15	$0	$1,500
155-Beach Breeze	Charles	Ryrie	Dec 2	Dec 9	$260	$1,300
155-Beach Breeze	Dwight	Pentecost	Dec 16	Dec 23	$280	$1,400
207-Beach Breeze	Lewis	Chafer	Dec 9	Dec 23	$280	$2,800
207-Beach Breeze	John	Walvoord	Dec 2	Dec 9	0	$1,500

Row Level Calculations

Row-level calculations are calculated at a row level, but you can choose to aggregate the results.

Example

Name the first Room with the following code:

```
SPLIT([Rental Property], "-", 1)
```

Then, create another calculated field named Building with the following code:

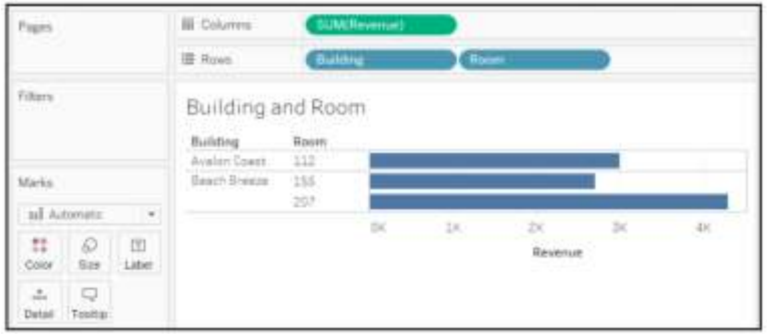
```
SPLIT([Rental Property], "-", 2)
```

Both of these functions use the Split() function, which splits a string into multiple values and keeps one of those values.

This function takes three arguments: the string, the delimiter (a character or set of characters that separate values), and the token number (which value to keep from the split, that is, 1st, 2nd, 3rd, and so on.)

Using the - (dash) as the delimiter, the Room is the first value and Building is the second.

Using the two calculated fields, create a bar chart of Revenue per Building and Room, similar to this:



Columns: SUM(Revenue)

Rows: Building, Room

Building and Room

Building	Room
Avalon Coast	112
Beach Breeze	155
	207

Example 2

Create a calculated field named Floor with the following code:

```
 IF LEFT([Room], 1) = "1"
 THEN "First Floor“
 ELSEIF LEFT([Room], 1) = "2“
 THEN "Second Floor“
 END
```

This code uses the LEFT() function to return the leftmost character of the room. Thus, 112 gives a result of 1; 207 gives a result of 2. The IF THEN END logic allows us to assign a result (either First Floor or Second Floor), depending on which case is true.

- **Example 3**

```
LEFT([Room], 1)
```

This code simply returns the leftmost character of the room number. We'll get 3 for 306 and 8 for 822.

Create a new calculated field called Full Name with the following code:

```
[First name] + " " + [Last name]
```

This code concatenates the strings of First name and Last Name with a space in between them.

We now have a single field that will display the full name of the individual renter.

Aggregate-level calculations

1. Create a calculation named Discount % with the following code:

SUM([Discount]) / SUM([Rent])

This code indicates that the sum of the Discount should be divided by the sum of Rent.
This means that all of the values of Discount will be added and all of the values of Rent will be added.
Only after the sums are calculated will the division occur.

2. Calculate the average occupancy rate of vacation rentals over a specific period.

SUM([Nights Booked]) / SUM([Total Available Nights])

3. Measure the revenue generated per available night for the rental property.

SUM([Revenue]) / SUM([Total Available Nights])

4. Calculate the average daily rate charged for occupied nights.

SUM([Revenue]) / SUM([Nights Booked])

5. Express the occupancy rate as a percentage

(SUM([Nights Booked]) / SUM([Total Available Nights])) * 100

•Calculate the average length of stay per booking.

6. Measure the average number of days between booking and check-in.

AVG(DATEDIFF('day', [Booking Date], [Check-In Date]))

7. Calculate the percentage of bookings that were canceled

(COUNT(IF [Booking Status] = 'Canceled' THEN 1 END) / COUNT([Booking ID])) * 100

8. Calculate the total revenue generated by different types of properties (e.g., condo, house).

{FIXED [Property Type]: SUM([Revenue])}

9. Calculate the average length of stay per booking.

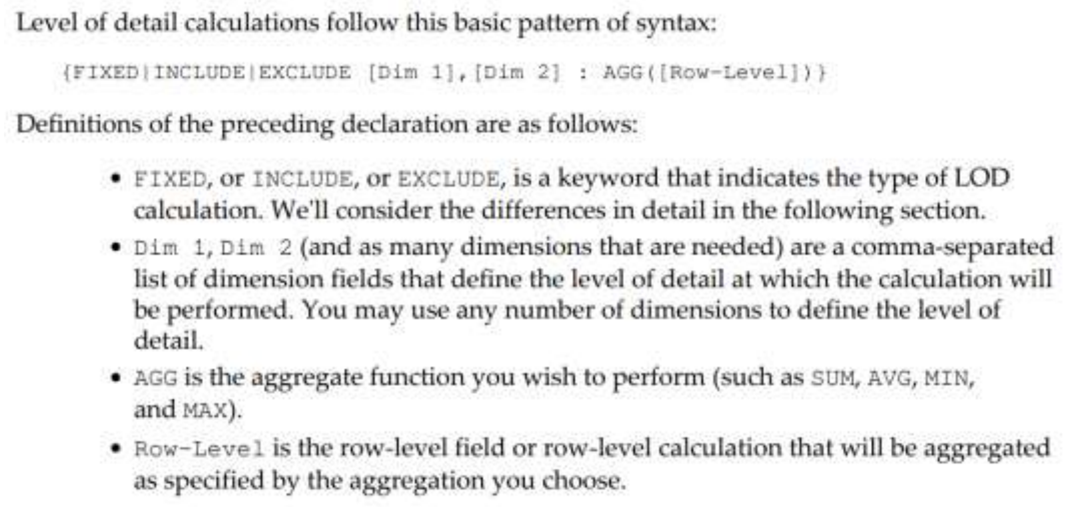
AVG(DATEDIFF('day', [Check-In Date], [Check-Out Date]))

10. Measure the growth rate of revenue over a specific period (e.g., month-over-month).

(SUM([Revenue]) - LOOKUP(SUM([Revenue]), -1)) / LOOKUP(SUM([Revenue]), -1)

Level Of Detail Calculations

- Level of detail calculations (sometimes referred to as LOD calcs or LOD expressions) are a fourth type of calculation that allows you to perform aggregations at a specified level of detail, which may be different from the level of detail that was defined in the view.
- **Level of detail syntax**



Level of detail calculations follow this basic pattern of syntax:

```
{FIXED|INCLUDE|EXCLUDE [Dim 1],[Dim 2] : AGG([Row-Level])}
```

Definitions of the preceding declaration are as follows:

- `FIXED`, or `INCLUDE`, or `EXCLUDE`, is a keyword that indicates the type of LOD calculation. We'll consider the differences in detail in the following section.
- `Dim 1`, `Dim 2` (and as many dimensions that are needed) are a comma-separated list of dimension fields that define the level of detail at which the calculation will be performed. You may use any number of dimensions to define the level of detail.
- `AGG` is the aggregate function you wish to perform (such as `SUM`, `AVG`, `MIN`, and `MAX`).
- `Row-Level` is the row-level field or row-level calculation that will be aggregated as specified by the aggregation you choose.

Level of detail types

Three types of level of detail calculations are used in Tableau: `FIXED`, `INCLUDE`, and `EXCLUDE`.

FIXED

Fixed level of detail expressions aggregate at the level of detail that's specified by the list of dimensions in the code, *regardless* of what dimensions are in the view. For example, the following code returns the average `Rent` per `Building`, regardless of what other dimensions are in the view:

```
{FIXED [Building] : AVG([Rent])}
```

The following two snippets of code represent a fixed calculation of the average rent for the entire data source (or the subset defined by a context filter):

```
{FIXED : AVG([Rent])}
```

This is the other code snippet you can use:

```
{AVG([Rent])}
```

INCLUDE

Include level of detail expressions aggregate at the level of detail that's determined by the dimensions in the view, *along with* the dimensions listed in the code. For example, the following code calculates the average rent at the level of detail that's defined by dimensions in the view, but includes the dimension `Room`, even if `Room` is not in the view:

```
{INCLUDE [Room] : AVG([Rent])}
```

EXCLUDE

Exclude level of detail expressions aggregate at the level of detail determined by the dimensions in the view, *excluding* any listed in the code. For example, the following code calculates the average price at the level of detail defined in the view, but does not include the `Apartment` dimension as part of the level of detail, even if `Apartment` is in the view:

```
{EXCLUDE [Apartment] : AVG([Price])}
```