Tableau Course Notes

What is Tableau?

-> Tableau is a Visualization Tool
-> It is a Business Intelligence Tool, allows any one to connect to the Data and analyzes the Data with in a few clicks in the form of Visuals
-> Tableau is easy as 'MS Excel', but powerful enough to satisfy an Experts even the most

complex analytical Problems
-> Tableau is derived from a French word 'TABLEAX' which means Pictures
-> Tableau is having its own Query Language called VIZQL (Visualization Query Language)
-> VIZQL is a Combination of SQL and descriptive language for Rendering Graphics
-> VIZQL automatically converts any data in the form of Text to Visuals

Where Tableau?

Source DBs -> ETL Transformations(ETL Tools) -> Data Warehouse -> Reports/Dashboards

(Reporting Tools)

Why Tableau?

Tableau is an In-Memory BI Tool

BI Tools

Traditional BI Tools

-> Perform all the Calculations based on Hard Disk Data
Ex:
Informatica
Datastage
SAP-BW
ODI
SAP-BO
OBIEE
Cognos
SSIS
SSRS
...
....

In-Memory BI Tools

-> Perform all the Calculations based on RAM Data

-> Compared to Traditional BI Tools, In-Memory BI Tools are very very Fast

-> Minimum 10 to 100 times Faster

Ex:

Qlikview

Tableau

Spotfire etc...

How Tableau different from other BI Tools?

1.Visual Analytics

-> In Tableau, we can analyze anything in the form of Visual

-> Tableau takes input as Textual Data and its output is in the form of Visuals

-> In the Background, it will convert Textual Data into Visuals by making use of SQL

-> Another great feature for Tablaeu is, It will generate the code by its own

-> In Tableau, no need of writing the code manually

2. Faster Performance

-> Tableau by making use of its brand new features like TABLEAU Data Engine, Performance

Recorder, Paralized Dashboards, It is very very Fast compared to other BI Tools

-> Its Performance is 100 times faster than any other BI Tool in the market currently

3. Any Data

-> By making use of Tableau, We can perfrom analysis on any kind of Data right from DWH to

Hadoop and from Hadoop to Excel and even from Excel to Salesforce data etc...

4. Business Integration

-> Tableau is tightly coupled with any kind of Business and It perform the analysis of any

Business very efficiently and accurately with in a lesser amount of Time

5. Web and Mobile Authoring

-> By making use of Tableau, We can securely Share the analysis Reports to anyone

according to that needs over the Web and even to their Mobiles also

Note:

-> In order to see the Report from the Mobile, the End User needs to install the Tableau

App in his Mobile

-> Currently Tableau App is supported on android and IOS Devices

Tableau Components

-> In Tableau, We have 5 Different components

1. Tableau Desktop
2. Tableau Server
3. Tableau Reader
4. Tabelau Public
5. Tableau Online

1. Tableau Desktop

-> In Tableau, If we want to design (or) create anything we need to use Tableau Desktop

-> Tabelau Desktop is the place where all the Development work will be done

-> It is the core platform for all Tableau Developers

2. Tableau Server

-> In Tableau, If we want to share anything globally, then we need to make use of Tableau

Server

-> It is a Web based Component

-> Tableau Server allows any user to share the Visual Analysis with any one by making use

of a Web Browser

3. Tableau Reader

-> In Tableau, If we want to share anything locally, we need to make use of Tableau Reader

-> We cannot create or design any Workbook

-> To access Tableau Workbooks locally, We need to install Tableau Reader software

Ex: similar to Adobe Reader to access PDF Documents

4. Tableau Public

-> Useful to share anything publicly
-> We can create any workbook by using Tableau Public
-> No Security. Any user can access these workbooks/Dashboards

5. Tableau Online

-> Hosted version of Tabelau Server in the Cloud
-> Allows any user to perfrom the Rapidfire Business Analytics in the cloud

Tabelau Architecture

-> Clients
-> Gateway/Load Balancer
-> Main componenets
    -> Appication Server
    -> Data Server
    -> VizQL Server

-> Data Connections
    -> SQL Connector
    -> MDX Connector
    -> Fast Data Engine
    -> Repository
-> Customer Data

Clients

-> Several ways to access Tableau Workbooks/Dashboards
-> End users can access data(Dashboards/Workbooks)
 By using Tableau Desktop
  (or)
 By using Web Browsers like IE, Firefox,Chrome etc...
  (or)
 By using Mobile Devices like IPhone, Android etc...

Gateway/Loadbalancer

-> It will take all Requests from Clients and send them to appropriate components like

Application Server, Data Server and VizQL Server

Application Server

-> It will take care of all Accesses and Permissions

Data Server

-> It will take care of all Database Connections

VizQL Server

-> It will take care of Visualizations (Graphics)

Data Connectors

-> Useful to pull Data from different kinds of Data Sources

-> We have several Data Connectors like SQL Connector, MDX Connector and Fast Data Engine

-> SQL Connector is useful to pull Data from Relational Data Sources like Oracle, SQL

SERVER,DB2, MySQL, Access etc...

-> MDX Connector is useful to Load Data from Multi Dimensional Cubes

-> Fast Data Engine is useful to extract complete Data from any kind of Data Source and

Store that entire Data Locally in the form of TDE File(Tableau Data Extract)

Repository

-> Tableau Repository Stores all Database Connections, Users Information and all Workbooks

and TDE Files etc...

Customer Data

-> We can Load Data from almost all types of Data Sources like Data Warehouses, Datamarts,

Text Files, Cubes etc...

Tableau Developer

-> Develop Tableau Workbooks/Reports/Dashboards

Input

Source - Data Warehouse, Database like Oracle, SQL Server, DB2.MySQL,Access etc...,

Datamart, Text, CSV, Excel Files etc... and SAP, Salesforce systems etc...

OutPut

Tableau Workbooks

Roles of Tableau Developer

-> Loading Data from Multiple Data Sources
-> Perfroming Some Transformations
-> Calculating Aggregations
-> Cha nging MetaData(Table name and Column Names)
-> Implementing Joins
-> Applying Conditions and Filters
-> Sorting Data
-> Drilling
    Year -> Quarter -> Month -> Week -> Day
    Region -> Country -> State -> District -> City
-> Craating Groups
-> Preparing Visualizations

Tabelau History

-> Invented by 3 people - Chris stolte, Pat Hanrahan and Christian Chabot
-> Professors in Stanford University
-> Their goal is to build a Tool for Business Analysis rendering Graphics
-> 10 Years Research
-> Released Tableau on 8th June 2008

Tableau Environment

Data Section
Workbooks Section

Data Section

-> The Data Sections lets any user to connect to new Data Source (or) Quickly opens a

saved connection
-> We can save any Data Connection
-> By Default all Data Sources will get stored in the Tabelau Repository under Data

Sources Folder

Workbooks Section

-> It contains all the workbooks which we have opened recently

-> First time this section will be blank
-> As we create new workbook and Save, it will be added to Workbooks Section

Workbook

-> Tableau Workbook is similar to MS Excel Workbook
-> Tableau Workbook contains multiple Sheets which can either be a Worksheet (or) a

Dashboard

WorkSheet

-> It is the smallest Development Object in Tableau
-> A WorkSheet Contains the Single View of Data which contains Shelves, Legends and Data

Window

Dashboard

-> A Dashboard is a collection of multiple Worksheets

Note:

-> within a Workbook, we can create new sheets, we can duplicate sheets, we can delete

sheets, even we can hide or show sheets

Data Window

-> In Tableau Work Space the first part is called as Data Window
-> Data Window Contains
1.Dimensions
2.Measures
3.Sets
4.Parameters

Show Me Option

-> It is the Place, where list of different Data Views are available. any user can select

appropriate view in the Show Me Window

Shelves

-> Shelves are the part of every worksheet

-> In Tableau, We have different Shelves

1. Column Shelves
2. Row Shelves

1. Column Shelves

-> It is the place where we can Drag and Drop the Data Fields

2. Row Shelves

-> It is the place where we can Drag and Drop the Data Fields

Filter Shelves

-> Filter Shelves is the Place where we can Drag and Drop the Data Fields in order to

Perform Filtering

Legends

1.Color Legend

-> Useful to display Data View in Different colors

2.Size Legend

-> Useful to display Data View in Different Sizes

3.Shape Legend

-> Useful to display Data View in Different Shapes

4.Map Legend

-> Useful to display Data View in the Form maps

Marks Card

-> It is the container which contains Different Shapes

Creating Data Connections in Tableau

-> In Tableau, We have 2 types of Data Connections
1. Live Connection
2. Extract Connection

1. Live Connection

-> If we define Connection Type as Live, then there will be a Live Communication between

Tableau and Data Source
-> It sends dynamic SQL or MDX Statements directly to the Source Database
-> It dont import all data
-> Keeps the detail data in Source System and sends the aggregate results of Queries to

Tableau
-> It can effctively handle unlimited amounts of Data

2. Extract Connection

-> The second type of connection in Tableau is Extract
-> The Extract first connect to the Data Source and takes the snapshot of the Data Source

and places the snapshot into Tableau Data Engine
-> This Tableau Data Engine reads data from the snapshot and it creates a local copy of

the Data Source in the form of Tableau Data Extract(TDE) file which gets stored in the

Data Sources folder of the Tabelau Repository

-> Whenever we make request, Tabelau interact with the TDE file and keep the data in RAM
-> It will do all the calculations and aggregations based on RAM Data

Note

-> If source system is not optimized then prefer Extract
-> If you have less amounts of data then prefer Extract
-> If you want to prepare visualizations on a weekly or monthly basis then prefer Extract

Dimensions

-> This section contains columns containing information related to Text and Dates

Measures

-> This section contains columns containing information related to numerical values like

Sales, Quantity, Profit,
Revenue etc...

Datatypes

-> Datatype reflects the kind of Information stored in that Field

Changing Data Type

Right Click on Column Name -> Change Data Type -> Give new Type
****

Joins

****

* Useful to establish relationship between two tables
* Joins are performed on Link Tables of data together within a single data source
* Any project database consists of multiple tables of data that are related in some way. Visualize you have been asked to analyze data in a Retail SuperSotre where the data is stored in heterogeneous data sources and multiple tables

- Banking data on Loan Analysis

* Loans * Customer *Geography * Product * Branch * Employee
* Interest Rate * Sales Channel * Loan DateTime * Maturity Period

- Retail data on Sale Analysis

* Sales * Customer * Product * Branch * Employee *Geography
* Sale DateTime

Note:

* When you connect to the database in Tableau, you'll see the tables listed on the left. Always start by adding the primary table.
* Tableau automatically create the joins as you add additional tables, if the referential integrity(common key fields) has been defined in the database

-> We can implement Join in 2 ways
-> By using same column
-> By using same datatype

Tableau Join Types: * Inner Join * Left Join * Right Join * Outer Join

****

Joins

****

Steps:

* Open Tableau --> Select Connect --> Excel File

* Load Excel Data Tables from 'C:\Tableau Daily Practice\Tableau Class Data'

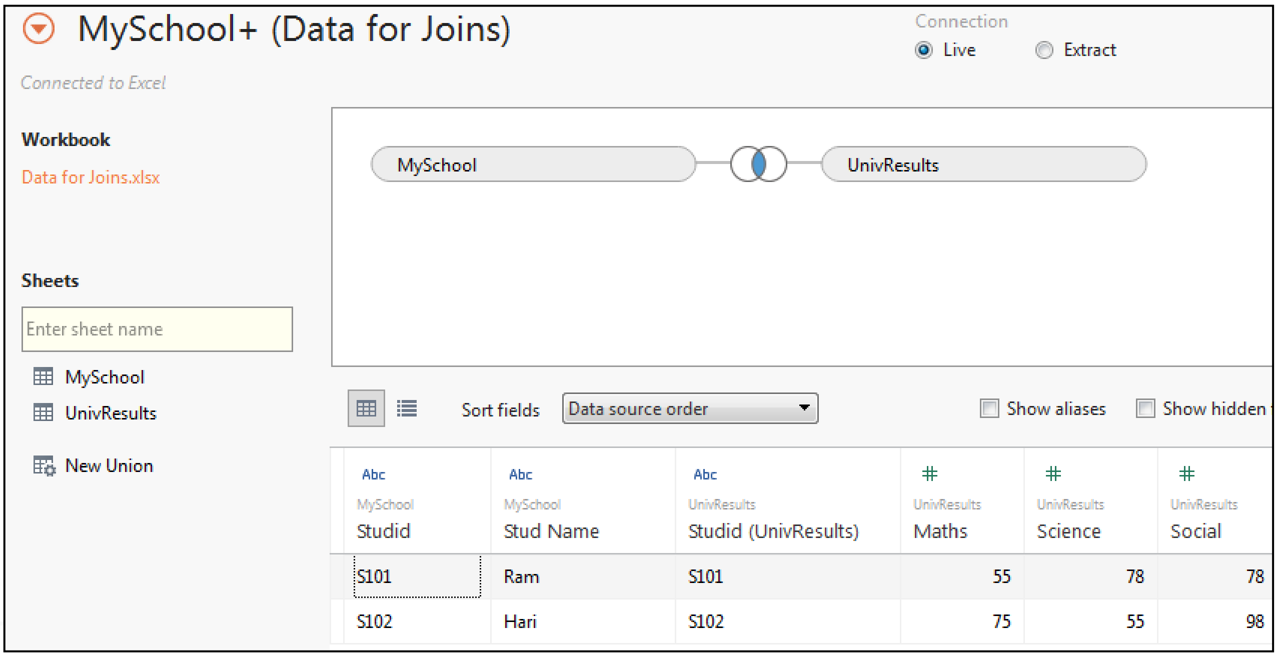
* MySchool.xlsx (Students from MySchool who appeared for the Exams) - Select as First Table

* UnivResults.xlsx (Board of Examination results of All the Schools in the City) - Select as Second Table

* Select Join Type and Create a New Work Sheet

ROWS : Studid / StudName

COLUMNS : Sum(Maths) / Sum(Science) / Sum(Social)



MySchool+ (Data for Joins)

Connected to Excel

Connection ● Live ○ Extract

Workbook

Data for Joins.xlsx

Sheets

Enter sheet name

MySchool

UnivResults

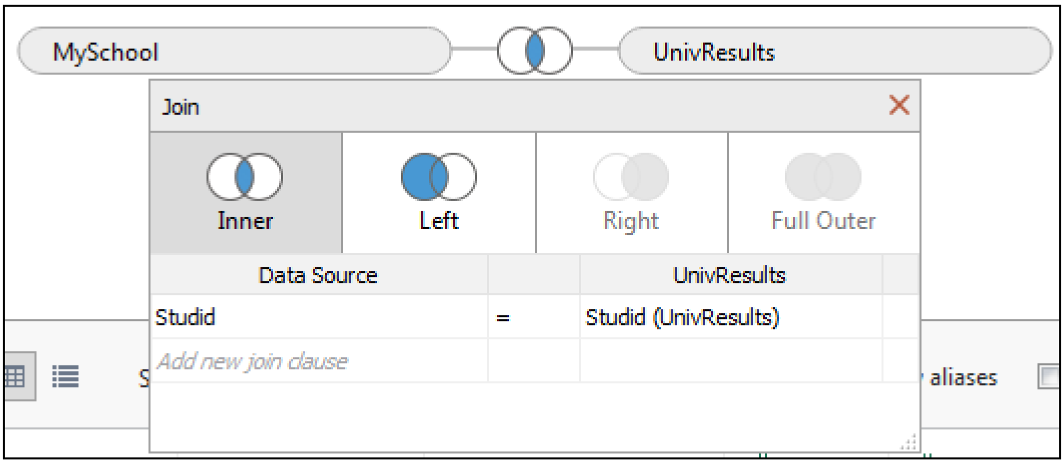
New Union

MySchool ⟷ UnivResults

Sort fields: Data source order

□ Show aliases □ Show hidden

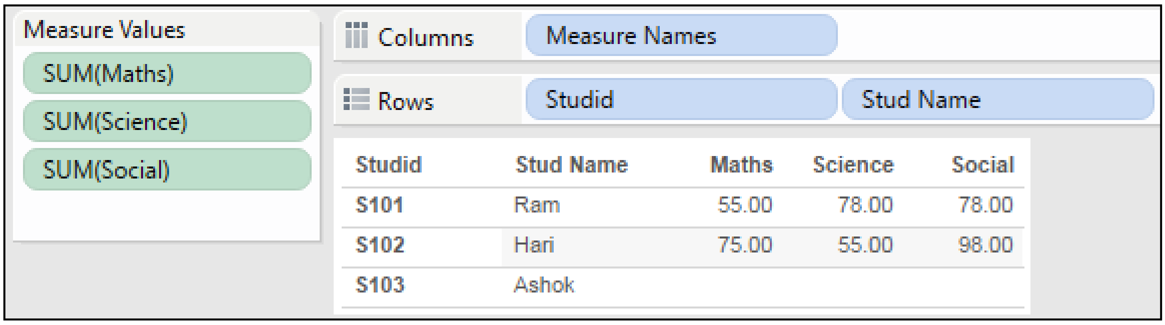
MySchool Studid	MySchool Stud Name	UnivResults Studid (UnivResults)	UnivResults Maths	UnivResults Science	UnivResults Social
S101	Ram	S101	55	78	78
S102	Hari	S102	75	55	98



MySchool ⟷ UnivResults

Join

Data Source		UnivResults
Studid	=	Studid (UnivResults)
Add new join clause		



Measure Values
- SUM(Maths)
- SUM(Science)
- SUM(Social)

Columns: Measure Names

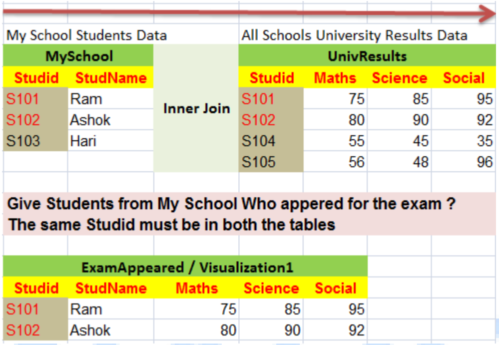
Rows: Studid | Stud Name

Studid	Stud Name	Maths	Science	Social
S101	Ram	55.00	78.00	78.00
S102	Hari	75.00	55.00	98.00
S103	Ashok			

Inner Join

* Reads only Records that Match the Join Condition from both the table on the left and the right

* If there is any matching condition then only records will be loaded



My School Students Data — Inner Join — All Schools University Results Data

MySchool			UnivResults			
Studid	StudName	Inner Join	Studid	Maths	Science	Social
S101	Ram		S101	75	85	95
S102	Ashok		S102	80	90	92
S103	Hari		S104	55	45	35
			S105	56	48	96

Give Students from My School Who appered for the exam ?
The same Studid must be in both the tables

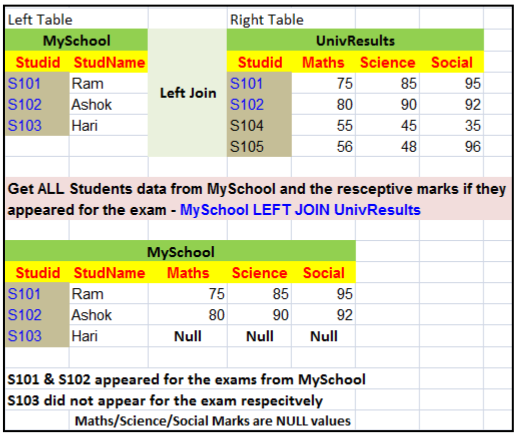
ExamAppeared / Visualization1				
Studid	StudName	Maths	Science	Social
S101	Ram	75	85	95
S102	Ashok	80	90	92

Left Join

* Load all the Records from Left side table

* Only matching Records from Right side Table

* For non matching, It will display NULLs



Left Table — Left Join — Right Table

MySchool			UnivResults			
Studid	StudName	Left Join	Studid	Maths	Science	Social
S101	Ram		S101	75	85	95
S102	Ashok		S102	80	90	92
S103	Hari		S104	55	45	35
			S105	56	48	96

Get ALL Students data from MySchool and the resceptive marks if they appeared for the exam - MySchool LEFT JOIN UnivResults

MySchool				
Studid	StudName	Maths	Science	Social
S101	Ram	75	85	95
S102	Ashok	80	90	92
S103	Hari	Null	Null	Null

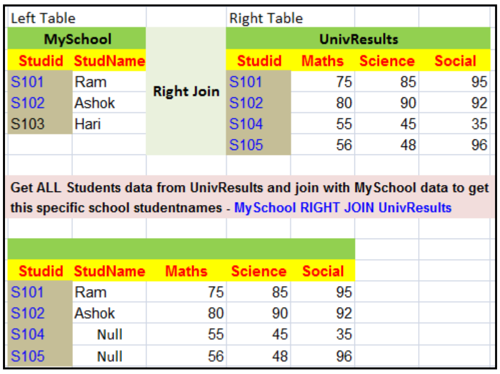
S101 & S102 appeared for the exams from MySchool
S103 did not appear for the exam respecitvely
Maths/Science/Social Marks are NULL values

Right Join

* Load all the Records from Right Side Table

* Only matching Records from Left side Table

* For non matching, It will display NULLs



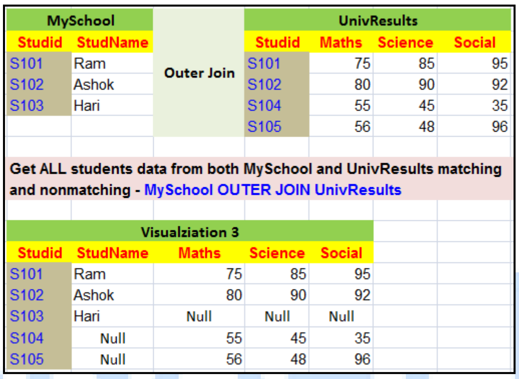
Left Table			Right Table			
MySchool			UnivResults			
Studid	StudName	Right Join	Studid	Maths	Science	Social
S101	Ram		S101	75	85	95
S102	Ashok		S102	80	90	92
S103	Hari		S104	55	45	35
			S105	56	48	96

Get ALL Students data from UnivResults and join with MySchool data to get this specific school studentnames - MySchool RIGHT JOIN UnivResults

Studid	StudName	Maths	Science	Social
S101	Ram	75	85	95
S102	Ashok	80	90	92
S104	Null	55	45	35
S105	Null	56	48	96

Outer Join (Left join + Right Join)

* Load all the Records from both tables matching as well as non matching



MySchool			UnivResults			
Studid	StudName	Outer Join	Studid	Maths	Science	Social
S101	Ram		S101	75	85	95
S102	Ashok		S102	80	90	92
S103	Hari		S104	55	45	35
			S105	56	48	96

Get ALL students data from both MySchool and UnivResults matching and nonmatching - MySchool OUTER JOIN UnivResults

Visualziation 3

Studid	StudName	Maths	Science	Social
S101	Ram	75	85	95
S102	Ashok	80	90	92
S103	Hari	Null	Null	Null
S104	Null	55	45	35
S105	Null	56	48	96

****

Data Blending or Blending Multiple Data Sources

****

- Blending Multiple Data Sources in the same View
- **Example:** Select Coffee Chain (Ms Access Db) and SuperStore Sales (Ms Excel) data sources in the same view
  These data sources are of different types
  * Blend Google Analytics Data with Oracle Data
  * Microsoft Sql Server with Salesforce, etc.
- Data Blending is made at an Aggregate Level and involves different queries sent to respective data source; unlike joining, which is done at a row level and involves a single query to a single data source.
- Steps Data Blending Process:
  1. Tableau issues a query to the primary data source
  2. The underlying data engine returns aggregate results
  3. Tableau issues another query to the secondary data source. This query is filtered based on the set of values returned from the primary source for dimensions that links the two data sources
  4. The underlying data engine returns aggregate results from the secondary data source

Note: * Tableau blends the results of the two queries together in the cache.
* Joins are accomplished in single query and results are matched row by row
* Data blending occurs by issuing two separate queries and then blending together the aggregate results

Rules: * There can be only one primary source, but there can be as many secondary sources as you desire.

* When all the aggregated results have been returned, tableau will match the aggregated rows based on linking field
* When more than one data source is used in Tableau workbook, whichever source used first in the view becomes the primary source
* Data Blending will be done based on the exact match of the dimension values (Ease- east)
* Blending is view specific i.e.
  - User can have one Data Source as the Primary View in one view and the same data source as the secondary in another.
  - All data sources can be used in a Blend but Olap Cubes (ssas) must be used as the primary source

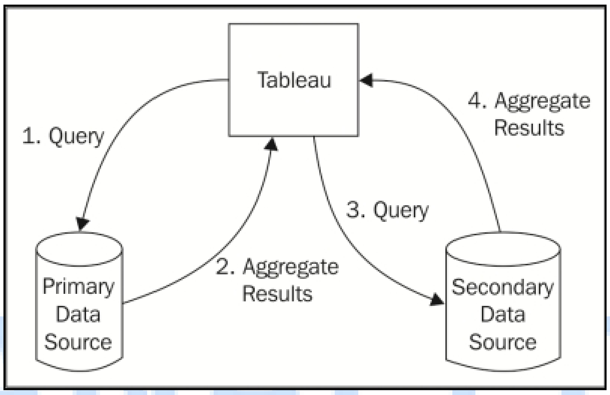


Figure: Tableau Blending Process

* **In Tableau, the Data Blending is performed in two ways**

1. Automated Way
2. Manual Way

* If the Dimension names are exactly matches, Tableau automatically performs the Blending
* if the Dimension names are not exactly matches, we need to perform blending in Manual way

1. Automated Way

Rules to Perform Data Blending:

* ***At least one common Dimension should exist*** between the multiple Data Sources
* In the ***common dimension at least one value should match***. If the one value is not matching it's not possible to perform the Data Blending

Linking Fields:

* Linking Fields are dimensions that are used to match data blended between the primary and secondary data sources.
* Linking field define the level of detail for the secondary source.
* Linking fields are automatically assigned if fields match by name and type between data sources.
* Manually assign relationships between fields by selecting, from the menu, Data --> Edit Relationships

Establishing Relationships between Blended Data Sources:

* Data --> Edit Relationships
* The Relationships window will display the relationships recognized between different data sources. Switch from Automatic to Custom to define user required own linking fields
* Linking fields used in view will generally be active by default, while other fields will not, however we can deactivate linking field by clicking on linking icon next to a linking field in the Data window

Example: Blending Ms Excel and Ms Access Data Sources

Steps:

* Define a connection to Sample - Superstore Excel Data Source(orders /Returns /User joined)
* Define a connection to Coffee Chain Ms Access Data Source (Coffee Chain)

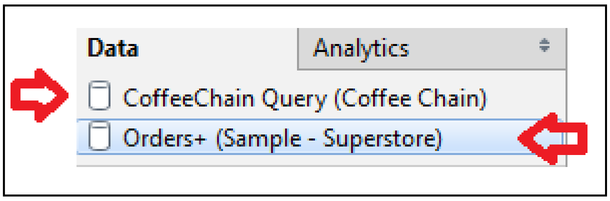
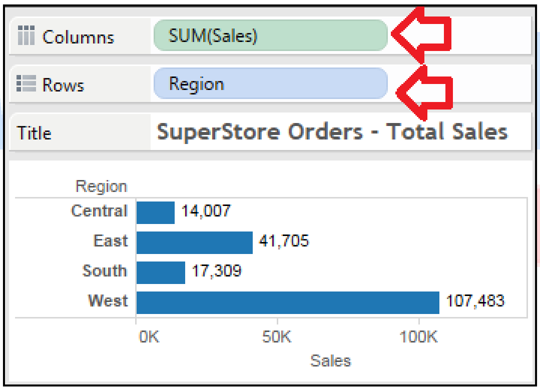
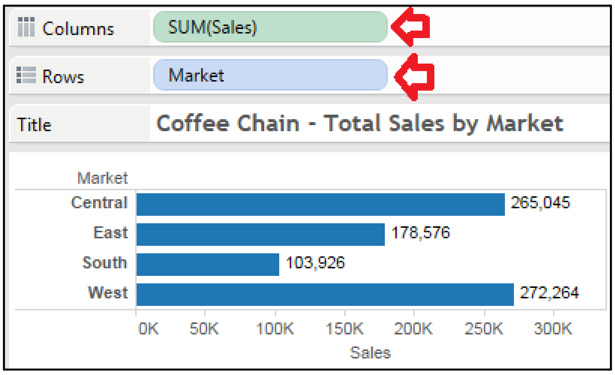

Data
Analytics
CoffeeChain Query (Coffee Chain)
Orders+ (Sample - Superstore)

Figure: Data window connection to Coffee Chain and SuperStore Multiple Data Sources

* Create a Data View From SuperStore Sales data (Region wise Total Sales) and analyze the output result



* Create a Data View From CoffeeChain Sales data (Market wise Total Sales) and analyze the output result



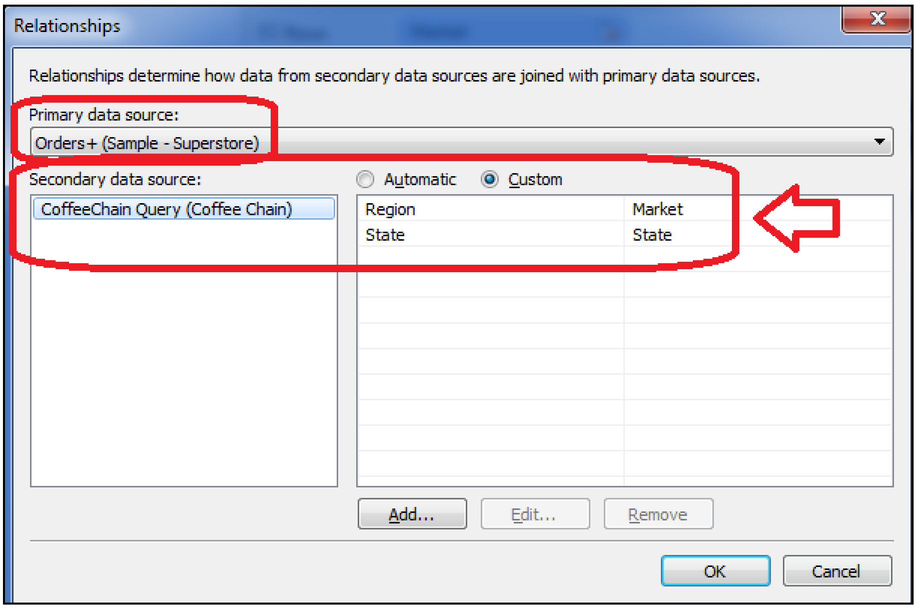
*** Create Relationship between primary and secondary data source**

- Data --- >Edit Relationships
- Primary Data Source = Orders Sample Superstore
  Secondary Data Source = CoffeeChain

Creates Linking fields automatically if fields match by name and type between data source

Create link between Region(SuperStore) and Market(CoffeeChain)

Custom = Region Market
State State



Relationships

Relationships determine how data from secondary data sources are joined with primary data sources.

Primary data source:
Orders+ (Sample - Superstore)

Secondary data source: ○ Automatic ● Custom
CoffeeChain Query (Coffee Chain)

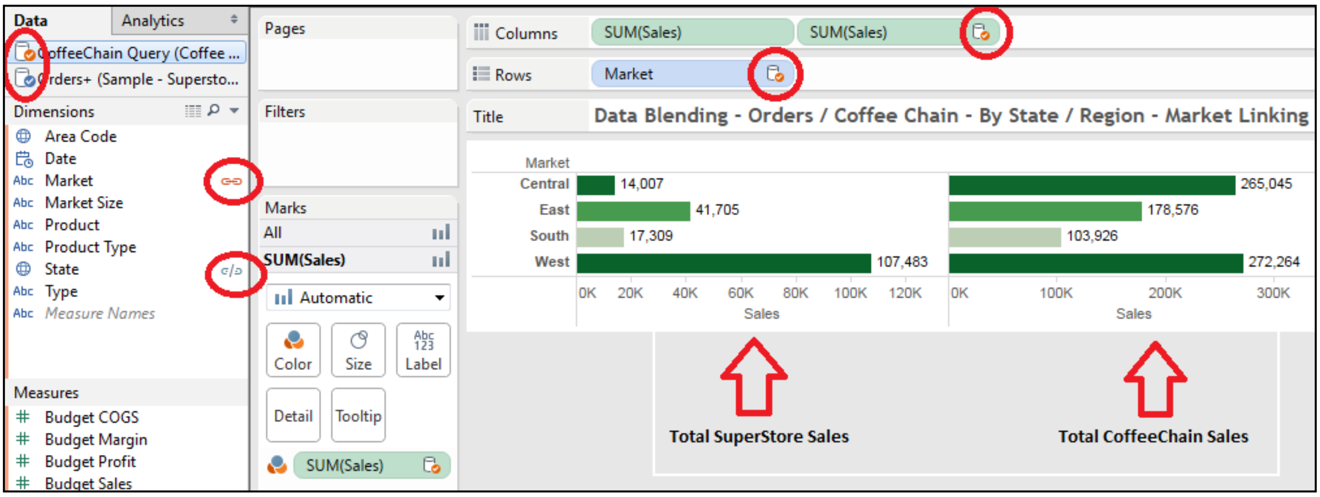
Region	Market
State	State

Figure: Automatic Linking between State fields

Creating link between Region(SuperStore) and Market(CoffeeChain)

*** Create a New Worksheet - Data Blending - Orders / Coffee Chain - By State / Region - Market Linking Field**

- Select Sum(Sales) from Sample Superstore and Drop it on COLUMNS
- Select Sum(Sales) from CoffChain and Drop it on COLUMNS
- Select Market (Dimension) from CoffeeChain and Drop it on ROWS
- Primary Data Source SuperStore in this view is indicated by **BLUE** checkmark
- Secondary Data Source CoffeeChain is indicated by **ORANGE** checkmark
- Linking fields Market and State
  - * The **Market** dimension is an **Active Linking Field (orange link icon)** on the data window
  - * The **State** dimension is another linking field but it is **NOT ACTIVE** in this view **(Grey Color)**



Data | Analytics

CoffeeChain Query (Coffee ...
Orders+ (Sample - Supersto...

Dimensions
Area Code
Date
Market
Market Size
Product
Product Type
State
Type
Measure Names

Measures
Budget COGS
Budget Margin
Budget Profit
Budget Sales

Pages

Filters

Marks
All
SUM(Sales)
Automatic

Columns: SUM(Sales) SUM(Sales)

Rows: Market

Title: Data Blending - Orders / Coffee Chain - By State / Region - Market Linking

Figure: Data Blending from Multiple Sources

Commonly occured issues while Performing Data Blending

* Without Defining the Relationship between the Data Sources, it is not possible to perform the Data Blending

* No Relationship to the Primary Data Source, If no match for the Dimension and no match for the Dimension values

Exercise: Creating Data Views

* Find Category wise Total Sales?

* Find the Regionwise Statewise Total Sales?

* Find the Regionwise Statewise Average Sales?

* Find the Regionwise Statewise Average Profit and encode each Region in a different color?

* Obtain Yearwise Total Profit and display each Year in a different color?

Nested Tables

-> In Tableau, If we keep one Field inside another Field then it is called as Nested Table

Design a View to display Yearwise, Quarter wise Total Profits?

Small Multiples

-> If we divide data into small small parts then such kind of a view is called as Small

Multiple

Obtain the Yearwise Quarterwise Total Profits in each Region?

Note:

-> If we drag any dimension to the right of a measure, it will automatically get placed to

the left of the measure. because measures wont accept any Dimensions to its Right

Building the Views automatically in Tableau

-> In Tableau, We can build the Views automatically without dragging fields to column and

row shelve

Slect Fields using Ctrl button and goto Show View and Slect the visualziation

Design a View for Title, Summary, Caption for Customer Segment wise Sales for every Year?

Different Parts of Data View

-> In Tableau, The Data View also called as Table
-> In Tableau, Each Data view contains some basic componenets. Those basic componenets are

categorized as Table   components
-> In Tableau, The Data View of Table is a collection of Rows and Columns and it contains

componenets like Headers,   Panes, field Lables, Mark Cards, Legends, title, Caption,

summary, Axes

Header

-> In Tableau, Data View Headers are automatically created when we place a Dimension Field

on to Row Shelve or   column shelve
-> The Header Displays Different names in the Fields of Data View
-> In Tablau, We can able to Hide or display Header. To Hide right click on existing

Header and Uncheck Show Header
-> If we want to Show Header, Right click on Header Field and Select Show Header

Analysis Tab --> Table Layout --> Show Field Labels for Rows/Columns

Axes

-> If we drag any measure field on to the Rows and column shelve, Tableau will

automatically create the Axes
-> By Default, the values of the measure field are displayed along that continuous axes
-> In Tableau, we can able to hide or display the axes
-> If you want to hide the axes, Right click on axes and Select uncheck the Show Header

check box
-> If you want to show the axes, Right click on the field and Select Show Header

Panes

-> Panes are created by the intersection of Rows and Columns in a Table
-> Depending on the Table Type, Pane might be created by the Intersection of axes
Ex: 2010 Box is one pane
2011 Box is one Pane

Cells

-> Cells are the smallest individual components in the Data View
-> Collection of Cells is called as pane
-> Cell is the intersection of Row and Column

Marks Card

-> If we drag any field to the Data View, the Data displays using marks card

Title

-> In Tableau, If you want to add a Tilte to a Data View, go to work sheet tab and Select

Show Title. Tableau adds title automatically at the Top of the Data View
-> similarly the other components like Summary, Caption etc...

Field Lables

-> If you place any field to the column shelve and row shelve Tableau displays the Field

lables
-> If you want to hide the field lables, Right click on field lable and Select Hide field

lable (or) Go to Analysis Tab -> Table Layout -> Uncheck Show field lables for Rows

and Columns

Legends

-> If you drag any field to the color, size, shape they will automatically create the

legends

Properties of Dimensions (Discete Fields)

-> Dimensions always contains discrete (Independent) data
-> Discrete means different values

-> In the Dimensions all the values are Indepenent

Properties of Measure (Continuous Fields)

-> Measure always contains the Dependent Data
-> It contains always continuous fields
-> We can convert any measure values to dimensions (vice-versa). It means we can
convert continuous data to discrete Data

Converting a Dimension to a Measure

-> First Right click on Dimension and Select Convert to Measure
-> It will automatically converts that Dimension to a Measure (or) Drag Diemsnion field on

to Measure Section

Working with multiple Measures
----

1. Individual Axes

-> If you create a seperate axes for each measure on the Data View, then such kind of view

is called as Individual axes

Examples

-> Design a view for Sub-Category Total Sales and Profits in a seperate axes
-> Design a view to display the Sub-Category Total Sales, Total Profits, Total Shipping
Cost and Unit Price and display the Sales with unit Price in the color

2. Axes Blending (or) Blended Axes

-> The concept of mixing the axes for the two measures is known as Axes Blending
-> Blending means simply combinng (or) mixing
-> If we blend the axes, all the measures will state to the single axes so that all the

marks show on a single pane
-> So in the blended axes, all the values for each measure is shown along with one

continuous axes
-> Inorder to belnd multiple measures, simply drag one measure to the existing axes of

another measure

-> When we blend the measure, Tableau creates two new fields with names 1. Measure Names

2. Measure Values

-> Measure Names contains all the names of the different measures which are participating

in blending

-> Measure values will gets stored in the Measure value field

Note

-> Blending Axes is the most appropriate way while comparing measures that have similar

scale and units

For Example, Profit and Sales are having same unit called as Currency

Design a View to display the Sales, Profit and Shipping Cost for Statewise

Dual Axes

-> The third way to compare multiple Measures in the Tableau is by creating a Dual axes

-> In the dual axes, we have two independent axes that are placed on top each other

-> Dual axes are useful, if you want to compare the measures having different scales and

units

For Example Profit and Discount

Ex:Design a view to display Customer segment wise Sales and Discount

Combination of Charts

-> If we customize, the default shapes in the marks card and if you create our own charts

then those kind of charts are called as Combination of Charts

-> In these charts in general, we combine two different charts in a single pane

Ex: Design a Combination Chart for Customer segmentwise Discount and Total Profit

----

Filters (SUBSETTING/WHERE Condition in other programming languages)

In Tableau, by using Filters, we can restrict the unwanted data by displaying only

required data

Textual Field ---> State Field --> 50 states / 10 states
Numeric Field ---> Sum(Sales) --> Minvalue and Maxvalue(Range)
>= 25000000 >= <= > < = <>
<= 25000000
Date Fields

Advantages

-> Filters will improve the performance and the Data view contains only the required data
-> In Tableau, we can apply filters on
Textual Data (or) Character Data
Numerical Data (or) Number Data
Date Data

Filtering Textual Data

-> In Tableau, Textual Data can be filtered directly from the Data View (or) By making use

of Filter Shelve

-> Filter directly from data view by right-click and select Include or Exclude on
Tool Tip

Ex: Display the Region wise Sales. Display Only the Central and North ?

-> Filtering Textual Data by adding Fields to Filter shelve.

Ex: Design a view for the Statewise Sales whose name starting with 'M'

While Filtering the Textual Dimension in Filter Shelve, we can filter the Dimension in 4

different ways
1.General
2.Wildcard
3.Condition
4.Top

1.General

-> In the General Tab, we can filter the Data in 3 ways
1.Select from List
2.Custom Value List
3.Use All

1.Select from List

-> By using this option, we can able to select the list of values, Manually all at a time,

none at a time, few at a time and even we can exclude all the values, few of the

values and none of the values

12-02-2016

2.Custom Value List

-> By using this option, we have to enter the list of values manually

3. Use All

-> If we want to use all the options, then we need to select use all

Wild Card

-> While Filtering Textual Data, if we don't know the complete name and if you know only

particular part of the string (or) single character and if you want to perform

filtering, then such kind of filtering is called as wild card filtering

-> As here, we are filtering data based on a single character, that single character known

as wild card

-> Using Wildcard, we can perform the Filtering in two ways
1. Including Wild Card
2. Excluding Wild Card

Ex: Display the Statewise Sales for those whose name contains "s"?

Note: While performing the Filter using the wild card, that filtering is not case

sensitive

While Filtering the Character Data using Wild Card, We can Filter it in 4 ways
1. Contains
2.Starts With
3. Ends with
4. Exactly Matches

Condition

-> In Tableau, If we want to filter the Textual Data based on Some Condition then we have

to make use of Condition Tab

Ex: Display the Statewise Sales whose Sales > 50000

Top

-> If we want to filter the top based on the Top N analysis, then we have to create the

filter based on Top Tab

Ex: Display Top 10 States based on Their Total Sales?

Filtering the Measure Data (Analysis Field/Numeric Field)

-> In Tableau, We can Filter the Measures by taking all individual values and also by

making use of the combined values

-> While Filtering the Measure on the Grouped Records or on the individual values, we can

perform filtering in 4 different ways
1. Range of Values
2.Atleast
3.Atmost
4.Special

1. Range of Values

-> If we want to filter any measure based on a Particular Range (based on Min and Max

value) then we should filter the measure using Range of Values

2. Atleast

-> If we know only the minimum value and if we dont know the maximum value of the measure,

then we should filter the measure using Atleast

3. Atmost

-> If we know only the maximum value and if we dont know the minimum value of the measure,

then we should filter the measure using atmost

4. Special

-> While handling the NULL Values in the measure, this special option is used

13-02-2016

Task: Design a View to display the Statewise Total Sales, whose Total Sales >20000
ATLEAST

Task: Design a View to display the Statewise Total Sales, whose Total Sales b/w 10000 and

30000 RANGE

Task: Design a View to display the Statewise Total Sales, whose Total Sales not more than

30000 ATMOST

Task: Design a View to display the Statewise Total Sales, whose INDIVIDUAL Sales more than

20000

Note:

-> It is not recommended to perform filtering on the Measures because as the measures

contain large amount of Data, it may degrade the Performance
-> So instead of applying Filters directly on Measures, Create SETS and Perform Filtering

Filtering Date Data

-> In Tableau, the Dates can be filtered in 3 ways

1. Relative Dates
2. Range of Dates
3. Discrete Dates

1. Relative Dates

-> In Tableau, if you want to filter the Dates for a Specific period of Time, then we have

to go with Relative Dates

2. Range of Dates

-> If we know the Starting and Ending values of the Date fields i.e if you know the fixed

range of dates then we have to   go with Range of Dates
-> If you dont know the Ending Date even we can filter the Date by using the Stating Date

option
-> If you dont know the Starting Date even we can filter the Date by using the Ending Date

option
-> In order handle NULLs we have to go with Special Values

3. Discrete Dates

-> If we want to filter each individual dates as oppose to the range of Dates then we have

to go with the discrete dates.
   This type of filter is called as Discrete Filter. Because we are defining descrete

values instead of Range

-> This discrete Date filtering can be done in 3 ways

1. Using General Tab
2. Using Condition Tab
3. Using Top

Quick filter

-> It allows the User to Filter the values dynamically from the Data view

-> This Quick Adds the interactivity and dynamism to the Data View
-> We can add N no. of Quick Filters for the Data View in Tableau
-> Quick filters can be enabled for the Textual Data, Numeric Data and Date Data

Example:

Design a View to display Statewise Sales for the Year 2013 whose Sales are in the Range

blw 10000 to 20000, for those which doesnt contain 'v' in their name using Quick Filter

Filter Sharing Options

-> In Tableau, we can share the filter globally across multiple Sheets
-> By making use of this feature, we can define the filter in one sheet and simple we can

apply the filter into respective sheets
-> Upto Tableau 7.0, We have only two options for sharing the filters
1. Local
2. Global
-> From Tableau 8.0 0n wards, we can Share Filters in 3 ways
1. Only this Worksheet
2. All using the Data source
3. Only to the Selected Work sheets

1. Only this Worksheet

-> It indicates the Filter get applied only to the Current Worksheet

2. All using the Data source

-> THis filter will be applied to the all the Sheets which are using the Data Source

3. Only to the Selected Work sheets

-> This option is useful to share the filter for the Selected Work Sheets

Example:

-> Design a Global filter for the Region wise, Monthwise Sales for the Months Jan and Jun

and apply this filter for all the sheets using this Data Source

19/10/2016

Context Filter

-> By Default, all the filters in Tableau are computed independently, It means each filter

access all the Rows with out depending on other filter

-> In Tableau, If we want to define Dependant filter then we have to make use of Context

Filters

-> Context Filters act as the independent filter and all other filters that we define or

set will act as Dependent filters, because they process only the data that passed to the

Context filters

Context Filter --- > query data source ---> subset data ---> create a separate table wiht

subsetted valued in data source or in the memory ---- >The normal filters always depend

on these context filters and later on subset

Advantages

-> The Context Filter is mainly used to create the dependant Top N filters

-> It improves the Performance

-> If you have lot of Filters (or) If you are having larger Data Source, the query may be

slow. We can set one or more Context Filters to improve the Performance

Note:

-> Even though if we define the normal filter first and the context filter next, Tableau

Computes Context Filter first and later it will compute the normal filter

-> In order to perform Context filtering a Temporary Table is created automatically by

Tableau when we set the Context

Examples

-> Design a View to display the Top 10 Products based on Sales for the Furniture Category

-> Design a View to display the Top 10 States for the Year 2010

-> Design a View to display the top 10 Products for the California State in the Year 2010

Note:

-> Context defines an Environment, where the similar group of resources are available

-> Always the Context filter indicated in the Gray Color

-> Context Filter always appear on top of the Filter shelve. We cant chant the position of

the Context filter in the Filter shelve

-> If we Remove context from the filter, the filter will appears on to filter shelve

itself by converting into the normal filter

-> We can add one or more Context filters for a Single view

Data Source Filter

-> From Tabelau 8.1 the new feature available in Tableau is Data Source Filters. This

allows the users to perfrom the filtering at Data source Level

Example

-> Perform the Filtering at the Data Source Level for the year 2013 Technology Data

Calculation Filters

-> We cannot see this Filter on the Data View

-> But Data View changes based on this Filter

Example:

-> Create a Data View by using Product sub Category and Sales. Apply Filter based on Year

2010

Sorting

-> In Tableau, We can able to perform sorting in 2 ways

1. Computed Sorting

2. Manual Sorting

1. Computed Sorting

-> In computed Sorting, Tabelau uses programatic rules to arrange the data either in

ascending or descending order

-> In this Sorting, we can sort Data in 4 Ways
1. Data Source Order
2. Alphabetical Order
3.Field
4.Manual

1. Data Source Order

-> In the Data source Order, Tableau will sort the Data according to the Data in the Data

Source

2. Alphabetical Order

-> In the Alphabetical Order, Tabelau will sort the Data based on First character of the

value

3.Field

-> In this, It will Sort data based on some condition

4. Manual

-> In the Manual way, we can sort the data accrding to our way

Examples

-> Perform the computed sorting on the Statewise Sales and Sort the States in all

possible available ways
-> Perform the computed sorting the yearwise Sub-Category wise Total Average discounts

and Sort the Data based on each Sub-Category
-> Perform the computed sorting Region wise Total Sales (Sum(Sales)) but sort data on
average sales (avg(sales))

Manual Sorting

-> In the Manual way, we need to arrange the Data values manually in our own order

-> This type of Sorting is helpful, If you want to arrange the data in some specific order

Example:

-> Manually Sort the State wise Sales by arranging the field values

Notes

-> By Default, Tableau Sort the Data according to the Data Source

-> We can Sort the Data even from the Axes or from the Headers

-> From the Toolbar also, we can Sort the Data Either in ascending or Descending order

Aliases

-> Alias is nothing but alternate name

-> In Tableau, We can create the Alias names to the FIELD VALUES

-> Aliases can be created only for Dimensions ******

Example:

-> Design a View in such a way that display West Region for West, East Region for East,

South Region for South and Central Region for Central

Note

-> We cannot create the Aliases for the Measures *************

-> We can able to create the Aliases for SETS *************

Grouping

-> A group is a combination of dimension members that make higher level categories

-> Groups are useful for

* Correcting data errors

Ex: Dimension coutry has the values USA, U.S.A, United States of America (combine into one single value)

* Answering What-If type questions by combining

Ex: What will be the total combined sales of West and South Regions

-> In Tableau, If you want to organize the data at Higher Level, we need to combine the

Data at the higher level in order to store all the similar objects at one place

(GROUPING)

-> A Group is a combination of all the similar Dimensions, Members and Measures that make

higher level Categories

--> Grouping Example:

* Basic Grouping (DATA VIEW WINDOW)

- What is the total sales for Sub-Category and group some dimension members into "Low Sales Group"

COLUMNS - Sum(Sales)

ROWS - Low Sales Group, Sub-Category

* Grouping on Customer Name Dimension (DATA WINDOW)

- Creating Groups on Customer Name starting with letter "A"
- Select Customer Name --> Right click --> Create Group

* Group Name: Group "A" and "Other" Customers

Note: Always create CALUCLATED GROUPING

* Visual Grouping on Multiple Dimensions

- Select few dimension members on and group the fields

COLUMNS - Sum(Sales)

ROWS - Category / Region / Sub-Category

a. Select 4 items Bookcases/Furnishing/Tablets/Chairs->
Select GROUP Members from Tool Tip --> Select Region
Analyze the result, all the items under CENTRAL region are hilighted
under Furniture / Office Supplies / Technology
Drag and Drop Created GROUPS onto new sheet and analyze results

* Grouping by using CALUCLATED FIELD

- Group All the STATES that with start with letter "A"

New Calucated FieldName - States Start with alphabet A

if left

* Grouping on Geo Map

- Create GEO MAP for STATE with sum(Sales)

LOGITUDE / LATTITUDE / STATE / SUM(SALES)

Divide the map into multiple groups (EAST & WEST) and drag and drop them on filters if required

In Tableau, Groups can be Created in multiple ways

-> Directly from the Data View make selections and perform grouping (BASIC GROUPING)

-> By Selecting the Marks in the Data View

-> From the Header in the Data View
-> From the Data Window also

Examples

-> Design a View with SubCategory group in Sub Category wise Sales Data View
-> Design a View to create B-group by right click on field name (Ex: Customer)
-> Design a View to create multiple Visualization Groups

-> In a single data view, we can create multiple groups and that field will get added in

the Data Window seperately (or)
In other words, If we create any group automatically a new field get added in the Data

View
-> On the newly created groupo field, we cannot perform any calculations

Note:

-> If we want to ungroup all the members at a time, we have one option Reset in the Edit

Group window. Click on Reset -> click on Apply -> ok
-> We can able to create a group on the multiple Dimensions but we can't create a SET on

multiple Dimensions

Limitations:
* GROUP can be created on dimension and measures
* SET can be created on measures but not on dimensions
verify by right-click on dimension and measures ---> create ---> Group/Set
* SET cannot be created on calculated fields but sets can be used in calculated fields

Difference between SETS / FILTERS / GROUPS ?

SETS

-> Sub set of some data is called as SET

-> SETs are the Custom Fields that subset or sub part of Data based on Some Condition

-> A Set can be based on some computed condition (or) based on some set of data

-> By using sets we can compare the data (or) Perform analysis based on some set of Data

-> Tableau Display Sets at the bottom of the Data Window and labels them with a Set icon

-> SETS can be used again and again throughout the workbook.

-> Since sets become part of the metadata, any workbook connected through that Saved Data

Source (or .tds file) can also utilize its functionality.

-> Using sets maintains consistency and saves time.

Filters

-> Filters apply to only current worksheet and does not store it as metadata

-> Filters must be re-created multiple times whereas SETS once created can be utilized
by any worksheet(reusability)

Business Requirement:

----

1. Top 5 States by Sales (Constant/Computed Set)
2. Top 5 Customers by Profit (Constant/Computed Set)
3. Customers whose sales > 50000(Constant/Computed Set)

-> In Tableau, we can able to define 2 Types of SETs

1) Constant Sets (Static)

2) Computed Sets (Dynamic)

1) Constant Sets

-> The values in the constant Set won't change. always they are static
(the user manually selects the data values)

2) Computed Sets

-> The values in the computed Sets changes when the underlying Data changes based on some

condition
(the system dynamically selects the data values whenever the underlying data changes)

-> When we drag any computed Sets or Constant Sets onto the Columns and Row shelves, it

will be displayed in the In/Out Mode

In - > Indicates in members of the Set (or) Members which are part of the Set
Out -> Indicates Out members of the Set (or) Members which are not part of the Set

Sub-Category	Sales
Papers	100
Tables	200
Pens	300
Erasers	200
Clips	100

Set1

Tables
Pens
Erasers

In -> 700
Out -> 200 (hide)

TASKS IN SET:
==============

1. Constant SET (Static) / Creating Charts with SETS (In/Out)
2. Combined SETS
3. Computed (Dynamic) SETS

****

CREATE A CONSTANT SET / CREATING CHARTS WITH SETS (IN/OUT)

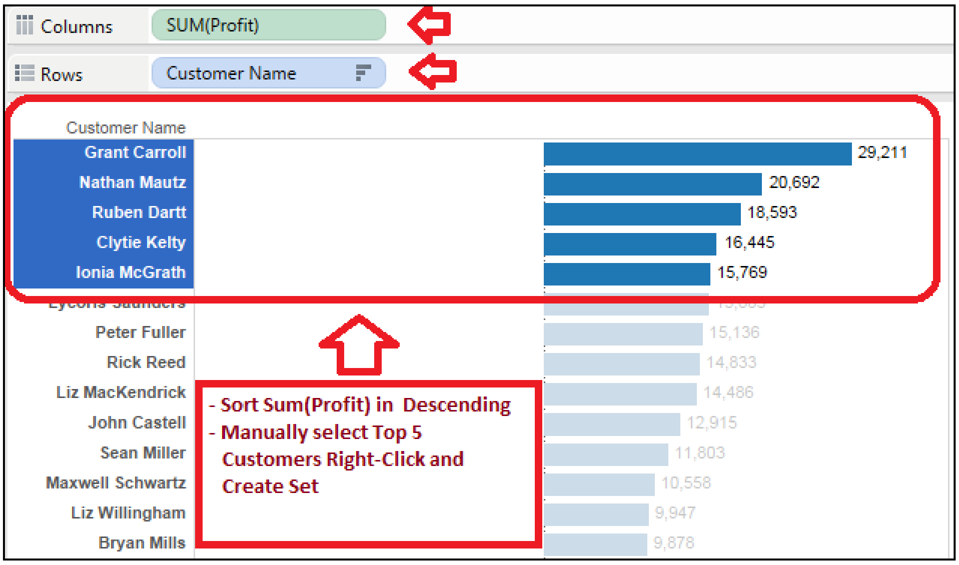
STATIC - sorting on sum(sales) by customer name and select the top 5 manually and create a set

****

Examples

-> Create a Constant SET in Tableau for Top 5 Customers based on Profit

COLUMNS - Sum(Profit)
ROWS - Customer Name
- Sort Sum(Profit) in Descending order and select top 5 rows -> Right click and select CREATE SET



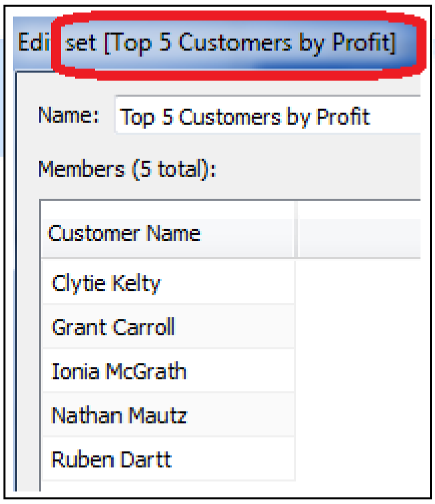
Columns: SUM(Profit)

Rows: Customer Name

Customer Name	SUM(Profit)
Grant Carroll	29,211
Nathan Mautz	20,692
Ruben Dartt	18,593
Clytie Kelty	16,445
Ionia McGrath	15,769
Peter Fuller	15,136
Rick Reed	14,833
Liz MacKendrick	14,486
John Castell	12,915
Sean Miller	11,803
Maxwell Schwartz	10,558
Liz Willingham	9,947
Bryan Mills	9,878

- Sort Sum(Profit) in Descending
- Manually select Top 5 Customers Right-Click and Create Set

SET Name- Top 5 Customers by Profit



Edi set [Top 5 Customers by Profit]

Name: Top 5 Customers by Profit

Members (5 total):

Customer Name
Clytie Kelty
Grant Carroll
Ionia McGrath
Nathan Mautz
Ruben Dartt

Figure: Create a Set for Top 5 Customers by Profit

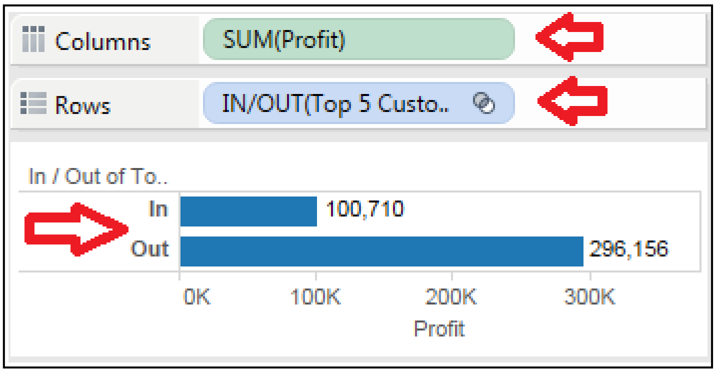
- Review the results by creating a New Sheet

COLUMNS - Sum(Profit)

ROWS - Top 5 Customers By Profit / Customer Name

In - > Indicates in members of the Set (or) Members which are part of the Set

Out -> Indicates Out members of the Set (or) Members which are not part of the Set



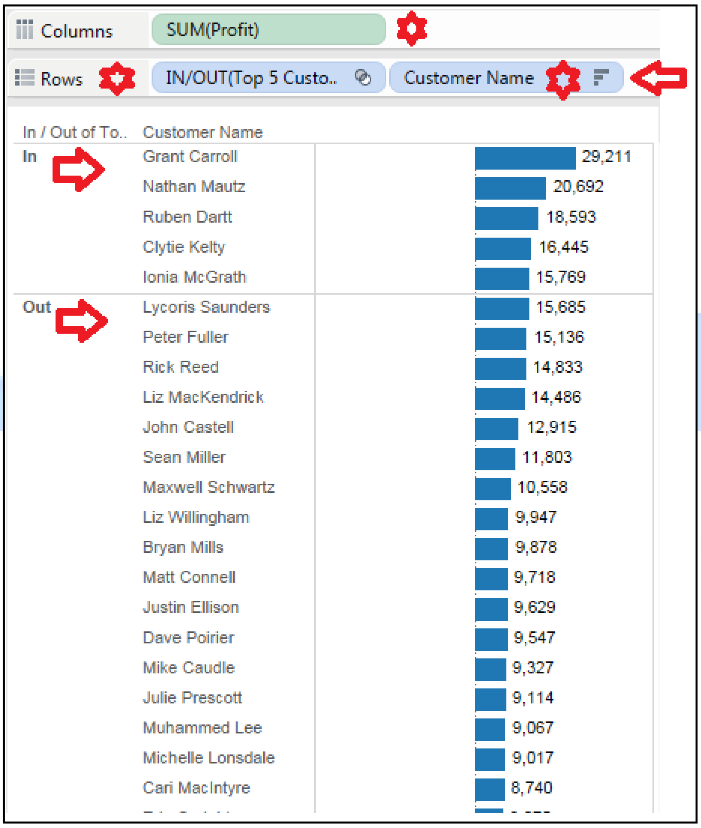
Columns: SUM(Profit)

Rows: IN/OUT(Top 5 Custo..

Figure: Total Profit by the Customers grouped by In & Out

IN --> Manually Selected 5 Customers

OUT --> All the Customers excluding 5 Customers in IN



Columns: SUM(Profit)

Rows: IN/OUT(Top 5 Custo.. Customer Name

Figure: Top 5 Customers by Profit

-> Create a Constant SET in Tableau for Top 5 Customers based on Sales

COLUMNS - Sum(Sales)

ROWS - Customer Name

Sort Sum(Sales) in Descending order and select top 5 rows -> Right click and select CREATE SET

SET Name - Top 5 Customers by Sales

Task: Hide Out (right click)

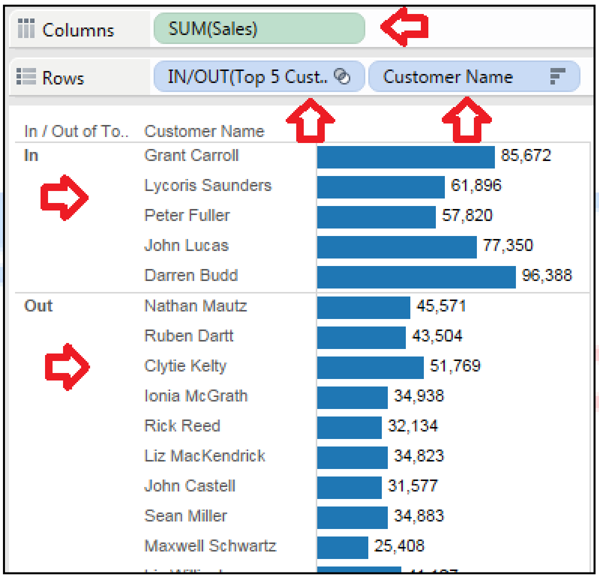
Header ---> IN ---> Edit Alias -----> TOP 5 Customers by Sales

---

Top 5 Customers by Sales
Customer Name
Darren Budd
Grant Carroll
John Lucas
Lycoris Saunders
Peter Fuller

Review the results by creating a New Sheet

COLUMNS - Sum(Sales)
ROWS - Top 5 Customers By Sales / Customer Name
Task: Hide Out (right click)
Header ---> IN ---> Edit Alias -----> TOP 5 Customers by Sales



Columns: SUM(Sales)
Rows: IN/OUT(Top 5 Cust.. | Customer Name

Figure: Top 5 Customers by Sales

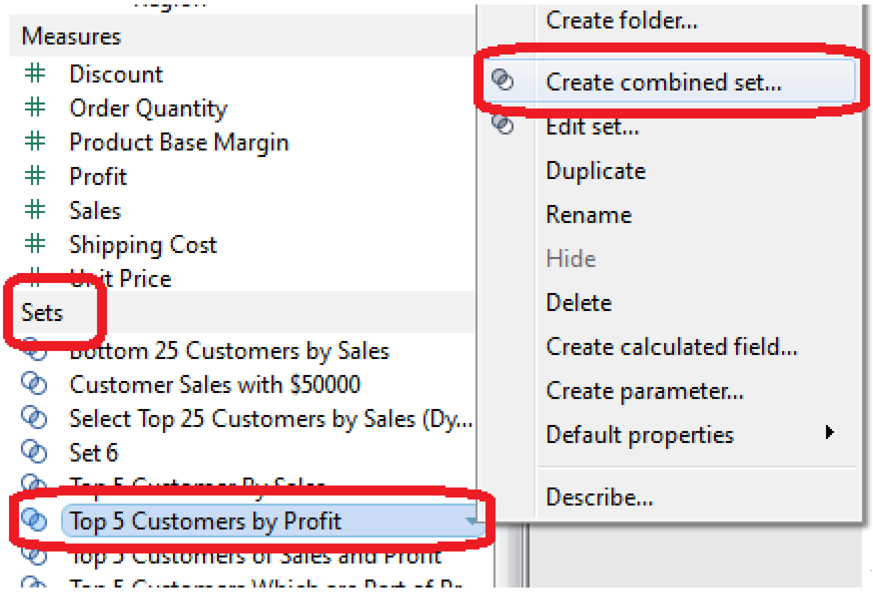
****

COMBINED SET

****

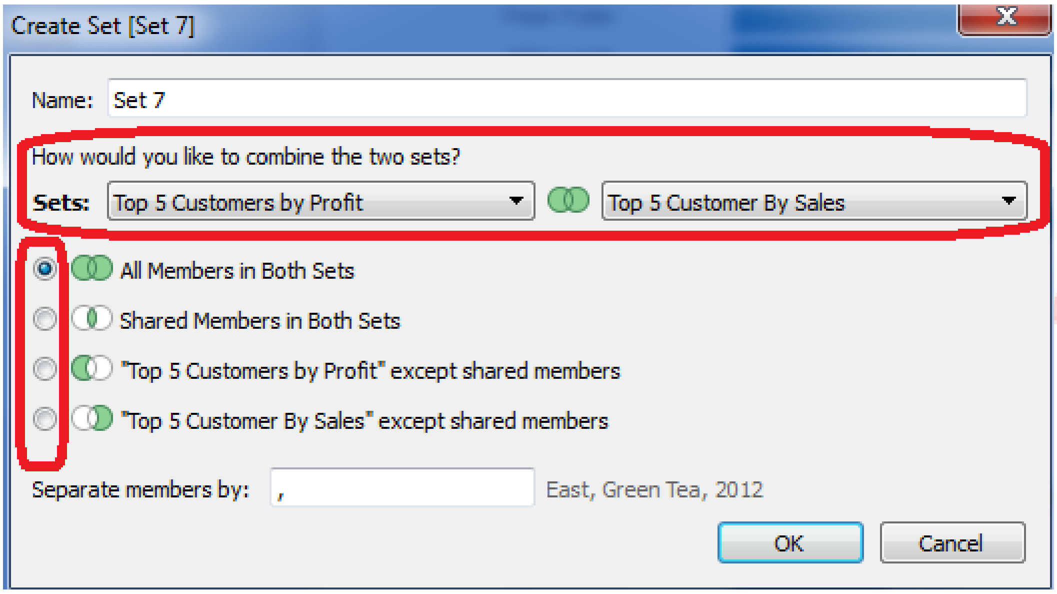
-> In Tabelau, We can combine the Sets based on Single Dimension in order to Compare the Members in both the Sets

Steps:
----
* Select Top 5 Customers by Profit SET from Data Window
* Right Click ---> Select Combined Set



* Match Sets

- Select Top 5 Customers by Profit / Top 5 Customer by Sales


Create Set [Set 7]

Name: Set 7

How would you like to combine the two sets?

Sets: Top 5 Customers by Profit ▾ Top 5 Customer By Sales ▾

● All Members in Both Sets

○ Shared Members in Both Sets

○ "Top 5 Customers by Profit" except shared members

○ "Top 5 Customer By Sales" except shared members

Separate members by: , East, Green Tea, 2012

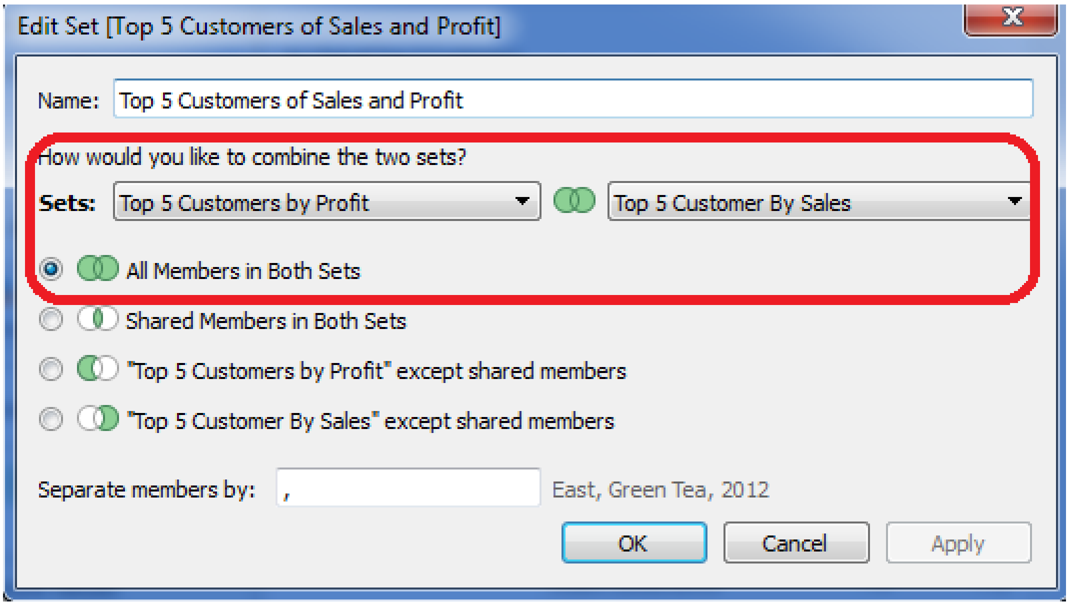
-> When we Combine two sets, It will create a new set in four Different ways

1. All Members in Both Sets

*** Generate a Visualization Top 5 Customers of Sales AND Top 5 Customers of Profit**

- SET NAME - Top 5 Customers OF Sales and Profit
- Analyze visualization on sheet

ROWS	-	Top 5 Customers OF Sales and Profit / Customer Name
COLUMNS	-	Sum(Sales)



Edit Set [Top 5 Customers of Sales and Profit]

Name: Top 5 Customers of Sales and Profit

How would you like to combine the two sets?

Sets: Top 5 Customers by Profit ▾ Top 5 Customer By Sales ▾

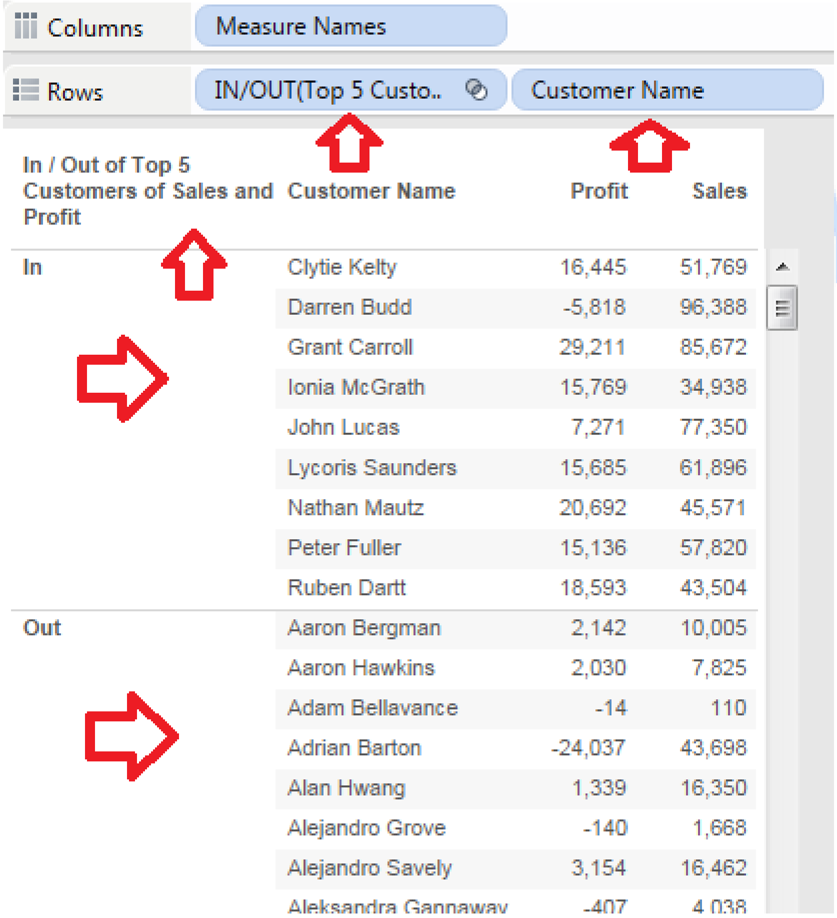
● All Members in Both Sets

○ Shared Members in Both Sets

○ "Top 5 Customers by Profit" except shared members

○ "Top 5 Customer By Sales" except shared members

Separate members by: , East, Green Tea, 2012



Columns: Measure Names

Rows: IN/OUT(Top 5 Custo.. | Customer Name

In / Out of Top 5 Customers of Sales and Profit	Customer Name	Profit	Sales
In	Clytie Kelty	16,445	51,769
	Darren Budd	-5,818	96,388
	Grant Carroll	29,211	85,672
	Ionia McGrath	15,769	34,938
	John Lucas	7,271	77,350
	Lycoris Saunders	15,685	61,896
	Nathan Mautz	20,692	45,571
	Peter Fuller	15,136	57,820
	Ruben Dartt	18,593	43,504
Out	Aaron Bergman	2,142	10,005
	Aaron Hawkins	2,030	7,825
	Adam Bellavance	-14	110
	Adrian Barton	-24,037	43,698
	Alan Hwang	1,339	16,350
	Alejandro Grove	-140	1,668
	Alejandro Savely	3,154	16,462
	Aleksandra Gannaway	-407	4,038

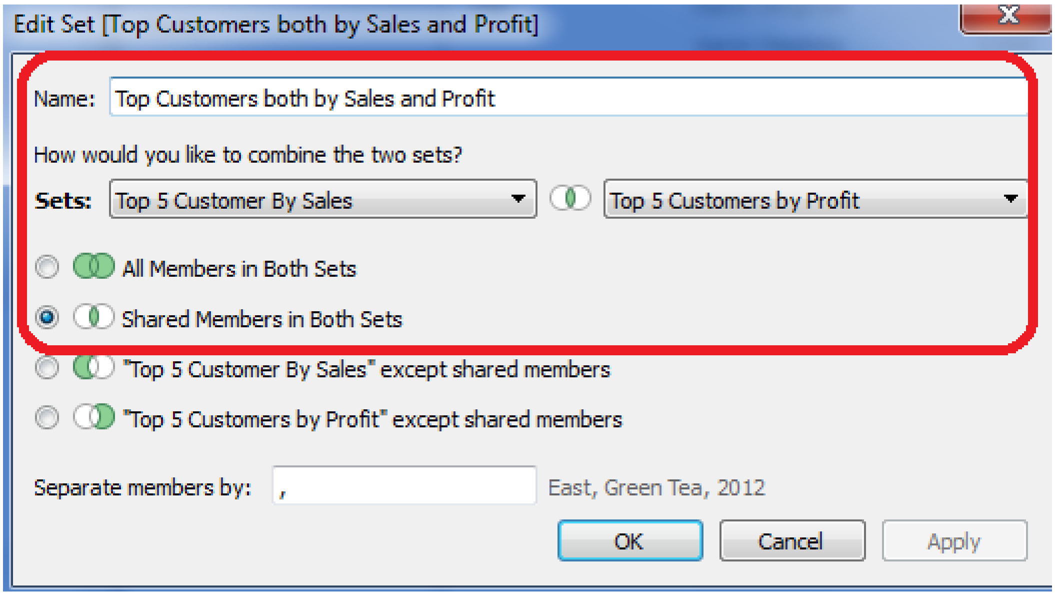
2. Only the shared members in both the Sets

*** Top Customers both by Sales and Profit**

- SET NAME - Top Customers both by Sales and Profit

- Analyze visualization on sheet

ROWS	-	Top Customers both by Sales and Profit / Customer Name
COLUMNS	-	Sum(Sales)



Edit Set [Top Customers both by Sales and Profit]

Name: Top Customers both by Sales and Profit

How would you like to combine the two sets?

Sets: Top 5 Customer By Sales ▾ Top 5 Customers by Profit ▾

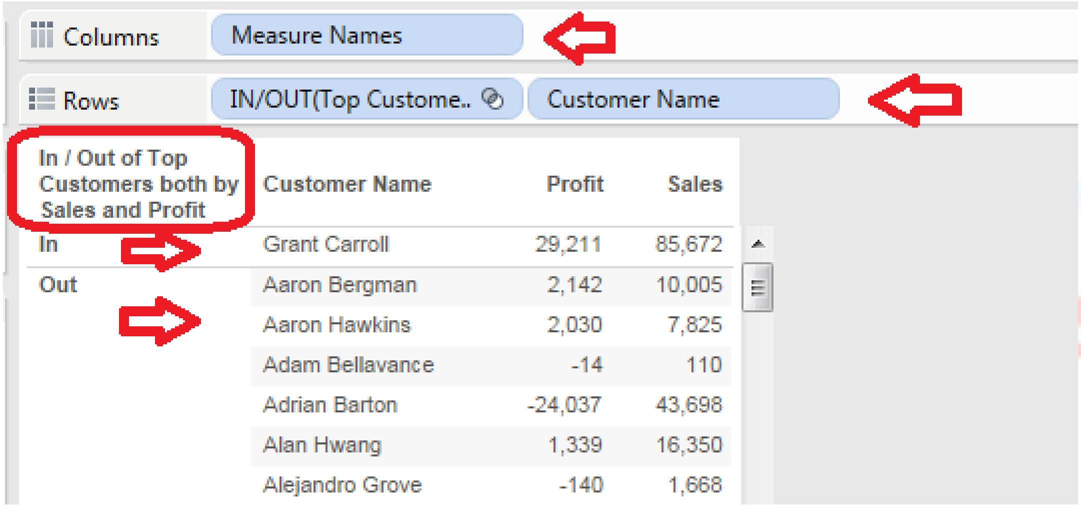
○ All Members in Both Sets

● Shared Members in Both Sets

○ "Top 5 Customer By Sales" except shared members

○ "Top 5 Customers by Profit" except shared members

Separate members by: , East, Green Tea, 2012



Columns: Measure Names

Rows: IN/OUT(Top Custome.. | Customer Name

In / Out of Top Customers both by Sales and Profit	Customer Name	Profit	Sales
In	Grant Carroll	29,211	85,672
Out	Aaron Bergman	2,142	10,005
	Aaron Hawkins	2,030	7,825
	Adam Bellavance	-14	110
	Adrian Barton	-24,037	43,698
	Alan Hwang	1,339	16,350
	Alejandro Grove	-140	1,668

3. Only the members which are part of the Left Set

*** Only the Top 5 Customers which are part of Sales**

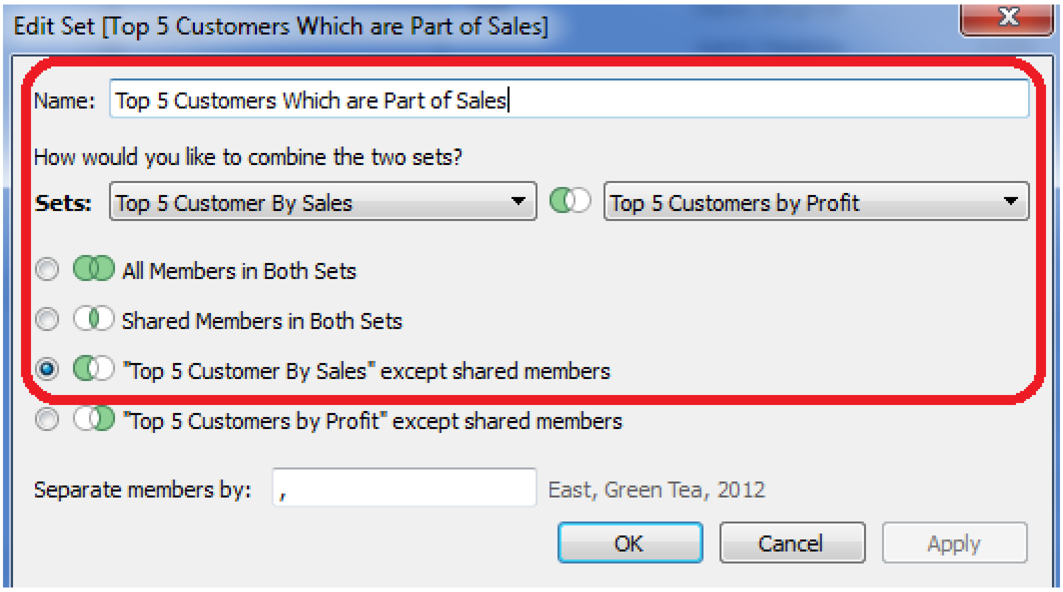
*** "Top 5 Customers by Sales" except Shared Members**

- SET NAME - Top 5 Customers Which are Part of Sales (Except Shared Members)

- Analyze visualization on sheet

ROWS - Top 5 Customers Which are Part of Sales / Customer Name

COLUMNS - Sum(Sales)



Edit Set [Top 5 Customers Which are Part of Sales]

Name: Top 5 Customers Which are Part of Sales

How would you like to combine the two sets?

Sets: Top 5 Customer By Sales ▼ | Top 5 Customers by Profit ▼

○ All Members in Both Sets

○ Shared Members in Both Sets

● "Top 5 Customer By Sales" except shared members

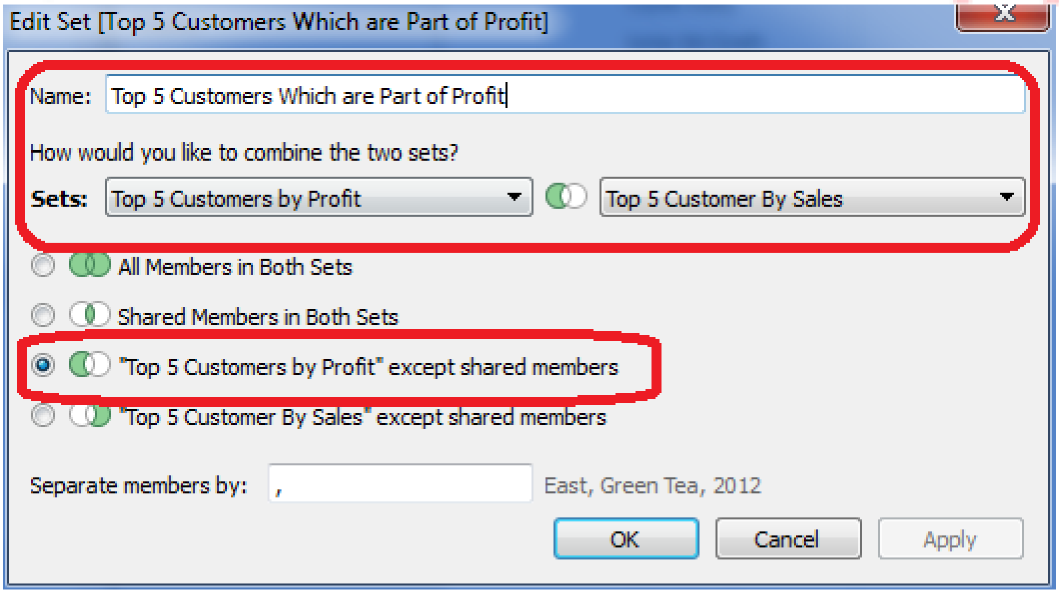
○ "Top 5 Customers by Profit" except shared members

Separate members by: , East, Green Tea, 2012

4. Only the members which are part of the Right Set

- **Only the Top 5 Customers which are part of Profit**
- **"Top 5 Customers by Profit" except Shared Members**
  - SET NAME - Top 5 Customers Which are Part of Profit (Except Shared Members)
  - Analyze visualization on sheet

ROWS	-	Top 5 Customers Which are Part of Profit / Customer Name
COLUMNS	-	Sum(Sales)



Edit Set [Top 5 Customers Which are Part of Profit]

Name: Top 5 Customers Which are Part of Profit

How would you like to combine the two sets?

Sets: Top 5 Customers by Profit ▼ | Top 5 Customer By Sales ▼

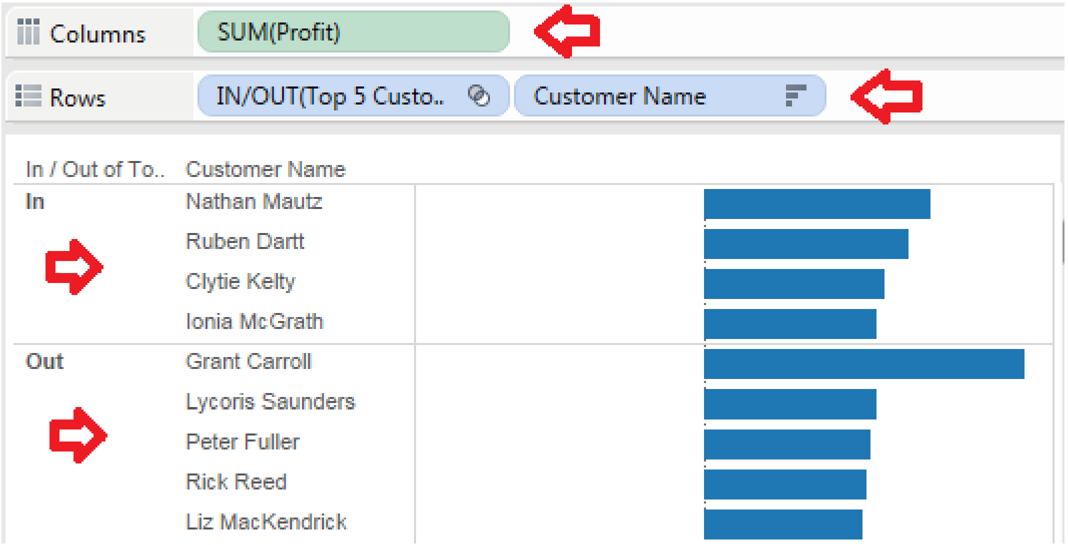
○ All Members in Both Sets

○ Shared Members in Both Sets

● "Top 5 Customers by Profit" except shared members

○ "Top 5 Customer By Sales" except shared members

Separate members by: , East, Green Tea, 2012



Columns: SUM(Profit)

Rows: IN/OUT(Top 5 Custo.. | Customer Name

In / Out of To..	Customer Name
In	Nathan Mautz
	Ruben Dartt
	Clytie Kelty
	Ionia McGrath
Out	Grant Carroll
	Lycoris Saunders
	Peter Fuller
	Rick Reed
	Liz MacKendrick

Exercise for Practice:

-> Obtain the States Information which are participating in

Both top 5 Sales and top 5 Profits

Only the Top 5 States which are part of Sales

Only the Top 5 States which are part of Profits

All the States in both Top 5 Sales and Profits

-> Design a View to display the Repeated Customers for the Years 2009 and 2011

* Drag and Drop Year Field onto Filter and Select 2009 and 2011

-> How do members of a set contribute to the Total?

-> How many Members of a Set exist in another Set? --> SUM(NUMBER OF RECORDS)

****

CREATE DYNAMIC SETS

DYNAMIC - Automatically Select the Top 25 Customer Names by Sales,
when underlying data changes automatically it selects Top 25 Customers by Sales without Tableau Developer intervention

****

* Select Top 25 Customers by Sales (Dynamic)

- Select Customer Name From Dimension -> Right-Click -> Create SET

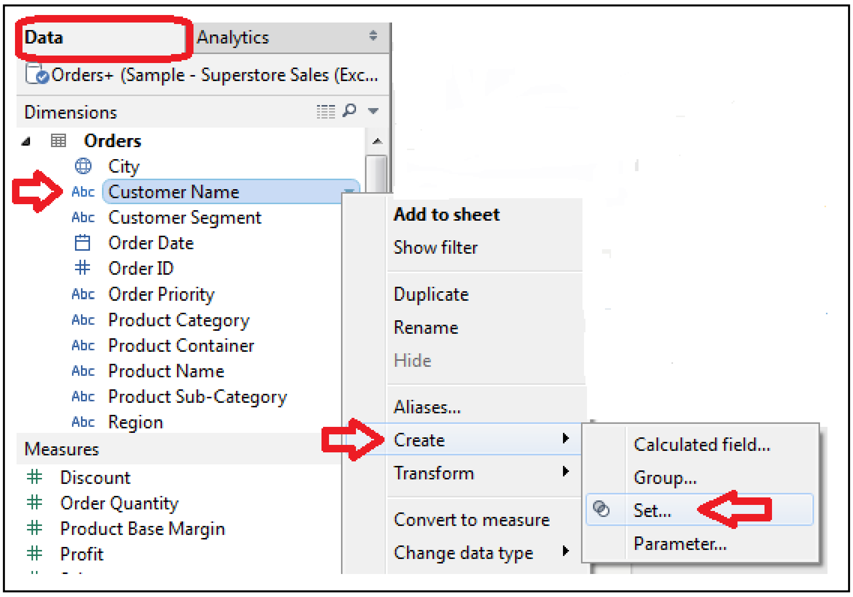
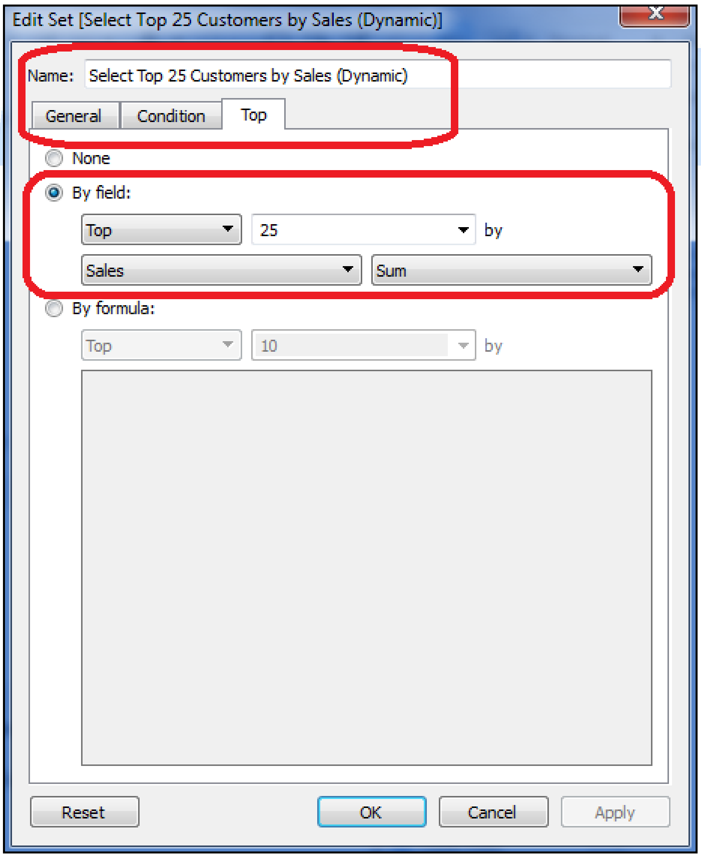


Figure: Creating DYNAMIC SET on Customer Name (Dimension / Discrete Field)

- SET NAME - Select Top 25 Customers by Sales (Dynamic)
- TOP (tab) - By Field - Top 25 by Sales Sum


Edit Set [Select Top 25 Customers by Sales (Dynamic)]

Name: Select Top 25 Customers by Sales (Dynamic)

○ None

● By field:

Top 25 by

Sales Sum

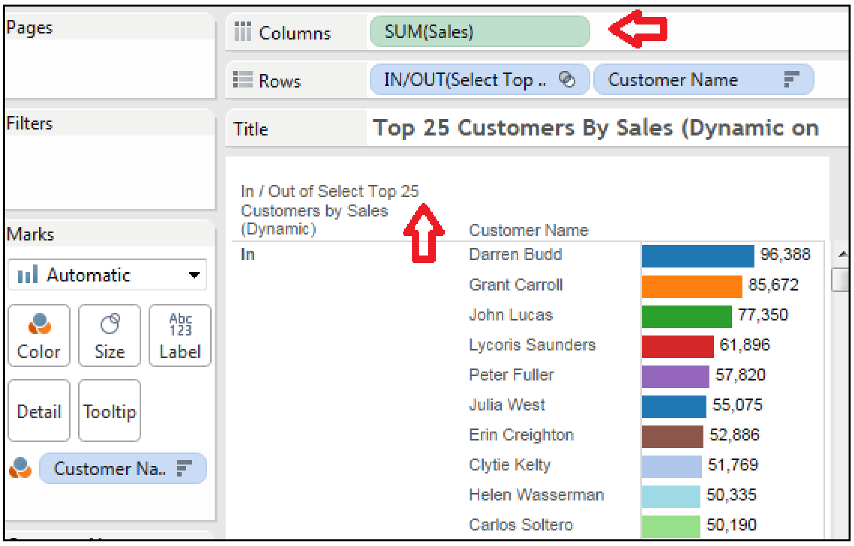
○ By formula:

Top 10 by

Figure: SET Expression on Customer Name Dimension

- Analyze visualization on sheet

ROWS	-	Select Top 25 Customers by Sales (Dynamic) / Customer Name
COLUMNS	-	Sum(Sales)



Columns: SUM(Sales)

Rows: IN/OUT(Select Top .. | Customer Name

Title: Top 25 Customers By Sales (Dynamic on

Figure: Dynamically sub setting on Top 25 Customers by Sales

*** Additional Task: Convert Visualization into Crosstab**

- Right Click Sheet - Select Duplicate as Crosstab
- New Sheet created with Cross Tabular Visualization

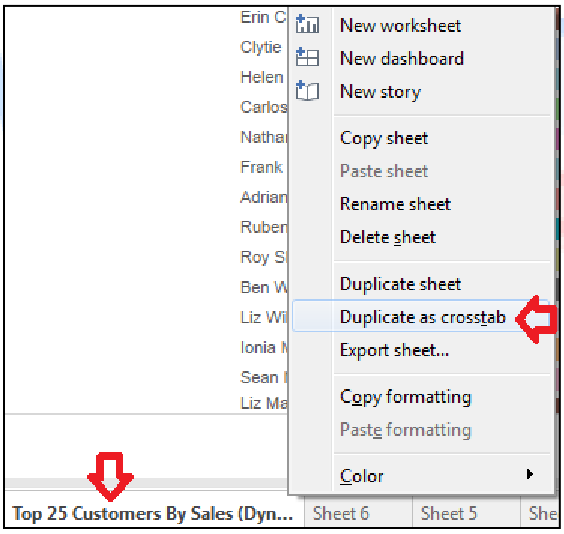
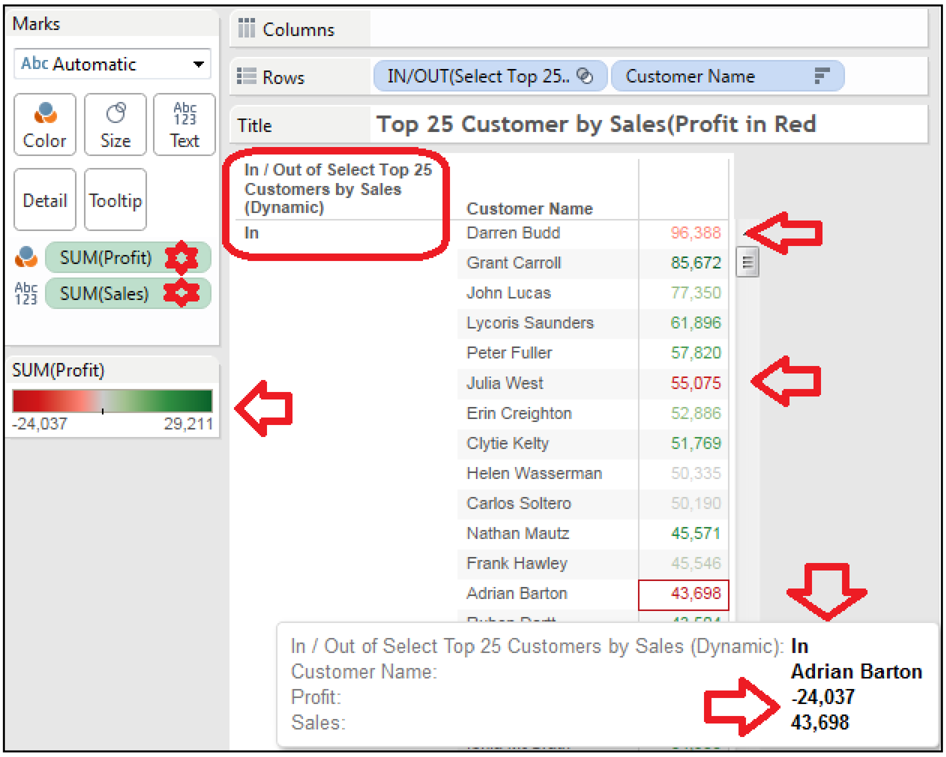


Figure: Convert to Cross Tabular Report

- Drag and Drop Sum(Profit) onto Color
- Analyze the output result on crosstab (if highlighted in red color then profit is in minus)
- Select Customer Name and Product Category onto ROWS and Analyze



Marks

Abc Automatic

SUM(Profit)

SUM(Sales)

SUM(Profit)

-24,037 29,211

Columns

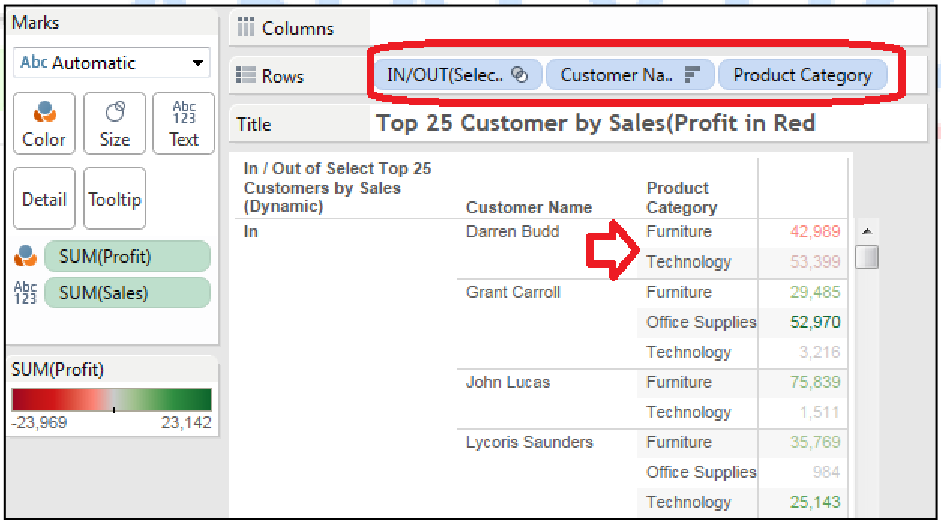
Rows: IN/OUT(Select Top 25.. | Customer Name

Title: Top 25 Customer by Sales(Profit in Red

In / Out of Select Top 25 Customers by Sales (Dynamic)	Customer Name	
In	Darren Budd	96,388
	Grant Carroll	85,672
	John Lucas	77,350
	Lycoris Saunders	61,896
	Peter Fuller	57,820
	Julia West	55,075
	Erin Creighton	52,886
	Clytie Kelty	51,769
	Helen Wasserman	50,335
	Carlos Soltero	50,190
	Nathan Mautz	45,571
	Frank Hawley	45,546
	Adrian Barton	43,698

In / Out of Select Top 25 Customers by Sales (Dynamic): **In**
Customer Name: **Adrian Barton**
Profit: **-24,037**
Sales: **43,698**

Figure: Top 25 Customers by Sales(Profit highlighted in red color)



Marks

Abc Automatic

SUM(Profit)

SUM(Sales)

SUM(Profit)

-23,969 23,142

Columns

Rows: IN/OUT(Selec.. | Customer Na.. | Product Category

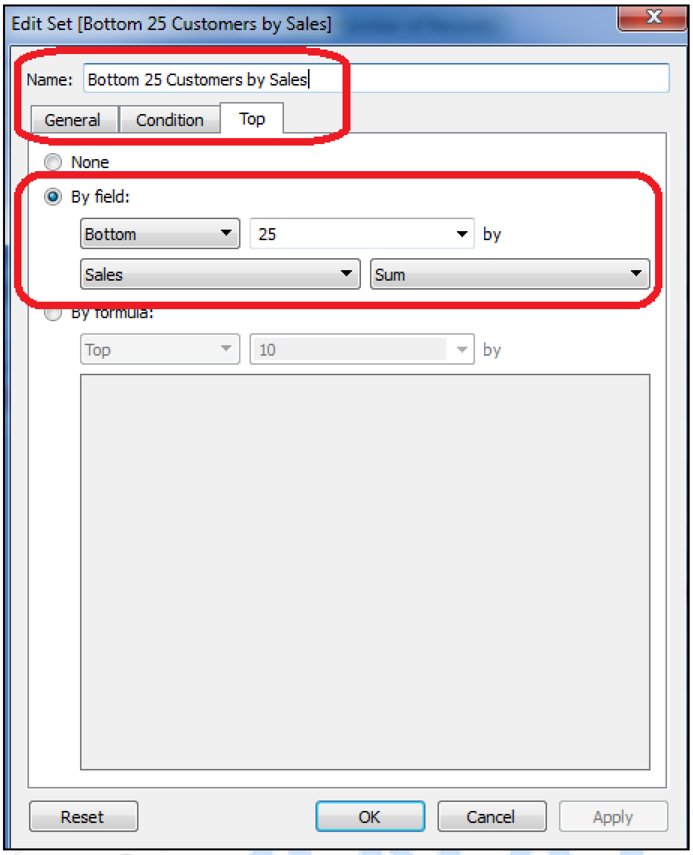
Title: Top 25 Customer by Sales(Profit in Red

In / Out of Select Top 25 Customers by Sales (Dynamic)	Customer Name	Product Category	
In	Darren Budd	Furniture	42,989
		Technology	53,399
	Grant Carroll	Furniture	29,485
		Office Supplies	52,970
		Technology	3,216
	John Lucas	Furniture	75,839
		Technology	1,511
	Lycoris Saunders	Furniture	35,769
		Office Supplies	984
		Technology	25,143

Figure: Top 25 Customers by Sales in Each Product Category (Profit highlighted in red color)

*** Select Bottom 25 Customers by Sales (Dynamic)**

- Select Customer Name From Dimension -> Right-Click -> Create SET
- SET NAME - Select Bottom 25 Customers by Sales (Dynamic)
- TOP (tab) - By Field - Bottom 25 by Sales Sum



Edit Set [Bottom 25 Customers by Sales]

Name: Bottom 25 Customers by Sales

○ None

● By field:

Bottom 25 by

Sales Sum

○ By formula:

Top 10 by

- Analyze visualization on sheet

ROWS	- Select Bottom 25 Customers by Sales (Dynamic) / Customer Name
COLUMNS	- Sum(Sales)

Task: Convert Visualization into Crosstab

- Right Click Sheet - Select Duplicate as Crosstab
- New Sheet created with Cross Tabular Visualization
- Drag and Drop Sum(Profit) onto Color
- Analyze the output result on crosstab (if highlighted in red color then profit is in minus)
- Select Customer Name and Product Category onto ROWS and Analyze

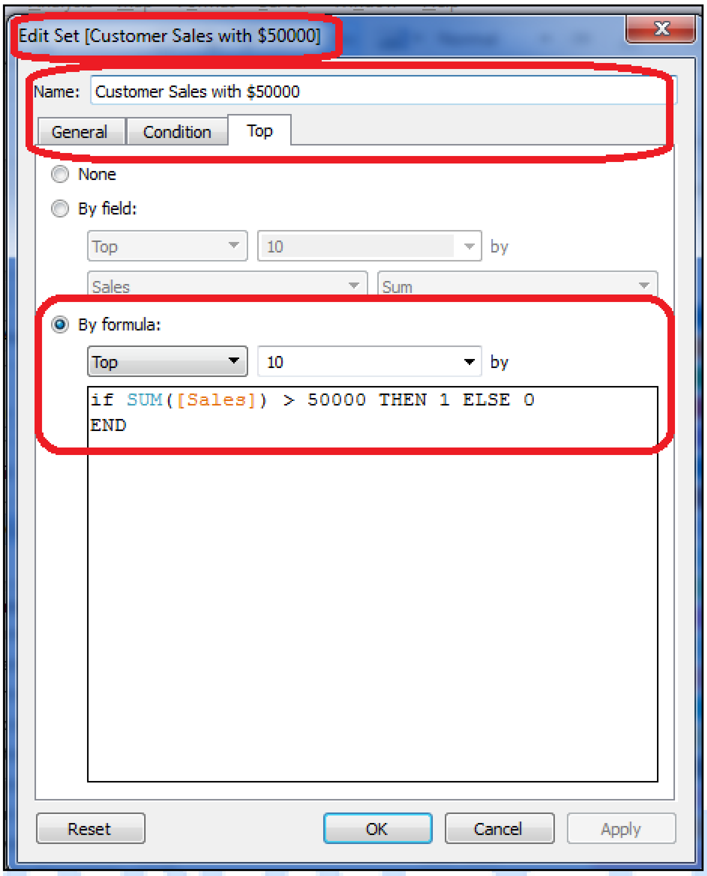
*** Select Top 10 Customers with Sales greater than 50000 (DYNAMIC)**

- Select Customer Name From Dimension -> Right-Click -> Create SET
- SET NAME - Customer Sales with $50000 (Dynamic)

- TOP (tab) - By Formula

  Top 10 By

  Expression: if SUM([Sales]) > 50000 THEN 1 ELSE 0 END



Edit Set [Customer Sales with $50000]

Name: Customer Sales with $50000

○ None

○ By field:

● By formula:

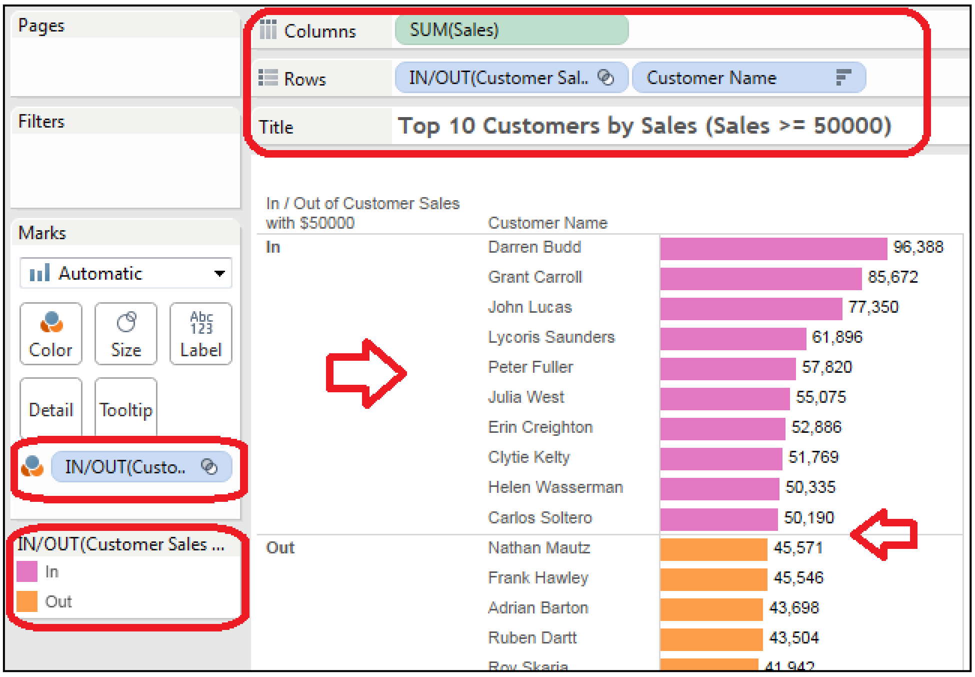
Top 10 by

```
if SUM([Sales]) > 50000 THEN 1 ELSE 0
END
```

Figure: Create Dynamic SET on Customer Name with Top 10 Customer by Sales > 50000

- Analyze visualization on sheet

ROWS	- Customer Sales with $50000 (Dynamic) / Customer Name
COLUMNS	- Sum(Sales)



Columns: SUM(Sales)

Rows: IN/OUT(Customer Sal.. | Customer Name

Title: Top 10 Customers by Sales (Sales >= 50000)

IN/OUT(Customer Sales ...
- 🟪 In
- 🟧 Out

In / Out of Customer Sales with $50000	Customer Name	
In	Darren Budd	96,388
	Grant Carroll	85,672
	John Lucas	77,350
	Lycoris Saunders	61,896
	Peter Fuller	57,820
	Julia West	55,075
	Erin Creighton	52,886
	Clytie Kelty	51,769
	Helen Wasserman	50,335
	Carlos Soltero	50,190
Out	Nathan Mautz	45,571
	Frank Hawley	45,546
	Adrian Barton	43,698
	Ruben Dartt	43,504
	Roy Skaria	41,942

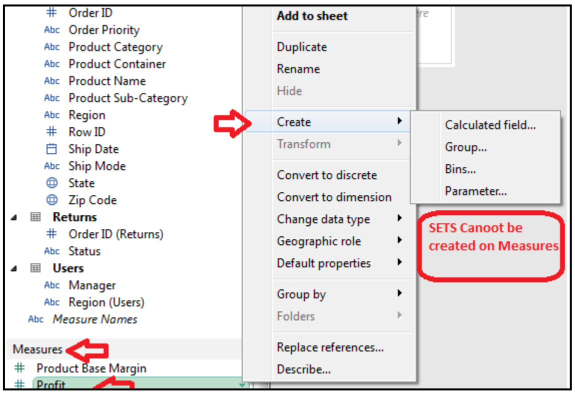
Figure: Top 10 Customers by Sales (Sales >= 50000)

****

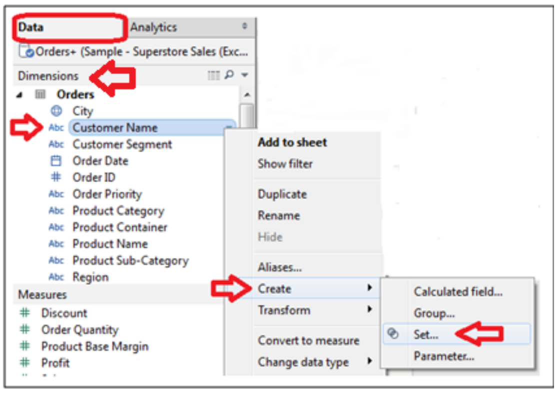
Notes:

****

-> Always Sets can be created only on dimensions. Not possible to create Sets on Measures. Because Measures always contains continuous data but dimensions contain discrete data



SETS Canoot be created on Measures

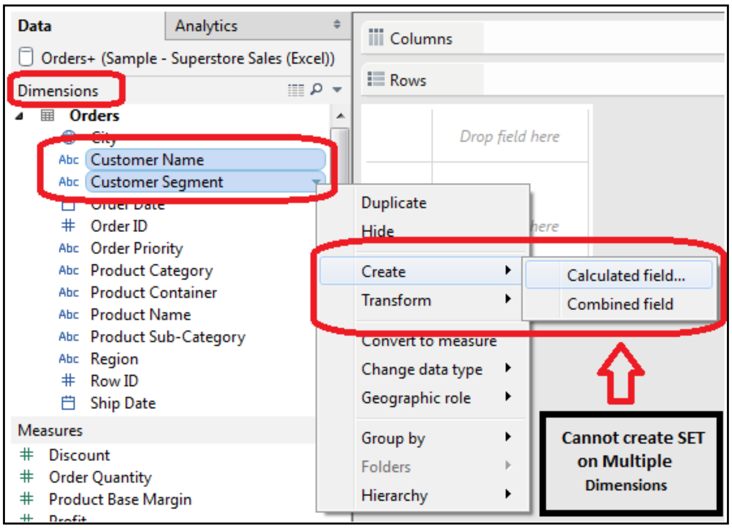


-> Even though, if you convert Measure to Discrete, it is not possible to create SET on Discrete Measure, If we convert a Measure to Dimension, then we can able to create SET on Dimension

Example: Convert **#Unit Price (Measure)** into #Unit Price (Discrete) -> Right-click and Select CREATE--> SET (NOT POSSIBLE)

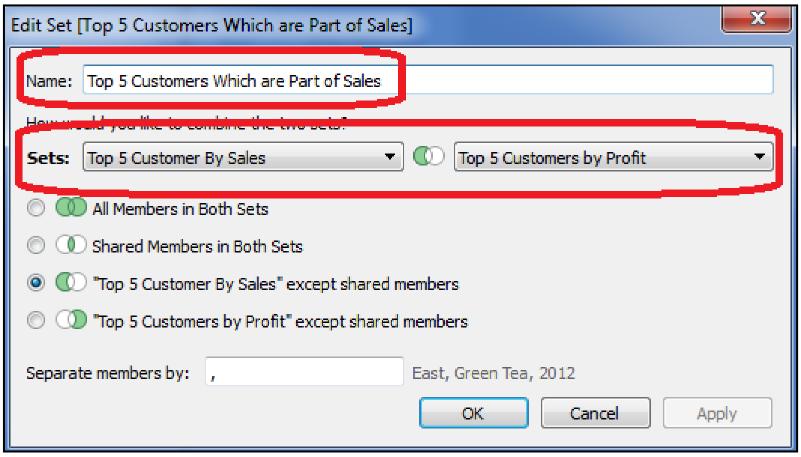
Convert **#Unit Price (Measure)** into #Unit Price (Continuous) -> Right-click and Select CREATE--> SET (POSSIBLE)

-> It is not possible to create a combine set on the Multiple Dimensions



Cannot create SET on Multiple Dimensions

-> We can create a Combined Set based on two other Combined Sets



Edit Set [Top 5 Customers Which are Part of Sales]

Name: Top 5 Customers Which are Part of Sales

How would you like to combine the two sets?

Sets: Top 5 Customer By Sales ▾ Top 5 Customers by Profit ▾

○ All Members in Both Sets

○ Shared Members in Both Sets

● "Top 5 Customer By Sales" except shared members

○ "Top 5 Customers by Profit" except shared members

Separate members by: , East, Green Tea, 2012

Figure: Combined Top 5 Customers By Sales & Top 5 Customers by Profit and Created a Combined Set Top 5 Customers Which are Part of Sales

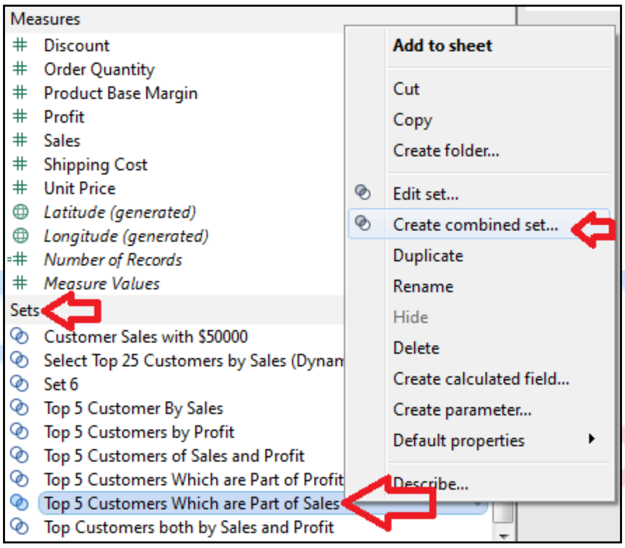
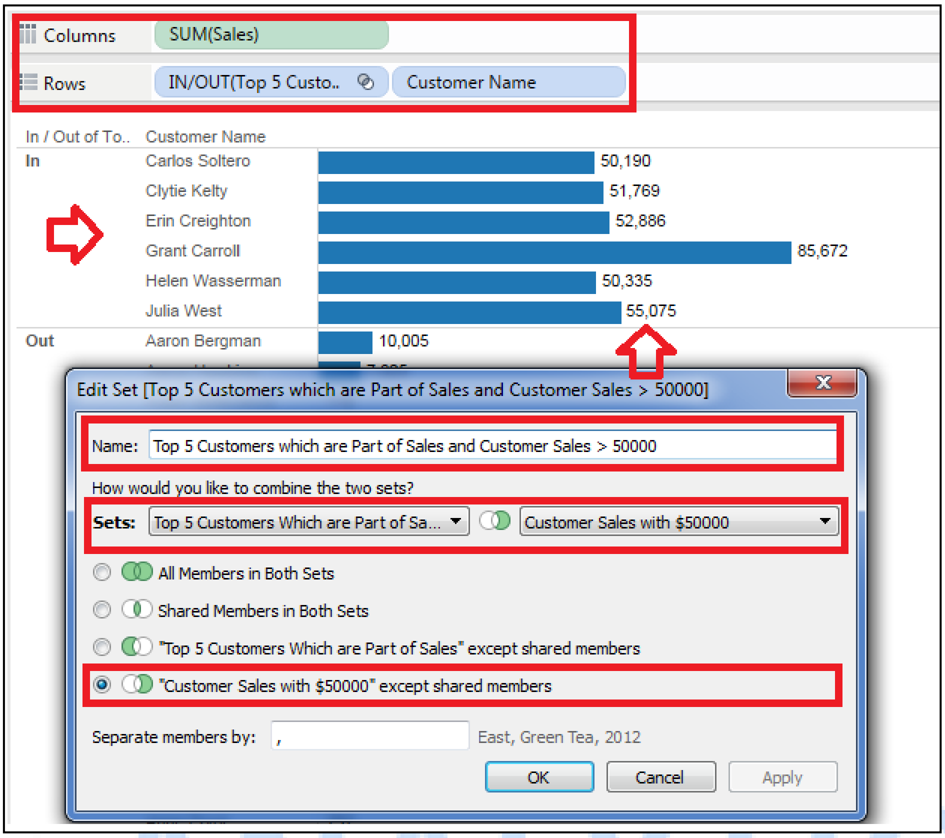


Figure: Creating a New Combined Set by Combining Top 5 Customers which are Part of Sales with Customer Sales with $50000



Columns: SUM(Sales)

Rows: IN/OUT(Top 5 Custo.. | Customer Name

In / Out of To..	Customer Name	
In	Carlos Soltero	50,190
	Clytie Kelty	51,769
	Erin Creighton	52,886
	Grant Carroll	85,672
	Helen Wasserman	50,335
	Julia West	55,075
Out	Aaron Bergman	10,005

Edit Set [Top 5 Customers which are Part of Sales and Customer Sales > 50000]

Name: Top 5 Customers which are Part of Sales and Customer Sales > 50000

How would you like to combine the two sets?

Sets: Top 5 Customers Which are Part of Sa... | Customer Sales with $50000

○ All Members in Both Sets
○ Shared Members in Both Sets
○ "Top 5 Customers Which are Part of Sales" except shared members
● "Customer Sales with $50000" except shared members

Separate members by: , East, Green Tea, 2012

Figure :Creating Combined SET on Two Other Combined SETS

Data Binning

-> In Tableau, It is not possible to create a set on Measures

-> If we want to create a set on Measure, we need to create some set kind of data called

Bin

-> Bins are useful to organize the values of Measure into discrete points

-> In Tableau, Bins are created only for the Relational Data Sources. This feature is not

supported for the multi Dimensional Data Sources

-> In Tableau if you create a Bin, it will automatically create a new dimension. This is

because we are creating the categories of data for the continuous numeric values

Example

-> Design a Bin for the Profit size of 10000 and Display the States falling in Different

Categories

Hierarchy

-> Defining one to many relationship between columns

Country -> Many States
State -> Many Districts
District -> Many Cities

-> The process of viewing data from Higher level to the lower level (or) from top to

bottom is known as Drill-down
-> The process of viewing data from Lower level to High level (or) from bottom to top is

known as Drill-up

28-02-2016

Differences Groups, Sets and Bins

Groups

-> We can create Groups on both Dimensions and Measures
-> If we create any group on the Dimension or Measure, It will create a new field on the

Dimension
-> We cant create any calculations in groups
-> on the Group, we can create a Set
-> In the Group, It is not possible to see all the values (or) members
-> Groups supports Parameters
-> We cant combine two groups directly.
-> If we combine two Groups the Result will not be a Group. It will create a new Dimension

or Field

SET

-> Sets can be created only on the Dimensions
-> If we create any Set, all the Sets will get seperately stored in the Sets of Data

Window

-> We can create calculations on Sets
-> We cant create any Groups and Bins on the Sets
-> In Sets, we can display all the Members in the form of In and Out
-> Sets supports Parameters
-> We can combine two Sets directly
-> If we combine two Sets, it will create a new Set

Bin

-> We can create Bins only on the Measures
-> If we create any Bin, Tableau will create a new Field for the Bin in the Dimension
-> We cant create calculations on Bins
-> We can create a Set on Bin but cant create Group on Bin
-> In Bins, we can see the value in lower or upper limit
-> Bins also supports Parameters
-> We cant combine two Bins directly. But we can combine two bins by Combined Bins option
-> If we combine two Bins the Result will be a Bin

Actions

-> In Tableau, Actions are mainly used to add the interactivity and context to the Data
-> In Tableau, By making use of Actions, we can add the interactivity between multiple

Worksheets. We can link Web pages to the Work sheets, Files to the Work sheets
-> Using these Actions, We can use the Data in one View and Filter the Data in another

view. After Filtering the Data, even we can call the attension to the specific

Results by Highlighting them

-> In Tabelau, There are 3 Types of Actions
1. Filter Action
2.URL Action
3.Highlight Action

1. Filter Action

-> Filter actions are used to enable the interactivity between multiple worksheets
-> This Filter Action allows the user to send information between worksheets
-> In Filter Action, We need to Define Source Sheet and Target Sheet
-> By using Filter Action, we can send information from source sheet to the Target Sheet

by displaying only the related information
-> Filter Actions adds the interactivity at the Worksheet Level and at the Dashboard Level

-> At the Worksheet level, it adds interactivity between multiple Work Sheets

Example

-> Design an Action to enable the Communication between two Work Sheets (by using Filter

Action), in such a way that If you click on first Work Sheet, it should display the

Data in the Second Work Sheet

-> Design an Action to display only Category wise each item Profit. by clicking on one

Category, it should display only the items belongs to the Categorywise Sales in sheet1

and Item wise Profit on Other Sheet

URL Action

-> URL Actions are mainly used to enable the communication and interactivity between

worksheet and web page
-> URL Action is Just a hyperlink that points to the web page,file or any other resources
-> URL Actions are mainly used to add more information to the data view in Tableau

Example:

Design an action in such a way that display the complete information of State having

highest Sales

HighLight Action

-> In Tableau, Highlight Actions allows to call the attension to the particular part of

the Data View and dimming the other part of the Data View
-> In Tableau, we can add the particular part of the Data View by selecting the marks in

the Data View
-> If we define a High Light Action and if we select the marks in the Data View, all other

marks will get dimmed by calling the attension to the part we selected
-> All Selections will be saved in the workbook

-> We can Perform the Highlight action in multiple ways
1. Using either work sheet or Dashboard
2. Using the color legend High lighting
3. Using the Toolbar Highlighting action

Example

-> Design a Dashboard in such a way that if we click on any state, it should high light

only the cities under that state

Building Maps in Tableau

In Tableau, Maps can be build in 2 ways
1. In the Automated Way
2. In the Manual Way

Building Maps Automatically in Tableau

-> In Tableau, we cna able to build Maps automatically by making use of geographical

fields like Country, State, city   etc...
-> If Tableau identifies the Geo graphical fields, it automatiocally creates 2 Fields

known as Latitude and Longtude
-> The Latitude and Longtude fields wont exist in the Data Source
-> Tableau automatically create these two fields in the Data Window

Example

-> Building a Symbol Map in Tablau for Different States of U.S.A?
-> Building Filled Map in Tableau for Different States of U.S.A?
-> Design a Filled Map to display the Statewise Sales and Profits of U.S.A and also enable

the User to select the required   state on the map at Run Time

Parameters

-> A parameter is a value that be changed by the user interacting with a view, rather

than your visualizations using a constant value.
-> Parameters allow you to give ursers control over visualization
-> Parameters are dynamic values that can replace constant values in Calculations,

filters and Reference Lines
-> Parameter
* The user selects/ or enter the value
* based on user selection the data view is displayed

Some Parameter Usages

- What - If analysis
- User - Control Thresholds
- Dynamic field, axis, title, etc.

Parameter Building Blocks:
----
1. Create the Parameter
2. Use the Parameter in either a Calculated Field, Reference Line, SET or FILTER

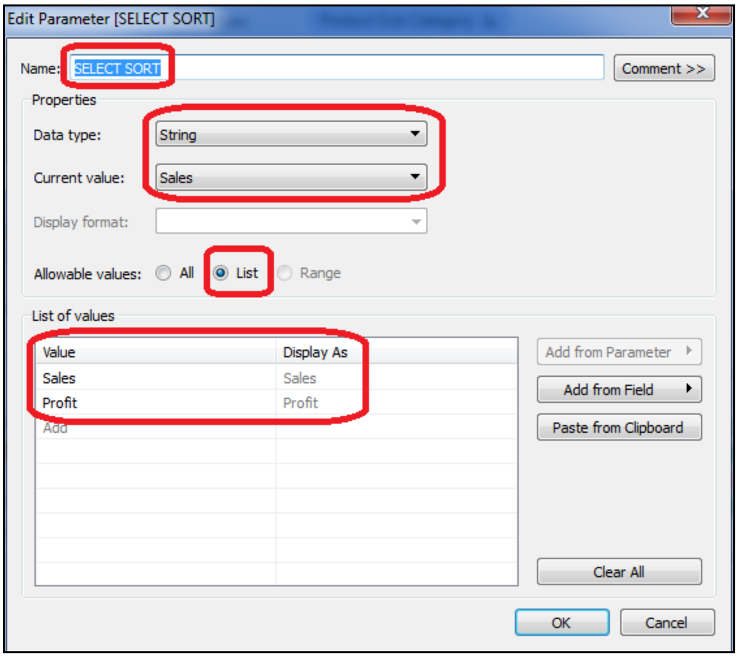
****
Example1: Sorting Dynamically on the Field Specified
****

0. Create a Data View
COLUMNS : Sum(Sales)
ROWS : Product Sub Category
Drag and Drop Sum(Profit) onto Color Legend on Marks Card
Analyze the data by sorting manually from menu
Sorts the data by Ascending or Descending on Sales
Create a Parameter that accepts the user specified field and sorts dynamically
User Specified Values: Sales / Profit

1. Create a Parameter Name: SELECT SORT

Data Type	: String
Current Value	: Sales
Allowable Values	: List
List of Values	: Sales Profit

Drag and Drop Parameter onto workspace area / data view window



Edit Parameter [SELECT SORT]

Name: SELECT SORT

Properties

Data type: String

Current value: Sales

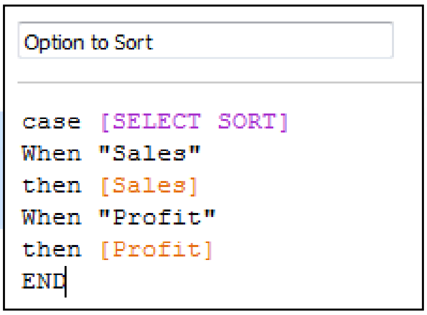
Display format:

Allowable values: ○ All ● List ○ Range

List of values

Value	Display As
Sales	Sales
Profit	Profit

2. Create New Calculated Field



Option to Sort

```
case [SELECT SORT]
When "Sales"
then [Sales]
When "Profit"
then [Profit]
END
```

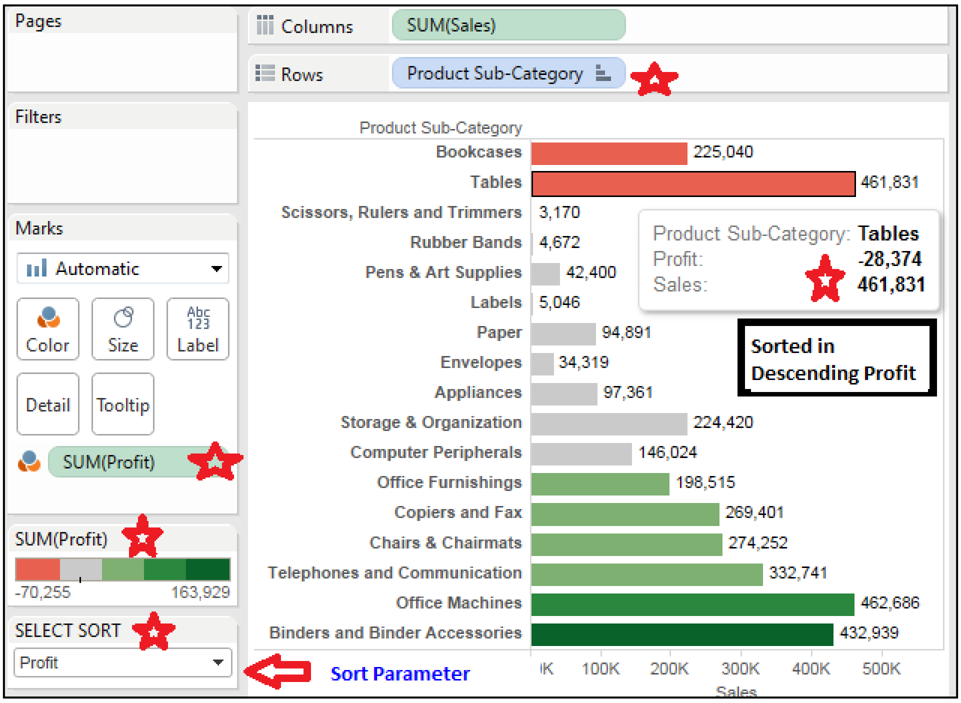
3. Create a link between data view and Parameter to dynamically display sorted values

Select Product Sub Category on ROWS Shelf and click pop-up menu

Select SORT

Sort Order	:	Ascending
Sort By	:	Field
		Option to Sort / Aggregation: Sum

4. Select Sales / Profit field from SELECT SORT Parameter on the view and analyze the sorted values on the Data View window



Columns: SUM(Sales)

Rows: Product Sub-Category

Marks: Automatic

SUM(Profit)

SELECT SORT: Profit

Product Sub-Category: Tables
Profit: -28,374
Sales: 461,831

Figure: Created a SELECT SORT parameter for the user to Select Sales / Profit field for sorting in descending order

****

Example2: WHAT - IF ANALYSIS

Create a Data View with Actual Sales and Projected Sales (Dual Axis)
The user state the PERCENTAGE by which the Projected Sales increase and analyze the results on data view (comparison)

****

Steps:

0. Create a Data View

COLUMNS : QUARTER(ORDER DATE) ---> Right click ---> Select Quarter Q2 2015 representation

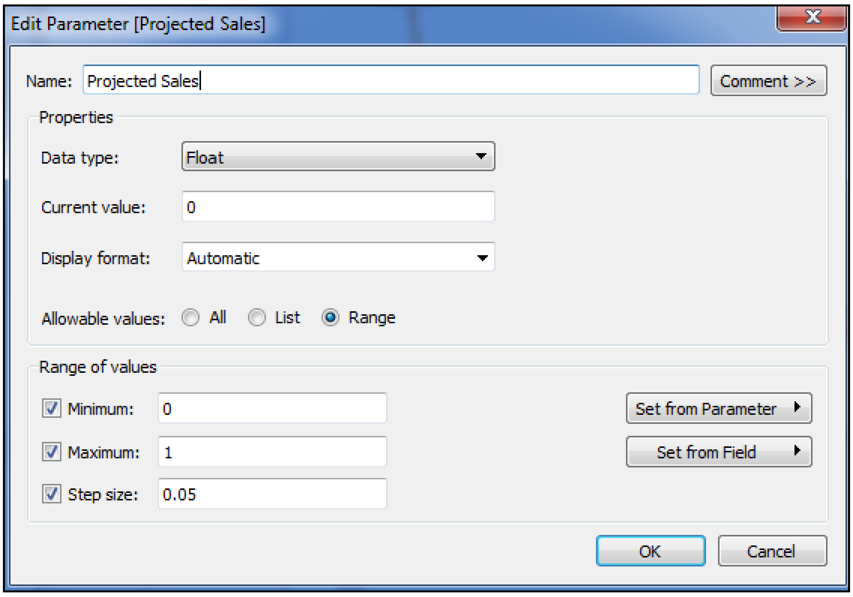
ROWS : Sum(Sales)

Generates a Line Chart what is the total sales for Each Quarter in a Year

1. Create a New Parameter: Projected Sales

Data Type	: Float
Current Value	: 0
Display Format	: Automatic
Allowable Values	: Range
Minimum	: 0
Maximum	: 1
Step Size	: 0.05

Select Projected Sales Parameter on Data View



Edit Parameter [Projected Sales]

Name: Projected Sales

Properties

Data type: Float

Current value: 0

Display format: Automatic

Allowable values: ○ All ○ List ● Range

Range of values

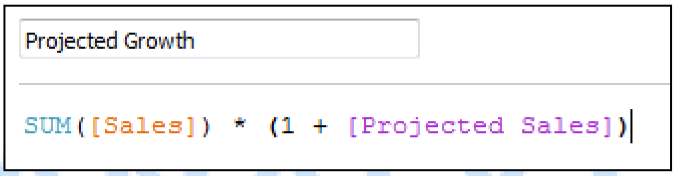
☑ Minimum: 0

☑ Maximum: 1

☑ Step size: 0.05

2. Create a New Calculated Field: Projected Growth

Expression : SUM([Sales]) * (1 + [Projected Sales])



Projected Growth

```
SUM([Sales]) * (1 + [Projected Sales])
```

3. Design a View

COLUMNS : QUARTER(ORDER DATE) ---> Right click ---> Select Quarter Q2 2015 representation

ROWS : Sum(Sales) / AGG(Projected Growth)

- Right Click AGG(Projected Growth) and Select DUAL AXIS
- Right Click Projected Growth on Data View and select Synchronize Axis
- Analyze the Dual Axis for Actual Sales and Projected Sales
- Select Projected Sales parameter with various values

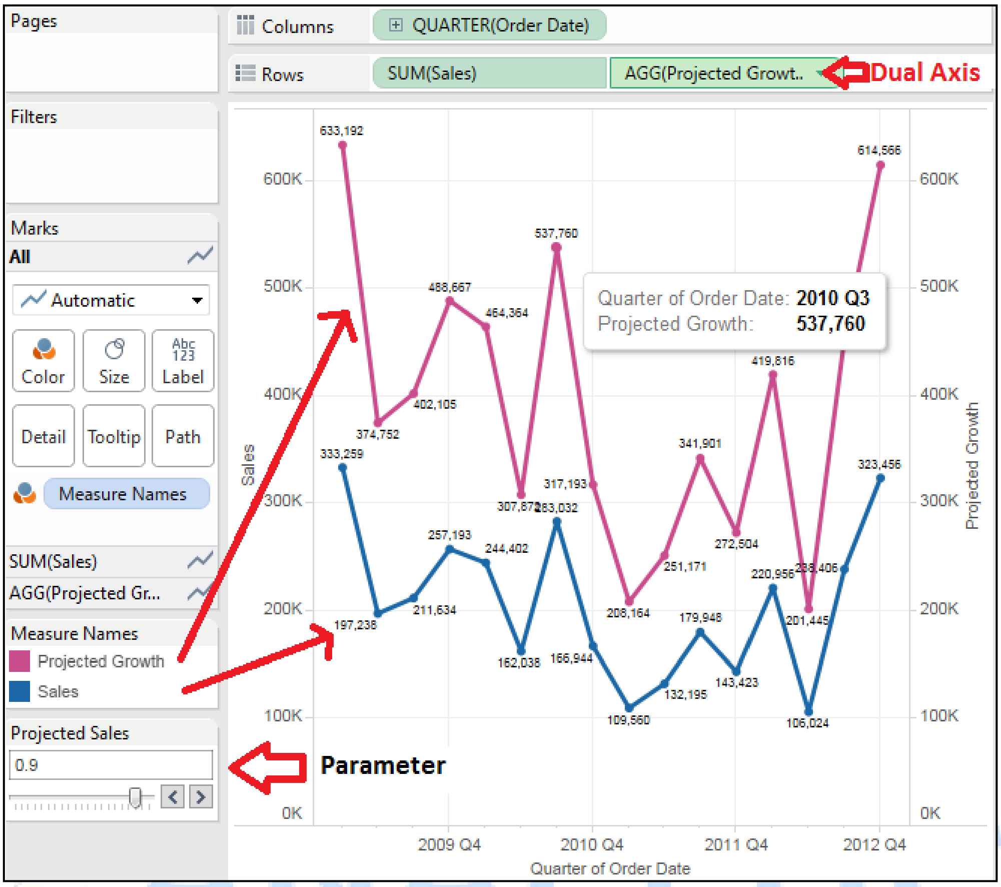


Figure: Actual Sales v/s Projected Sales (Parameter --> Projected Sales)

****

Example3: Generate Top N Ranks by Sum(Sales) for each Product Category

Report User selects the Number of Ranks

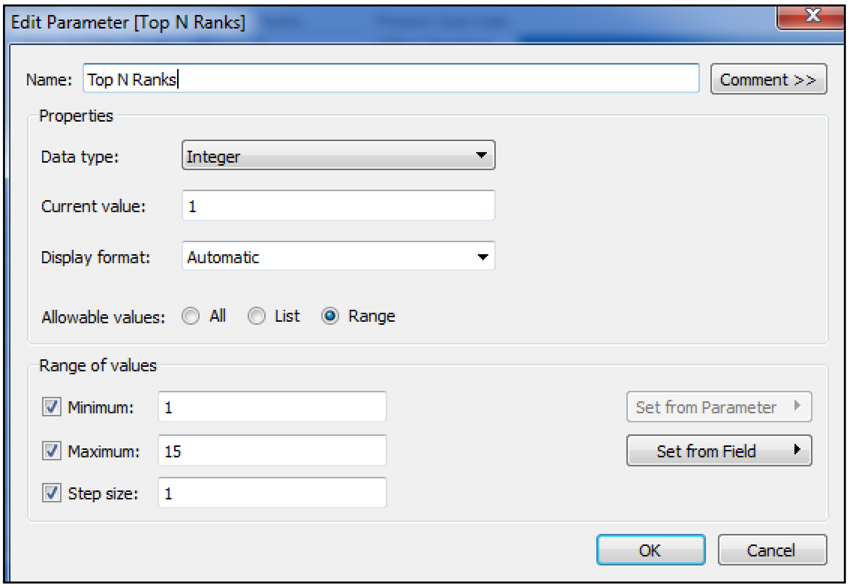
****

0. Create a New Calculated Field

Rank : Index()

Returns the index of the current row in the partition

1. Create a New Parameter: Top N Ranks



Edit Parameter [Top N Ranks]

Name: Top N Ranks

Properties

Data type: Integer

Current value: 1

Display format: Automatic

Allowable values: ○ All ○ List ● Range

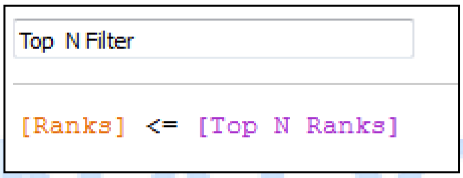
Range of values

☑ Minimum: 1

☑ Maximum: 15

☑ Step size: 1

2. Create a New Calculated Field: Top N Filter



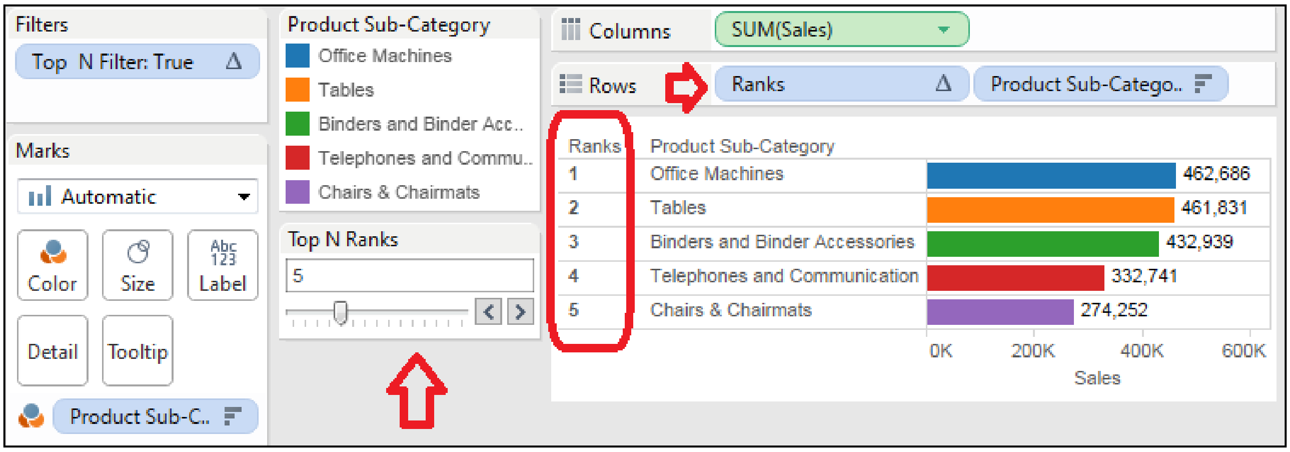
Top N Filter

```
[Ranks] <= [Top N Ranks]
```

3. Drag and Drop Top N Filters onto Filters Shelf --> Right click Edit Filter and Select True

COLUMNS : Sum(Sales)

ROWS : Rank / Product Category



Filters: Top N Filter: True

Columns: SUM(Sales)

Rows: Ranks / Product Sub-Catego..

Top N Ranks: 5

Ranks	Product Sub-Category	Sales
1	Office Machines	462,686
2	Tables	461,831
3	Binders and Binder Accessories	432,939
4	Telephones and Communication	332,741
5	Chairs & Chairmats	274,252

Exercise: Alter the above to Create Bottom N Ranks

-> Multiple Data Views using Single Work Sheet

-> Create a Data View
-> Create a Parameter
-> Create a Calculated Field
-> Drag Calculated Field on to Data View Field

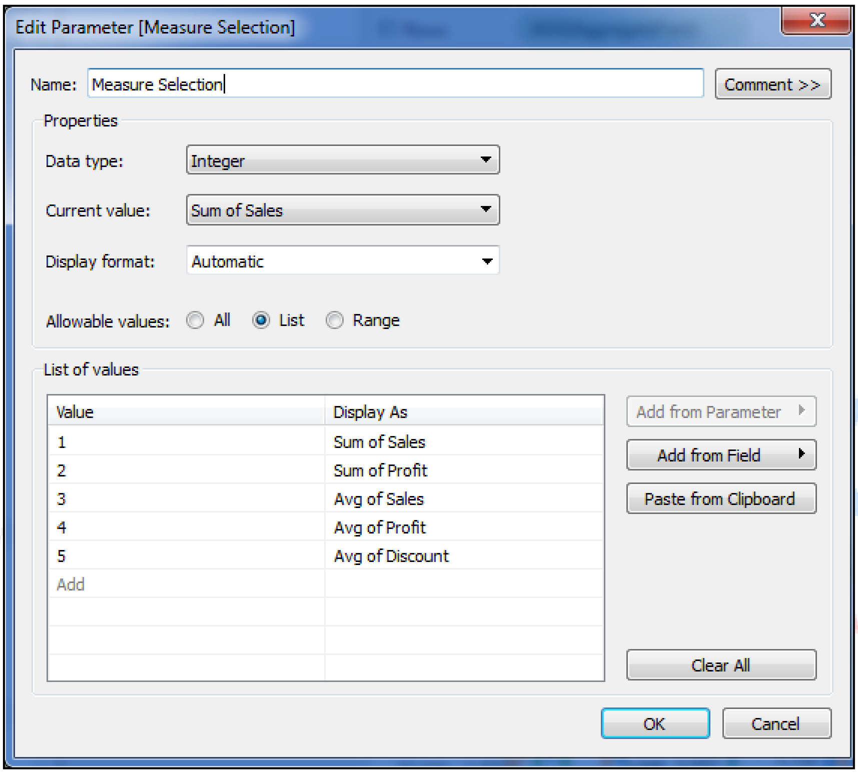
****

Example4: Dynamic Measure Selection with Parameters

End-User makes a selection of Aggregation and Analysis Field on which visualization to be generated

****

0. Create a New Parameter: Measure Selection



Edit Parameter [Measure Selection]

Name: Measure Selection

Properties

Data type: Integer

Current value: Sum of Sales

Display format: Automatic

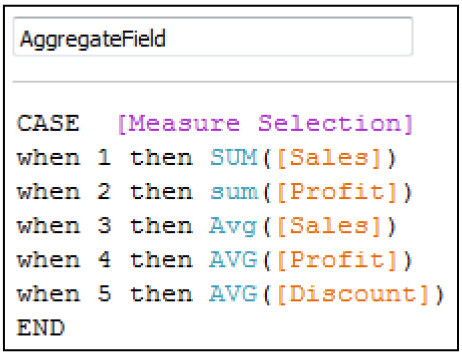
Allowable values: ○ All ● List ○ Range

List of values

Value	Display As
1	Sum of Sales
2	Sum of Profit
3	Avg of Sales
4	Avg of Profit
5	Avg of Discount
Add	

- Select Measure Selection Parameter on Data View (Show Parameter Control)
- Click on the menu option for the Measure Selection Control, and change it to a single value list

1. Create a New Calculated Field:



AggregateField

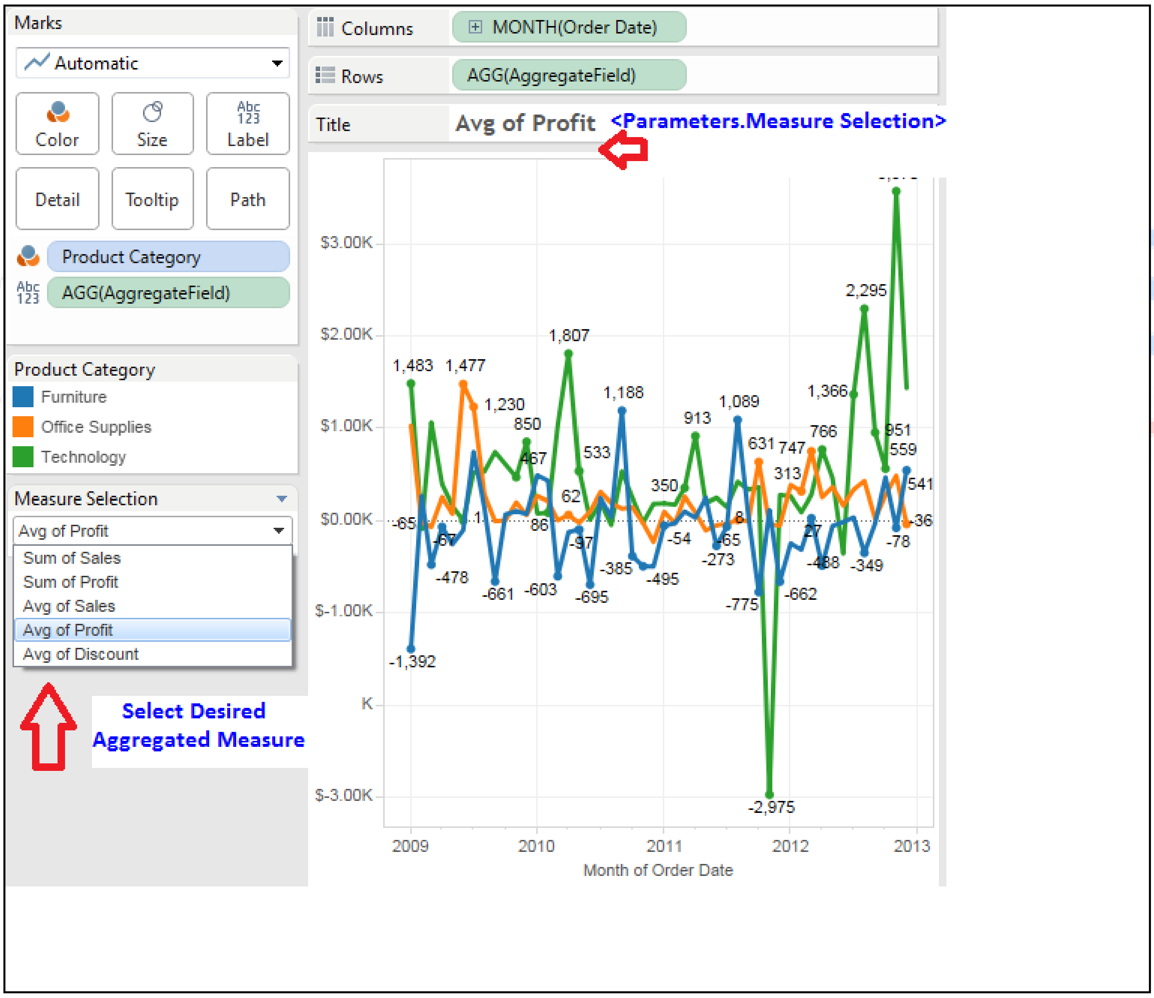
```
CASE  [Measure Selection]
when 1 then SUM([Sales])
when 2 then sum([Profit])
when 3 then Avg([Sales])
when 4 then AVG([Profit])
when 5 then AVG([Discount])
END
```

2. Create a Data View

COLUMNS : Month(ORDER DATE)
ROWS : AggregateField
Drag and Drop Product Category onto Color Legend
Generates a Line Chart

3. Right-click on the axis for AggregateField and choose Edit Axis. Clear the title.
- Right-click on the axis and choose FORMAT to display the scale as currency to Thousands
- Select Worksheet --> Show Title, and double click the title to edit it.
- Clear the sheet name , and use the Insert Menu to call the variable for <Parameters.Measure Selection>

4. Rename the Worksheet to 'Dynamic Measure'
Analyse the output result by selecting various Aggregated Measure in Select Box



Columns: MONTH(Order Date)
Rows: AGG(AggregateField)
Title: Avg of Profit <Parameters.Measure Selection>
Measure Selection: Avg of Profit

Figure: Dynamically select summarized measure from Parameter 'Measure Selection'

****

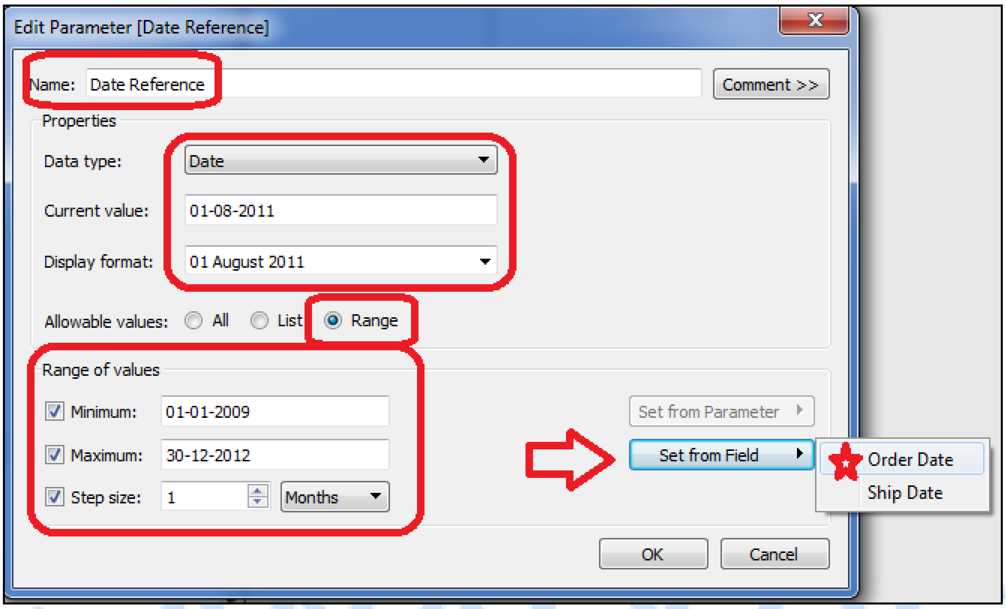
Example5: Dynamic Reference Line

Business Report User would like to move the point of reference line by date and show before and after based on their selection

****

0.Create a duplicate worksheet of "Dynamic Measure" and Re-name to 'Dynamic Reference Line' (Previous Visualization)

1. Create a New Parameter: : Date Reference



Edit Parameter [Date Reference]

Name: Date Reference

Properties

Data type: Date

Current value: 01-08-2011

Display format: 01 August 2011

Allowable values: ○ All ○ List ● Range

Range of values

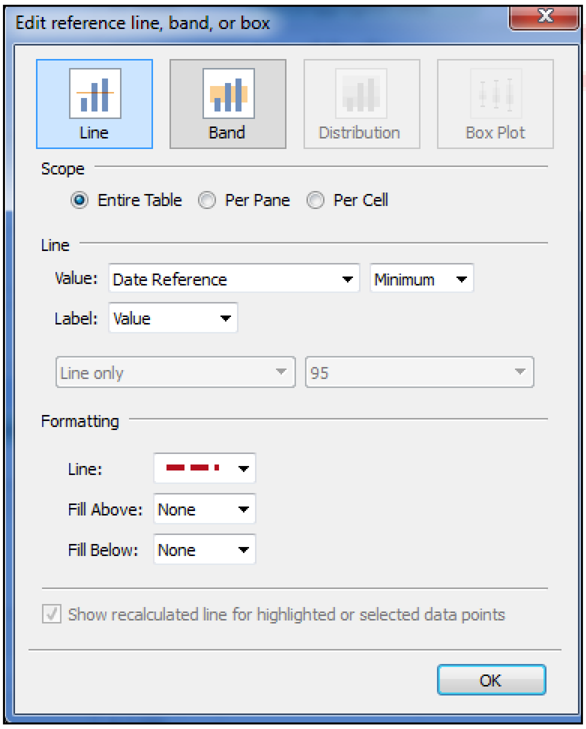
☑ Minimum: 01-01-2009

☑ Maximum: 30-12-2012

☑ Step size: 1 Months

Set from Field: Order Date

2. Add Reference Line from Analytics Tab, drag and drop it on the axis



Edit reference line, band, or box

Line

Scope: ● Entire Table ○ Per Pane ○ Per Cell

Line

Value: Date Reference Minimum

Label: Value

Formatting

Line: ▬ ▬ ▪

Fill Above: None

Fill Below: None

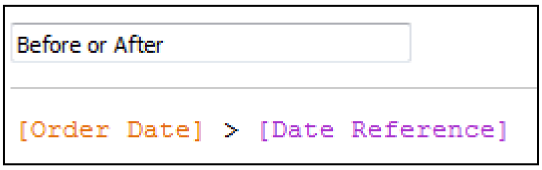
☑ Show recalculated line for highlighted or selected data points

3. Select 'Date Reference' Parameter onto Data View Window by 'Show Parameter Control'

4. Create 'New Calculated Field' to determine whether the Order Date is before or after

Reference Date (Create Field that accepts Boolean (T/F)

New Calculated Field : Before or After

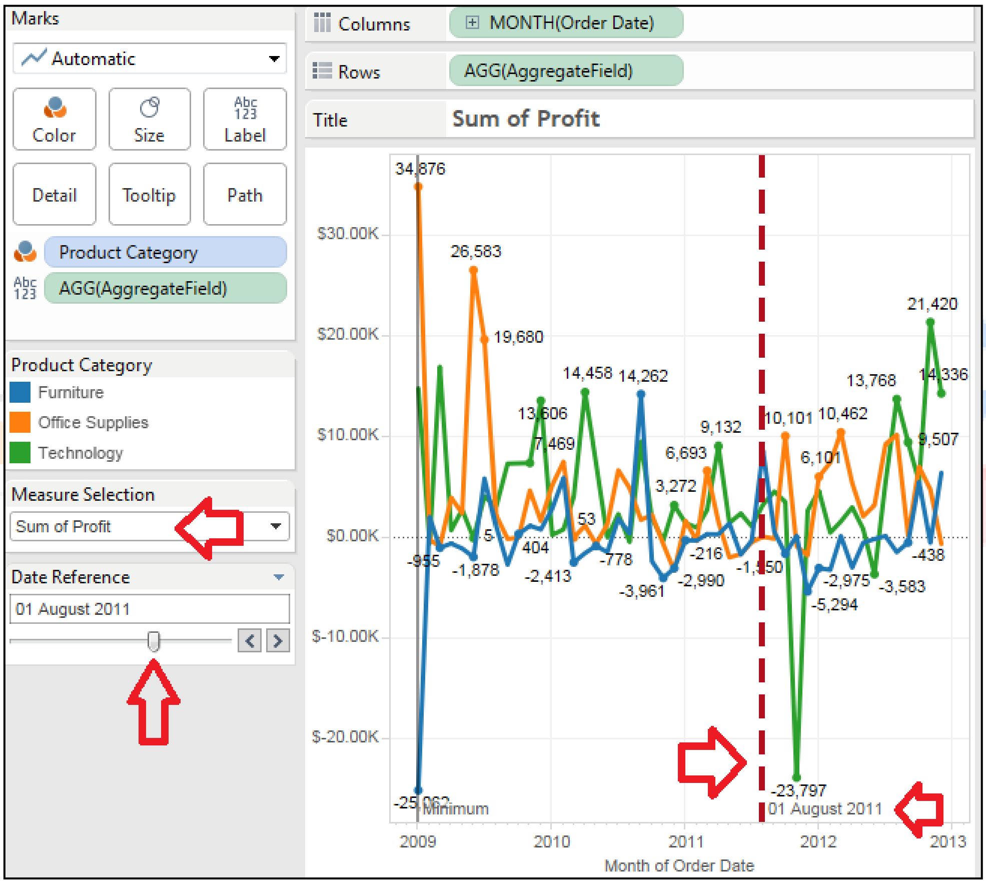


Before or After

```
[Order Date] > [Date Reference]
```

5. Drag and Drop Category field and Before or After onto Color legend once it appears as a Detail, click on the menu to the left of the field, and apply it to color

6. Analyse the output result by selecting various Date References and how the data is divided by reference line



Marks: Automatic

Product Category

AGG(AggregateField)

Product Category

🟦	Furniture
🟧	Office Supplies
🟩	Technology

Measure Selection: Sum of Profit

Date Reference: 01 August 2011

Columns: MONTH(Order Date)

Rows: AGG(AggregateField)

Title: Sum of Profit

Figure: Created Date Reference Parameter for the use to select Date and the visualization generated display the data divide into two

Clip Boards

-> Even in the Tableau, we can analyse the Data Directly with out creating any Data

Connection

-> Some times if you want to pull the Data from outside Data Sources for quick analysis,

directly we can copy the data and paste the Data Into Tableau
-> Tabelau automatically creates a Data Source in the form of Clipboard and we can perform

analysis directly from clipboard
-> When we save the work book, the Clip board data also get saved as Tab Delimited Text

file into our Tableau Repository
-> We can create the Clipboards by copying and Pasting the Data from various applications

like MS Excel, MS Word and even from HTML Pages also
-> When we copy and paste the Data inot Tableau, Tableau makes use of Microsoft JET Engine
-> Microsoft JET Engine is a database engine on which several Microsoft Products have been

built
-> JET Stands for Joint Engine Technology
-> This Microsoft JET Engine copies the Data and represent the Data in Tableau in the form

of clipboard

Example

-> Copy the Data
-> go to Data Tab
-> Click on Paste Data

04-03-2016

Charts

Bar Charts
Line Charts
Pie Charts

Straight Tables
Pivot Tables

Atleast One Dimension and One Measure or one Expression

Yearwise Sales

Year
Sum(Sales)

Countrywise Sales

Country
Sum(Sales)

Bar Chart

-> Useful to display a set of items side by side
-> Useful for Comparision

Ex:
Yearwise Sales

Countrywise Sales

Country (200+)
Sum(Sales)

Note: It dont have better visibility in Charts like Bar, Line Pie chart etc...

Pie Chart

-> Useful to display Total Share
-> It will take entire Pie as 100% and Display each item share

Ex:

Categorywise Sales
Department wise Sales
Top 5 Countries
Top 10 Customers

Line Chart

-> Useful to display changes/Growth/Trends
-> Useful for Comparision Purpose

-> to display more number of items properly we need depend on Tables

-> In Tabelau, we can able to create 3 Different types of Charts
1. Uni-varite charts
2. Bi-Varite Charts
3. Multi-Varite Charts

1. Uni-varite charts

-> Analysis involving the single measure in TABELAU is known as uni varite charts

Text Tables (or) Cross Tables and Highlight Table

-> Tables in the Tableau are great way to represent large amount of data in the smaller

area
-> In Tableau, we are having 2 types of tables
1. Text Tables
2. Highlight Tables

1. Text Tables

-> For Text Tables, minimum requirements or 1 or more dimensions and 1 or more measures

2. Highlight Tables

-> 1 or more Dimensions
1 measure

-> The Highlight table highlighting the value of measures based on value range in Tableau

Note

-> In the output view, Text Table will display all the measure values where as highlight

table highlights the value of a measure based on the value range in the table

-> Both the highlight and Text Tables are uni varite charts

Bar Charts (or) Histograms

-> In Tableau, we are having 2 types of Bar Charts
1. Horizontal Bar Chart

2. Vertical Bar Chart

Again the vertical Bar Chart is divided into two types
1. Stacked Bar Chart
2. Side by Side Bar Chart

- > Bar Chart is the great way to compare the Data across various categories
-> In the Bar Chart, length of the each bar indicates value of the particular category
-> Bar Chart is the easy way to understand the different categorial information such as

Country, State, Region, Continent, Years, Months etc...

Horizontal Bar Chart

-> The minimum requrement for hirizontal Bar Chart is zero or more Dimensions and 1 or

more measures

Stacked Bar Chart

-> 1 or more Dimensions
    1 or more Measures

Side by Side Bar Chart
----
1 or more Dimensions and 1 or more measures

Note

-> In Horizontal Bar Chart, each value in a given category is represented with a seperate

bar where as in Stacked bar all the values of a given category is represented in a

Single Bar vertically
-> Side by Side by requires atleast 3 fields

-> The horizontal bar chart and vertical bar charts are uni varite charts

Line Charts

-> Line Chart is the great way to compare the categorial data over the period of time
-> It is mainly used to analyze the trends over the period of time.

Types of Line Charts

-> we have 3 different types of Line charts

1. Continuous Line Chart
2. Decrete Line Chart
3. Dual Line chart

05-03-2016

----

1. Continuous Line Chart

-> It requires atleast one Date field, o (or) more Dimensions and 1 (or) more Measures

i) Descrete Line Chart

-> It requires atleast one Date field, 0 (or) more Dimensions and 1 (or) more measures
-> It requires one Discrete Date

Dual Lines

-> It requires atleast one Date Field which is of either continuous (or) descrete, 0 (or)

more Dimensions and 2 measures

Note:

-> The Continuous line chart requires continuous Date Field where Discrete Line Chart

requires discrete Data Field. Dual lines requires two measures

-> In the Line Charts Continuous Line Chart and descrete Line Chart are of Type Uni-Varite

Area Chart

-> Area Charts are mainly used to indicate the growth of particular category
-> Area Chart is an extension of a Line Chart
-> An Area Chart shows the Line of a Measure how it growing and also fills the area below

the line based on the value of a measure
-> Area Charts are the great way to compare the growth of a particular category

-> In Tableau, we have 2 Types of area charts
1.Continuous Area Chart

2.Descrete Area Chart

1.Continuous Area Chart

-> It requires one Date Field , 0 or more dimensions and 1 or more measures

2. Descrete Area Chart

-> It requires one Date Field , 0 or more dimensions and 1 or more measures

Note:

-> The Descrete area chart and the continuous area charts are the uni-varity charts
-> Descrete area chart requires descrete Date field where as Continuous area chart

requires continuous Date Field

Circle Charts

-> In Tableau, we have 2 types of circle charts
1. Circle Views
2. Side by side circles

-> Circle charts are another wat to represent the categorial data in the form of circles

1. Circle Views

-> In order To create Circle views in the Tableau, it requires 1 or more Dimensions and 1

or more measures

2. Side by side circles

-> In order To create Side by side circles in the Tableau, it requires 1 or more

Dimensions and 1 or more measures
-> It require atlease 3 Fields

-> Both Circle Views and Side by Side charts are uni-varite Charts

Packed Bubbles

-> Packed Bubbles is the another way to represent the each categorieal data based on their

measure value

-> It requires atleast one or more dimensions and 1 or 2 measures

Pie Chart

-> It is the another way to represent the data in the form of slices

Note:

-> For the effective look of Pie chart and for the best practice dont create a Pie Chart

with more than 5 slices

Heat Maps

-> The another way to represent the categorial Data is Heat Maps. A Heat map is a visual

representation of number in a table (or) in the form of grid in such a way that the

bigger numbers are encoded by dark colors (or) bigger sizes and the smaller numbers

are encoded by light colors (or) smaller Sizes
-> This type of representation of data makes the readers pattern very easier and the

reader can able to analyse the data very fastly

-> The Minimum requirement for the Heat map is 1 or more Dimensions and 1 or 2 measures

Tree maps

-> The tree map is another way to represent categorial Data. Treemap fills the portion by

thick color if the value is more and it fills with light color if the field value is

less
-> Tree maps requires 1 or more Dimensions and 1 or 2 measures

Gantt View

-> Gantt view is the modified representation of the Bar Chart

-> In order to create Gantt view it requires atleast one Date , 1 or more dimensions and 0

to 2 measures

07-03-2016

Box-and-whisker Plots

-> Box-and-whisker plots diaplys the Distribution of the measure values for a given

category in the form of Box

-> Box-and-whisker plot also known as Box-Plot

-> In order to this chart, we need o or more dimensions and 1 or more measures

Maps

-> In Tableau, Maps also comes under uni-varite charts

-> We have 2 types of Maps

1.Symbol maps

2. Filled Maps

Bi-Varity Charts

-> Analysis involving in two measures in the Tableau is known as Bi-Varity Charts

-> The Different Bi-Varite Charts available in Tableau are

1. Text Tables

2.Heat Maps

3.Pie Charts

4.Symbol Maps

5. Horizontal Bars

6.Stacked Bars

7.Side by side Bars

8.side by side circles

9.Circle Views

10.Bullet Graphs
11.All line charts
12.All area Charts
13. Combination chart
14. Packed Bubbles
15. Box and Whisker plots
16. Gannt Views
17. Scatter Plots

Scatter Plots

-> These are oftenly used to identify and to observe the relationship between two measures

and any two variables. By looking into the scatter plots any user can quickly observe

the trends
-> It requires atleast o or more dimensions and 2 to 4 measures
-> The scatter plot displays the relationship in terms of small circle called scatter

Bullet Graphs

-> Bullet graphs are the modified view of the Bar Chart.
-> Bullet graphs requires o or more Dimensions and 2 measures

Multi-Varite Charts

-> Analysis involving more than two measures are known as multi varite charts

-> The Different multi-varite charts are
1.Text Tables
2.Horizontal,stacked,side-by-side Bars
3.Circle view and side-by-side circle
4. continuous and discrete line chart
5.Diffrent types of Area Charts
6. Scatter plot
7. Box-and-whisker plots

Reference Lines,Bands,Box,Distribution

-> In Tableau, the reference lines are used to mark a specific value on axes and also we

can shade (or) color the particular area along the axes and also we can display the

distribution values and full values

-> In Tableau, we can add unlimited reference lines, Bands and Boxes

Types of Reference lines

-> Line
-> Band
-> Boxes
-> Box Plots

Line

-> Reference line add a line at a value on the axes. It will add the line for the constant

value (or) for the computed value

Bands

-> Bands shades or colors an area in the data view b/w two constant (or) computed values

along the axes

Distribution

-> In the distribution, we can add the coloring or shading slong the axes to indicate the

dsitribution of the values

-> Distribution of values can be defined by confidence, intervals,percentages,percemtiles

(or) standard deviation.

-> These type of reference line mainly used to create bullet chart

Box Plot

-> Box plot also describes the distribution of values by adding the box plots. This box

plot shows quartiles, viscus

-> Tabelau provides different Bo plot styles and allows you to configure the location of

the viscus

Ex:

-> Add a reference line in order to indicate the states for those states whose total

profit is more than the average profit of all the states

-> Design a reference line for the department wise total sales for each year and display a

line for those States whose sales is more than average Sale of all the

departments for one year

09-03-2016

-> Design a reference band for yearwise, category wise total profits and fill the band in

between average total profit and maximum total profit?

-> Create a Constant Band for the Dept wise, Year wise total Profits?

-> Design a band for the Customer segment wise, Dept wise Total Sales and for each

customer segment display a band the Constant value and change the value dynamically

Table Calculation

-> Table Calculations are the calculations that are applied to the entire values in the

table

-> In Tableau, the table calculations can be performed in 2 ways

1. Automatic way
2. Manual way

-> What type of calculation we are performing

-> Where we are performing the calculation or computation

-> Again where we are performing the calculations is having 2 parts

1. Addressing Field
2. Partitioning Field

Addressing Field

-> It indicates how we are addressing the calculation field

-> Addressing field defines part of the table where we are applying the calculation
-> Addressing fields mostly be dimensions

Partitioning Fields

-> The Dimensions that define how we are dividing the Data View to perform the calculation

(or) How we are grouping the Data Fields to perform the calculations are known as

Partitioning Fields

Addressing Options

->In Tableau, we are having the list of addressing options

1. Table Across
2. Table Down
3. Pane Across
4. Pane Down
5. Table Across then Down
6. Pane Across then Down
7. Cell

1. Table Across

-> If we select the addressing option as Table Across Tableau perfroms (or) address the

calculation along the entire table moving horizontally

2. Table Down

-> If we select the addressing option as Table Down Tableau perfrom the calculation along

the entire table vertically

3. Pane Across

-> If we select the addressing option as Pane across Tabelau perform the calculation

across each pane horizontally

4. Pane Down

-> If we select the Adressing option as Pane down, Tableau perform the calculation across

each pane vertically

5. Table across then Down

-> If we select the addressing option as table across then down, Tableau computes or

addresses the table across the   horizontally first then down the table vertically

6. Pane across then down

-> If we select the addressing option as Pane across then down, Tableau computes or

addresses the Pane across the   horizontally first then down the pane vertically

7. Cell

-> If you select the addressing option as cell , Tableau perfrom the computation across

each cell in the Data View

******************************Analytics Path***********************
Advanced Visualizations

Bar in Bar - To compare two similar values within the same view
Example: Annual Sales Comparison of Two different years (2012 and 2011)
****

0. Create Two New Calculated Fields for each Year

Name	: 2012 Sales
Expression	: if YEAR([Order Date]) = 2012 then [Sales] END
Name	: 2011 Sales
Expression	: if YEAR([Order Date]) = 2011 then [Sales] END

2. Create the Data View

ROWS	: Product Category / Product Sub Category
COLUMNS	: 2012 Sales

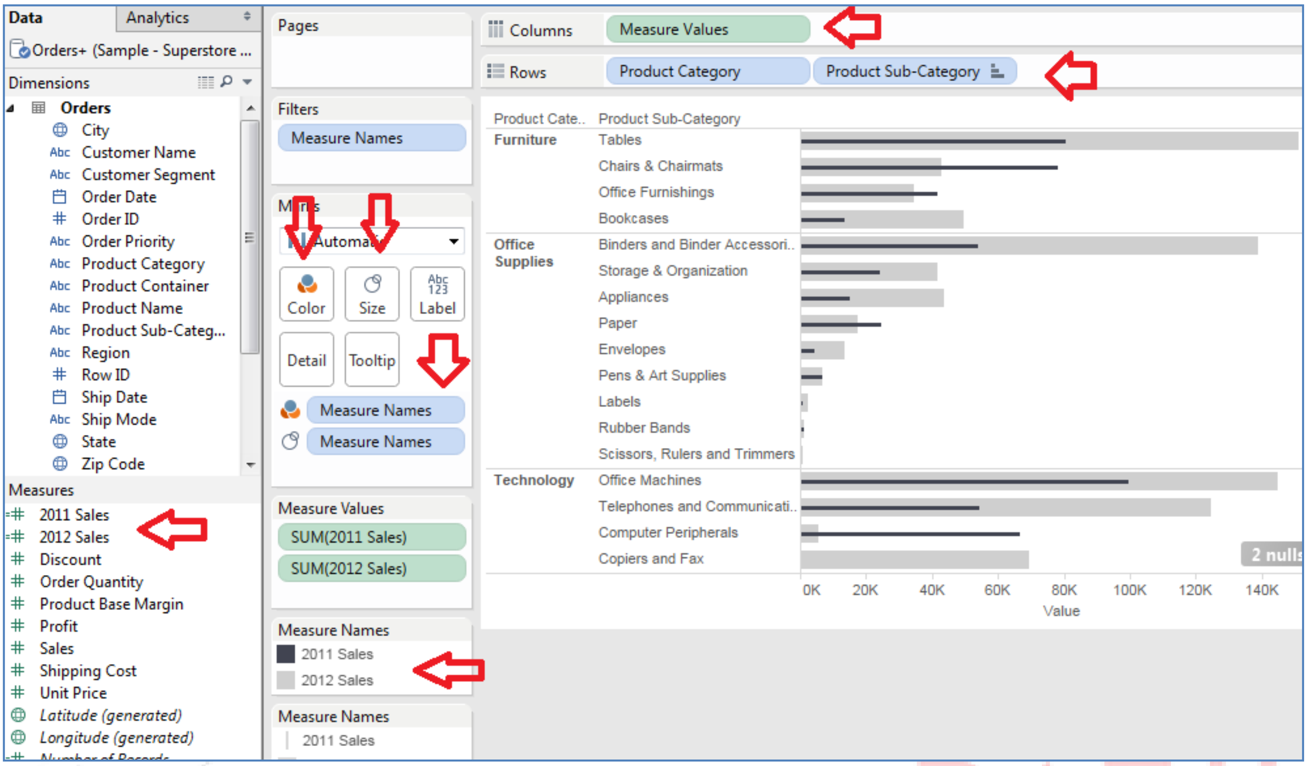
- Drag 2011 Sales to the axis, for 2011 Sales to create a Combined AxisView
- From the Row Shelf, drag Measure Names to Color on the Marks Card, which will stack the marks
- Select Measure Names to drag a copy to Size on the Mark Card
- Select Analysis from Menu, choose Stack Marks > Off to start both bars at the 0 position
- View Formatting:
    * Adjust size legend

* Sort the bars in descending order

* Edit the Color Legend to change colors for 2012 and 2011

* Format currency values on the axis to thousands

- Rename worksheet to Yearly Sales Comparison



Columns: Measure Values

Rows: Product Category, Product Sub-Category

Filters: Measure Names

Marks: Automatic; Color: Measure Names; Size: Measure Names

Measure Values: SUM(2011 Sales), SUM(2012 Sales)

Measure Names: ■ 2011 Sales, □ 2012 Sales

Figure: Annual Sales Comparison (Bar in Bar Visualization)

******************************Analytics Path***********************

Advanced Visualizations

Bullet Chart: Actual Sales versus Budgeted Sales

Generate a Bullet Chart visualization to compare Actual Sales v/s Budgeted Sales

Create a Reference Line for Avg(Budget Sales) for each Cell

Create a Reference Lines for 50, 75, 100% of Avg(Budget Sales) to analyze Actual Sales values in in the range of 50, 75, 100%

****

0. Create a Connection to Data Source - Coffee Chain.mdb and select Coffee Chain Query Table

1. Create a New Worksheet named Actual Sales v.s. Budget and create the data view

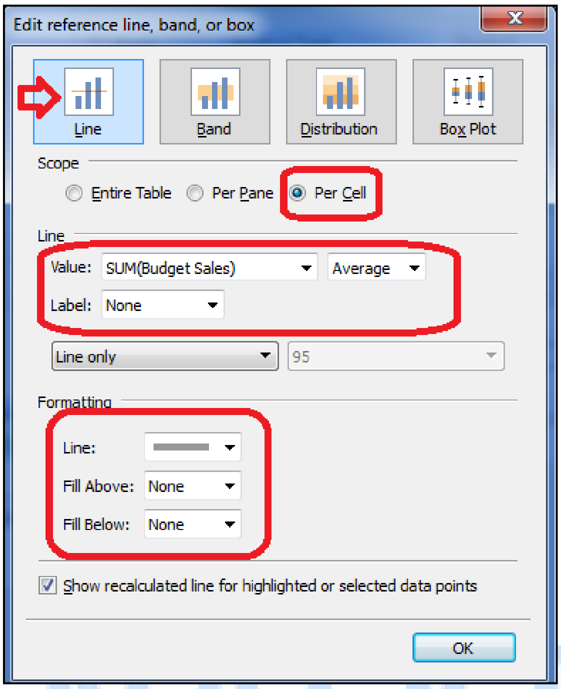
COLUMNS	: Sum(Sales)
ROWS	: Product Type / Product

- Sort bars in descending order by sales
- Drag and Drop Budget Sales to Detail on Marks Card

2. Create a Reference Line on the Sales axis for the Average Budget Sales per cell (Right-click sales axis and select add new reference line)

Type	: Line

Scope	: Per cell
Value	: Sum(Budget Sales) (Average)
Label	: None
Formatting	: Bold Line



Edit reference line, band, or box

Scope

○ Entire Table ○ Per Pane ● Per Cell

Line

Value: SUM(Budget Sales) Average

Label: None

Line only 95

Formatting

Line:

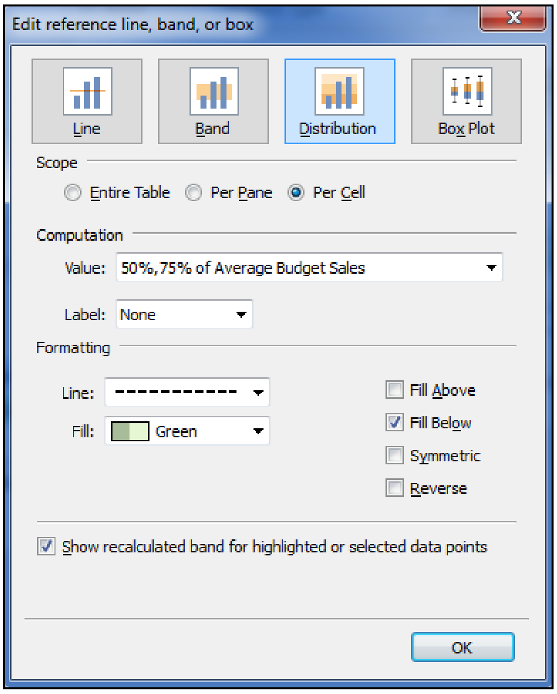
Fill Above: None

Fill Below: None

☑ Show recalculated line for highlighted or selected data points

3. Create another Reference Line on the Sales axis to analyze Actual Sales values in the range of 50, 75, 100% (Right-click sales axis and select add new reference line)

Type	: Distribution
Scope	: Per cell
Value	: 50, 75, 100% of Budget Sales (Avg)
Label	: None
Formatting	: Bold Line and Green Fill



Edit reference line, band, or box

Scope

○ Entire Table ○ Per Pane ● Per Cell

Computation

Value: 50%,75% of Average Budget Sales

Label: None

Formatting

Line: -----------

Fill: Green

□ Fill Above

☑ Fill Below

□ Symmetric

□ Reverse

☑ Show recalculated band for highlighted or selected data points

OK

4. Create a New Calculated Field to find out if Sales is below Budged

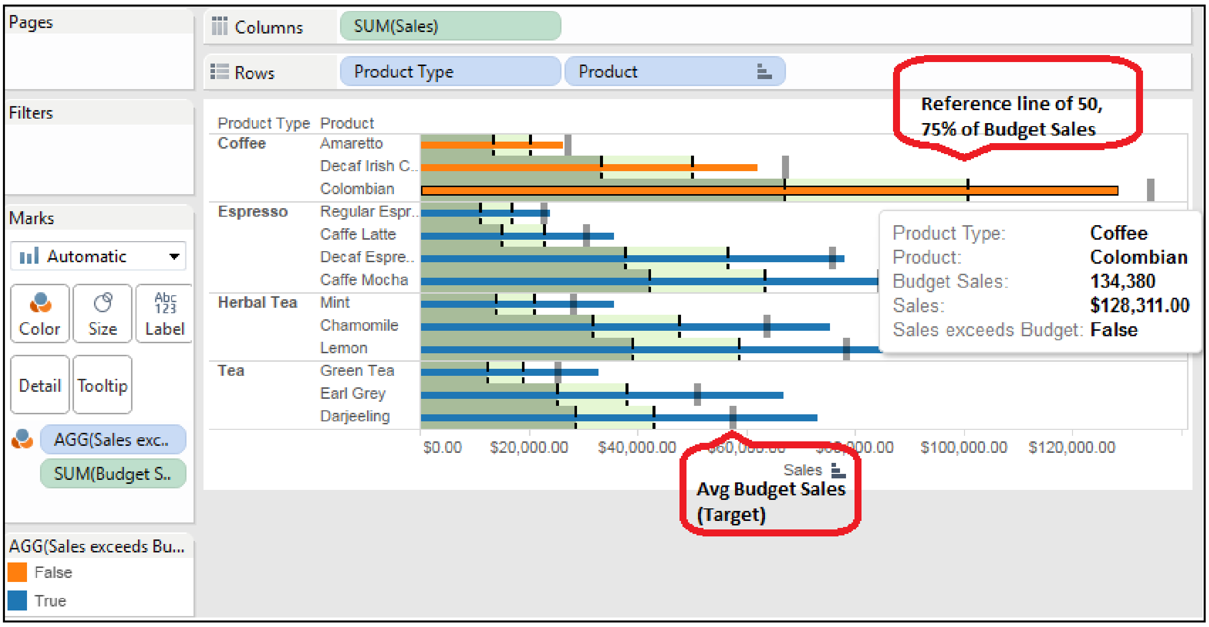
Name : Sales exceeds Budget

Expression : SUM([Sales]) >= SUM([Budget Sales])

5. Drag Sales exceeds Budget onto Marks card -> Color and adjust the colors: True(Blue), False(Orange)

6. Formatting Data View by

- Sizing of the bars down for clear visuals
- Change format to United States Currency on axis



Pages

Filters

Marks

Automatic

Columns: SUM(Sales)

Rows: Product Type, Product

AGG(Sales exc..

SUM(Budget S..

AGG(Sales exceeds Bu...

🟧 False

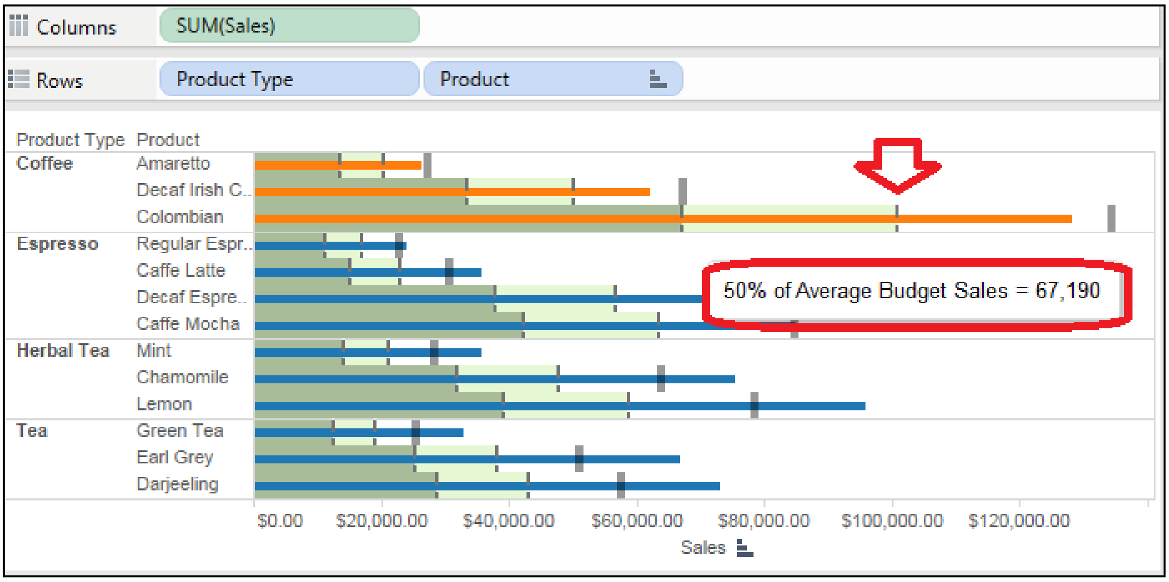
🟦 True

Reference line of 50, 75% of Budget Sales

Product Type: Coffee
Product: Colombian
Budget Sales: 134,380
Sales: $128,311.00
Sales exceeds Budget: False

Avg Budget Sales (Target)

Figure: Actual Sales v/s Budgeted Sales (Bullet Chart Visualization)



50% of Average Budget Sales = 67,190

##

STRING FUNCTIONS

##

Contains(string, substring)

Returns true if the given string contains the specified substring.

Example:

CONTAINS(“Analytics Path Technologies”, “echno”) = true

CONTAINS(“Analytics Path Technologies”, “Z”) = true

CONTAINS(“Analytics Path Technologies”, “Z”) = false

****

Conditionally creating a new field Central Region by using CONTAINS function

NewField:	Expression

Central Region	if(CONTAINS([Region],"Central"))
	then "Central"
	Else "Other Regions"
	END

Selecting Multiple Values using OR operator

Note: IN operator not applicablein TABLEAU Unlike oter softwares

```
CustomerType                    if [Customer Segment] = 'Home Office' OR
                                [Customer Segment] = 'Consumer'
                                THEN 'Personal Customer'
                                ELSE 'Corporate Clients'
                                END
```

ENDSWITH(string, substring)

Returns true if the given string ends with the specified substring. Trailing white spaces are ignored.

Example:

```
ENDSWITH("Analytics Path Technologies","s")      =     True
ENDSWITH("Analytics Path Technologies","S")      =     False (Case Sensitive)
ENDSWITH("Analytics Path Technologies","ies")    =     True
```

****

Create Sales Commission 10% for State dimension values ending with letter "a"
else 20% as Sales Commission

New Field   Expression

****

```
Sales Commission                IF (ENDSWITH([State],"a"))
                                THEN [Sales]*.10
                                ELSE [Sales]*.20
                                END
```

****

FIND(string, substring, [start])

Returns the index position of substring in string, or 0 if the substring isn't found. If the optional argument start is added, the function ignores any instances of substring that appear before the index position start. The first character in the string is position 1.

Examples:

```
FIND("Analytics Path Technologies", "l") = 3
FIND("Analytics Path Technologies", "l") = 0
FIND("Analytics Path Technologies", "ll") = 3
FIND("Analytics Path Technologies",LOWER("L"),5)
FIND("Analytics Path Technologies","l",3)
```

```
New Field                    Expression
*******************************************************************************************
Region Separation            IF  FIND([NewRegion],"Central") > 0
                             THEN "Central Region"
                             ELSEIF  FIND([NewRegion],"East") > 0
                             THEN "East Region"
                             ELSEIF  FIND([NewRegion],"West") > 0
                             THEN "West Region"
                             ELSE "South Region"
                             END

                             -------------------------------------
                             if FIND([Region],"Central") > 0
                             THEN "Central Region"
                             ELSE "Other Region"
                             END
*******************************************************************************************
```

Concatenate 2 or more fields values in Tableau using + Operator

----

```
New Field                    Expression
*******************************************************************************************
NewRegion                          [Region]+ '  ' + [State]
FullName                           [FirstName] + ", " + [LastName]
                                   [FirstName] + "  " + [LastName]
*******************************************************************************************
```

FINDNTH(string, substring, occurrence)

Returns the position of the nth occurrence of substring within the specified string, where n is defined by the occurrence argument.

Example

FINDNTH("Analytics Path Technologies", "l", 2)	=	4 (2nd occurence of letter "l")
FINDNTH("Analytics Path Technologies", "l", 3)	=	13
FINDNTH("Analytics Path Technologies", "l", 5)	=	0 (no occurence returns 0 as nth occurence)

LEFT(string, number)

Returns the left-most number of characters in the string.

RIGHT(string, number)

Returns the right-most number of characters in string.

Example

LEFT("Analytics Path Technologies", 5) = Analytics Path

["Analytics Path"] + ", " + Left("Technologies",1) = Analytics Path Technologies

RIGHT("Analytics Path Technologies",5) = ogies

NewVariable	Expression
**********************************	**********************************
FullName1	[FirstName] + ", " + Left([LastName],1)
FullName1	[FirstName] + ", " + Left([LastName],1)

****

LEN(string)

Returns the length of the string.

Example

LEN("Analytics Path Technologies") = 18

LOWER(string)

Returns string, with all characters lowercase.

Example

LOWER("ANALYTICS PATH TECHNOLOGIES") = "Analytics Path technologies"

LTRIM(string)

Returns the string with any leading spaces removed.

RTRIM(string)

Returns string with any trailing spaces removed.

Example

LTRIM(" ANALYTICS PATH ") = "ANALYTICS PATH "

RTRIM(" ANALYTICS PATH ") = " ANALYTICS PATH"

MAX(a, b)

MIN(a, b)

Returns the maximum of a and b (which must be of the same type). This function is usually used to compare numbers, but also works on strings. With strings, MAX finds the value that is highest in the sort sequence defined by the database for that column. It returns Null if either argument is Null.

Example

MAX ("Analytics Path","Technologies") = "Technologies"

MIN ("Analytics Path","Technologies") = "Technologies"

MID(string, start, [length])

Returns the string starting at index position start. The first character in the string is position 1. If the optional argument length is added, the returned string includes only that number of characters.

Examples

MID("Analytics Path Technologies", 3) = "lly Technologies"

MID("Calculation", 3, 3) = "lly"

REPLACE(string, substring, replacement)

Searches string for substring and replaces it with replacement. If substring is not found, the string is not changed.

Example

REPLACE("Analytics Path Technologies", "Analytics Path", "Wipro") = "Wipro Technologies"

NewVariable	Expression
**	
NewRegion	[Region]+ ' ' + [State]
Replace Central	REPLACE([NewRegion],"Central", "CTRL")
**	

SPACE(number)

Returns a string that is composed of the specified number of repeated spaces.

Example

SPACE(1) = " " - Returns TRUE if a single blank space exists
ROW - NewRegion

SPACE(2) = " " - Returns FALSE if no duble blank space exists
ROW - NewRegion

NewVariable Expression
****

SpacesExistorNot Space(1) = " "

SPLIT(string, delimiter, token number)
Returns a substring from a string, using a delimiter character to divide the string into a sequence of tokens.

The string is interpreted as an alternating sequence of delimiters and tokens. So for the string abc-defgh-i-jkl, where the delimiter character is '-', the tokens are abc, defgh, i, and jlk. Think of these as tokens 1 through 4. SPLIT returns the token corresponding to the token number. When the token number is positive, tokens are counted starting from the left end of the string; when the token number is negative, tokens are counted starting from the right.

Examples
SPLIT("Central-Illinois-Bolingbrook-60440", '-', 3) = Bolingbrook
SPLIT("Central-Illinois-Bolingbrook-60440", '-', -3) = Illinois
SPLIT("Central-Illinois-Danville-61832", 'll', 3) = e-61832
SPLIT("Central-Illinois-Bolingbrook-60440", '0', 1) = Central-Illinois-Bolingbrook-6

SPLIT ('a-b-c-d', '-', 2) = 'b'
SPLIT ('a|b|c|d', '|', -2) = 'c'

NewVariable Expression
****

NewRegion_ [Region]+ '-' + [State] + '-' + [City] + '-' + [Zip Code]

```
if contains([Product Sub-Category],"o") = TRUE Then "Exists"
else "Not Exists"
END

if [Customer Segment] = 'Home Office' OR
  [Customer Segment] = 'Consumer'
THEN 'Personal Customer'
ELSE 'Corporate Clients'
END
```

##

DATE FUNCTIONS

##

date_part values
year / quarter / month / dayofyear / day / weekday / week / hour / minute / second

Replace with the relavant date_part values and analyze the output result:

DATENAME(date_part, date, [start_of_week])
DATENAME('month',[Order Date]) = MonthName
DATENAME('day',[Order Date]) = DayName
DATENAME('quarter',[Order Date]) = QtrName
DATENAME('dayofyear',[Order Date]) = DayName
DATENAME('year',[Order Date]) = YEarName

Replace with the relavant date_part values and analyze the output result:

DATEADD(date_part, increment, date)
DATEADD('year',3,[Order Date])
DATEADD('month',3,[Order Date])
DATEADD('day',3,[Order Date])

if contains([Customer Segment],"Consumer") = TRUE
Then "Customer"
else "Corporate"
END

DATEDIFF(date_part, date1, date2, [start_of_week])
DATEDIFF('year',[Order Date],[Ship Date])
DATEDIFF('month',[Order Date],[Ship Date])
DATEDIFF('day',[Order Date],[Ship Date])
DATEDIFF('year',[Odate1],TODAY())

DATEPART(date_part, date, [start_of_week])
DATEPART('month',[Order Date]) = MonthName
DATEPART('day',[Order Date]) = DayName
DATEPART('quarter',[Order Date]) = QtrName
DATEPART('dayofyear',[Order Date]) = DayName
DATEPART('year',[Order Date]) = YearName

Day([Order Date])

MAKEDATE(2004, 4, 15)
MAKEDATETIME([Odate1],#02:02:20#)
MAKETIME(14, 52, 40)
TODAY( )
YEAR(#2004-15#)
NOW( )

###

NUMBER FUNCTIONS

###

ABS(number)
Returns the absolute value of the given number

Example:
Abs(-3200.24) = 3200.24

CEILING(number)
Rounds a number to the nearest integer of equal or greater value.
Example:

Ceiling(3.99) = 4
Ceiling(3.01) = 4

FLOOR(number)
Rounds a number to the nearest integer of equal or lesser value.

Example:

Floor(3.99) = 3
Floor(3.01) = 3

DIV(integer1, integer2)
Returns the integer part of a division operation, in which integer1 is divided by integer2.

Example
Div(21,2) = 1

MAX(number, number)

Returns the maximum of the two arguments, which must be of the same type. Returns Null if either argument is Null. MAX can also be applied to a single field in an aggregate calculation.

Example:

Max(7,2)	= 2
Max(Sum(Sales),Sum(Profit))	= whichever value is greater
Max([First Name],[Last Name])	= whichever value is maximum

POWER(number, power)

Raises the number to the specified power.

Example:

Power(5,2)	= 25
5^2	= 25

ROUND(number, [decimals])

Rounds numbers to a specified number of digits. The decimals argument specifies how many decimal points of precision to include in the final result. If decimals is omitted, number is rounded to the nearest integer.

Round(3.27,1)	= 3.30
Round(3.24,1)	= 3.20
Round(3.25,1)	= 3.30
ROUND(3.01,.10)	= 3.00
ROUND(3.99,1)	= 4.00
Round([Sales])	

ZN(expression)

Returns the expression if it is not null, otherwise returns zero. Use this function to use zero values instead of null values.

Example

ZN([Sales])	= [Sales]

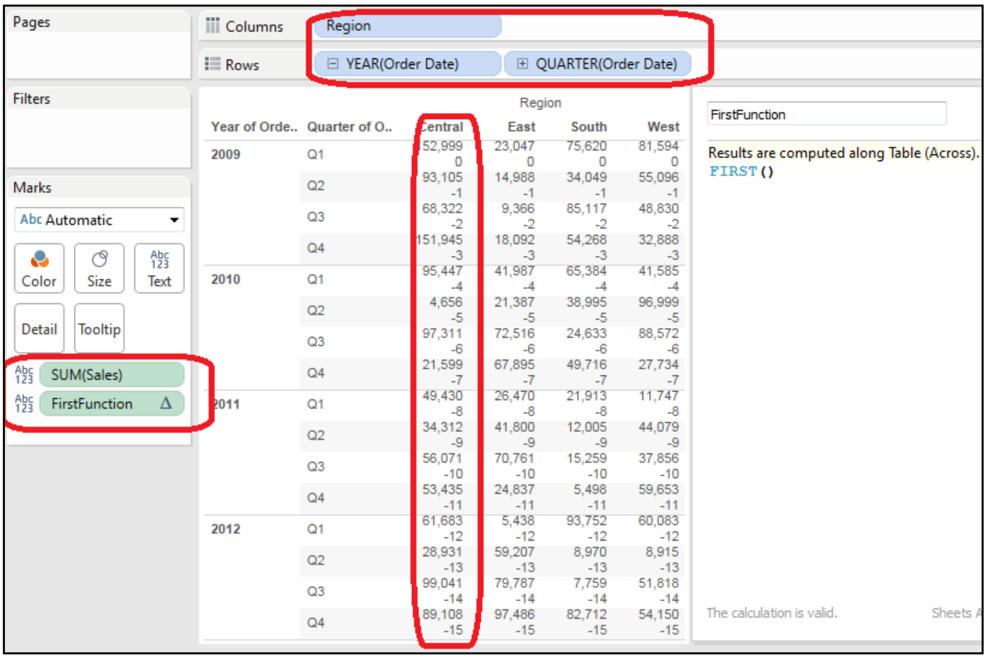
##

Table Calculation FUNCTIONS

##

1. Create a Data View

    COLUMNS - Region
    ROWS - Year(Order Date) / Quarter (Order Date)
    New Function Variable Names ----> Text Label



Columns: Region

Rows: YEAR(Order Date) | QUARTER(Order Date)

Year of Orde..	Quarter of O..	Central	East	South	West
2009	Q1	152,999 0	23,047 0	75,620 0	81,594 0
	Q2	93,105 -1	14,988 -1	34,049 -1	55,096 -1
	Q3	68,322 -2	9,366 -2	85,117 -2	48,830 -2
	Q4	151,945 -3	18,092 -3	54,268 -3	32,888 -3
2010	Q1	95,447 -4	41,987 -4	65,384 -4	41,585 -4
	Q2	4,656 -5	21,387 -5	38,995 -5	96,999 -5
	Q3	97,311 -6	72,516 -6	24,633 -6	88,572 -6
	Q4	21,599 -7	67,895 -7	49,716 -7	27,734 -7
2011	Q1	49,430 -8	26,470 -8	21,913 -8	11,747 -8
	Q2	34,312 -9	41,800 -9	12,005 -9	44,079 -9
	Q3	56,071 -10	70,761 -10	15,259 -10	37,856 -10
	Q4	53,435 -11	24,837 -11	5,498 -11	59,653 -11
2012	Q1	61,683 -12	5,438 -12	93,752 -12	60,083 -12
	Q2	28,931 -13	59,207 -13	8,970 -13	8,915 -13
	Q3	99,041 -14	79,787 -14	7,759 -14	51,818 -14
	Q4	89,108 -15	97,486 -15	82,712 -15	54,150 -15

FirstFunction

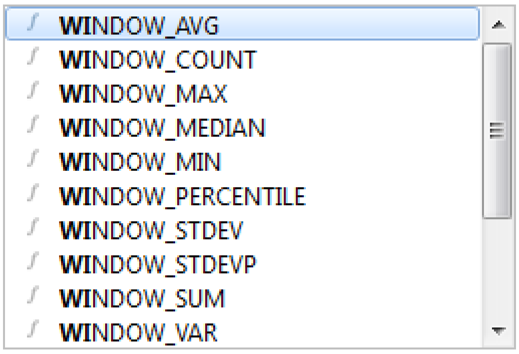
Results are computed along Table (Across).

```
FIRST()
```

The calculation is valid.

Sheets A

NewVariable	**Expression**
********************************	**
FirstFunc	First()
LastFunc	Last()
IndexFunc	Index()
Running_Sum	RUNNING_SUM(SUM([Sales]))
Running_Avg	RUNNING_AVG(SUM([Sales]))
Running_Count	RUNNING_COUNT(SUM([Sales]))
Running_Max	RUNNING_Max(SUM([Sales]))
Running_Min	RUNNING_Min(SUM([Sales]))
Total_Func	TOTAL(SUM([Sales]))
Size_Func	Size()
Window_Func	WINDOW_MAX(SUM([Sales]))
	Try:

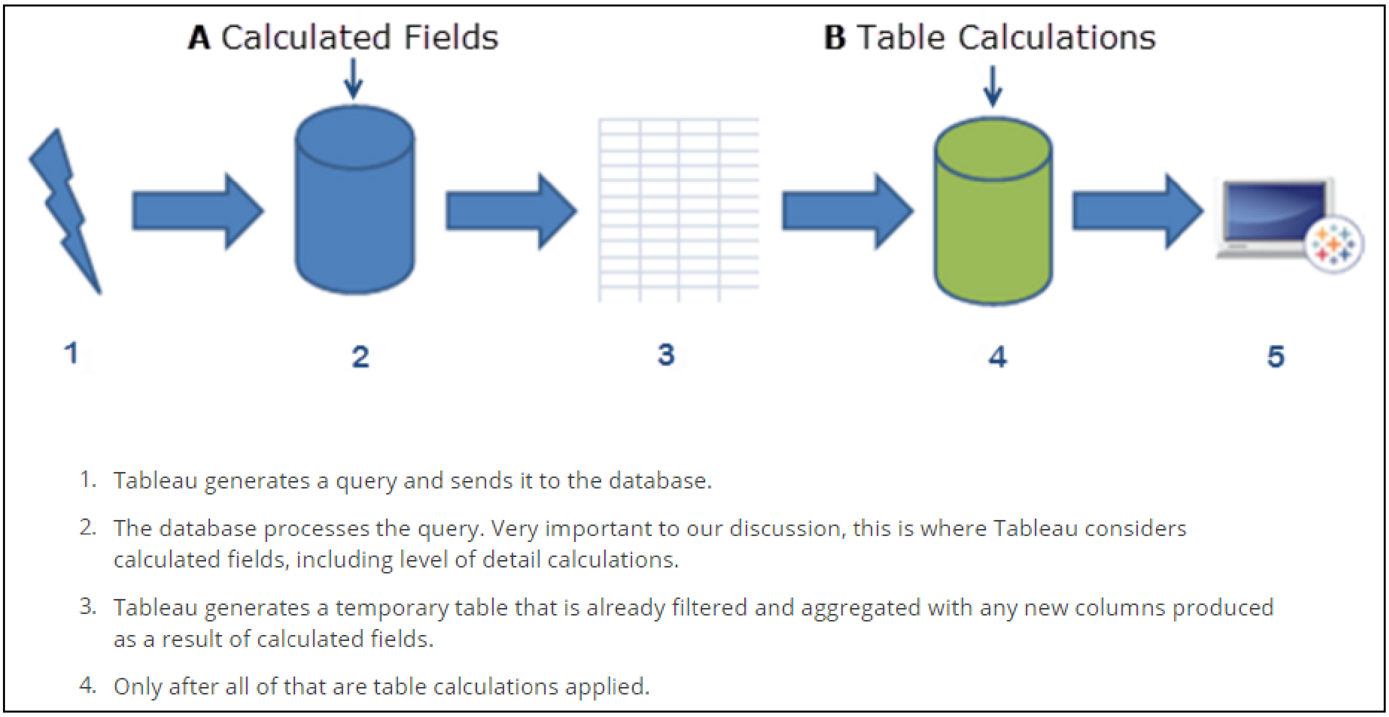

WINDOW_AVG
WINDOW_COUNT
WINDOW_MAX
WINDOW_MEDIAN
WINDOW_MIN
WINDOW_PERCENTILE
WINDOW_STDEV
WINDOW_STDEVP
WINDOW_SUM
WINDOW_VAR

*******************************Analytics Path*************************

TABLE CALCULATIONS from Workarea Window

****

***** Understanding where Calculations occur for ***Calculated Fields and Table Calculations***
we have to understand how Tableau processes calculated fields, filters and table calculations. This is the order of actions that Tableau takes every time the visualization is rendered:



1. Tableau generates a query and sends it to the database.
2. The database processes the query. Very important to our discussion, this is where Tableau considers calculated fields, including level of detail calculations.
3. Tableau generates a temporary table that is already filtered and aggregated with any new columns produced as a result of calculated fields.
4. Only after all of that are table calculations applied.