Tableau is one of the fastest evolving Business Intelligence (BI) and data visualization tool. It is very fast to deploy, easy to learn and very intuitive to use for a customer.

Tableau offers five main products catering to diverse visualization needs for professionals and organizations. They are:

* **Tableau Desktop:** Made for individual use
* **Tableau Server:** Collaboration for any organization
* **Tableau Online:** Business Intelligence in the Cloud
* **Tableau Reader:** Let you read files saved in Tableau Desktop.
* **Tableau Public:** For journalists or anyone to publish interactive data online.

### Quick Features

1. Tableau Public and Tableau Reader are free to use, while both Tableau Server and Tableau Desktop come with a 14 days fully functional free trial period, after which the user must pay for the software.
2. Tableau Desktop comes in both a Professional and a lower cost Personal edition. Tableau Online is available with an annual subscription for a single user, and scales to support thousands of users. Tableau have gone through different versions, here we will discuss the learning curve of Tableau Desktop 9.0.

Step-3: Connecting With Data

1. Tableau can connect with various data sources such as text, excel file, databases to big data queries also. In this section, we will look at the basics and advance feature of data connectivity with different sources. Here we will also look at Join types, Data Blending, connection with cubes, custom sql and Google Analytics.

Step-4: Creating Views and Analysis

1. Tableau has multiple options to represent data in different views, applying filters /drill downs /formatting, creating sets, groups, generating trend lines and performing forecasting. Start exploring !
2. You have now looked at various objects to visualize data. One big dilemma that you will face while creating data visualization is which object should you choose to represent data.
3. The snapshot below will help you to choose and decide the type of visualization. However, the feature of automatic selection of views available in tableau takes care of this issue largely. This feature automatically activates best views for selected dimension(s) and measure(s). Hence, you need no worry. You are in safe hands !

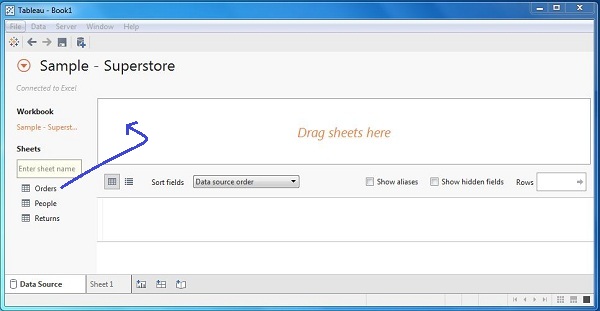
Tableau Features

Tableau provides solutions for all kinds of industries, departments and data environments. Below are the unique features which enable tableau handle so many diverse scenarios.

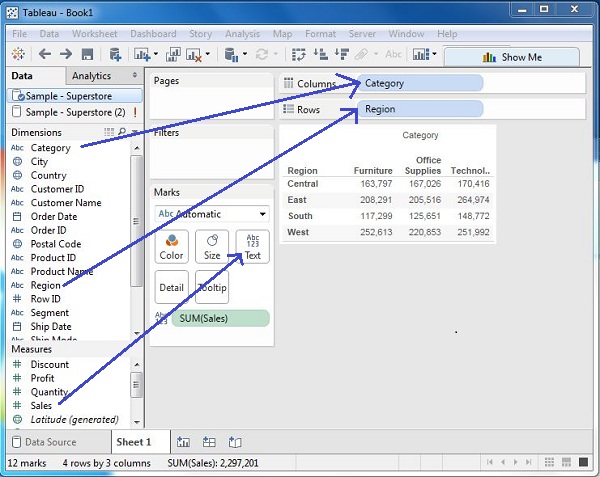
* **Speed of Analysis** - As it does not need high level of programming expertise, any computer user with access to data can start using it to derive value from the data.
* **Self-Reliant** - Tableau does not need a complex software setup. The desktop version which is used by most users is easily installed and contains all the features needed to start and complete data analysis.
* **Visual Discovery** - The user explores and analyses the data by using visual tools like colours, trend lines, charts and graphs. There is very little script to be written as nearly everything is done by drag and drop.
* **Blend Diverse Data Sets** - Tableau allows you to blend different relational, semi-structured and raw data sources in real time, without expensive up-front integration costs. The users don’t need to know the details of how data is stored.
* **Architecture Agnostic** - Tableau works in all kinds of devices where data flows. So the user need not worry about specific hardware or software requirements to use Tableau.
* **Real Time Collaboration** - Tableau can filter, sort, and discuss data on the fly and embed a live dashboard in portals like SharePoint site or Salesforce. You can save your view of data and allow colleagues to subscribe to your interactive dashboards so they see the very latest data just by refreshing their web browser.
* **Centralized Data** - The tableau server provides a centralized location to manage all of the organization’s published data sources. You can delete, change permissions, add tags, and manage schedules in one convenient location. It’s easy to schedule extract refreshes and manage them in the data server. Administrators can centrally define a schedule for extracts on the server for both incremental and full refreshes.

Connect to a Data Source

One opening Tableau we get the start page Showing various data sources. Under the header Connect, we have options to choose a file or server or saved data source. Under Files we choose excel. Then navigate to the file “Sample – Superstore.xls” as mentioned above. The excel file has three sheets named orders, people and Returns. We choose Orders.



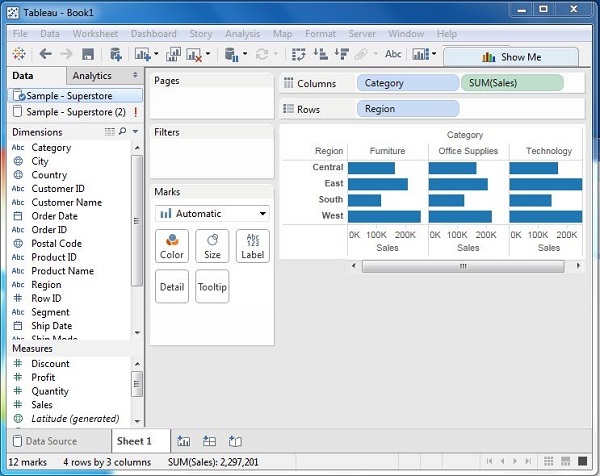
Choose the Dimensions and Measures

Next we choose the data to be analyzed by deciding on the dimensions and measures. Dimensions are the descriptive data while measures are numeric data. When put together, they help us visualize the performance of the dimensional data with respect to the data which are measures. We choose category and region as the dimensions and sales as the measure. Drag and drop them as shown below. The result shows the total sales in each category for each region. 

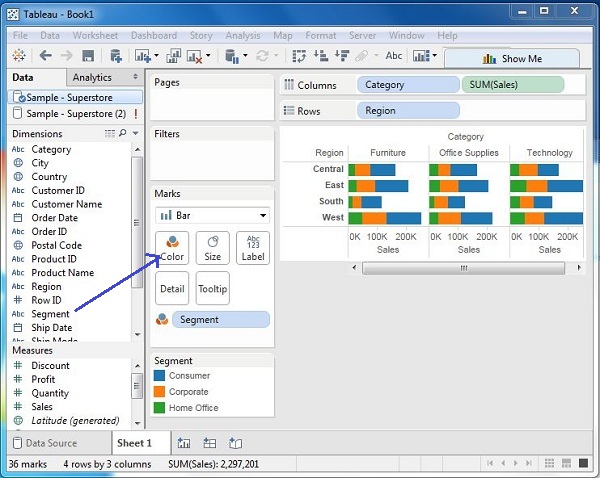
Apply Visualization Technique

In the previous step we see that the data is available only as numbers. We have to read and calculate each of the values to judge the performance. But we can see them as graphs or charts with different colours to get a quicker judgment.

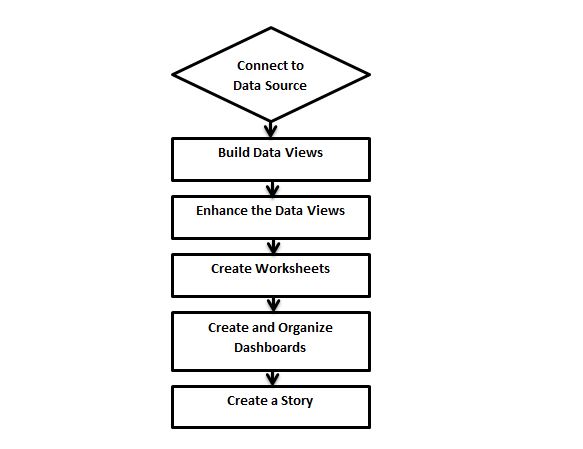
We drag and drop the sum(sales) column from the Marks tab to the Columns shelf. The table showing the numeric values of sales now turns into a bar chart automatically.



We can apply a further technique of adding another dimension to the existing data and that will add more colours to the existing bar chart as shown below.



**Design Flow**



## Connect to Data Source

Tableau connects to all popular data sources. It has inbuilt connectors which take care of establishing the connection once the connection parameters are supplied. be it**Simple text files, Relational sources, No Sql sources or Cloud data bases**, tableau connects to nearly every data source.

## Build Data Views

After connecting to a data source, you get all the column and data available in the Tableau environment. You classify them as dimensions, measures and create any hierarchy required. Using these you build views which are traditionally known as Reports. Tableau provides easy drag and drop feature to build views.

## Enhance the Views

The views created above needs to be enhanced further by use of filters, aggregations, Labeling of Axes, Formatting of colors and borders etc.

## Create Worksheets

We create different worksheets to create different views on the same data or different data.

## Create and Organize Dashboards

Dashboards contain multiple worksheets which are linked it. So the action in any of the worksheet can change the result in the dashboard accordingly.

## Create a Story

A story is a sheet that contains a sequence of worksheets or dashboards that work together to convey information. You can create stories to show how facts are connected, provide context, demonstrate how decisions relate to outcomes, or simply make a compelling case

|  |  |  |
| --- | --- | --- |
| **File Type** | **File Extension** | **Purpose** |
| **Tableau Workbook** | .twb | It contains information on each sheet and dashboard that is present in a workbook. It has the details of the fields which are used in each view and the formula applied to the aggregation of the measures. It also has the formatting and styles applied. It also contains the data source connection information and any metadata information created for that connection. |
| **Tableau Packaged Workbook** | .twbx | This file format contains the details of workbook as well as the local data that is used in the analysis. Its purpose is to be share with other Tableau desktop or Tableau reader users assuming it does not need data from the server. |
| **Tableau Data source** | .tds | The details of the connection used to create the tableau report are stored in this file. In the connection details it stores the source type(excel/relational/sap etc.) as well as the data types of the columns. |
| **Tableau Packaged Data source** | .tdsx | This file is similar to the .tds file with the addition of data along with the connection details. |
| **Tableau Data Extract** | .tde | This file contains the data used in a .twb file in a highly compressed columnar data format. This helps in storage optimization. It also saves the aggregated calculations that are applied in the analysis. This file should be refreshed to get the updated data form the source. |
| **Tableau Bookmark** | .tbm | These files contain a single worksheet that is shared easily to be pasted into other workbooks. |
| **Tableau Preferences** | .tps | This file stores the colour preference used across all the workbooks. It is mainly used for consistent look and feel across the users. |

Any data analysis involves a lot of calculations. In Tableau the calculation editor is used to apply calculations to the fields being analyzed. Tableau has a number of inbuilt functions which help in creating expressions for complex calculations.

The description of different categories of functions are given below.

* Number Functions
* String Functions
* Date Functions
* Logical Functions
* Aggregate Functions

## Number Functions

These are the functions used for numeric calculations. They only take numbers as inputs.Below are some examples of important number functions.

|  |  |  |
| --- | --- | --- |
| **Function** | **Description** | **Example** |
| **CEILING(number)** | Rounds a number to the nearest integer of equal or greater value. | CEILING(2.145) = 3 |
| **POWER(number, power)** | Raises the number to the specified power. | POWER(5,3) = 125 |
| **ROUND(number, [decimals])** | Rounds numbers to a specified number of digits. | ROUND(3.14152,2) = 3.14 |

## String Functions

String Functions are used for string manipulation. Below are some important string functions with examples.

|  |  |  |
| --- | --- | --- |
| **Function** | **Description** | **Example** |
| **LEN(string)** | Returns the length of the string. | LEN("Tableau") = 7 |
| **LTRIM(string)** | Returns the string with any leading spaces removed. | LTRIM(" Tableau ") = "Tableau" |
| **REPLACE(string, substring, replacement)** | Searches string for substring and replaces it with replacement. If substring is not found, the string is not changed. | REPLACE("GreenBlueGreen", "Blue", "Red") = "GreenRedGreen" |
| **UPPER(string)** | Returns string, with all characters uppercase. | UPPER("Tableau") = "TABLEAU" |

## Date Functions

Tableau has a variety of date functions to carry out calculations involving dates. All the date functions use the date\_part which is a string indicating the part of the date like - month, day or year. Below are the examples of some of the important date functions.

|  |  |  |
| --- | --- | --- |
| **Function** | **Description** | **Example** |
| **DATEADD(date\_part, increment, date)** | Returns an increment added to date. The type of increment is specified in date\_part. | DATEADD('month', 3, #2004-04-15#) = 2004-07-15 12:00:00 AM |
| **DATENAME(date\_part, date, [start\_of\_week])** | Returns date\_part of date as a string. The start\_of\_week parameter is optional. | DATENAME('month', #2004-04-15#) = "April" |
| **DAY(date)** | Returns the day of the given date as an integer. | DAY(#2004-04-12#) = 12 |
| **NOW( )** | Returns the current date and time. | NOW( ) = 2004-04-15 1:08:21 PM |

## Logical Functions

These functions evaluate some single value or result of an expression and give a boolean output.

|  |  |  |
| --- | --- | --- |
| **Function** | **Description** | **Example** |
| **IFNULL(expression1, expression2)** | The IFNULL function returns the first expression if the result is not null, and returns the second expression if it is null. | IFNULL([Sales], 0) = [Sales] |
| **ISDATE(string)** | The ISDATE function returns TRUE if the string argument can be converted to a date and FALSEif it cannot. | ISDATE("11/05/98") = TRUE ISDATE("14/05/98") = FALSE |
| **MIN(expression)** | The MIN function returns the minimum of an expression across all records or the minimum of two expressions for each record. |  |
|  |  |  |

## Aggregate Functions

|  |  |  |
| --- | --- | --- |
| **Function** | **Description** | **Example** |
| **AVG(expression)** | Returns the average of all the values in the expression. AVG can be used with numeric fields only. Null values are ignored. |  |
| **COUNT(expression)** | Returns the number of items in a group. Null values are not counted. |  |
| **MEDIAN(expression)** | Returns the median of an expression across all records. Median can only be used with numeric fields. Null values are ignored. |  |
| **STDEV(expression)** | Returns the statistical standard deviation of all values in the given expression based on a sample of the population. |  |