Power BI:

What is power bi?

Power BI is a business intelligence tool that we can use for preparing the data, building relationships between the data tables, and creating visualization and dashboards.
OR
This tool we can use for loading, modeling, and visualizing data.

Different connectivity modes in Power BI:

We have three different kinds of connectivity modes:

1] Import Mode:

Import mode is an easy, fast, and frequently used mode to get the data in Power BI. In import mode, we connect with different data sources and try to get the data inside Power BI. We use import mode when the data volume is low.

2] Direct Query Mode:

Direct Query Mode is another method of importing data with a **query** to retrieve data from a different data source. When the data volume is too large, we use DirectQuery to avoid refreshing data as it can take a long time.

3] Composite Mode:

In composite mode, we use both import mode and direct query mode to get the data from different sources.

Type of source data in Power BI:

Flat files & Folders:- (CSV, txt, Xls, etc) •
Databases:- (SQL, Access, Oracle, IBM, Azure, etc) •
Online Services:- (Sharepoint, GitHub, Dynamics 365, Google Analytics, Salesforce, Power BI Service, etc) •
Others:- (Web feeds, R scripts, Spark, Hadoop, etc).

Where is the data stored in Power BI?

Primarily, PowerBI uses two repositories to store its data: Azure Blob Storage and Azure SQL Database. Azure Blob Storage typically stores the data that the users upload. Azure SQL Database stores all the metadata and artifacts for the system itself.

What is the maximum data limit per client for the free version of Power BI?

With a Power BI Free license, a user can use 10 GB of storage in the cloud for hosting Power BI reports. The maximum size a Power BI report can be used in the cloud is 1GB.

The building block of Power BI:

Visualizations - A visual representation of data like a chart or map.

Data Sets - A collection of data that Power BI uses to create visualizations.

Reports - A collection of visualizations that appear together on one or more pages.

Dashboard - A Power BI dashboard collects visuals from a single page that you can share with others. Often, it's a selected group of visuals that provide quick insight into the data or story that you're trying to present.

Tiles - It's a single visualization on a report or a dashboard. It's a rectangular box that holds an individual visual.

Components of PowerBI:

Power BI has Three main components:

1] Power Query

Power Query is a kind of ETL(Extract Transform, Load) tool that we use to get the data from different sources and after that transform and clean the data and then load that data into Power Pivot. It can be accessed in Power BI Desktop by going to the **Transform data** option in the **Home** Menu.

2] Power Pivot

Power Pivot is mostly used for data modeling and data analysis.
In Power pivot, we build the relationships between the tables and create new columns and measures using DAX(Data Analysis Expression).

3] Power View:

The power view component is used to provide data insights using reports and dashboards and to provide interactive charts.

4] Other Power BI components:

1] Power Map
2] Power BI Q&A: Natural language-based question and answering engine.
3] Power BI Desktop

What is DAX:

Data Analysis Expression is a language used to perform multiple calculations and create new columns and measures based on the data.

What is M-code or M-language:

The M stands for data Mash-up. The M language is the data transformation language of Power Query. Anything that happens in the query is ultimately written in M.
The M Language is a case-sensitive and functional language that consists of 'let' and 'in' block statements that contain variables,

Tables:

We have two kinds of tables

1] Fact Table:

Fact Table holds the main data or business data like Sales data, Revenue, Profit, and Order data. This is the main table in data modeling.

2] Dimension or Lookup Table:

These are lookup tables that we use to get reference data. Ex- Customer,, Product table.
Note: Dimension Table mostly provides descriptive data and contains Categorical data like the product category and subcategory.

Example of Tables:

Suppose there is Sales data that holds all data about the sales and we also have customer_id and product_id columns in this data.
We have another two tables Customer and Product Table.
Now In Sales data, we use these Customer and product tables to get the details about customers and products.

So, here Sales data is a Fact Table because it has the main data, and Customer, Product tables are dimension tables that we are using to get data about customers.

Star & Snowflake Schema:

Star Schema:

In a star schema, every dimension table will be connected to the fact table.

Snowflake Schema:

In the snowflake schema, Fact Table is connected to the dimension table and the dimension table can also be connected to the other few dimension tables. So here dimension tables can be connected to the Fact table indirectly.

Power Query:

Transformers:

We can apply the following transformers:

1] Basic Table transformations:

1] Choose or remove columns, Keep or remove rows, Sort values, Change data type, Promote header row

2] Number-based transformations

Statistics functions allow you to evaluate basic stats for the selected column (sum, min/max, average, count, countdistinct, etc)
Standard, Scientific and Trigonometry tools allow you to apply standard operations (addition, multiplication, division, etc.)

3] Text-based transformations

Split a text column, Extract characters from a text column, Format a text column

3] Date-based transformations

Age: Difference between the current time and the date in each row

Year/Month/Quarter/Week/Day: Extracts individual components from a date field (Time-specific options include Hour, Minute, Second, etc.) •
Earliest/Latest: Evaluates the earliest or latest date from a column as a single value (can only be accessed from the "Transform" menu) .

Advanced & Important Transformers:
Conditional Columns allow you to define new fields based on logical rules and conditions (IF/THEN statements)

Group By allows you to aggregate your data at a different level (i.e. transform daily data into monthly, roll up transaction-level data by store, etc)

"Pivoting" is a fancy way to describe the process of turning distinct row values into columns ("pivoting") or turning columns into rows ("unpivoting").

Appending queries allows you to combine (or stack) tables that share the exact same column structure and data types.

Merging: This is the way to join two or multiple tables.
Power Pivot:

In Power Pivot We have three kinds of Views.

Report View: It is the default view that shows the visualization of the data in reports. You can create multiple reports and visualizations here.

Data View: Data view shows the transformed data in a table format with columns and rows. It also allows you to create new calculated columns for further insights.

Model View: Also called, Relationship View, helps to create relationships between data models. All the models created in the data can be seen in this view and accordingly, you can compare or create diagrams based on subsets of the model.

1] Model View

Also called, Relationship View, helps to create relationships between data models. All the models created in the data can be seen in this view and accordingly, you can compare or create diagrams based on subsets of the model.

What is Data Model & Data Modeling:

Data Modeling is the process of making relationships between tables or connecting multiple tables with a common column.

The Data Model is data we get after joining multiple tables using a common column. Or The data model is the result of the data modeling process.

Normalization: is the process of organizing the tables and columns in a relational database to reduce redundancy and preserve data integrity.

It's commonly used to:

1] Eliminate redundant data to decrease table sizes and improve processing speed & efficiency.
2] Minimize errors and anomalies from data modifications (inserting, updating or deleting records) .
3]Simplify queries and structure the database for meaningful analysis

Primary Key & Foreign Key:

A primary key is used to identify the records uniquely.
A foreign key constraint is used to prevent actions that would destroy links between two tables.
a foreign key is a field or a column that is used to establish a link between two tables.
Actually foreign key is present in the fact table lookup table contains the primary key.
Using these keys we connect two tables.
In simple words, you can say that a foreign key in one table is used to point primary key in another table.

Row Context

In the Row context, we calculate values for each row. Calculated columns have row context because here we calculate a new value for each row.

Row Context is related to current rows. If you create a calculation using the calculated column, the row context involves the values of all columns from the current row.

Ex - Amount = Sales[Quantity] * Sales[Price]

Here we are adding a new column called Amount using calculated columns and it is calculating the amount for each row based on the quantity and product price.

Ex- Is_Parent = Customers[Parent]>0,"Yes","No":
Here we are adding a new column called Is_Parent using the calculated column and it is calculating the value based on the condition.

Filter Context:

When we are applying filters on the set of values of columns or tables using DAX calculations, that is known as Filter Context.

Filter context is present when we calculate the values based on the filters. Measure has filter context and based on these filters measure calculates the value and whenever filter changes it recalculates the values again.

Ex - Amount = Sales[Quantity] * Sales[Price]
We can't use this expression to create a new measure called Amount because we don't have filters.

We can calculate this column using SUMX() function because it is an iterator function that iterates over each row and then applies the aggregation function.
Ex - Amount = SUMX(Sales , Sales[Quantity] * Sales[Price])

If we apply the above same function using calculated columns then for each it will replicate the same value.

We have Query Conext in which The combination of row and filters create the final query for DAX

Calculated Columns:

Calculated columns allow you to add new or formula-based columns to tables. Calculated columns generate values for each row, which are visible within tables in the Data view

Calculated columns understand row context; they’re great for defining properties based on information in each row but generally useless for aggregation (SUM, COUNT, etc).

One more benefit of the calculated column is that if there is no unique column inside the table then we can create a new unique column using other columns.

Note:1] As a rule of thumb, use calculated columns when you want to “stamp” static, fixed values to each row in a table (or use the Query Editor!)
2] DO NOT use calculated columns for aggregation formulas, or to calculate fields for the “Values” area of a visualization (use measures instead)
3] Calculated columns are typically used for filtering data, rather than creating numerical values

Example of using calculated columns:
In this case, we’ve added a calculated column named “Parent”, which equals “Yes” if the [TotalChildren] field is greater than 0, and “No” otherwise (just like Excel!)

Since calculated columns understand row context, a new value is calculated in each row based on the value in the [TotalChildren] column •

This is a valid use of calculated columns; it creates a new row “property” that we can now use to filter or segment any related data within the model

Example of NOT using calculated columns:
Here we’re using an aggregation function (SUM) to calculate a new column named TotalQuantity
• Since calculated columns do not understand filter context, the same grand total is returned in every single row of the table
• This is not a valid use of calculated columns; these values are statically “stamped” onto the table and can’t be filtered, sliced, subdivided, etc.

Measures:
Measures are DAX formulas used to generate new calculated values.

Like calculated columns, measures reference entire tables or columns

Unlike calculated columns, measure values aren’t visible within tables; they can only be “seen” within a visualization like a chart or a matrix (similar to a calculated field in an Excel pivot)

Measures are evaluated based on filter context, which means they recalculate when the fields or filters around them change (like when a new row or column labels are pulled into a matrix or when new filters are applied to a report)

Note: As a rule of thumb, use measures (vs. calculated columns) when a single row can't give you the answer (in other words, when you need to aggregate)
2] Use measures to create numerical, calculated values that can be analyzed in the "values" field of a report visual

Calculated Columns VS Measures:

Calculated Columns: Values are calculated based on information from each row of a table (has row context)
Measures: Values are calculated based on information from any filters in the report (has filter context)

Calculated Columns: Appends static values to each row in a table and stores them in the model (which increases file size
Measures: Does not create new data in the tables themselves (doesn't increase file size)

Calculated Columns: Primarily used as rows, columns, slicers, or filters
Measures: Almost always used within the values field of a visual

Most Important DAX Functions in Power - BI:

1] RELATED(): Returns related values in each row of a table based on relationships with other tables.
Ex- Product_Lookup[ProductName]
Since this function requires row context, it can only be used as a calculated column or as part of an iterator function that cycles through all rows in a table (FILTER, SUMX, MAXX, etc)

2] CALCULATE() : It calculates the values based on the given condition or filter. IT works like the excel SUMIF function that's it.
Ex- calculate(SUM(Sales[amount]) , Territory_Lookup[Country] = "USA")

We can use measure inside the calculate function

Here firstly it will filter the table based on the country USA then it will apply the sum function.

Now it will just give us the one number as the Sum of sales amount for the USA but When we drag this value in a matrix with product category then again one filter is applied and it will show us USA sales for each country.

Instead of creating a calculated column like Amount in the table, we can create Amount Measure to save space and we can measure later also in calculate function if required.

CALCULATETABLE(): calculate table.
Calculate table is almost similar to calculate but calculate table returns the table as result as instead of a value.

3] ALL() : This function returns all the values of the columns or tables ignoring the passed filters.

Ex- ALL(Product , Product[color] = 'Red')
This function returns the table ignoring the red color filter.

We use All function in Calculate and Filter functions.

Ex- Calculate([sales_amount], ALL(Sales['color']='red'))
Here Sales_amount is measure and All function return all sales data ignoring the color filter and calculate function sales_amount without applying any filters.

3] ALLNOBLANKROW() - all no blank row.
This function also returns the values ignoring the passed filters but also ignores blank rows.

3] ALLEXCEPT() - all except
As the name suggested This function also ignores all the applied filters but keeps the filter on the provided column name.

3] ALLSELCTED() – allselected -
This function returns all the rows or columns by ignoring all the filters that have applied inside the query but keeps the filters that come from outside.

SImple it ignores chart or matrix filters and keeps the filter coming from the slicer or outside.

4] FILTER(): Returns a table that represents a subset of another table or expression

EX- FILTER(Sales,Terrotary[Country] = “USA”)

Here This function returns the Sales table which has USA as the country.

Note: 1] FILTER is used to add new filter context, and can handle more complex filter expressions than CALCULATE (by referencing measures, for example)

2] Since FILTER returns an entire table, it’s almost always used as an input to other functions, like CALCULATE or SUMX

3] Since FILTER iterates through each row in a table, it can be slow and processor-intensive; don’t use FILTER if a CALCULATE function will accomplish the same thing

5] ITERATOR Functions

Iterator (or “X”) functions allow you to loop through the same calculation or expression on each row of a table and then apply some sort of aggregation to the results (SUM, MAX, etc)

Imagine the function adding a temporary new column to the table, calculating the value in each row (based on the expression), and then applying the aggregation to that new column (like SUMPRODUCT)

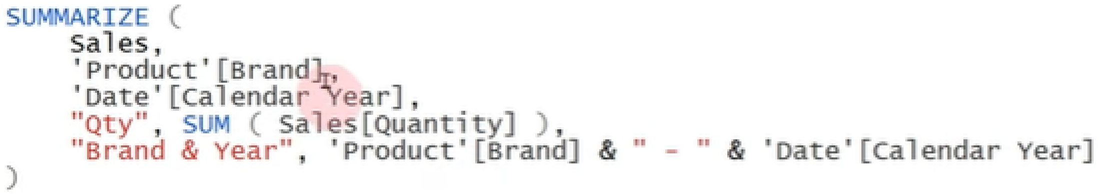
1**] SUMX() -** Ex - SUMX(Sales , Sales[product_cost] - Sales[product_sales_price])
2] AVERAGEX()
3] COUNTX()
4] RANKX()

6] SELECTEDVALUE():

This function returns a value when there is one value in the column.

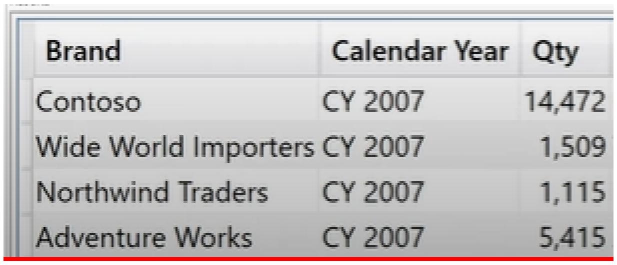
6] SUMMARIZE() - summarize function

Summarize group the table based on the passed column and return the columns.
We use summarize function to group the column and apply some aggregator functions and return the results.
Ex - Find the sum of sales for each brand over the calender year..



```
SUMMARIZE (
    Sales,
    'Product'[Brand],
    'Date'[Calendar Year],
    "Qty", SUM ( Sales[Quantity] ),
    "Brand & Year", 'Product'[Brand] & " - " & 'Date'[Calendar Year]
)
```

The result of the above query is below:



Brand	Calendar Year	Qty
Contoso	CY 2007	14,472
Wide World Importers	CY 2007	1,509
Northwind Traders	CY 2007	1,115
Adventure Works	CY 2007	5,415

Time Intelligence Functions:

These functions are designed to return the table to be easily used in a filter contexts.

The below all functions return the dates from 1st January of the specified year to the last date of the specified year.

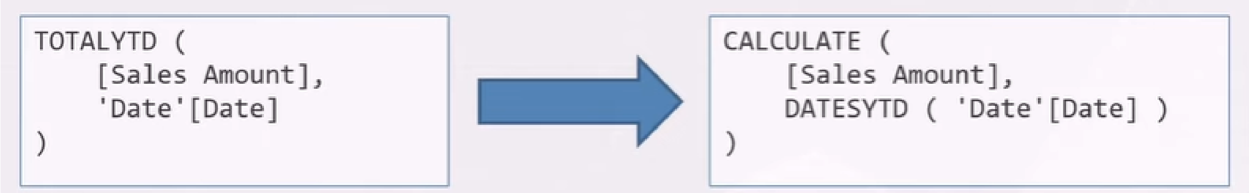
1] DATESYTD - dates year to date.
2] DATESQTD - dates quarter to date.
2] DATESMTD - dates Month to date.

Ex - date - 12-5-2008
CALCULATE([Total Sales] , DATESYTD(date_table[date]))
Here we are calculating the sales from 1-1-2008 to 12-5-2008

DATESQTD(date_table[date])) - returns dates from 1-4-2008 to 12-5-2008 because new 2nd quarter starts from April.

DATESMTD(date_table[date])) - returns dates from 1-12-2008 to 12-5-2008.

We have some coated functions Like TOTALYTD but they are not recommended.



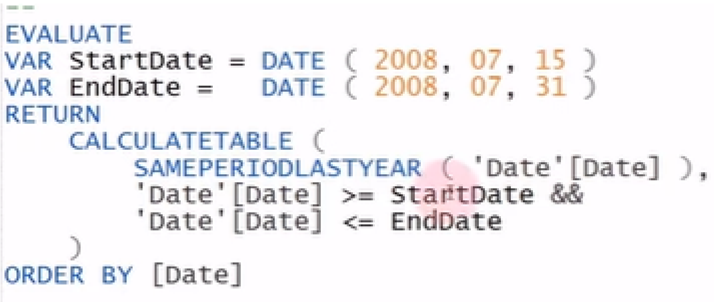
```
TOTALYTD (
    [Sales Amount],
    'Date'[Date]
)
```

```
CALCULATE (
    [Sales Amount],
    DATESYTD ( 'Date'[Date] )
)
```

3] DATESINPERIOD - to calculate the running total.

1] SAMEPERIODLASTYEAR - same period last year

This function helps us to generate dates for the same period in the last year.



```
--
EVALUATE
VAR StartDate = DATE ( 2008, 07, 15 )
VAR EndDate =   DATE ( 2008, 07, 31 )
RETURN
    CALCULATETABLE (
        SAMEPERIODLASTYEAR ( 'Date'[Date] ),
        'Date'[Date] >= StartDate &&
        'Date'[Date] <= EndDate
    )
ORDER BY [Date]
```

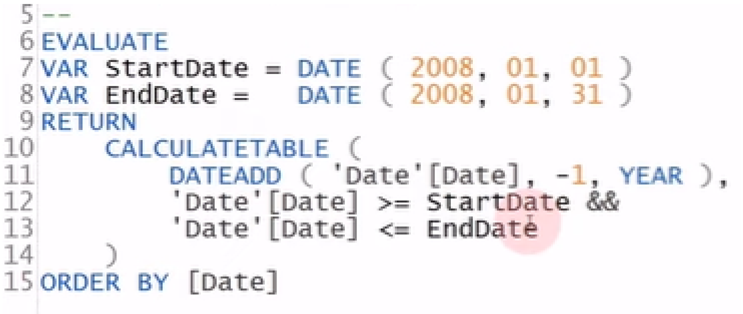
The drawback of this function is it generates the dates for the last 1 year only.

Using this function we can compare This year's December Month sales with last year's December month sales.

DATEADD():

This function is more generic and we can use month, year, and quarter here to generate dates.

Using this function we go back and forth using + & - signs.



```
--
EVALUATE
VAR StartDate = DATE ( 2008, 01, 01 )
VAR EndDate =   DATE ( 2008, 01, 31 )
RETURN
    CALCULATETABLE (
        DATEADD ( 'Date'[Date], -1, YEAR ),
        'Date'[Date] >= StartDate &&
        'Date'[Date] <= EndDate
    )
ORDER BY [Date]
```

The above query returns all the dates between 2007-1-1 to 2007-1-31.

Accessing Calculated Columns & Measures in Power BI:

Option 1: Select “New Measure” or “New Column” from the Home tab*

Option 2: Right-click within the table (in the Data view) or the Field List (in either the Data or Report view)

Quick Measures:

Quick Measures are pre-built formula templates that allow you to drag and drop fields, rather than write DAX from scratch
While these tools can be helpful for defining more complex measures (like weighted averages or time intelligence formulas),

Implicit Measures & Explicit Measures:

Implicit measures are created when you drag raw numerical fields (like “OrderQuantity”) into the values pane of a visual and manually select the aggregation mode (Sum, Average, Min/Max, etc)

Implicit measures are only accessible within the specific visualization in which it was created, and cannot be referenced elsewhere

Explicit measures are created by actually entering DAX functions (or adding “quick measures”) to define calculated columns or measures.

Explicit measures can be used anywhere in the report, and referenced within other DAX calculations to create “measure trees

Commonly Used DAX Functions:(Have a look at Course pdf page No 81.)

Power View:
The power view component is used to provide data insights using reports and dashboards and to provide interactive charts.

Types of charts we have in Power View:

1] Bar Charts & Column Charts.

Bar charts are useful when we want to see numerical data with categorical values.
Ex- No sales happened across states or countries.

In the Bar chart and column chart, we can have one numerical column and one or two categorical columns.

In these charts, we need to put the Numerical column in the values section and one categorical column in the axis section, and another category in the legend section but we can’t add two categorical columns in the axis section.

Both these charts are recommended when we need to show limited categories because if we add a categorical column with more categories then will get a scroller in the chart and that is not recommended.

In Bar and Column charts we can't show two numerical columns but we can show two categorical columns but this will just give us a count of each category.

Bar VS Column Charts:
A bar chart is a horizontal chart and categories are shown on the y axis. This chart is recommended when we have long category names.
.
A column chart is a vertical chart and categories are shown on the x-axis. This is chart is recommended when we have short category names.

We have six kinds of Bar charts and Column Charts.

All these charts are used to compare one numerical column and one or two categorical columns.
Ex- when we want to see how sales happened across countries and category-wise.

Note: We can use these charts when we have -ve values also.

1] Stacked Bar & Column charts VS Clustered Bar & Column charts.

Both these charts serve the same purpose but we need to use them based on the use cases.
The main thing is when there is one category and numerical column then it doesn't matter which chart we use but when there are two categorical and one numerical column then we need to pick a chart based on how categories values are distributed.

In the Stacked charts, all categories are divided into single bars or columns but in stacked cluster charts for each category single sub-bar is created.

Use Stacked Clustered charts when two categories' proportion is indistinguishable.
When there are more categories in the legend or categorical column then stacked clustered is not suitable.

When we use Clustered chart we don't need to compare the values with other by looking at column height we can understand everything.

Thumb Rule:
When we have fewer or more categories and the proportion of the category is distinguishable then use stacked charts.

When we have fewer category categories and the proportion is indistinguishable then use stacked clustered charts.

When we have more categories and the proportion is indistinguishable then use 100% stacked charts.

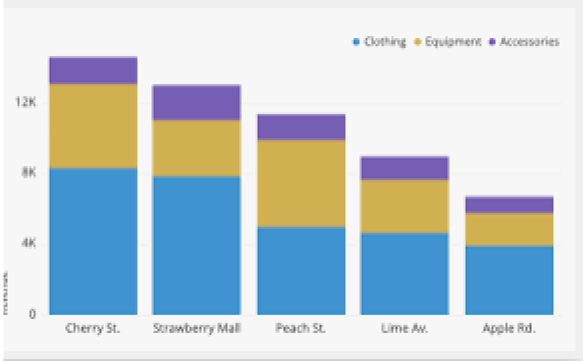
Example: **Suppose we have Sales as a Numerical column and States, product - category as categorical columns.**
Now we want to see how sales are happening across the states product category-wise.
We need to put states on the x-axis because there might be more categories and product categories in the legends because of fewer categories.

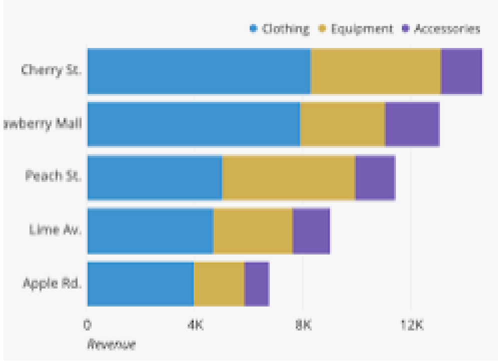
We can use stacked charts when the sales values are distinguishable across the product category.

We can use Stacked clustered charts when sales values are indistinguishable across the product category and there should be fewer categories.

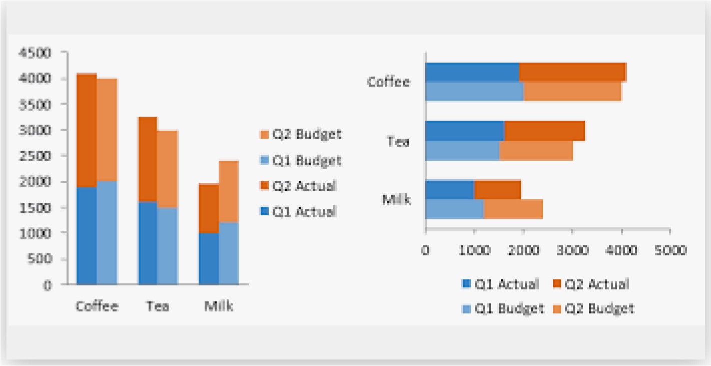
We can use 100% stacked chart when sales values are indistinguishable across the product category and if there are more categories.

Stacked Column & Bar Charts:





Stacked Clustered Column & Bar Charts:



2] Area Charts & Stacked area charts

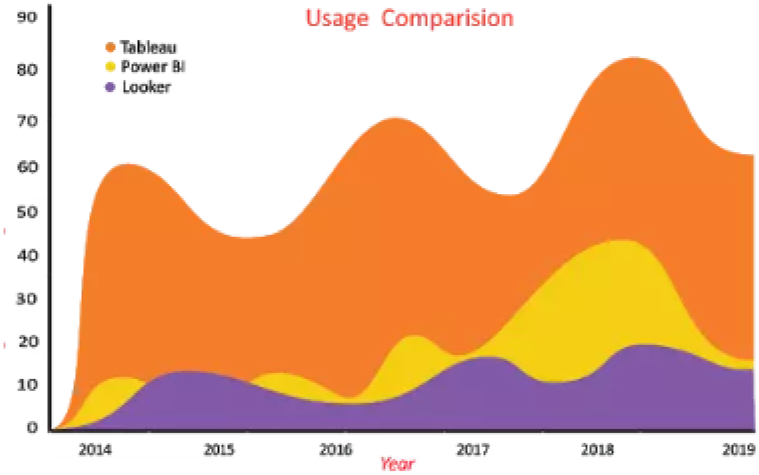
We use area charts when we need to show trend changes over time.
In Area charts, we always use date column on the x-axis and Numerical columns on the y-axis and also we can use one categorical column which is optional.

Ex- If we want to see how Yearly sales are happing.OR If we want to see how Yearly sales are happing across the product category or state.

Simple area chart VS Stacked area chart:

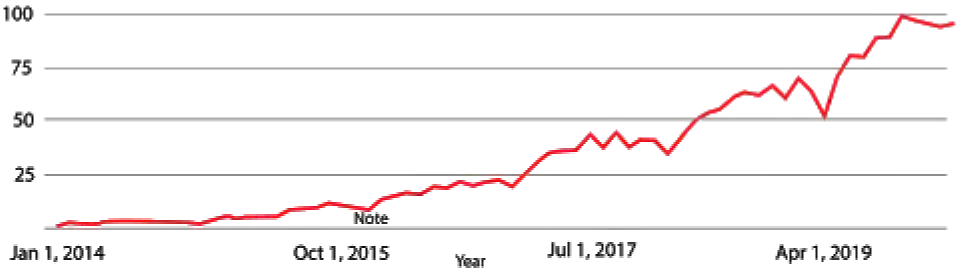
The difference between a simple area chart and a stacked area chart is that in a stacked area chart the area under the line is shaded. Most people prefer a Stacked area chart over simple area chart.

When we use a simple area chart with multiple categories in the categorical column then all the category lines will get mixed up and In the stacked area chart it doesn't happen all category lines are clear.



3] Line Charts

We use line charts when we need to show trend changes over time.
Ex- below chart shows the popularity of the Microsoft Power BI keyword in Google search across the world. It's clearly indicating that the popularity of Power BI has been increasing gradually since its inception.



In line charts, we can use two or more numerical columns and categorical columns on the x-axis.

Line Chart VS Area Chart:(https://www.edrawsoft.com/choose-line-area-chart.html)

If we want to compare multiple values then go with line charts because charts without shaded areas would be effective to show the comparison between different groups.

Use area charts when we want to compare some part of the value to the whole value.
Ex- when we want to compare out of all total hotel revenue how much revenue made by hotel-2

Filters VS Slicers

Filters and slicers both are used to filter the data and change the charts accordingly.
We have three kinds of filters

1) Page level
2) All pages
3) Visual level

Difference is that when we use slicers we can restrict filters for whichever charts we want.
Ex - suppose there are multiple charts about the sales in the report.Now I am adding slicer and enabling the filter for only orders and returns chart.
And disabling the filter for remaining charts.

Here I would have used visual filters but again I need to add one filter for each visual chart and I have three charts then I need three filters.

So with one slicer I can filter two charts or 3 or more charts.

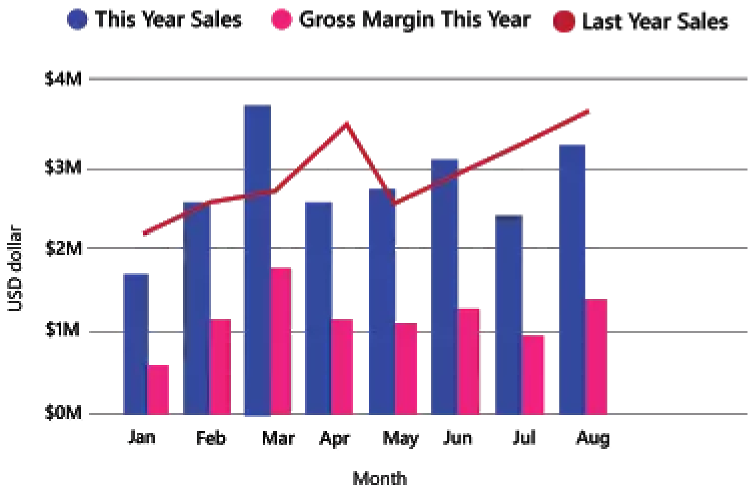
Important thing is that slicers are placed as charts in the report but filters are placed in the filter section.
So even non technical people can use slicers but it would be difficult for them to use filters.

4] Combo Charts

A combo chart is a combination of both the column charts and line charts that help you to make a quicker comparison of the data.

Combo charts are useful when we need to compare two or three numerical columns over one categorical column.
Ex- If we want to see how this year's sales are going on compared to the previous year's sales month-wise.
In the below charts we are using three numerical and one categorical column.

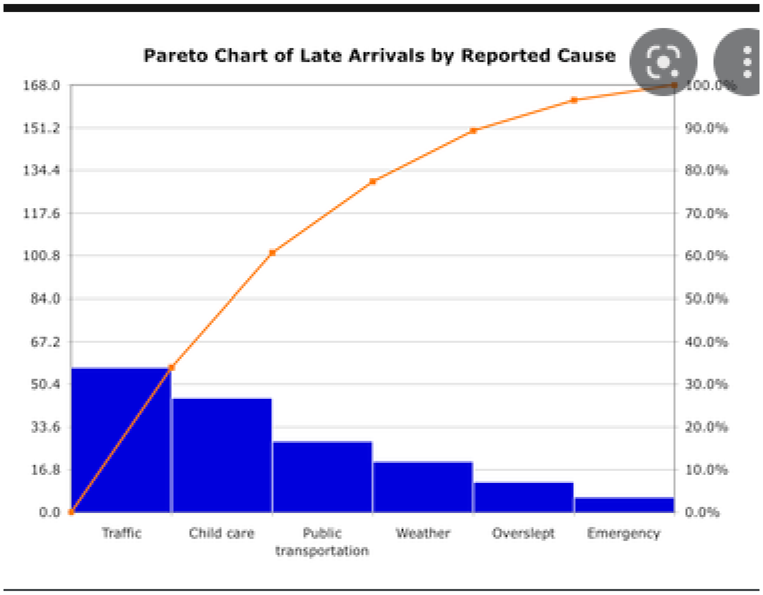
As we can see only in March month this year sales crossed the previous year's sales and also we can see current year gross margin.



Pareto Chart:

This chart is also a combination of column and line chart. In this chart, the columns are shown in decreasing order of the values and the line shows the cumulative percentage of categories.

In this chart, we can use only one numerical column and one categorical column.



If we have more categories between 4-6 then we can use this chart instead of pie chart. This chart also shows category-wise sales divided into percentages.

If we have more indistinguishable categories then don't use this chart.

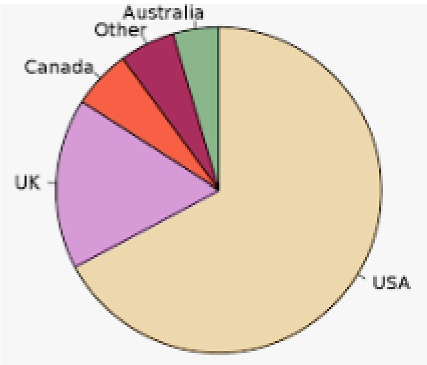
6. Pie Charts

Pie charts divide the whole data in parts. Each portion of a pie chart represents the percentages and the sum of all parts should be equal to 100%.

Pie charts are helpful when we want to see the values in the percentage or portion. Ex - if we want to see how much percent of sales happened across the regions.

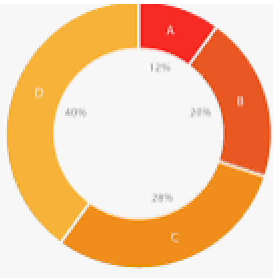
Again in the pie chart we can see one category's sales in percentage.
If we want to see sales percentage across regions with product categories then we need to use a 100% stacked bar chart.

In pie charts we can have only one numerical and one categorical column.
Use pie charts when we have less than 5 categories in the column.



7. Doughnut Charts

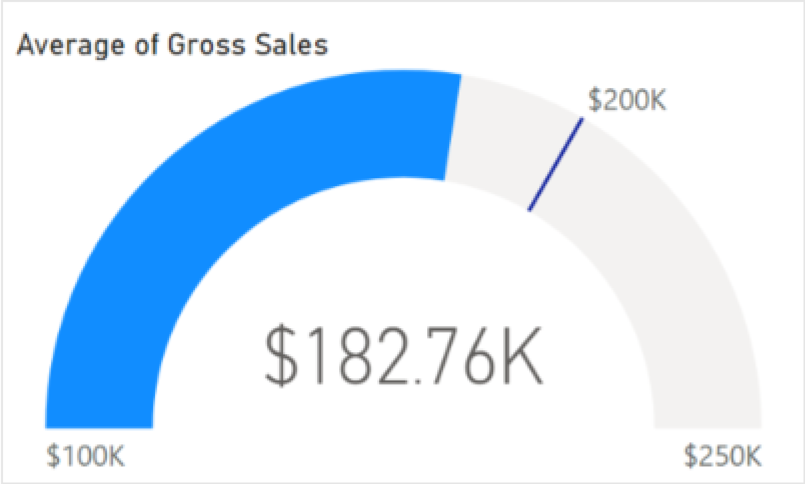
Doughnuts are similar to pie charts with hole in the center .doughnut charts show the whole data into the proposition.



8. Gauge Charts

We use Gauge charts to see progress towards the goal.

In the gauge chart, we provide one numerical column and one target value.
Ex- If we want to see current sales progress and setup the target.

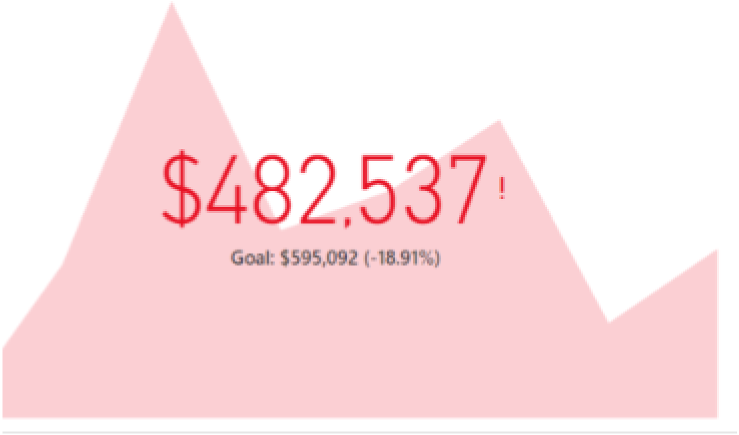


9] KPI

A Key Performance Indicator (KPI) is a visual cue that communicates the amount of progress made toward a measurable goal.
Use KPI to measure progress (what am I ahead or behind on?).

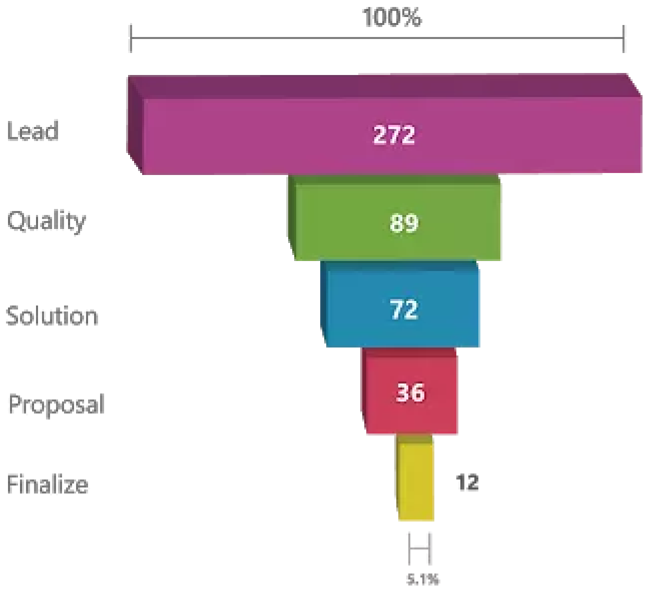
Gauge Chart VS KPI:

The gauge chart just shows the target we want to achieve and it helps to understand whether the target is achieved or not.
KPI charts show the target and also it helps to understand by how much value or percent we are ahead or back with the signal color red and green.



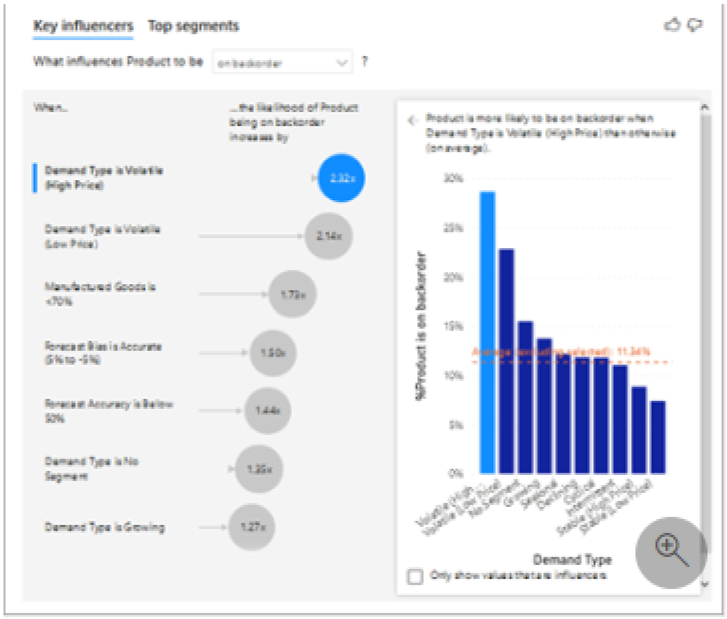
9] Funnel Charts

Funnels help visualize a process that has stages, and items flow sequentially from one stage to the next. One example is a sales process that starts with leads and ends with purchase fulfillment.



Key influencers chart

Key influencers are a great choice to help you understand the factors that influence a key metric. For example, *what influences customers to place a second order* or *why were sales so high last June*.


Key influencers
Top segments
What influences Product to be
on backorder
?
When...
...the likelihood of Product being on backorder increases by
Demand Type is Volatile (High Price)
2.32x
Demand Type is Volatile (Low Price)
2.14x
Manufactured Goods is <70%
1.72x
Forecast Bias is Accurate (5% to -5%)
1.50x
Forecast Accuracy is Below 50%
1.44x
Demand Type is No Segment
1.35x
Demand Type is Growing
1.27x
30%
25%
20%
15%
10%
5%
0%
%Product is on backorder
Average (excluding selected): 11.34%
Volatile (High Price)
Volatile (Low Price)
No Segment
Growing
Seasonal
Declining
Cyclical
Intermittent
Stable (High Price)
Stable (Low Price)
Demand Type

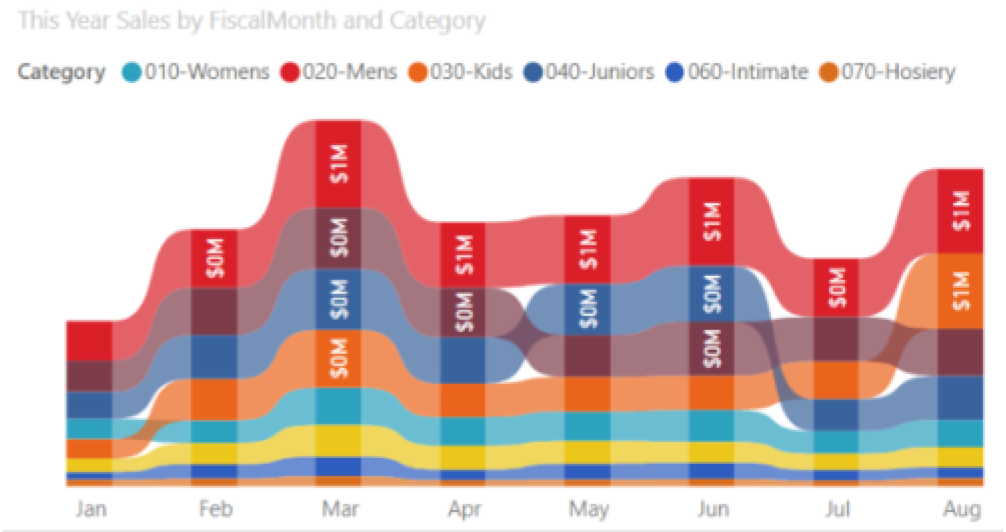
Product is more likely to be on backorder when Demand Type is Volatile (High Price) than otherwise (on average).

□ Only show values that are influencers

Ribbon chart:

A ribbon chart is used to visualize the data and quickly identify which data category has the highest rank (largest value).

Ribbon charts show which data category has the highest rank (largest value). Ribbon charts are effective at showing rank change, with the highest range (value) always displayed on top for each **time period.**



15. Tree Maps:

Treemap is used to display large amounts of hierarchical data in Rectangle boxes.

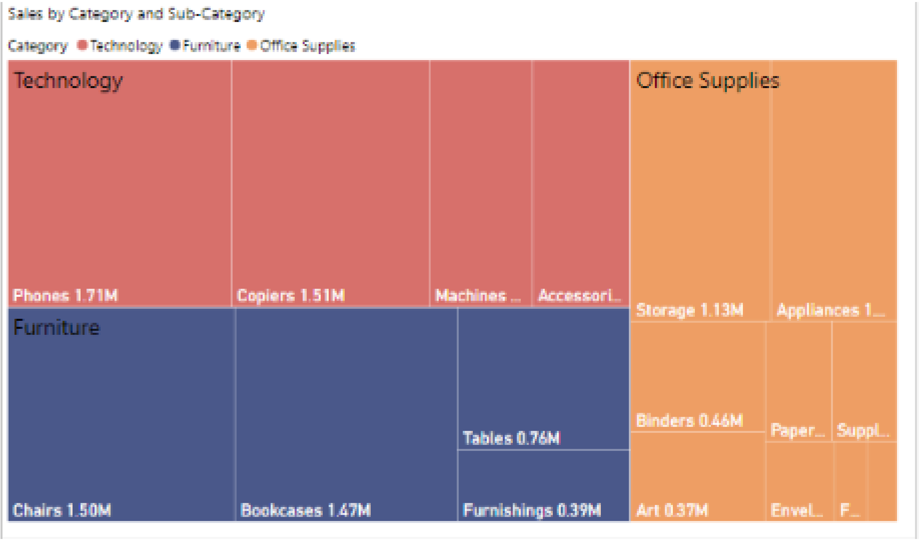
Hierarchical means data is divided into categories and sub-categories then it displays the values in descending order of the values.
The first rectangle is large and the last rectangle is low. we can easily identify them.

In tree maps, we can have one numerical column and one categorical or two categorical columns.

Ex-When we want to see how sales are happening across the categories and subcategories and a number of sub-categories are more.

When data is large and categories are more and numerical columns values are indistinguishable in bar charts we can use Treemaps because in treemap we can easily identify which category has made more sales because sales values are ordered in descending order.

With tree maps, we can see how sales are distributed across the categories.



Power Maps:

In power BI we have 4 kinds of maps.
1]Basic map
2]Filled map
3]shape map
4] ArcGIS map

The basic map just shows the bubble or points on the locations but the filled map shade or fill the area of the location.

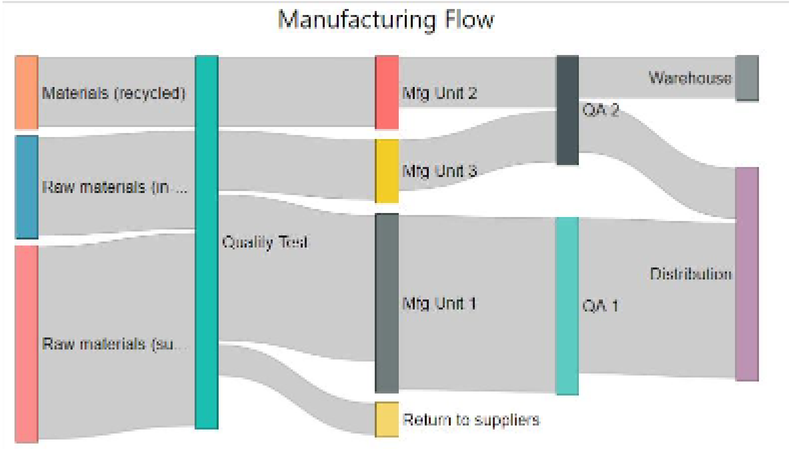
Custom Charts:

We can access these charts from the report Marketplace option or from the Home tabs From marketplace option in Power BI.

1] Histogram chart
2] Sankey chart
3] Duel KPI

Sankey Chart(Custom Chart):

Sankey charts are used to show the flow between source and target.



Power BI Services:

Power BI services are responsible for publishing the reports to the service or use.

We have three kinds of licenses in Power BI:

1] Free - This is available for everyone
2] Pro - This is available to data analysts working in the companies.
3] Premium - we can have only one Premium Account.

My Workspace & Workspace:

My workspace is a personal space where we create reports and charts and then publish them in the workspace. Workspace is like a general or global space where everyone can publish their reports in the organization.

Roles available in the Workspace:

1] Admin - This is the highest role who can access everything and add and remove the people and grant access.
2] Member: Member is one level above Contributor. He can do all the things except some things that the admin can do like adding a few people.
2] Contributor: The contributor is responsible for creating reports & dashboards
4] Viewer - The viewer can just see the reports and dashboards.
Can not see the data and can't modify the data.

RLS - Row Level Security:

Using RLS we can restrict data to users.
Ex- For Indian people, we can restrict USA data and Vice-versa.

We have two types of RLS:

1] Static - Here we define user names and data to restrict. For doing this we need to go from Power BI desktop to service. This is a bit hectic process.

2] Dynamic - Here we use the User principle name as DAX function which is responsible for capturing the system name or user name based on that we will give the access. After this step, we have the entitlement step where we give tables to access.

Reports & Dashboards:
We can have multiple reports to show all the insights from different data but we must have one dashboard to show all the important reports.

Reports are interactive where we can use filters and the dashboard is a snapshot of reports.

Gateway & Its Type:
Gateway is responsible for moving our data from on-prem to the services or desktop. The gateway is a connectivity mode between data and service.

1] Standard: Standard gateway is used in the organization between colleagues.
2] Personal: Personal gateway is for personal use.

Data Flows:

Data Flows are used to transform and create datasets in the Power BI service. Whatever transformers we are applying in the power query we can apply them in the Data power bi service.