Power BI

Agenda

- Introduction to Power BI
- Power Query Editor
- Data Analysis Expression
- Power BI Relationships and KPI
- Administration and Security in Power BI
- Formatting and Best Practices in Power BI

- Exploratory Data Analysis

Introduction to Power BI

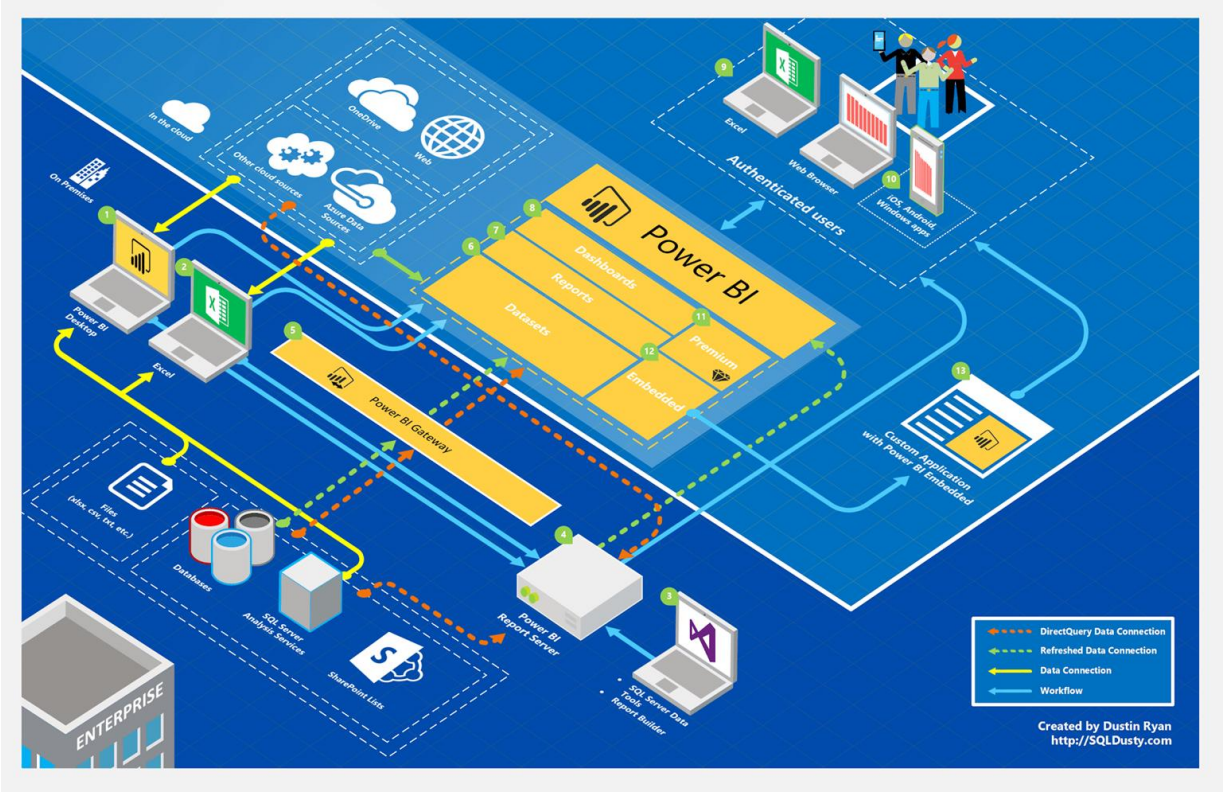
Introduction

- Power BI is a business analytics service developed by Microsoft. It allows users to visualize and analyze data with interactive dashboards, reports, and data visualizations.
- Power BI connects to a wide range of data sources, including Excel spreadsheets, cloud-based and on-premises data sources, and third-party applications.

Components

- **Power BI Desktop:**
  - This is a Windows application that allows users to create and publish reports and dashboards.
  - It provides a range of data visualization and data modeling tools, as well as features for data transformation and integration.
- **Power BI Service:**
  - This is a cloud-based service that allows users to share and collaborate on reports and dashboards.
  - It provides a web interface for accessing and interacting with data visualizations, as well as features for data sharing and collaboration.
- **Power BI Mobile:**
  - This is a mobile app that allows users to view and interact with reports and dashboards on their mobile devices.
  - It provides a responsive design that adapts to different screen sizes, as well as features for offline access and mobile-specific interactions.

Common Workflow of Power BI



- Begins by connecting to data sources and building a report in Power BI Desktop
- Publish report from Power BI Desktop to the Power BI service
- Share it to end users with the Power BI Service
- Mobile Devices can view and interact with the report

Power BI Basics

Power BI Desktop

Power BI Desktop is a Windows application that allows users to create interactive data visualizations, reports, and dashboards. It is a part of the Power BI suite of tools and provides a comprehensive set of features for data modeling, data preparation, data analysis, and data visualization.

Features:

1. **Get Data**-Easily connect, clean, and mashup data
   - ➢ Connect to 80+ data sources, both on-premises and cloud
   - ➢ Shape, transform, and clean data for analysis
   - ➢ Live connectivity to on-premises and cloud data sources
   - ➢ Extend with custom data connectors for any data source
   - ➢ Prep your data using the familiar Power Query experience on the web
   - ➢ Get started quickly with a common data model
   - ➢ Extend self-service prep to Azure Data Lake Storage

1. **Analyze-**Build powerful models and flexible measures
   - ➢ Automatically create model when connecting to data
   - ➢ High performance, in-memory engine
   - ➢ Point and click analysis with Quick measures, clustering & binning
   - ➢ Create powerful measures with familiar DAX (Data Analysis Expressions) formulas

1. **Visualise-**Create stunning interactive reports
   - ➢ Author reports using 150+ visuals via a drag-drop canvas
   - ➢ Explore data across multiple interactive visualizations
   - ➢ Provide insights in the context of the business with Custom Visuals
   - ➢ Visualize data story with bookmarks and customer navigation

1. **Publish-**Share insights with others
   - ➢ Publish directly to the cloud or on-premises
   - ➢ Automatic data refresh, so the reports are always up to date
   - ➢ Package your reports in apps for easy consumption and control
   - ➢ Manage analytics content with admin and governance tools

1. **Collaborate-**Empower your organization with self-service analytics

   Power BI Desktop allows users to collaborate and share reports and dashboards with other team members.

Power BI Services & Integration with other Apps

- Power BI is a business intelligence platform developed by Microsoft that allows users to create interactive visualizations and reports from various data sources.
- Power BI Services is the cloud-based service provided by Microsoft to host and share Power BI reports, dashboards, and data visualizations with others.

Integration with Power BI

Power BI can be integrated with a variety of tools and platforms to extend its capabilities and provide users with additional functionalities.

Excel and Power BI:

Power BI and Excel are two powerful tools developed by Microsoft that can be used together to provide a comprehensive solution for data analysis and visualization. Here are some ways in which Power BI can be used with Excel:

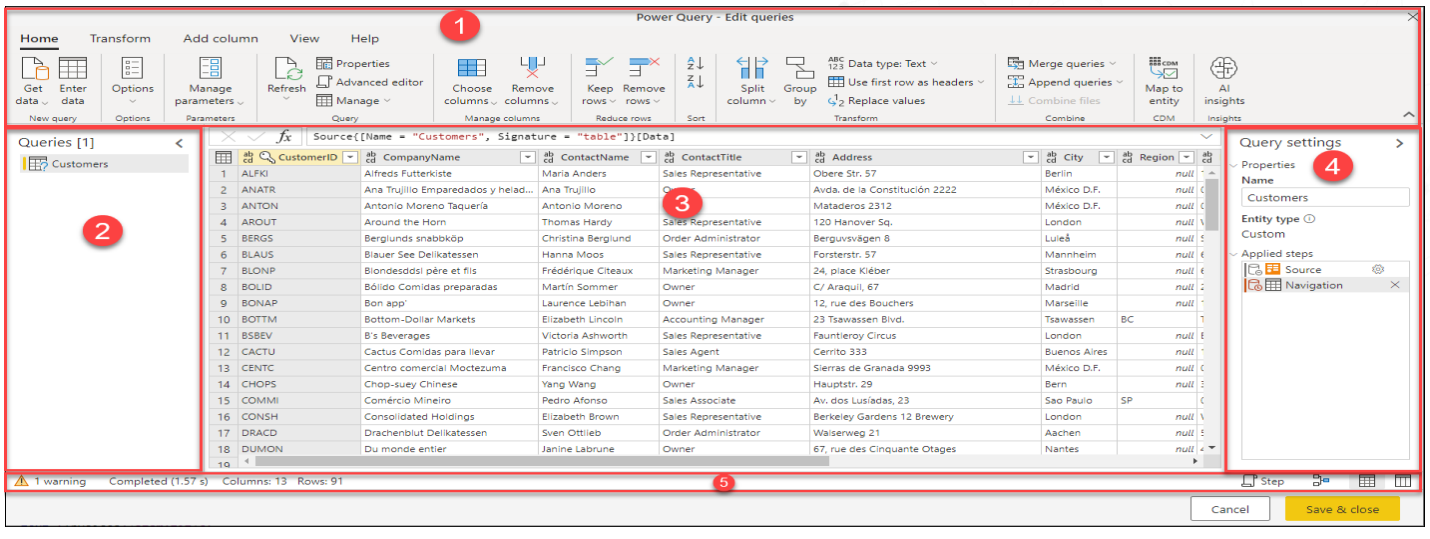
- Analyze in Excel
- Use Excel to view and interact with a dataset you have in Power BI
- Import Excel data into Power BI
- Connect to the data in your workbook so you can create Power BI report and dashboards
- Upload your Excel file to Power BI
- Bring your Excel file into Power BI to view and interact with it just as you would in Excel Online. Pin ranges to Dashboards

Power Query Editor

Power Query Editor

- Power Query Editor is a data transformation and cleansing tool that is built into Power BI, Excel, and other Microsoft products.
- It provides users with a powerful set of tools for transforming and shaping their data before it is imported into a data model or visualization.

User Interface:



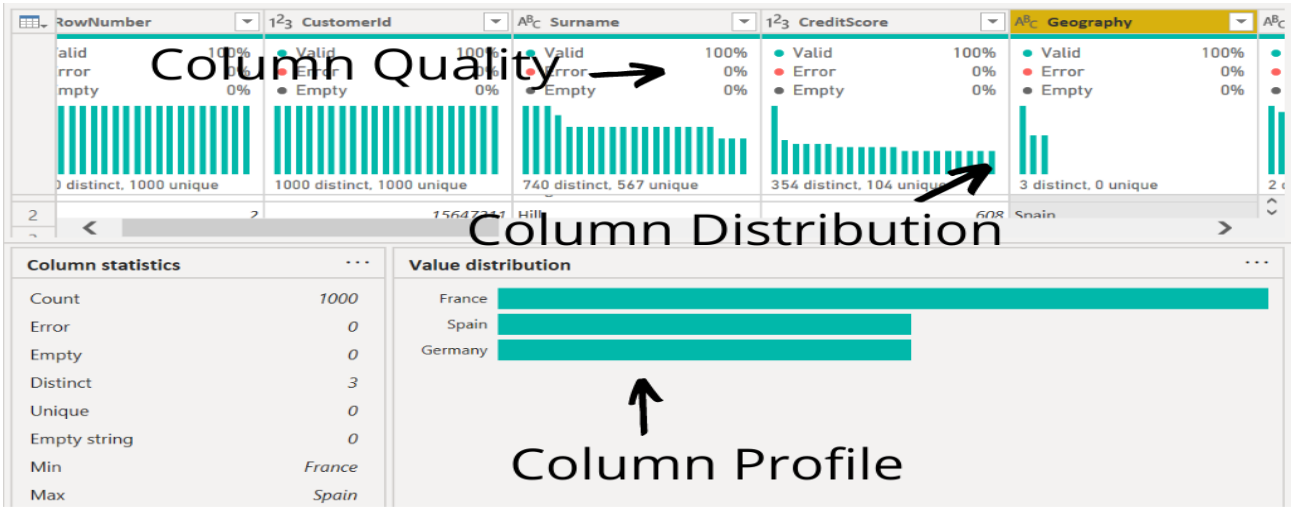
1. **Ribbon:** The Power Query Editor interface includes a ribbon that provides users with quick access to a range of tools and functions. The ribbon includes tabs for Home, Transform, Add Column, and View, among others.
2. **Queries:** Contain Queries that gets processed.
3. **Data preview:** The Power Query Editor interface includes a data preview pane that allows users to preview their data as they transform and shape it. The preview pane shows a sample of the data, which can be refreshed to show changes in real-time.
4. **Query settings:** The Power Query Editor interface includes a pane that shows query settings, including the data source, data types, and other settings.
5. **Footer-** Contains number of row columns and query completion taken time.

Data Profiling Tools

The Data Profiling tab in Power BI provides users with a range of data profiling tools, Provide new and intuitive ways to clean, transform, and understand data

Includes:

- Column Quality
- Column Distribution
- Column Profile



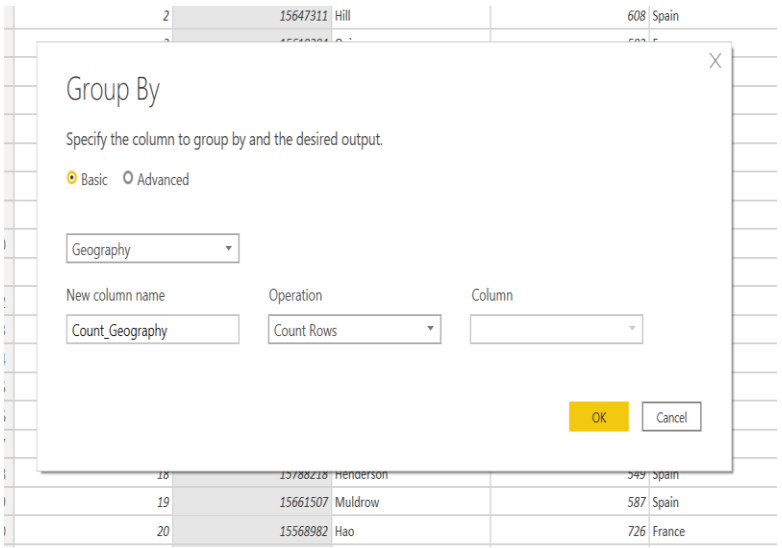
Power Query Editor

Group By Dialog

To group data in Power BI, you can use the "Group By" feature in the "Transform Data" dialog. Here's how to access it:

- Select the table you want to group in the "Fields" pane.
- Click on the "Transform Data" button in the "Home" tab of the ribbon.
- This will open the "Query Editor" window. In the "Query Editor", select the column(s) you want to group by.
- Go to the "Transform" tab in the ribbon and click on "Group By".
- In the "Group By" dialog, you can specify the columns to group by and the aggregate calculations you want to perform.
- Click "OK" to apply the grouping.

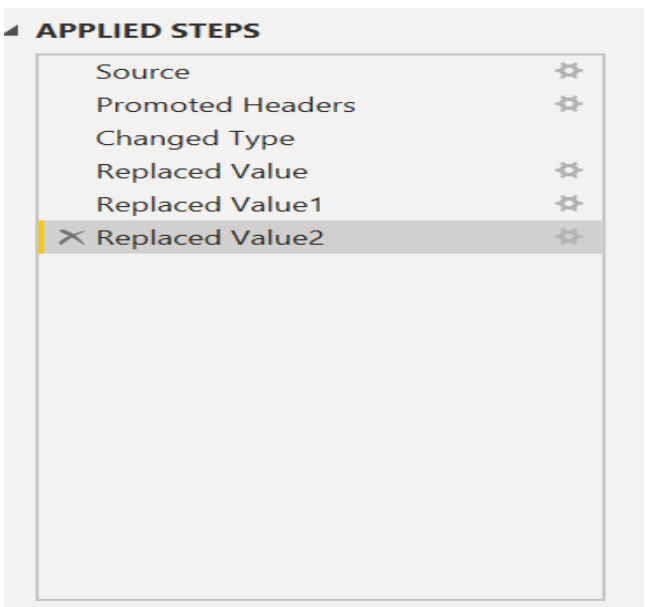
Once you have applied the group by transformation, you can see the results in the "Fields" pane as a new table.



	Geography	Count_Geography
1	France	5014
2	Spain	2477
3	Germany	2509

Applied Steps

- Any steps performed in Power BI is logged under the Applied Steps.
- Steps can be added or deleted anytime during the process.



Power Query Editor

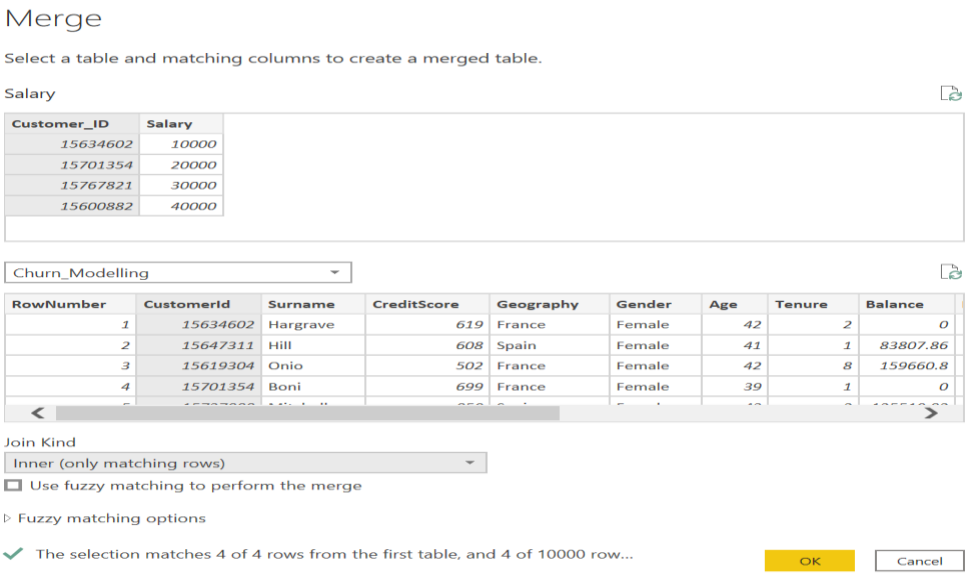
Appending Vs Merging

Merging

- When you have one or more columns that you'd like to add to another query.
- To merge data in Power BI, follow these steps:
    - Select the two tables you want to merge.
    - Click on the "Merge Queries" button in the "Home" tab of the ribbon.
    - In the "Merge Queries" dialog, select the common column(s) on which to merge the tables.
    - Choose the join type you want to use (inner, left outer, right outer, or full outer).
    - Specify any additional columns that you want to include in the merged table.
    - Click "OK" to apply the merge transformation.

Appending

- When you have additional rows of data that you'd like to add to an existing query.
- To append data in Power BI, follow these steps:
    - Go to the "Home" tab in the ribbon and select "Append Queries".
    - In the "Append Queries" dialog, select the tables you want to append together.
    - Arrange the order of the tables as you want them to appear in the final result.
    - Specify any column mappings that are needed.
    - Click "OK" to apply the append transformation.



Building Blocks of Power BI

Visualization

- A visual representation of data is called visualization.
- For example, a chart, or a graph can be used to represent data visually.
- PowerBI provides 150+ charts to represent data, that are dynamic in nature.

Datasets

- A dataset is a collection of data or information, that can be used to represent visually to draw insights.

Reports

- A collection of visualizations that appear together on one or more pages.
- It is a collection of items that have common motive.
- A report can also be part of a dashboard chart.

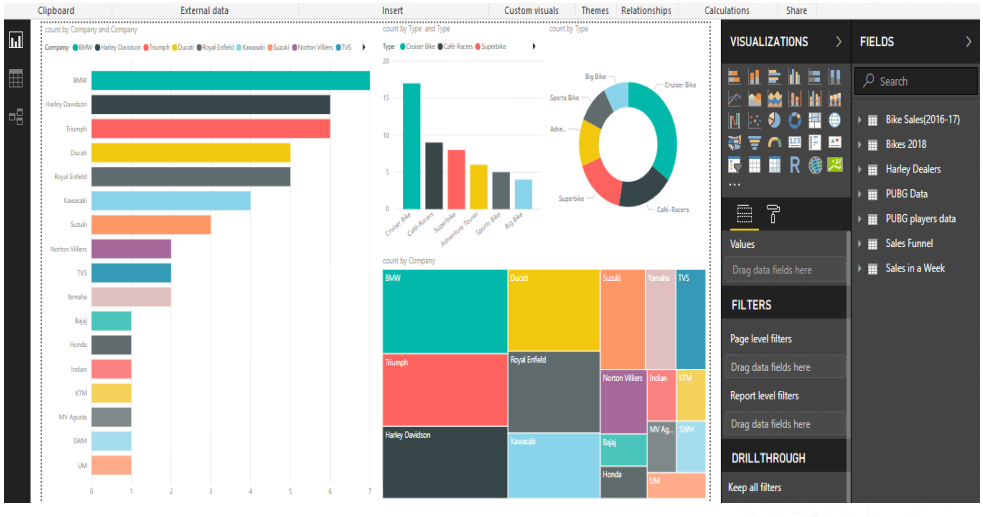
Dashboards

- A single page interface that uses the most important elements of a report to tell a story.
- Users can create dashboards that provide an overview of key metrics or create detailed reports that explore specific aspects of their data.

Tiles

- A tile is a single visualization found in a report or on a dashboard.
- On a dashboard, a page from report can be pinned as a tile in the dashboard.

Different Charts in Power BI



How to create charts in Power BI?

Here are the basic steps to create a chart in Power BI:

1. **Connect to your data source:** Power BI allows you to connect to various data sources such as Excel, SQL Server, and many others. Once you connect to your data source, you can import the data into Power BI.
2. **Create a new report:** Click on the "Create" button on the left-hand side of the screen, then select "Report".
3. **Choose a visualization:** Click on the "Visualizations" button on the right-hand side of the screen, then select the type of chart you want to create (e.g., bar chart, line chart, pie chart, etc.).
4. **Add fields to the chart:** Drag and drop the fields from your data set into the appropriate areas of the chart. For example, if you want to create a bar chart showing sales by region, drag the "Region" field to the "Axis" area and the "Sales" field to the "Values" area.
5. **Customize the chart:** You can customize the chart by changing the colors, fonts, and other settings. You can also add titles, legends, and other features to the chart.
6. **Save and share the report:** Once you are done creating the chart, you can save the report and share it with others. You can also publish the report to the Power BI service and embed it in a website or application.

Few Types of Chart in Power BI

- ➔ **Column chart:** A column chart is used to compare values across categories. It is effective for showing trends over time or comparing multiple categories at once.
- ➔ **Bar chart:** Similar to a column chart, a bar chart is used to compare values across categories, but it is ideal for horizontal orientation.
- ➔ **Line chart:** A line chart is used to show trends over time or to track changes in data over time. It is effective for displaying continuous data.
- ➔ **Area chart:** An area chart is similar to a line chart, but it shows the area beneath the line. It is useful for showing the total value of a set of data.
- ➔ **Pie chart:** A pie chart is used to show the proportion of each category to the whole. It is useful when the data has clear categories and the total can be easily defined.
- ➔ **Donut chart:** A donut chart is similar to a pie chart, but it has a hole in the middle. It is useful when you want to emphasize the proportions of the categories.

Data Analysis Expression

Data Analysis Expression

DAX (Data Analysis Expressions) is a formula language used in Power BI to create custom calculations, aggregations, and measures.
Here are some basic concepts of DAX in Power BI:

Calculated columns:

- You can create new columns in your data table by defining a formula using DAX.
- The formula can be based on one or more columns in the table.
- Calculated columns are computed at the row level within the table it belongs to.
- Represents a single value per row.
- It is computed at compile time.
- It provides dynamic results, based on Rows.

Example: Tenure_Months := Churn[Tenure]*12

How to create a calculated columns?

To create a calculated column in Power BI, you can follow these steps:

- Open the Power BI Desktop application and go to the "Data" view.
- Select the table you want to add the calculated column to.
- Click on the "Modeling" tab in the ribbon.
- Click on "New Column" in the "Calculations" group.
- Enter the formula for the calculated column in the formula bar.
- Press "Enter" to create the column.

Measures:

- Measures are calculations that aggregate data from a table or multiple tables.
- Measures are used to perform calculations such as sum, average, count, or any other custom calculation.
- Evaluated in the context of the cell evaluated in a report or in a DAX query
- Measures represents a single value per data model
- It is computed at run time
- It provides Dynamic results, based on filters

Example: TotalQuantity := SUM(Sales[Quantity])

How to create a measure?

To create a measure using DAX in Power BI, you can follow these steps:

- Open the Power BI desktop application and load your data into the report.
- Click on the "New measure" button in the "Fields" pane.
- Enter a name for your measure.
- Write your DAX formula in the formula bar. For example, you can create a measure to calculate the sum of sales by writing the formula "Total Sales = SUM(Sales[Amount])".
- Press "Enter" to save the formula.

Data Analysis Expression

Implicit measures

- Measures that are not explicitly defined in a dataset but are instead derived from the values in the dataset.
- In Power BI, implicit measures can be created using the "New Measure" option in the "Modeling" tab.

To create an implicit measure in Power BI, you can follow these steps:

1. Click on the "New Measure" option in the "Modeling" tab.
2. In the formula bar, write the formula for the implicit measure using the values in the dataset. For example, if you want to create an implicit measure that calculates the average sales per month, you can write the formula "AVERAGE(Sales[Sales Amount])/12".
3. Press Enter to create the measure.
4. The measure will be added to the "Fields" pane and can be used in visuals and other calculations.

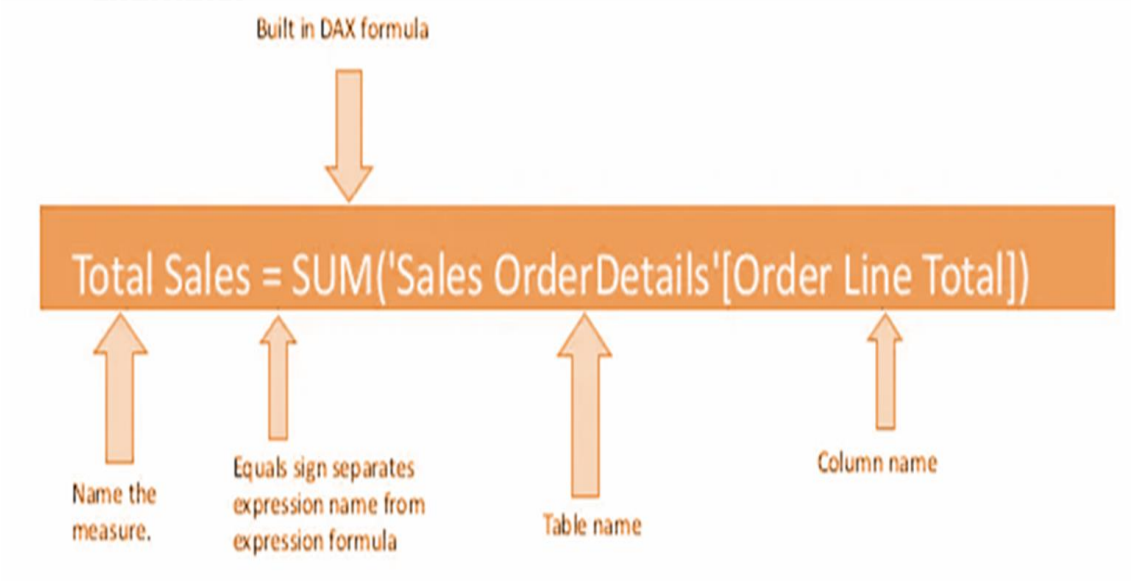
- Implicit measures can be useful when you need to perform calculations that are not explicitly defined in the dataset.
- They can also be used to simplify complex calculations and make them more manageable.
- However, it's important to note that implicit measures can be less intuitive and may require additional documentation to explain their purpose and how they are calculated.

Elements of DAX:

A DAX formula, generally comprises of the following:

- **Measure Name**
- **DAX function**
- **Table Name**
- **Column Name**

Example:



Basic DAX Functions

DAX (Data Analysis Expressions) is the formula language used to create custom calculations in Power BI. The following discusses few basic DAX functions:

- **SUM:** Calculates the sum of a column or expression.
  - ***Syntax:***

    SUM(<column or expression>)
  - ***Example:*** SUM(Sales[Revenue])

- **AVERAGE:** Calculates the average of a column or expression.
  - ***Syntax:***

    AVERAGE(<column or expression>)
  - ***Example:*** AVERAGE(Sales[Revenue])

- **MIN:** Returns the smallest value in a column or expression.
  - **Syntax:**

    MIN(<column or expression>)
  - **Example:** MIN(Sales[Revenue])

- **MAX:** Returns the largest value in a column or expression.
  - **Syntax:**

    MAX(<column or expression>)
  - **Example:** MAX(Sales[Revenue])

- **COUNT:** Counts the number of rows in a table, or the number of non-blank values in a column or expression.
  - ***Syntax:***

    COUNT(<column or expression>)
  - **Example:** COUNT(Sales[Revenue])

- **COUNTROWS:** Counts the number of rows in a column or expression.
  - ***Syntax:***

    COUNTROWS(<column or expression>)
  - **Example:** COUNTROWS(Sales[Product])

Basic DAX Functions

Date Functions

Power BI DAX provides several functions to work with dates and time values. Here are some of the commonly used date functions in Power BI DAX along with their syntax and examples:

TODAY()

This function returns the current date in the system's local time zone.

Syntax: TODAY()

Example: TODAY()

-returns 3/29/2023 if the current date is March 29, 2023.

DATEDIFF()

This function returns the differences between two dates.

Syntax:

DATEDIFF(start_date,end_date,unit)

Example:

DATEDIFF([Date1],[Date2],"day")

-where [Date1] and [Date2] are the columns in your dataset that contain the two dates you want to compare.

The DATEDIFF function returns an integer value, so if you want to display the result as a decimal or percentage, you will need to format the result accordingly.

DATEADD()

In Power BI DAX (Data Analysis Expressions), the DATEADD function can be used to add a specified number of intervals (such as days, months, or years) to a given date.

Syntax:

DATEADD(<start_date>, <number_of_intervals>, <interval>)

Example:

Final_Date= DATEADD('Table'[Date], 30, DAY)

You can also use negative numbers for the *'number_of_intervals'* parameter to subtract intervals from the starting date.

Example: To subtract 2 months from a given date

Final_Date = DATEADD('Table'[Date], -2, MONTH)

Basic DAX Functions

Calendar Functions

- In Power BI, you can create a calendar table to facilitate time-based analysis and visualization.
- A calendar table is a table that contains a row for each date in a specified range, along with additional columns that provide various attributes and calculations related to each date.

  Here are the general steps to create a calendar table in Power BI:

  1. Create a new blank query in the Power Query Editor.
  2. In the Home tab, click on the "New Source" dropdown and select "Blank Query".
  3. In the formula bar, enter the following formula to generate a list of dates:

     = List.Dates(#date(2021,1,1),365,#duration(1,0,0,0))

  1. This formula generates a list of 365 dates starting from January 1st, 2021.
  2. Rename the query to "Calendar".
  3. In the "Add Column" tab, click on "Date" and select "From List".
  4. Select the date column you generated in step 3.
  5. In the "Modeling" tab, click on "New Table" to create a new table.
  6. In the formula bar, enter the following formula to create the calendar table:

     Calendar = CALENDAR(DATE(2021,1,1), DATE(2021,12,31))

     -This formula creates a calendar table with a row for each date between January 1st, 2021 and December 31st, 2021.

  1. Expand the "Calendar" table to include additional columns such as day of week, week number, month name, quarter, and year.

- Once you have created a calendar table, you can use it in your data model to create relationships with other tables that contain time-based data.
- You can also use it to create time-based calculations such as year-to-date, month-to-date, and rolling averages.

Basic DAX Functions

Get Year, Month name, Month number and day of the week

- To display the year, month number, month name, and day of the week in a calendar in Power BI, you can use a combination of DAX expressions and custom formatting.
- Here are the steps:
  - Create a new calculated column in your date table by selecting the "New Column" option from the "Modeling" tab.
  - Use the following DAX expression to extract the year, month number, month name, and day of the week from the date column:
    - Year = YEAR('Date'[Date])
    - Month Number = MONTH('Date'[Date])
    - Month Name = FORMAT('Date'[Date], "MMMM")
    - Day of the Week = FORMAT('Date'[Date], "dddd")
  - Once you have created these columns, you can use them in a calendar visual in Power BI.
  - Select the calendar visual and go to the "Format" pane.
  - In the "Data colors" section, select "Year" and choose a color for the year.
  - In the "Date headers" section, select "Month" and choose "Month Name" as the format.
  - In the "Weekday headers" section, select "Weekday" and choose "Day of the Week" as the format.
  - Adjust other formatting options as desired and your calendar visual should now display the year, month number, month name, and day of the week for each date.

Basic DAX Functions

Context

- Context in Power BI refers to the subset of data that is currently being considered or evaluated by a DAX formula.
- There are two types of context in Power BI: **Row context** and **Filter context.**

Row context:

- ❏ Created for each individual row in a table, and it determines the values of all columns for that row.
- ❏ When a DAX formula is evaluated in row context, it uses the value in the current row to perform calculations.

Filter context:

- ❏ Created by any filters that have been applied to a visual or a calculation.
- ❏ The filters can be explicit (set by the user) or implicit (applied automatically by Power BI).
- ❏ When a DAX formula is evaluated in filter context, it uses the values of the columns that are included in the filter context.

Example:

- Suppose you have a table that contains sales data for different products, and you want to create a measure that calculates the total sales for a specific product.
- If you create the measure using the SUM function, like this:

  Total Sales = SUM(Sales[Amount])

  then the result will depend on the current context.

  -If you use this measure in a visual that includes a filter for a specific product, then the measure will be evaluated in filter context and will only return the total sales for that product.

  -On the other hand, if you use the measure in a visual that does not have any filters applied, then the measure will be evaluated in row context and will return the total sales for all products.

Basic DAX Functions

Calculate

- The CALCULATE function is used to modify the filter context of a calculation by applying additional filters to the data.
- **Syntax:**

  CALCULATE(<expression>, <filter1>, <filter2>, …)

  - ❖ The first argument is the expression or calculation that you want to modify the filter context for, and the additional arguments are the filters that you want to apply.
  - ❖ The filters can be specified using column names, relationships, or DAX expressions.

- **Example:**
  - ➔ Suppose you have a sales table that includes a column for sales amount and you want to calculate the total sales amount for a specific category.
  - ➔ You can use the CALCULATE function as follows:

    Total Sales = CALCULATE(SUM(Sales[Sales Amount]), Sales[Category] = "Electronics")

    -This will return the sum of the sales amount column, but only for the rows where the category is "Electronics".

Filter

- The FILTER function is used to create a virtual table that includes only the rows that meet a specific condition.
- **Syntax:**

  FILTER(<table>, <condition>)

  - ❖ The first argument is the name of the table or the expression that you want to filter, and the second argument is the condition that you want to apply.
  - ❖ The condition can be specified using a column name or a DAX expression.

- **Example:**
  - ➔ Suppose you have a sales table that includes a column for sales amount and you want to calculate the total sales amount for the months of January and February.
  - ➔ You can use the FILTER function as follows:

    Total Sales = CALCULATE(SUM(Sales[Sales Amount]), FILTER(Sales, MONTH(Sales[Sales Date]) <= 2))

    -This will return the sum of the sales amount column, but only for the rows where the sales date is in January or February.

Basic DAX Functions

If else and Nested If Blocks

If-Else

- In Power BI, the IF-ELSE statement is used to evaluate a condition and return a value based on whether the condition is true or false.
- **Syntax:**

  IF(<condition>, <value if true>, <value if false>)

  The first argument is the condition that you want to evaluate, and the second and third arguments are the values that you want to return based on whether the condition is true or false.
- **Example:**
  - ➔ Suppose you have a sales table that includes a column for sales amount, and you want to create a new column that categorizes the sales amount as "High" or "Low" based on a threshold value.
  - ➔ You can use the IF-ELSE statement as follows:

    Sales Category = IF(Sales[Sales Amount] > 1000, "High", "Low")

    -This will return the value "High" if the sales amount is greater than 1000, and "Low" otherwise.

Nested If Blocks

- Nested IF blocks can be used to evaluate multiple conditions and return different values based on each condition.
- **Syntax:**

  IF(<condition1>, <value if true1>, IF(<condition2>, <value if true2>, <value if false2>))

  Explanation:
  - Here, the second IF statement is nested within the first IF statement.
  - If the first condition is true, then the value specified in the second argument will be returned.
  - Otherwise, the second IF statement is evaluated, and if the second condition is true, the value specified in the first argument of the second IF statement will be returned.
  - Otherwise, the value specified in the third argument of the second IF statement will be returned.
- **Example:**
  - ➔ Suppose you have a sales table that includes a column for sales amount, and you want to categorize the sales amount as "High", "Medium", or "Low" based on different threshold values.
  - ➔ You can use nested IF statements as follows:

    Sales Category = IF(Sales[Sales Amount] > 1000, "High", IF(Sales[Sales Amount] > 500, "Medium", "Low"))

    -This will return "High" if the sales amount is greater than 1000, "Medium" if it is between 500 and 1000, and "Low" otherwise.

Basic DAX Functions

Time Intelligence Function

- Time Intelligence functions in Power BI allow you to perform calculations and analysis on time-based data, such as dates, months, quarters, and years.
- Some of the commonly used time intelligence functions in Power BI are:

TOTALYTD:

- Calculates the total value of an expression from the beginning of the year up to the specified date.
- **Syntax:**

  TOTALYTD(<expression>, <date_column>)
- **Example:**
  - Suppose you have a sales table that includes columns for sales amount and sales date, and you want to calculate the total sales amount for each year.
  - You can use the TOTALYTD function as follows:

    Total Sales YTD = TOTALYTD(SUM(Sales[Sales Amount]), Sales[Sales Date])

TOTALMTD:

- Calculates the total value of an expression from the beginning of the month up to the specified date.
- **Syntax:**

  TOTALMTD(<expression>, <date_column>)
- **Example:**
  - Suppose you have a sales table that includes columns for sales amount and sales date, and you want to calculate the total sales amount for each month.
  - You can use the TOTALMTD function as follows:

    Total Sales MTD = TOTALMTD(SUM(Sales[Sales Amount]), Sales[Sales Date])

SAMEPERIODLASTYEAR:

- The SAMEPERIODLASTYEAR function returns a table that contains the corresponding date or time period from the previous year for each date or time period in the specified column.
- Syntax:

  SAMEPERIODLASTYEAR(<date column>)
- Example:
  - Suppose you have a sales table that includes a column for sales amount and a column for sales date, and you want to compare the sales amount for each date to the sales amount for the same date in the previous year.
  - You can use the SAMEPERIODLASTYEAR function as follows:

    Sales YoY Change = CALCULATE(SUM(Sales[Sales Amount]), SAMEPERIODLASTYEAR(Sales[Sales Date]))

    -This will return the sum of the sales amount column, but only for the rows where the sales date is the same date in the previous year.

Basic DAX Functions

TOTALQTD

- Calculate the total value of a measure from the beginning of the quarter up to and including the current date.
- **Syntax:**

  TOTALQTD(<expression>, <date_column>)
- **Example:**
  - Suppose you have a sales table that includes columns for sales amount and a column for sales date, and you want to calculate the total sales amount for the current quarter.
  - You can use the TOTALYTD function as follows:

    Total Sales QTD = TOTALQTD(SUM(Sales[Sales Amount]), Sales[Sales Date])

PREVIOUSMONTH:

- The PREVIOUSMONTH function in Power BI is used to return the first date of the previous month based on a specified date.
- **Syntax:**

  PREVIOUSMONTH(<date>)
- **Example:**
  - Suppose you have a sales table that includes columns for sales amount and a column for sales date, and you want to create a measure that calculates the sales amount for the previous month.
  - You can use the PREVIOUSMONTH function as follows:

    Previous Month Sales = CALCULATE(SUM(Sales[Sales Amount]), Sales[Sales Date] = PREVIOUSMONTH(TODAY()))

    -This will return the sum of the sales amount column, but only for the rows where the sales date is the first date of the previous month based on today's date.

NEXTMONTH:

- The NEXTMONTH function in Power BI is used to return the first date of the next month based on a specified date.
- **Syntax:**

  NEXTMONTH(<date>)
- **Example:**
  - Suppose you have a sales table that includes a column for sales amount and a column for sales date, and create a measure that calculates the sales amount for the next month.
  - You can use the NEXTMONTH function as follows:

    Next Month Sales = CALCULATE(SUM(Sales[Sales Amount]), Sales[Sales Date] = NEXTMONTH(TODAY()))

    -This will return the sum of the sales amount column, but only for the rows where the sales date is the first date of the next month based on today's date.

Basic DAX Functions

X Functions

- "X functions" are functions that iterate over a table or a column, such as SUMX, AVERAGEX, MINX, and MAXX.
- These functions perform a calculation for each row in a table or column and then aggregate the results into a final value.
- X functions are particularly useful when working with tables and need to perform calculations at a granular level.

  Example:

  Total Sales SUMX =SUMX('Sales OrderDetails','Sales OrderDetails'[qty]'Sales OrderDetails'[unitprice])*

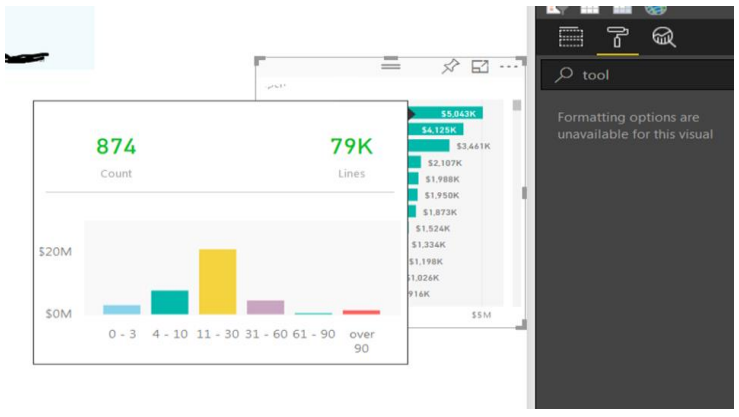
Non-X Functions

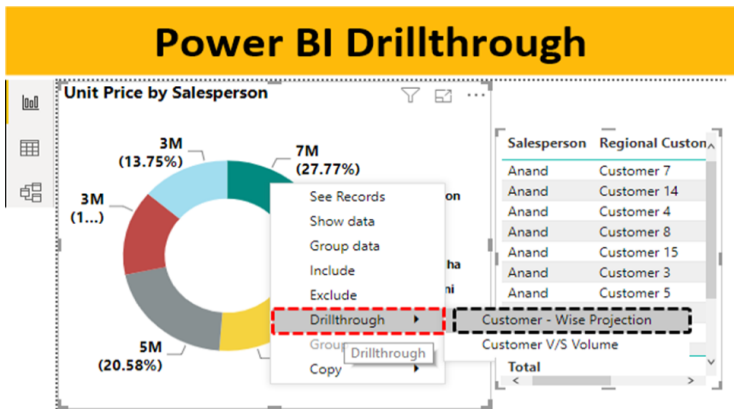
- "Non-x functions," on the other hand, are functions that do not iterate over a table or column, such as SUM, AVERAGE, MIN, and MAX.
- These functions perform calculations on a set of values without iterating over each row or column.
- Non-x functions are useful for working with individual values or groups of values.

  Example:

  Total Sales =sum('Sales OrderDetails'[Order Line Total])

Tool Tips & Drill Throughs





In Power BI, Tooltips provide additional information about data points on a visual, without cluttering the visual with too much information. Drill down allows users to view more detailed information about a specific data point in a visual.

To create a tooltip, follow these steps:

- Select the visual that you want to add a tooltip to.
- Click on the ellipsis (...) in the visual's top right corner and select "Tooltip" from the dropdown menu.
- Add the fields you want to include in the tooltip in the "Tooltip" pane.

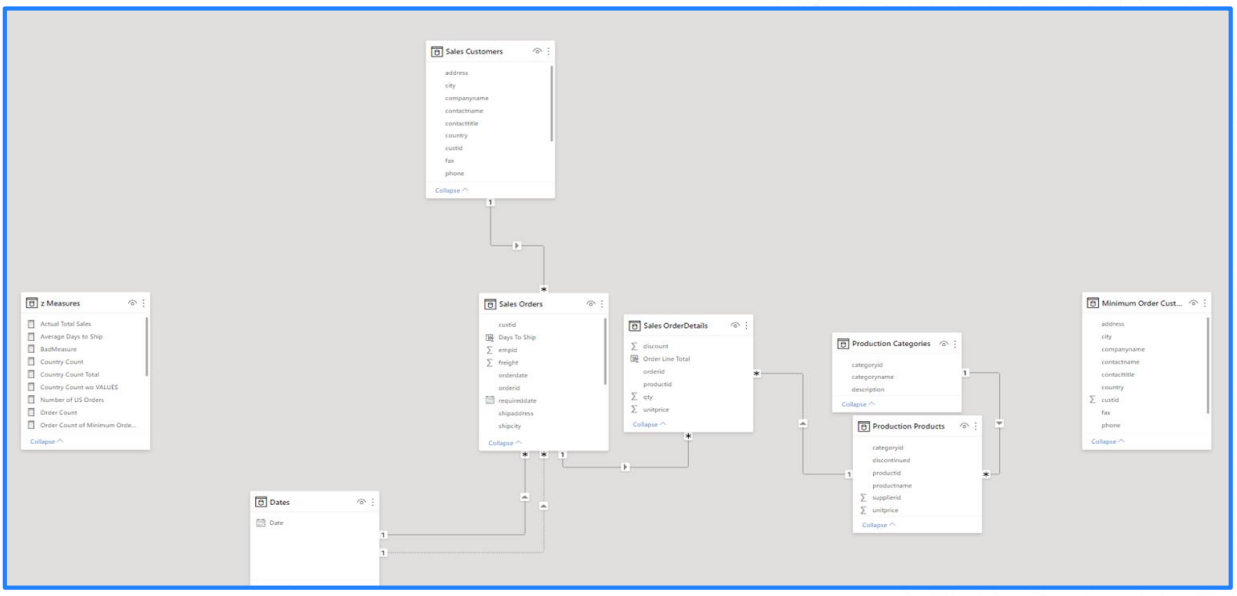
To enable drill down, follow these steps:

- Select the visual that you want to enable drill down for.
- Click on the ellipsis (...) in the visual's top right corner and select "Drill down" from the dropdown menu.
- Add the fields you want to allow users to drill down on in the "Drill down" pane.

Power BI Relationships and KPI

Power BI Relationships

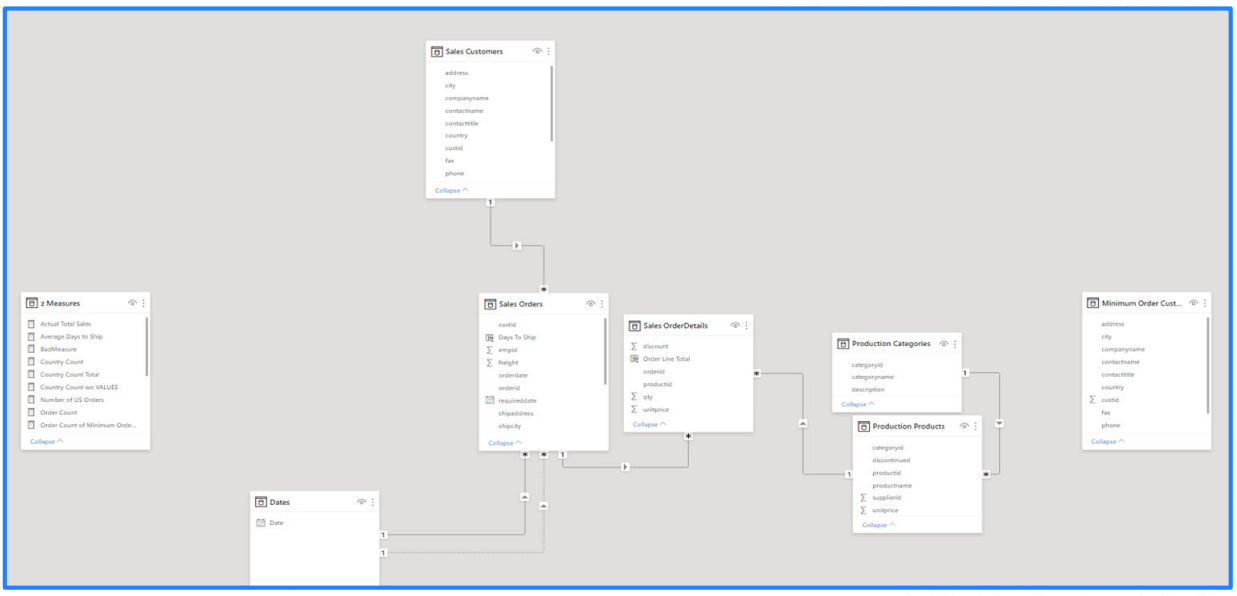
Relations in Power BI



- In Power BI, relationships are used to connect different tables of data together based on common fields or columns.
- By creating relationships between tables, you can combine data from multiple sources and create more comprehensive reports and visualizations.
- In Power BI, there are three types of relationships you can create between tables:
  - **One-to-many (1:N) relationship:**
    - This is the most common type of relationship, where one row in the primary table can have multiple related rows in the related table.
    - For example, one customer can have multiple orders, so you would create a one-to-many relationship between the customer table and the orders table.
  - **Many-to-one (N:1) relationship:**
    - This relationship is the inverse of the one-to-many relationship, where multiple rows in the related table can be related to a single row in the primary table.
    - For example, multiple products can belong to a single category, so you would create a many-to-one relationship between the products table and the categories table.
  - **Many-to-many (N:N) relationship:**
    - This type of relationship occurs when multiple rows in the primary table can be related to multiple rows in the related table.
    - However, Power BI does not allow you to create a direct many-to-many relationship.
    - Instead, you must create a bridge table that connects the two tables through a one-to-many relationship.
    - For example, if you have a table of customers and a table of products, and each customer can have multiple favorite products, you would create a bridge table that includes the customer ID and product ID fields, and then create one-to-many relationships between the bridge table and both the customers and products tables.

Power BI Relationships

Relations in Power BI



To create a relationship in Power BI, follow these steps:

1. Open your Power BI report and select "Data" from the left-hand pane.
2. Select the table you want to use as the primary table (i.e., the one you want to build the relationship from).
3. Click on "Manage relationships" under the "Table Tools" section of the ribbon.
4. Click "New" to create a new relationship.
5. Choose the related table from the dropdown list.
6. Select the related field or column from the related table.
7. Choose the related field or column from the primary table.
8. Choose the type of relationship (e.g., one-to-many, many-to-one, etc.).
9. Click "OK" to create the relationship.

Key Performance Indicator

- KPI stands for Key Performance Indicator, which is a measurable value that helps you track progress towards a specific business goal or objective.
- In Power BI, KPIs can be used to visually represent this progress and highlight areas that require attention.

To create a KPI in Power BI, follow these steps:

1. Open your Power BI report and select "Data" from the left-hand pane.
2. Select the table that contains the data for your KPI.
3. Click on "New measure" under the "Modeling" section of the ribbon.
4. Enter a name for your measure (e.g., "Sales KPI").
5. Write the formula for your KPI. This could be a simple calculation, such as a sum or average, or a more complex calculation based on multiple measures or tables.
6. Choose the target value for your KPI. This is the value that represents the goal or objective you are tracking.
7. Choose the trend axis for your KPI. This is the direction that indicates whether your KPI is trending positively or negatively.
8. Choose the status threshold values for your KPI. These are the values that indicate whether your KPI is on track, in danger, or off track.
9. Choose the visual format for your KPI. You can choose from a variety of visualizations, such as a gauge, card, or indicator.

- Once you have created a KPI, you can add it to your report and use it to track progress towards your business goals.
- The KPI will update dynamically based on changes to your underlying data, allowing you to monitor performance in real-time.

Administration and Security in Power BI

Administration in Power BI

In Power BI, there are four types of roles for users, each with a different set of permissions and responsibilities: Admin, Member, Contributor, and Viewer.

Admin:

- The Admin role has full control over the Power BI environment.
- They can create, delete, and manage workspaces, add or remove users, assign roles, set up data sources and gateways, and manage security and compliance settings.
- They also have access to all data and reports within the environment.
- Only a few users should have the Admin role, as they have complete control over the organization's data.

Member:

- Members are users who have access to the workspaces and content within the environment, but they do not have administrative privileges.
- They can view and edit reports, dashboards, and datasets within their assigned workspaces and can collaborate with other members.
- They cannot add or remove users, create new workspaces, or manage security settings.

Contributor:

- Contributors have more limited permissions than Members.
- They can create, edit and publish reports and dashboards, but they cannot create new workspaces or manage security settings.
- They have access to the content within their assigned workspace but cannot view or edit other workspaces.

Viewer:

- Viewers are users who can only view reports and dashboards.
- They cannot edit or publish any content and do not have access to the Power BI service to create or manage content.
- They can only view and interact with content that is shared with them.

It's important to note that users can have multiple roles within a single environment, depending on the access they require. For example, a user might be a Member in one workspace and a Viewer in another.

Overall, the different roles in Power BI enable organizations to control and manage access to their data and ensure that users have the appropriate permissions to complete their work efficiently and securely.

Security in Power BI

- ❖ Row-Level Security (RLS) is a feature in Power BI that enables you to restrict data access based on a user's role or permission level.
- ❖ RLS allows you to control what data users can see, interact with, and modify within a report or dashboard.

To implement RLS in Power BI, follow these steps:

- **Define your roles:** Define the roles for your users, such as sales reps, managers, or regional directors.
- **Define your rules:** Define the rules that control what data each role can see. For example, you might limit a sales rep to only seeing data for their territory, or a manager to only seeing data for their team.
- **Implement your rules:** Implement your RLS rules by creating DAX expressions that define the rules for each role. These expressions will be used to filter the data that each user sees.
- **Test your rules:** Test your RLS rules by previewing the report or dashboard as each role. Ensure that each role only sees the data that they are supposed to see.

Here's an example of how to implement RLS in Power BI:

1. **Create a new role:** Click on the Manage Roles option in the Modeling tab and create a new role, such as "Sales Rep".
2. **Define your rules:** Create a DAX expression that filters the data based on the Sales Rep's territory. For example: =Sales[Territory]=USERPRINCIPALNAME()
3. **Implement your rules:** Apply the Sales Rep role to the report or dashboard and ensure that the Sales Rep only sees data related to their territory.
4. **Test your rules:** Preview the report or dashboard as the Sales Rep and ensure that they only see data related to their territory.

- ❖ By implementing RLS in Power BI, you can ensure that your users only see the data that they are authorized to see.
- ❖ This can help you to maintain data security and privacy, and ensure that your reports and dashboards are accurate and reliable.

Formatting and Best Practices in Power BI

Formatting Options

Power BI provides a wide range of formatting options that allow you to customize the appearance of your reports and dashboards.
Here are some of the formatting options available in Power BI:

Visual formatting: You can customize the appearance of each visual element in your report, including fonts, colors, borders, backgrounds, and shapes. You can also add and format images and icons.

Data formatting: You can format the data within each visual element to improve its readability, including decimal places, currency symbols, percentages, date formats, and time formats.

Page formatting: You can customize the layout and formatting of each report page, including backgrounds, margins, page orientation, and page size.

Theme formatting: You can apply a pre-designed theme to your report or dashboard to quickly change the colors, fonts, and other formatting options across all visuals.

Conditional formatting: You can set rules that apply formatting changes based on data values, such as highlighting cells that meet certain criteria or displaying data bars that show relative values.

Drill-through formatting: You can customize the behavior of drill-through actions, such as controlling what data is displayed in the drill-through page, and formatting the appearance of the drill-through page.

Report-level formatting: You can apply formatting changes to the entire report, such as setting a background color, adding a logo or title, and adjusting margins.

Overall, the formatting options in Power BI allow you to create visually appealing and informative reports and dashboards that are easy to read and understand. By using formatting options effectively, you can improve the user experience and make your data more accessible and actionable.

Best Practices

Here are some best practices to follow when working with Power BI:

- **Plan your data model:** Invest time in planning your data model before creating your report or dashboard. Ensure that your data is structured and organized correctly, and that relationships between tables are set up properly.

- **Use measures instead of calculated columns:** Avoid using calculated columns where possible, as they can slow down performance. Instead, use measures to calculate data on the fly and improve the speed of your report.

- **Use visuals effectively:** Choose visuals that effectively communicate the data you want to show, and ensure they are formatted appropriately. Use colors, text, and other design elements to make your visuals easy to read and understand.

- **Optimize performance:** Minimize the number of visuals on a single page to improve performance. Use filters and slicers to limit the amount of data displayed in each visual, and avoid using unnecessary visuals or elements.

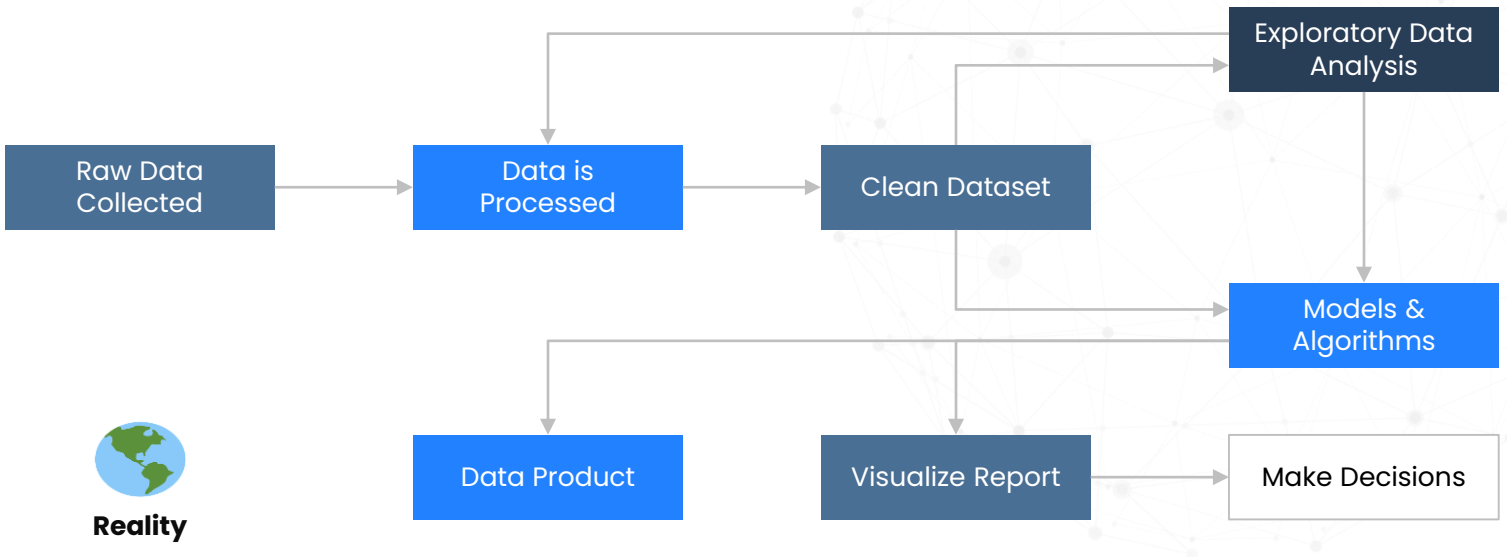
- **Use RLS for security:** Implement row-level security to ensure that users only see the data they are authorized to see. This can help you maintain data security and privacy.

- **Use themes:** Use themes to ensure that your report or dashboard has a consistent look and feel, and to save time when applying formatting changes.

Exploratory Data Analysis

Exploratory Data Analysis

Data Analytics Process

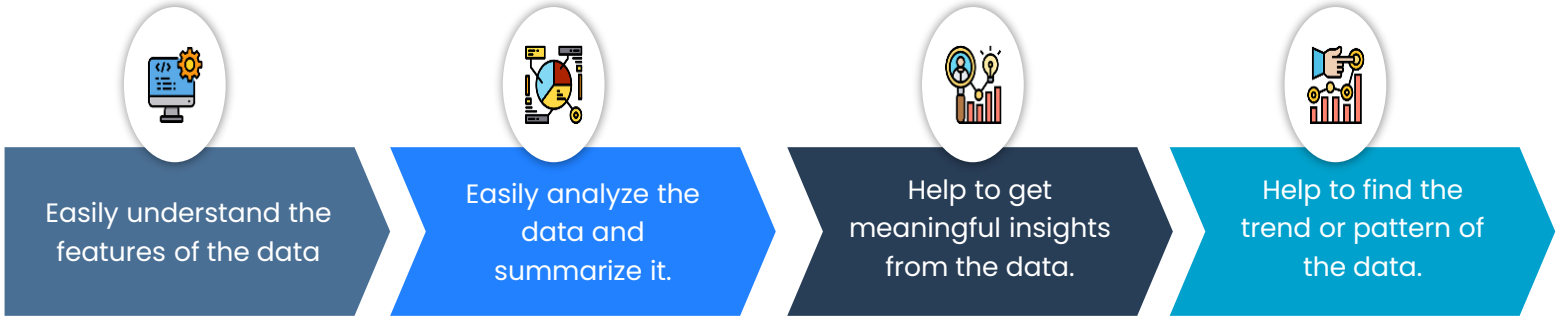


What is EDA?

- Exploratory Data Analysis is an approach to analyze the datasets to summarize their main characteristics in form of visual methods.
- EDA is nothing but a data exploration technique to understand various aspects of the data.
- The main aim of EDA is to obtain confidence in a data to an extent where we are ready to engage a machine learning model.
- EDA is important to analyze the data it's a first steps in data analysis process.
- EDA give a basic idea to understand the data and make sense of the data to figure out the question you need to ask and find out the best way to manipulate the dataset to get the answer to your question.
- Exploratory data analysis help us to finding the errors, discovering data, mapping out data structure, finding out anomalies.
- Exploratory data analysis is important for business process because we are preparing dataset for deep thorough analysis that will detect you business problem.
- EDA help to build a quick and dirty model, or a baseline model, which can serve as a comparison against later models that you will build.

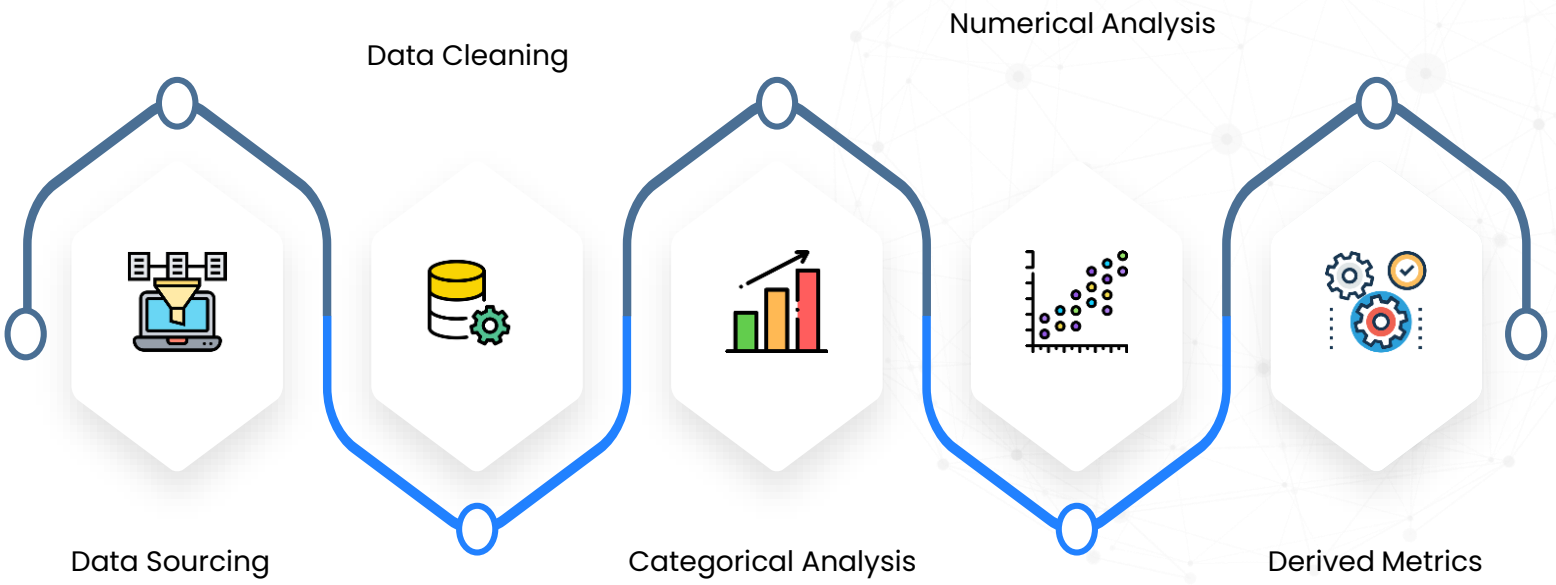
Visualization

Visualization is the presentation of the data in the graphical or visual form to understand the data more clearly. Visualization is easy to understand the data



Exploratory Data Analysis

Steps in EDA



Data Sourcing

- Data Sourcing is the process of gathering data from multiple sources as external or internal data collection.
- There are two major kind of data which can be classified according to the source:

1. **Public data**
    - The data which is easy to access without taking any permission from the agencies is called public data. The agencies made the data public for the purpose of the research,
    - **Example:** government and other public sector or ecommerce sites made the data public.

1. **Private data**
    - The data which is not available on public platform and to access the data we have to take the permission of organisation is called private data.
    - **Example:** Banking ,telecom ,retail sector are there which not made their data publicly available.

Data Cleaning

- After collecting the data , the next step is data cleaning. Data cleaning means that you get rid of any information that doesn't need to be there and clean up by mistake.
- Data Cleaning is the process of clean the data to improve the quality of the data for further data analysis and building a machine learning model.
- Data Cleaning is the process of clean the data to improve the quality of the data for further data analysis and building a machine learning model.
- The following are some steps involve in Data Cleaning:
    - Handling Missing Values
    - Standardization of Data
    - Handling Invalid Data
    - Outlier Treatment

Exploratory Data Analysis

Handling Missing Values

- **Delete Rows/Columns**
  - This method we commonly used to handle missing values. Rows can be deleted if it has insignificant number of missing value Columns can be delete if it has more than 75% of missing value
- **Replacing with mean/ median/mode**
  - This method can be used on independent variable when it has numerical variables.
  - On categorical feature we apply mode method to fill the missing value.
- **Algorithm Imputation**
  - Some machine learning algorithm supports to handle missing value in the datasets. Like KNN, Naïve Bayes, Random forest.
- **Predicting the missing values**
  - Prediction model is one of the advanced method to handle missing values.
  - In this method dataset with no missing value become training set and dataset with missing value become the test set and the missing values is treated as target variable.

Standardization/Feature Scaling

- Feature scaling is the method to rescale the values present in the features. In feature scaling we convert the scale of different measurement into a single scale.
- It standardize the whole dataset in one range.
- **Importance of Feature Scaling**

  When we are dealing with independent variable or features that differ from each other in terms of range of values or units of the features, then we have to normalize/standardize the data so that the difference in range of values doesn't affect the outcome of the data.
- Methods:
  - **Standard Scaler**
    - Standard scaler ensures that for each feature, the mean is zero and the standard deviation is 1, bringing all feature to the same magnitude.
    - In simple words Standardization helps you to scale down your feature based on the standard normal distribution.

$$Z = \frac{x - \mu}{\sigma}$$

Score, Mean, SD

  - **Min-Max Scaler**
    - Normalization helps you to scale down your features between a range 0 to 1.

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Exploratory Data Analysis

Outlier Treatment

- Outliers are the most extremes values in the data.
- It is an abnormal observations that deviate from the norm.
- Outliers do not fit in the normal behavior of the data.
- Outliers can be detected using the following methods:
  - Boxplot
  - Histogram
  - Scatter plot
  - Z-score
  - Interquartile range(values out of 1.5 time of IQR)
- Outliers can be handled using one of the following methods:
  - Remove the outliers.
  - Replace outlier with suitable values by using following methods:-
  - Quantile method
  - Interquartile range
  - Use that ML model which are not sensitive to outliers
  - Like:-KNN, Decision Tree, SVM, Naïve Bayes, Ensemble methods.

Handle Invalid Value:

- **Encode Unicode properly**
  In case the data is being read as junk characters, try to change encoding, E.g. CP1252 instead of UTF-8.
- **Convert incorrect data types**
  - Correct the incorrect data types to the correct data types for ease of analysis.
  - E.g. if numeric values are stored as strings, it would not be possible to calculate metrics such as mean, median, etc.
  - Some of the common data type corrections are — string to number: "12,300" to “12300”; string to date: "2013-Aug" to “2013/08”; number to string: “PIN Code 110001” to "110001"; etc.
- **Correct values that go beyond range**
  - If some of the values are beyond logical range, e.g. temperature less than -273° C (0° K), you would need to correct them as required.
  - A close look would help you check if there is scope for correction, or if the value needs to be removed.

- **Correct wrong structure**
  - Values that don’t follow a defined structure can be removed.
  - E.g. In a data set containing pin codes of Indian cities, a pin code of 12 digits would be an invalid value and needs to be removed. Similarly, a phone number of 12 digits would be an invalid value

Exploratory Data Analysis

Types of Data

- In exploratory data analysis (EDA), data can be classified into different types based on their nature and characteristics. The common types of data in EDA are:
    - **Qualitative**

      A variable which able to describe quality of the population. (Categorical values)
        - **Nominal**
            - It represent qualitative information without order. Value represent a discrete units.
            - Like Gender: Male/Female ,Eye colour.
        - **Ordinal**
            - It represent qualitative information with order. It indicate the measurement classification are different and can be ranked.
            - Lets say Economic status: high/ medium /low which can ordered as low, medium, high.
    - **Quantitative**

      A variable which quantify the population (Numerical values)
        - **Discrete**
            - It has a discrete value that means it take only counted value not a decimal values. Like count of student in class
        - **Continuous**
            - A number within a range of a value is usually measured, such as height

Types of Analysis

Univariate analysis: This type of analysis involves examining the distribution and summary statistics of a single variable. It can help identify outliers, missing values, and patterns in the data.

Bivariate analysis: This type of analysis involves exploring the relationship between two variables. It can help identify correlations, dependencies, and potential cause-and-effect relationships.

Multivariate analysis: This type of analysis involves examining the relationship between three or more variables. It can help identify complex patterns, interactions, and dependencies in the data.

Numerical analysis:

- We also perform various analysis over Numerical data.
- For example, dealing with a single numerical variable, we might be interested in knowing their statistical information such as mean, median, 25th Percentile, 75th Percentile, min, max etc.
- Similarly, while analyzing multiple features, we might be interested in knowing their correlation with each other.

Exploratory Data Analysis

Derived Metrics

- Derived metrics create a new variable from the existing variable to get a insightful information from the data by analyzing the data.
- Derived metrics can provide additional insight and understanding beyond what is available in the original data, helping analysts to identify patterns, trends, and relationships that may not be immediately apparent.

Feature Binning

- Feature binning and feature encoding are two common techniques used in data preprocessing to transform categorical or continuous variables into numerical features that can be used by machine learning algorithms.
- Feature binning, also known as discretization, is the process of dividing continuous numerical variables into a finite number of bins or categories.
- This is often done to reduce noise and handle outliers, and to make the data more interpretable.
- There are several methods of binning, such as equal-width binning, equal-frequency binning, and equal width binning.
    - Equal Frequency Binning
        - Equal frequency separate the continuous variable into several categories having approximately same number of values.
    - Equal Width Binning
        - Equal width separate the continuous variable to several categories having same range of width.

Feature Encoding

- Feature encoding, on the other hand, is the process of converting categorical variables into numerical features that can be used by machine learning algorithms.
- This is often done because many machine learning algorithms cannot handle categorical variables directly.
- There are several methods of encoding, such as one-hot encoding, and label encoding.
    - **One-Hot Encoding**
        - One-hot encoding is a common method of encoding categorical variables.
        - In this method, each category is assigned a binary variable, and the variable takes a value of 1 if the sample belongs to that category and 0 otherwise.
    - **Label Encoding**
        - Label encoding is technique to transform categorical variables into numerical variables by assigning a numerical value to each of the categories.

Thank you