**🡪 Data visualization** is the practice of translating information into a visual context, such as a map or graph, to make data easier for the human brain to understand and pull insights from.

It is the representation of information and data through use of common graphics, such as charts, plots, infographics, and animations. Data visualization is a powerful way for people, especially data professionals, to display data so that it can be interpreted easily.

Benefits of Data Visualization:

1. It is easy to understand the information with graphics

2. It made data to be represented in attractive way

3. Shows complex relationships

4. Helps to process large datasets

5. Useful for identifying trends

6. Minimizes ambiguity

🡪 The following are some common types of data visualizations:

• **Table:** A table is data displayed in rows and columns, which can be easily created in a Word document or Excel spreadsheet.

• **Chart or graph:** Information is presented in tabular form with data displayed along an x and y axis, usually with bars, points, or lines, to represent data in comparison.

• **Geospatial visualization**: Data is depicted in map form with shapes and colours that illustrate the relationship between specific locations, such as a choropleth or heat map.

• **Dashboard**: Data and visualizations are displayed, usually for business purposes, to help analysts understand and present data.

🡪 Following are some of the basic data visualization plots

1. Line plots

2. Area plots

3. Histograms

4. Bar charts

5. Pie charts

6. Box plots

7. Scatter plots

**Line Plots:** A line plot is used to represent quantitative values over a continuous interval or time period. It is generally used to depict trends on how the data has changed over time.

**Histograms:** Histograms represents the frequency distribution of a dataset. It is a graph showing the number of observations within each given interval.

**Bar Charts**: A Bar chart or bar graph is a chart or graph that presents categorical data with rectangular bars with heights or lengths proportional to the values that they represent. A bar plot is a way of representing data where the length of the bars represents the magnitude/size of the feature/variable.

**Pie Charts**: A Pie chart is a circular statistical chart, which is divided into sectors to illustrate numerical proportion.

**Box Plots**: A Box plot (or box-and-whisker plot) shows the distribution of quantitative data in a way that facilitates comparisons between variables or across levels of a categorical variable. Box plot shows the quartiles of the dataset while the whiskers extend encompass the rest of the distribution but leave out the points that are the outliers.

**Scatter Plots**: A Scatter chart, also called a scatter plot, is a chart that shows the relationship between two variables.

* **Matplotlib** is an amazing visualization library in [**Python**](https://www.geeksforgeeks.org/python-programming-language/)for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on Numpy arrays

**Types of Matplotlib**

Matplotlib comes with a wide variety of plots. Plots help to understand trends, and patterns, and to make correlations. They’re typically instruments for reasoning about quantitative information. Some of the sample plots are

* Matplotlib Line Plot
* Matplotlib Bar Plot
* Matplotlib Histograms Plot
* Matplotlib Scatter Plot
* Matplotlib Pie Charts
* **Seaborn** is an amazing visualization library for statistical graphics plotting in Python. It provides beautiful default styles and color palettes to make statistical plots more attractive. It is built on top matplotlib library and is also closely integrated with the data structures from [pandas](https://www.geeksforgeeks.org/introduction-to-pandas-in-python/).

Seaborn divides the plot into the below categories –

* **Relational plots:** This plot is used to understand the relation between two variables.
* [**Categorical plots:**](https://www.geeksforgeeks.org/seaborn-categorical-plots/)This plot deals with categorical variables and how they can be visualized.
* [**Distribution plots:**](https://www.geeksforgeeks.org/seaborn-distribution-plots/)This plot is used for examining univariate and bivariate distributions
* [**Regression plots:**](https://www.geeksforgeeks.org/seaborn-regression-plots/)The regression plots in Seaborn are primarily intended to add a visual guide that helps to emphasize patterns in a dataset during exploratory data analyses.
* [**Matrix plots:**](https://www.geeksforgeeks.org/ml-matrix-plots-in-seaborn/) A matrix plot is an array of scatterplots.
* **Multi-plot grids:**It is a useful approach to draw multiple instances of the same plot on different subsets of the dataset.
* **Bokeh**is a data visualization library in Python that provides high-performance interactive charts and plots. Bokeh output can be obtained in various mediums like notebook, html and server.

Bokeh provides two visualization interfaces to users:

**bokeh.models** : A low level interface that provides high flexibility to application developers.

**bokeh.plotting** : A high level interface for creating visual glyphs.

* The **Plotly Python**library is an interactive open-source library. This can be a very helpful tool for data visualization and understanding the data simply and easily. plotly graph objects are a high-level interface to plotly which are easy to use. It can plot various types of graphs and charts like scatter plots, line charts, bar charts, box plots, histograms, pie charts, etc.

#### **Why do we need plotly when there is matplotlib, seaborn like libraries in python?**

All these three libraries do the same task, which is to plot a graph or chart for data visualization. But the difference because of which the library Plotly stands out are:

* It is visually attractive that can be accepted by a wide range of audiences. Many themes are available in plotly to make your charts and graph more appealing.
* It allows you for the endless customization of your graphs that makes your plot more meaningful and understandable for others.
* Built on top of the Plotly JavaScript library (plotly.js), plotly enables Python users to create beautiful interactive web-based visualizations that can be displayed in Jupyter notebooks, saved to standalone HTML files, or served as part of pure Python-built web applications using Dash. The plotly Python library is sometimes referred to as “plotly.py” to differentiate it from the JavaScript library.
* The plotly graphs are stored in Javascript Object Notation (JSON) data format so that they can be read using scripts of other programming languages such as R, Julia, MATLAB etc.
* The plotly graphs can be exported in various formats such as PNG, SVG, PDF and HTML to your local machine.

🡪 A legend is an area describing the elements of the graph. In the [Matplotlib library](https://www.geeksforgeeks.org/matplotlib-tutorial/), there’s a function called **legend()** which is used to place a legend on the axes.

🡪 **linspace()** has two required parameters, **start** and **stop**, which you can use to set the beginning and end of the range: ex: np.linspace(1, 10, num=10)

🡪 **Distribution Plot:** A distribution plot shows the distribution of a univariate dataset.It often combines a histogram and a KDE(Kernel Density Estimation)plot. Example: Visualizing the distribution of sepal lengths in the Iris dataset.

🡪**Joint Plot:** A joint plot combines multiple plots to visualize the relationship between two variables. It typically includes a scatterplot and marginal histograms or KDE plots.

* **Pair Plot**: A pair plot visualizes pairwise relationships in a dataset by creating scatterplots for all combinations of numeric variables.
* **Rug Plot**: A rug plot place small vertical ticks at datapoints along an axis.  
  🡪 **ViolinPlot**: A violinplot combines a boxplot with a KDEplot to visualize the distribution of data.
* **Strip Plot**: A strip plot displays individual data points along a single axis to show their distribution.
* **Swarm Plot:** :A swarm plot is similar to a strip plot but prevents data points from overlapping.