**VISUALIZATION**

Visualizations is a way of representing data in a graphical form to get the better insight of the data in the form of graphs, charts, maps or plots rather than in the tabular form. It helps in faster decision-making process and identify relationships between variables.

There are many libraries that is used for visualization task like matplotlib, seaborn, plotly, bokeh, ggplot etc.

In this document let us dive deep into the seaborn and matplotlib.

**MATPLOTLIB**

Matplotlib is one of the python library used for 2D creation of static and animated images.

It mainly contains :

1. Pyplot ( from matplotlib.pyplot):

-provides function to create charts and graphs

-most commonly used part of this library

2. Figure:

-the whole image of the

3. Axes:

-the individual plot

There are some common customization tools like color, labels, legend, title, gridlines, tick marks, font sizes etc.. use of these can make our plot more attractive and professional.

It also allows us to create multiple plots in one figure use of the subplot function that is available in matplotlib.

There are some of the basic charts that can be used to plot the raw data to meaningful insights.

**LINE PLOT:**

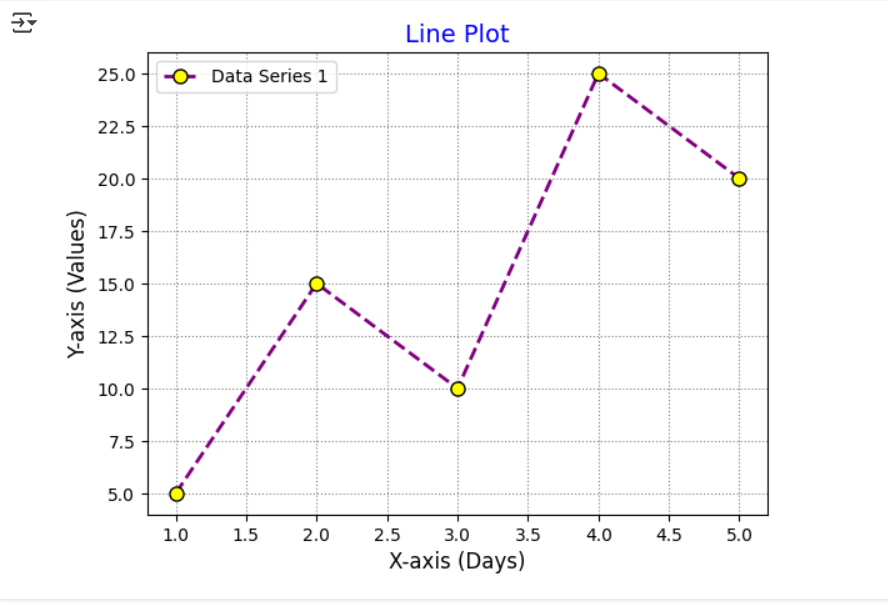
A **line plot** connects data points using **straight lines**. It is mainly used to show trends over a continuous variable, like time or sequence.

It is created using the function plt.plot(x,y).x corresponds to the time, value and y corresponds to the sales, temperature etc.

It contains various features like markers, linestyle, linewidth, line color .

Let’s see an example by applying all these features





**BAR CHART:**

It is used for the comparison categories.

It uses the rectangular bars for the plotting where the length and height depend on the values or the count.

Plotted using a function called plt.bar(x,height)

There are different type of bar charts that include:

-horizontal bar plot(plt.barh)

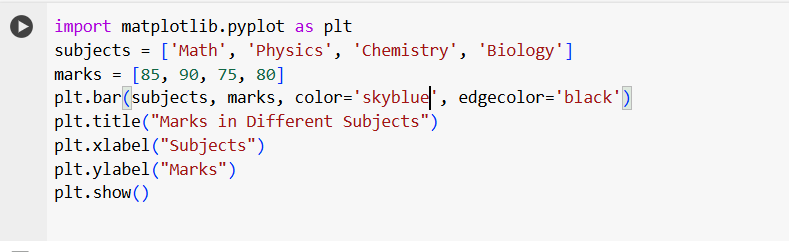
-vertical bar(standard and common one plt.bar)

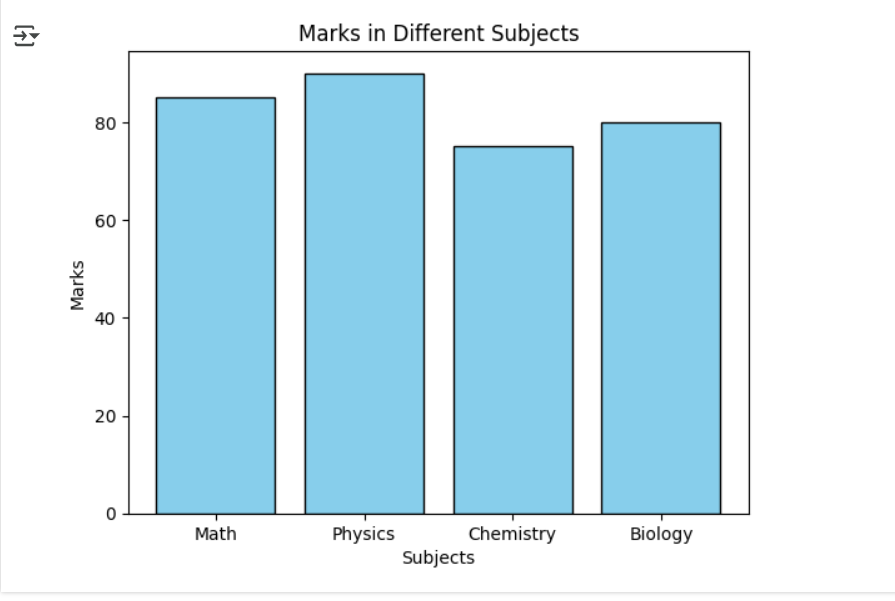
-stacked bar chart(plt.bar with bottom=)

-grouped bar chart(plt.bar)

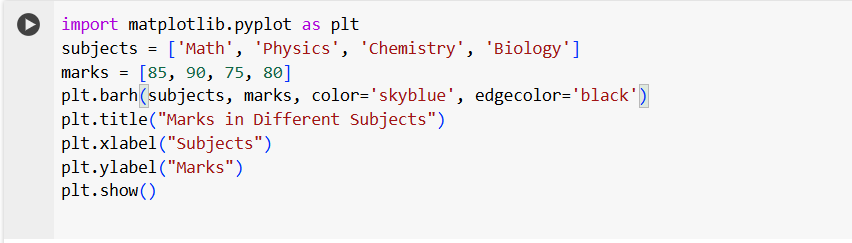
Lets see an example for each bar graph type:

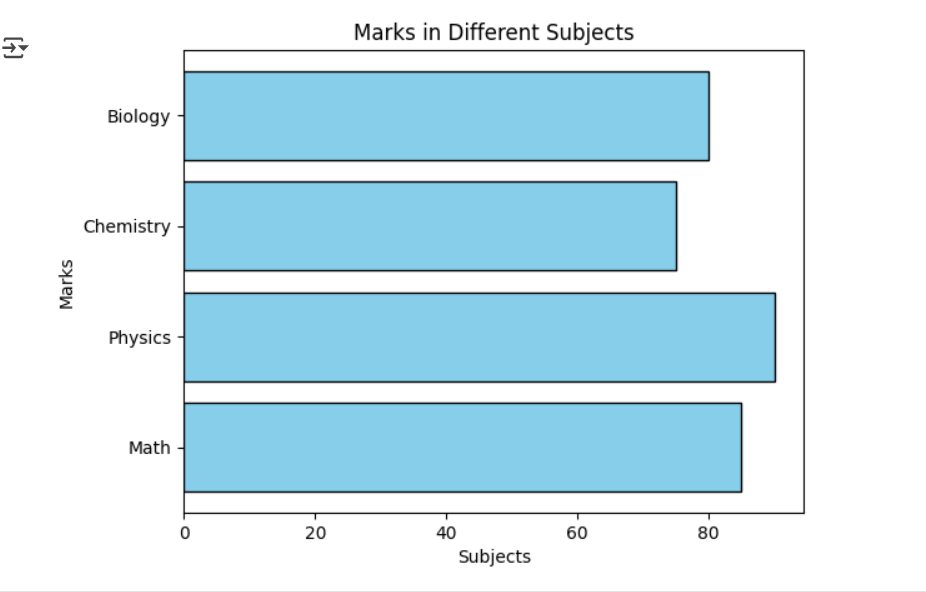
**Vertical Bar**



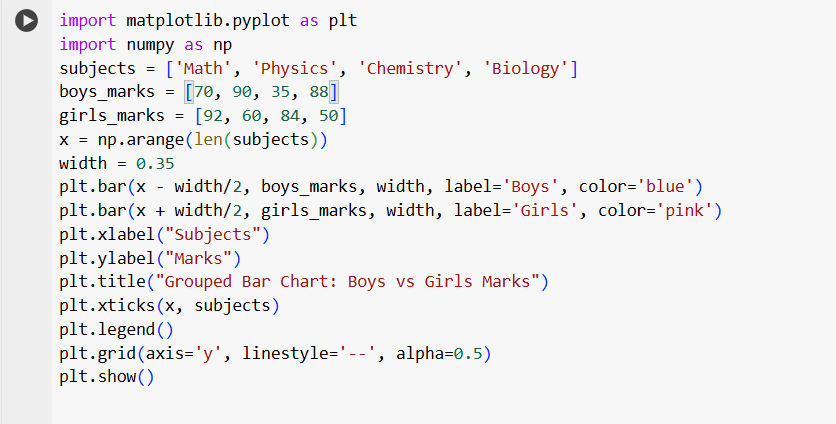


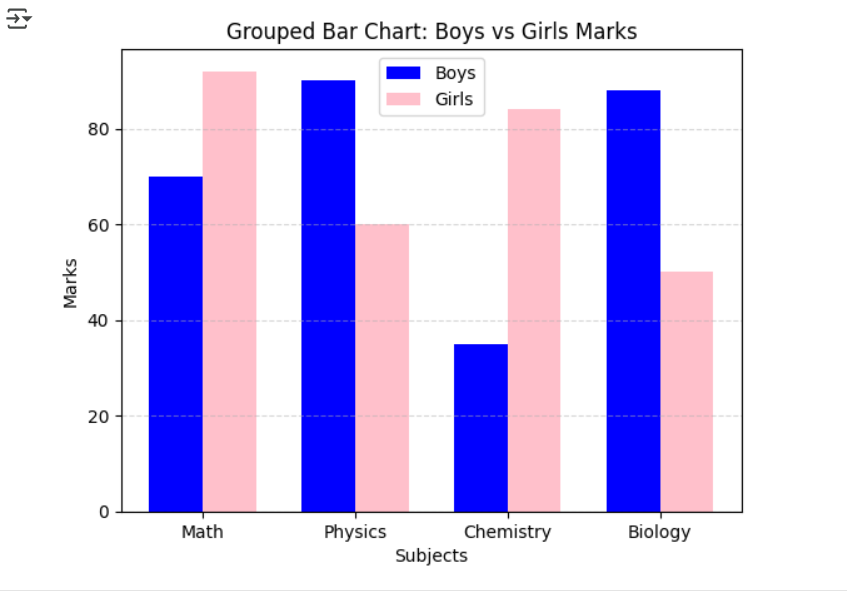
**Horizontal Bar**



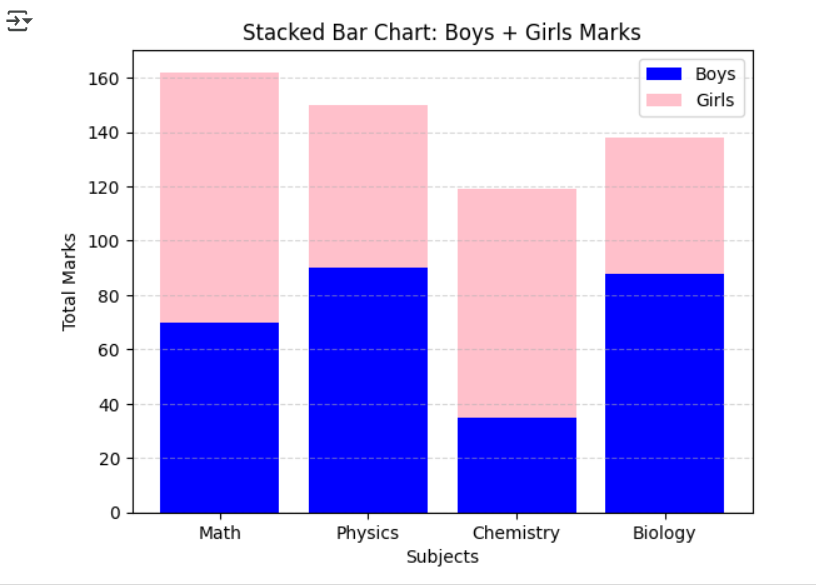


**Grouped bar chart**





**Stacked bar chart**



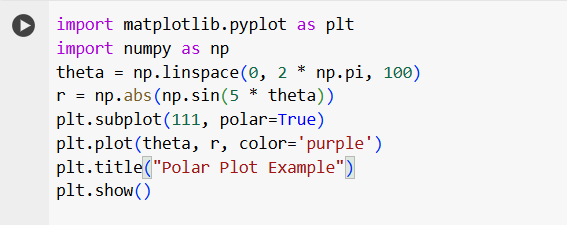
**POLAR PLOT:**

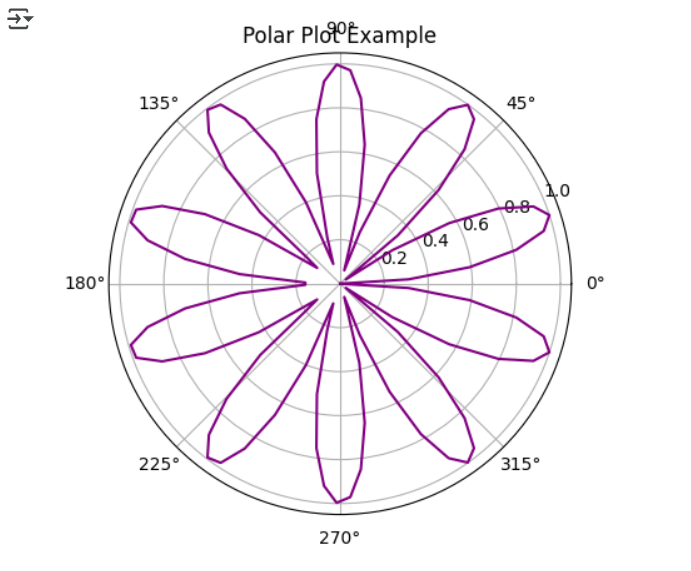
A Polar Plot is a type of plot where data is displayed using polar coordinates instead of the usual X-Y Cartesian coordinates.

* It uses an angle (θ) and a radius (r) to define each point.
* Think of it like plotting in circularspace, where:
  + θ (theta) = angle in degrees or radians (0 to 360° or 0 to 2π)
  + r = distance from the center (radius)

Polar plots are useful when:

* Your data is **cyclical** or involves **angles**.
* You want to represent **directional data** or **periodic functions**.
* Common in: Wind directions & speeds, Radar signals, Sound waves, Trigonometric functions, Electromagnetic fields





**HISTOGRAM CHART:**

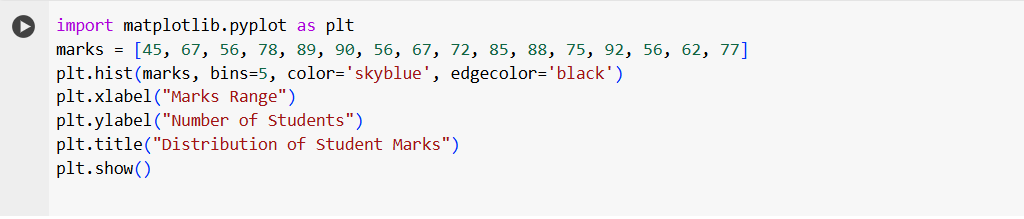
A histogram is a graphical representation that shows how data is distributed over intervals (called bins).

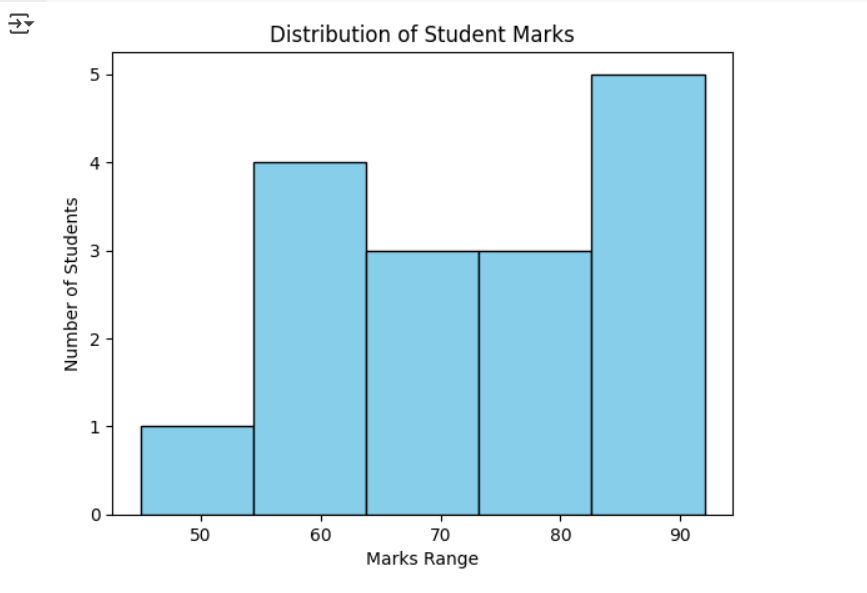
* It looks similar to a bar chart.
* But instead of categories, it shows how many data points fall intospecific ranges.

Gives a quick overview of how your data is spread out.

Helps you decide how to clean or transform data.

Often the first step in Exploratory Data Analysis (EDA)





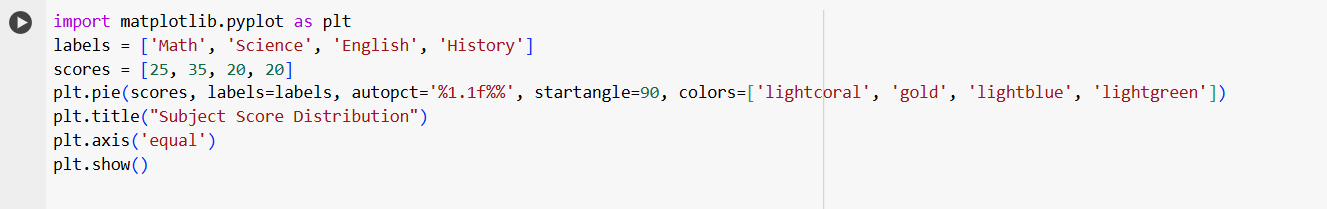
**PIE CHART:**

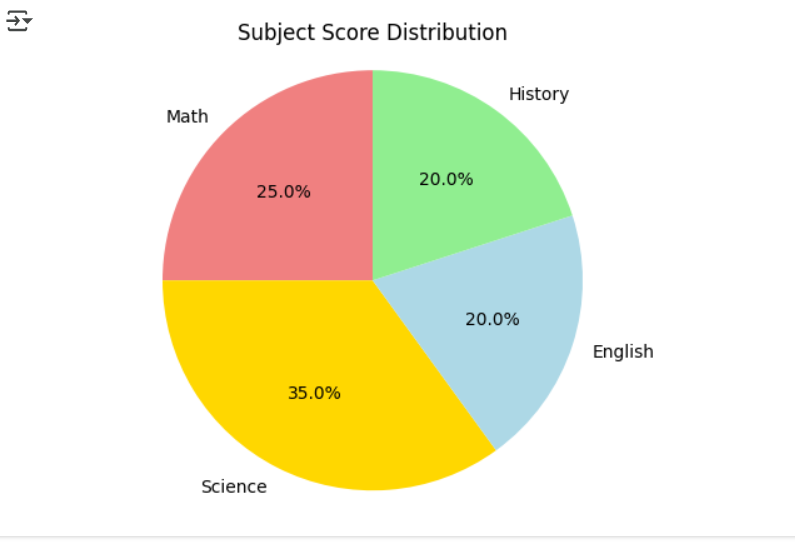
A pie chart is a circular statistical graphic that's divided into slices, where each slice represents a portion (percentage or fraction) of the whole.

* Each wedge = a category
* Size of wedge = proportion of total

It is useful for showing relative percentages of a few categories. Commonly used in business, media, and reports to show simple part-to-whole relationships.

Ideal when you have 3 to 6 categories; too many slices can make it hard to read. Colors are often used to differentiate each slice.





**COMPARISON OF THE ABOVE CHARTS BASED ON HOW AND WHEN TO USE:**

**1. Line Plot**

**When to use**

* To show trends or changes over time
* To visualize continuous data
* Useful in time series and sensor data

**How to use**  
Use plt.plot(x, y)  
Usually with numerical x and y values

**Example use**  
Plotting temperature changes across days

**2. Bar Chart**

**When to use**

* To compare quantities between categories
* Useful for discrete or grouped data
* Can be vertical or horizontal

**How to use**  
Use plt.bar(x, height) or plt.barh() for horizontal  
x = categories, height = values

**Example use**  
Comparing marks of students in different subjects

**3. Polar Plot**

**When to use**

* To plot circular or angular data
* Used in radar data, wind direction, or trigonometric functions
* Great for periodic or directional values

**How to use**  
Use plt.subplot(111, polar=True) and then plt.plot(theta, r)  
theta = angle (radians), r = radius

**Example use**  
Wind direction and speed visualization

**4. Histogram**

**When to use**

* To show the frequency distribution of continuous data
* Helps in understanding data spread and shape
* Useful in data analysis and preprocessing

**How to use**  
Use plt.hist(data, bins=n)  
data = list of numbers, bins = number of intervals

**Example use**  
Visualizing how students' marks are distributed

**5. Pie Chart**

**When to use**

* To show proportion or percentage parts of a whole
* Best for limited categories (ideally 3–6)
* Visually appealing, but not great for precise comparison

**How to use**  
Use plt.pie(sizes, labels=labels, autopct='%.1f%%')  
sizes = values, labels = categories

**Example use**  
Displaying budget distribution in a project

**SEABORN**

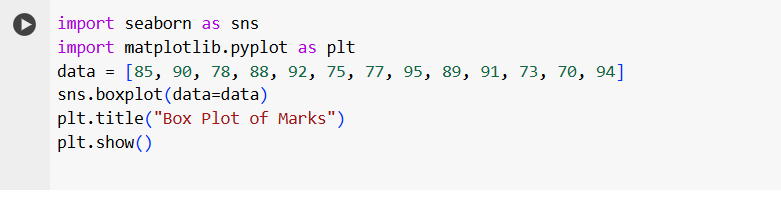
Seaborn is a library for making statistical graphics in Python. It builds on top of [matplotlib](https://matplotlib.org/) and integrates closely with [pandas](https://pandas.pydata.org/) data structures.

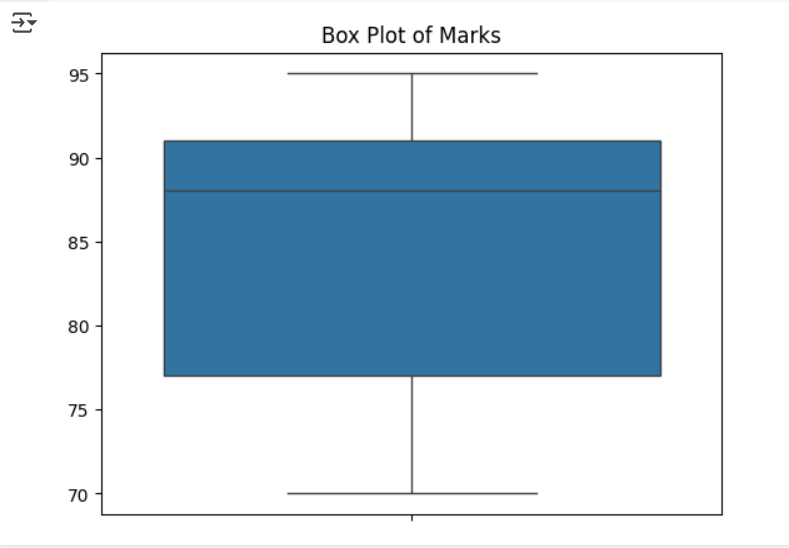
Seaborn helps you explore and understand your data. Its plotting functions operate on data frames and arrays containing whole datasets which makes perfect for Data Analytics and Machine learning.

**BOX PLOT:**

A box plot (also called a box-and-whisker plot) is a statistical graph that shows the distribution of a dataset. It displays the minimum, first quartile (Q1), median (Q2), third quartile (Q3), and maximum. These five-number summaries help you understand the spread and detect outliers in the data.

It’s useful for comparing distributions across multiple groups or categories and is especially handy in spotting outliers, skewness, and data symmetry.



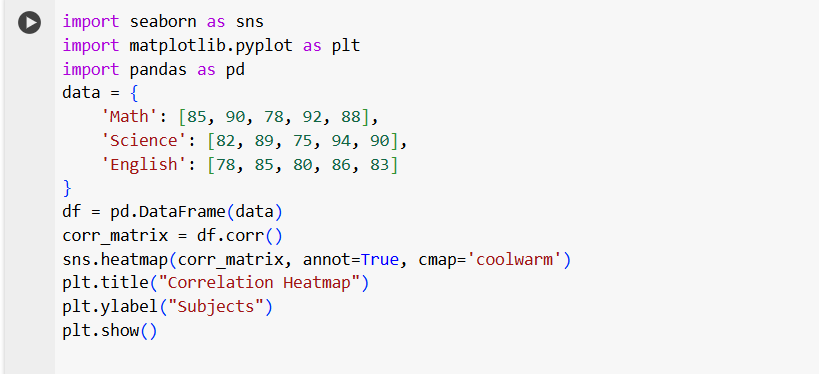


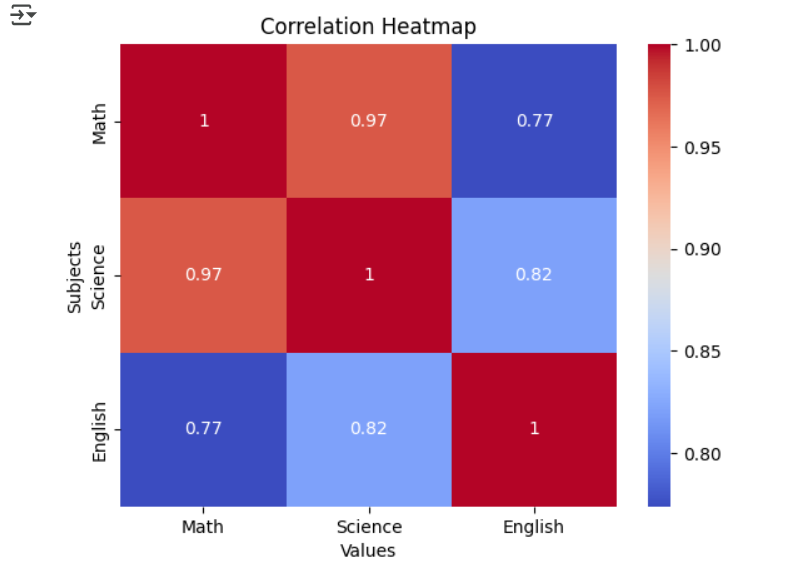
**HEATMAP:**

A heatmap is a graphical representation of data where individual values are represented by colors. It is typically used to visualize relationships between two variables in matrix form, where color intensity indicates the magnitude of values.

Use cases include:

* Showing correlation between features in a dataset
* Visualizing missing data patterns
* Representing confusion matrices in machine learning
* Comparing multiple variables or categories in compact form



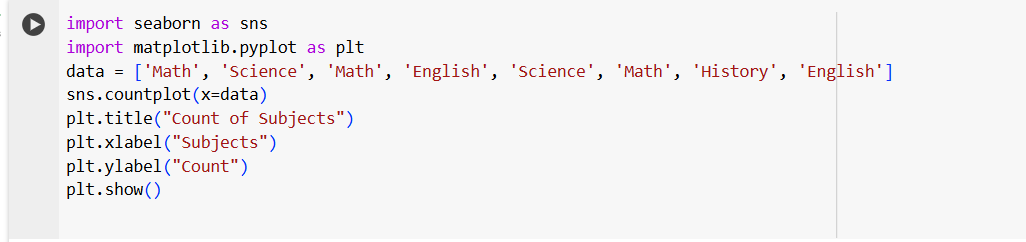


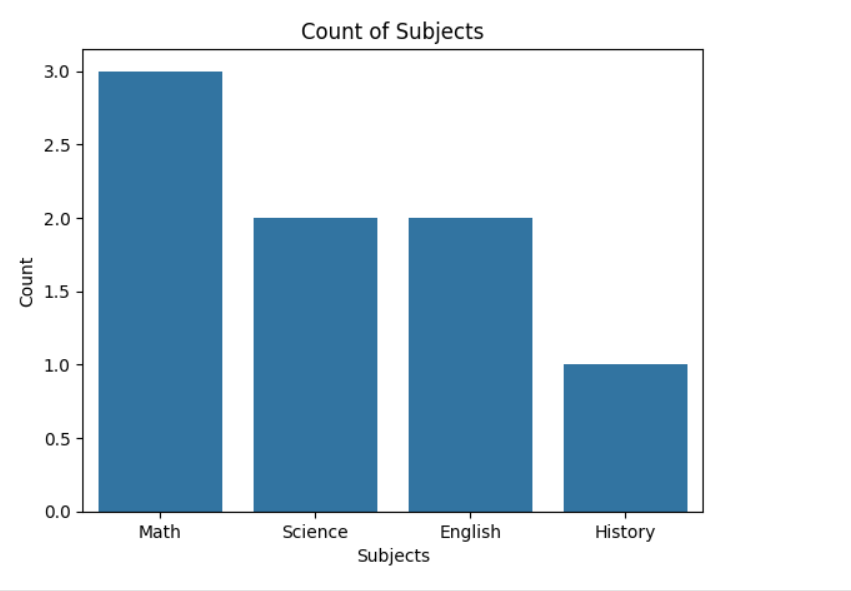
**COUNT PLOT:**

A count plot is a type of bar plot that shows the frequency or count of observations in each category of a categorical variable. It is commonly used in exploratory data analysis to quickly understand the distribution of a categorical feature.

Use cases include:

* Visualizing how many times each category appears in a dataset
* Comparing class balance in classification tasks
* Analyzing survey responses or grouped data



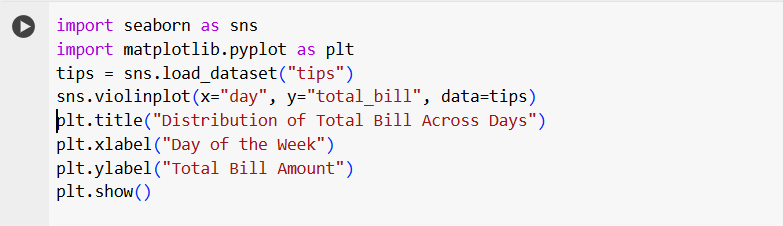


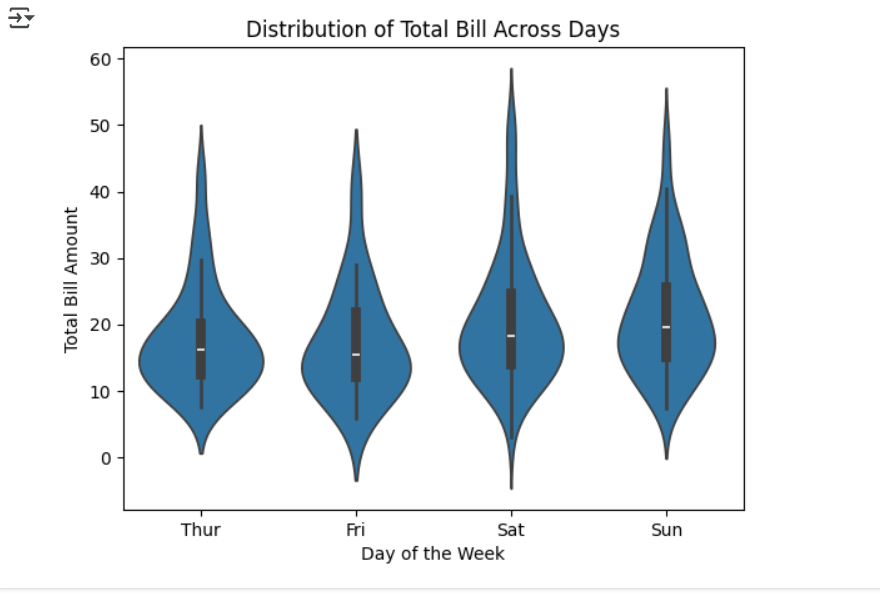
**VIOLIN PLOT:**

A violin plot is a combination of a box plot and a kernel density plot. It shows the distribution of the data across different categories and gives more information about the density and probability of data at different values.

The uses include:

* Comparing the distribution of a numeric variable across multiple categories
* Understanding both summary statistics (like in a box plot) and the shape of the data
* Spotting multimodal distributions and data spread
* Useful in statistical analysis, especially for comparing groups

****

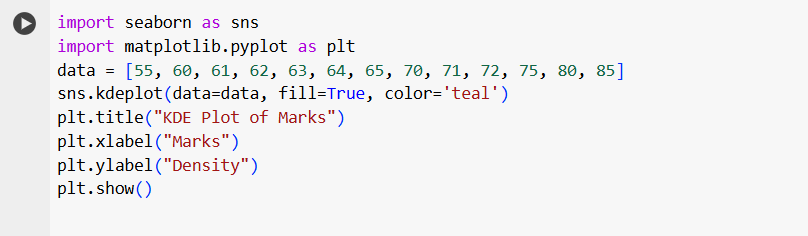
****

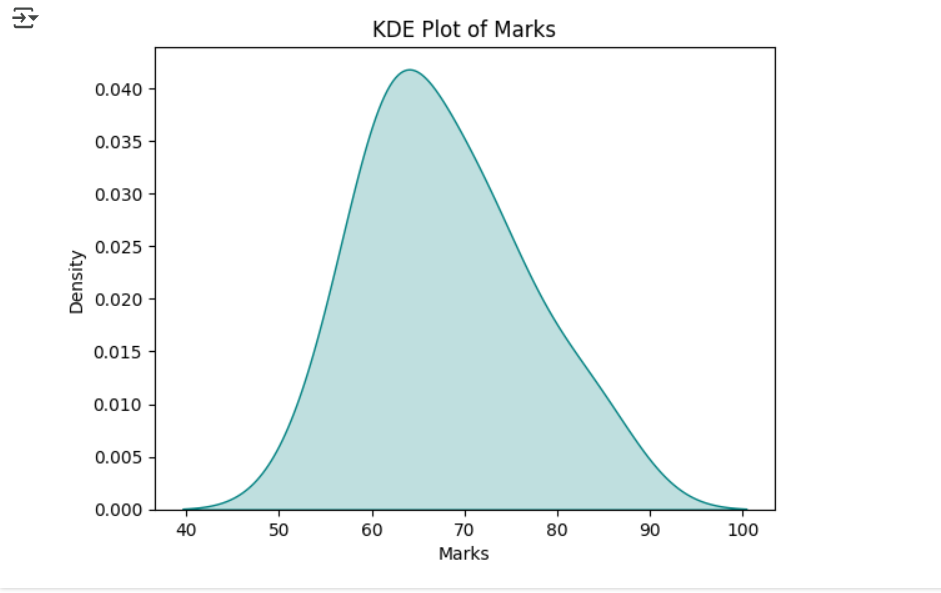
**KDE PLOT:**

A kdeplot, or Kernel Density Estimate plot, is used to visualize the probability density of a continuous variable. It is a smoothed version of a histogram and helps understand the distribution shape of data. Instead of showing frequencies in bins, it shows a continuous curve that represents the likelihood of a value occurring.

The use cases include:

* Understanding the distribution shape (e.g., unimodal, bimodal)
* Comparing the distribution of variables or groups
* Identifying skewness, peaks, and spread
* Replacing or enhancing histograms in EDA





**COMPARISON OF THE ABOVE CHARTS BASED ON HOW AND WHEN TO USE:**

**1. Box Plot**

**When to use:**

* To summarize the distribution of a numeric variable
* To detect outliers and understand data spread
* Great for comparing distributions across categories

**How to use:**  
Use sns.boxplot(x=category, y=value, data=df)  
or just sns.boxplot(data=numeric\_list)

**Example use:**  
Compare students’ scores across subjects

**2. Heatmap**

**When to use:**

* To visualize relationships in matrix or 2D data
* Common for correlation matrices or confusion matrices
* Good for spotting patterns and high/low values

**How to use:**  
Use sns.heatmap(data, annot=True, cmap='coolwarm')  
Pass a 2D matrix or DataFrame

**Example use:**  
Visualizing correlation between features in a dataset

**3. Count Plot**

**When to use:**

* To count the number of occurrences in each category
* Best for analyzing class balance or frequency of categorical data

**How to use:**  
Use sns.countplot(x=column, data=df)

**Example use:**  
Counting number of male vs female passengers in Titanic dataset

**4. KDE Plot**

**When to use:**

* To estimate and visualize the probability density of a continuous variable
* Helps understand distribution shape, especially for smooth curves
* Better than histogram for comparing smooth distributions

**How to use:**  
Use sns.kdeplot(data=your\_list, fill=True)

**Example use:**  
Visualizing the distribution of students’ marks

**5. Violin Plot**

**When to use:**

* To see both distribution (like KDE) and summary stats (like boxplot)
* Useful when comparing distributions between groups
* Great when data might have multiple peaks or non-symmetric shapes

**How to use:**  
Use sns.violinplot(x=category, y=value, data=df)

**Example use:**  
Compare total bill amounts across different days in a restaurant dataset.