# VISUALIZATION

**Visualization** is a way to show information. **Data visualization** means showing data using charts or graphs so it's easier to understand. It helps us see patterns and trends clearly.

In Python, there are many libraries for making visualizations, like **Matplotlib**, **Seaborn**, **Plotly**, and more.

In this guide, we’ll take a closer look at **Matplotlib** and **Seaborn**.

.

## MATPLOTLIB

Matplotlib is a popular plotting library introduced in 2002 by John Hunter. It’s mainly used for making 2D plots and is built on top of numpy.

A typical Matplotlib plot includes these main parts:

Figure

Axes

Axis

Artists

**Figure**

Think of the Figure as the main window or container where everything gets drawn. It can hold one or many plots.

**Axes**

Axes are the actual plots inside the figure. Each one is like a graph area with x and y axes. You can use functions like set\_xlabel() and set\_ylabel() to name the axes.

**Axis**

The Axis deals with things like setting tick marks and limits on the x or y-axis.

**Artists**

Everything you see in the figure (like lines, texts, legends) are called Artists—they’re the visual parts of the plot.

## .

## PYPLOT

**Pyplot** is a sub-module in Matplotlib that contains most of the functions we use to create plots. It makes it easier to build different types of graphs like:

* Bar graphs
* Scatter plots
* Pie charts
* Histograms
* Area charts

We import pyplot from matplotlib as following:



And we also import **NumPy** along with it, since it helps in creating data for plotting:



Some of the plots in matplotlib:

### **LINE CHART**

**Line plots** are one of the most basic types of charts. They connect data points using a straight line to show trends over time or continuous data.

We usually use the plot() function to draw a line chart.

Code snippet:

A screenshot of a computer code

Description automatically generated

Output:

A graph with a line

Description automatically generated

**Description:**

### np.array() is used to create arrays (lists of numbers) that we want to plot.

### plot() is the function that draws the line graph.

### title() sets the title at the top of the graph.

### xlabel() and ylabel() are used to name the x-axis and y-axis.

### show() actually displays the plot window with the graph.

### **BAR GRAPH**

**Bar charts** use rectangular bars to represent values. The **height** of each bar shows how much or how often something happens.

To draw a bar chart, we use the bar() function in Matplotlib. It takes a few common parameters like:

bar(x, y, color, width)

Code snippet:

A screenshot of a computer code

Description automatically generated

Output:

A graph with green bars

Description automatically generated

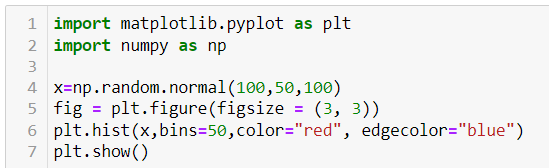
**Description:**

* The bar() function is used to make the bar chart.
* The width parameter sets how thick each bar looks.
* The color parameter changes the color of the bars.

**HISTOGRAM**

**Histogram** is like a bar chart but shows data grouped into ranges (or bins).  
We use hist() to plot it, with parameters like x, bins, color, and edgecolor..

Code snippet:



Output:

A graph with red and blue lines

Description automatically generated

**Description:**

* x is a randomly generated array of 100 values, with a standard deviation of 50.
* bins decides how many groups (sections) the data is split into.
* color sets the fill color of the bars.
* edgecolor adds borders to the bars to separate them clearly.
* A **histogram** is similar to a bar chart but shows grouped data.

### **SCATTER PLOT**

Scatter plot uses dots to depict the relationship between the data. scatter() is used for plotting scatter plot and scatter plot can be done for multiple datasets at the same time by differentiating it with colors. Legends help to achieve this difference.

Code snippet:

A screenshot of a computer code

Description automatically generated

Output:

A graph with red and green dots

Description automatically generated

**Description:**

### x1, y1 are one set of data, and x2, y2 are another.

### The scatter() function is used to make the scatter plot.

### x and y are the data points, and c sets the color.

### legend() adds labels to help tell the plots apart.

### **PIECHART**

Pie charts are used to plot data of same kind which means the same series of data where the different elements are divide based on their percentage.

Code snippet:



Output:

A colorful circle with white text

Description automatically generated

**Description:**

* time is used as the x-values in the pie chart.
* labels are set using the labels keyword to name each slice.
* wedgeprops is used to set the line width and border color between slices.
* The color list gives each slice a different color.

### **AREA CHART**

In area chart, the area under the line to x axis is filled or shaded. It can be done by using fill\_between() function or stackplot() function.

Code snippet:

Using fill\_between() function

A screenshot of a computer code

Description automatically generatedA blue graph with numbers

Description automatically generated

Using stackplot() function

A screenshot of a computer program

Description automatically generated A graph with different colored triangles

Description automatically generated with medium confidence

**Description:**

* fill\_between() takes x and y to fill the area under a line.
* stackplot() uses x and multiple y values (like y1, y2, y3) to show different groups stacked together.
* Legends are added using the labels parameter, and plt.legend(loc=...) sets where the legend appears.

### **SEABORN**

**Seaborn** is a Python library built on top of **Matplotlib**. It’s easier to use and comes with better-looking styles by default.

It supports many types of plots, grouped into categories:

**1. Relational Plots** – show relationships

* scatterplot(), lineplot(), relplot()

**2. Categorical Plots** – compare categories

* barplot(), countplot(), boxplot(), violinplot(), swarmplot(), pointplot(), catplot()

**3. Distribution Plots** – show data spread

* histplot(), kdeplot(), rugplot(), distplot()

**4. Regression Plots** – show trends

* regplot(), lmplot()

**5. Matrix Plots** – visualize matrices

* heatmap(), clustermap()

Here are some sample codes for some of the graphs.

### Scatter plot:

A screenshot of a computer code

Description automatically generated

Output:

A graph of different colored dots

Description automatically generated

**Description:**

Here the different colors on sex is due to the parameter hue. Since hue is on “sex” column there is a difference for male and female.

### LINE PLOT

A computer screen shot of a math problem

Description automatically generated

Output:

A graph of a person and person

Description automatically generated

### BAR PLOT

A screenshot of a computer code

Description automatically generated

Output:

A graph of different colored bars

Description automatically generated with medium confidence

### COUNT PLOT

A computer code with text

Description automatically generated

Output:

A graph of a number of species

Description automatically generated

**Description:**

* Count plots show how many times each category appears.
* x is the category (like species), and hue splits it by another feature (like sex).
* palette='Set1' sets the colors, and it's one of Seaborn’s default styles

### BOX PLOT

A **box plot** shows how data is spread out.  
Values inside the box are in the middle range (interquartile range),  
and any points outside the “whiskers” are seen as **outliers**Code snippet:

A computer code with text

Description automatically generated with medium confidence

Output:

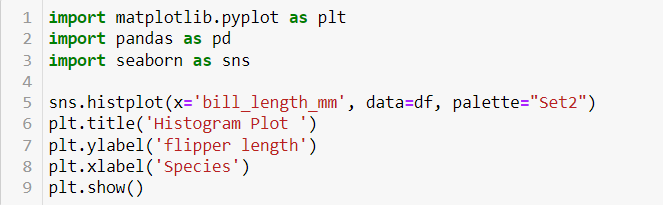
A diagram of a box plot

Description automatically generated

**Description:**

The parameter x should be numeric as the plot shows the difference between categories.

### HISTOPLOT



Output:

A graph of a number of species

Description automatically generated

### KDEPLOT

KDE stands for kernel density estimate. It creates a curve as based on probability. It creates single graph for multiple data samples.

Code snippet:

A white background with black text

Description automatically generated

Output:

A graph of different species

Description automatically generated

### HEATMAP

Heatmaps uses correlation matrix and visualizes data. The datapoints where the higher values get brighter colors and lower values get darker colors.

Code snippet:

A computer code with text

Description automatically generated with medium confidence

Output:

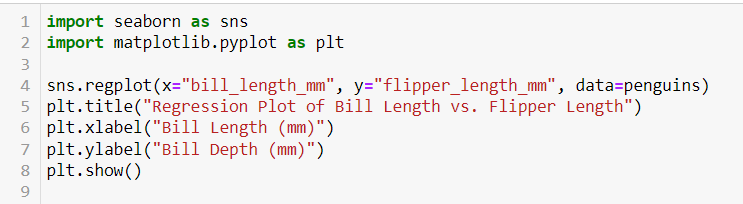
A screenshot of a graph

Description automatically generated

### REGPLOT

Regression plots shows the relationship between variables along with the regression line.

Code snippet:



Output:

A graph of a graph showing a line between a bill length and a bill length

Description automatically generated

### COMPARISON OF MATPLOTLIB AND SEABORN

**Matplotlib:**

* A core plotting library in Python.
* Gives full control over every part of the plot.
* Supports many plot types — even 3D and maps.
* Highly customizable (colors, styles, fonts, etc.).
* Works well with NumPy, Pandas, and SciPy.
* Has been around for a long time and is very stable.

**Advantages of Matplotlib:**

1. Great for detailed, custom plots.
2. Huge range of chart types.
3. Strong community and documentation.
4. Good for creating professional-quality graphs.

.

**Seaborn:**

* Built on top of Matplotlib, but easier to use.
* Focused on **statistical plots** (like box, violin, heatmaps).
* Has nice default styles and color themes.
* Works directly with Pandas DataFrames.
* Best for quick and clean visualizations.

**Advantages of Seaborn:**

1. Less code to write.
2. Beautiful plots by default.
3. Great for data analysis and exploring trends.
4. Integrates easily with Pandas.

Overall, the choice between **Matplotlib** and **Seaborn** depends on what kind of visualization you need. Matplotlib is better when you want more control and customization over your plots, while Seaborn is great for quick, attractive, and statistical visualizations. Since both have their own strengths, many people use them together to get the best results.