

## Task 1: Basic Plotting

- Objective: Introduce the basics of plotting using Matplotlib.
- Instructions:
  1. Create a simple line plot for the function  $y = 2x + 3$ , where  $x$  ranges from -10 to 10.
  2. Label the x-axis as "X-axis" and the y-axis as "Y-axis."
  3. Add a title to the plot: "Basic Line Plot."
- Expected Output: A line graph with proper labels and title.

## Task 2: Customizing Plots

- Objective: Learn how to customize plots using different parameters.
- Instructions:
  1. Plot the functions  $y = x^2$  and  $y = x^3$  on the same graph.
  2. Customize the lines by setting different colors, line styles, and markers for each function.
  3. Add a legend to differentiate between the two functions.
  4. Set gridlines and choose a suitable style (e.g., dashed grid lines).
- Expected Output: A graph showing two functions with different line styles, colors, and a proper legend.

## Task 3: Subplots

- Objective: Practice creating multiple plots in one figure using subplots.
- Instructions:
  1. Create a 2x2 grid of subplots.
  2. Plot the following functions on each subplot:
    - Top-left:  $y = \sin(x)$
    - Top-right:  $y = \cos(x)$
    - Bottom-left:  $y = \tan(x)$
    - Bottom-right:  $y = e^x$
  3. Set the range for  $x$  from  $-2\pi$  to  $2\pi$ , except for  $y = e^x$  where  $x$  ranges from 0 to 5.
- Expected Output: A figure with four subplots showing different mathematical functions.

## Task 4: Bar Chart and Pie Chart

- Objective: Learn how to create bar and pie charts for categorical data.

- Instructions:

1. Create a bar chart for the following data representing the number of students enrolled in each department:

- CS: 150, IT: 120, EE: 100, ME: 80

2. Label the axes and add a title "Department-wise Enrollment."

3. Create a pie chart for the same data, showing the percentage of students in each department.

4. Add labels to the pie chart slices.

- Expected Output: A bar chart and pie chart for enrollment data.

## Task 5: Histogram

- Objective: Understand how to visualize data distributions with histograms.

- Instructions:

1. Generate a dataset of 500 random numbers from a normal distribution (use `np.random.randn(500)`).

2. Plot a histogram of the data with 20 bins.

3. Customize the histogram by changing the color and edge color of the bars.

4. Add a title "Histogram of Random Data" and label the axes.

- Expected Output: A well-labeled histogram showing the distribution of the random data.

## Task 6: Scatter Plot

- Objective: Visualize relationships between two variables using scatter plots.

- Instructions:

1. Generate a random dataset of 100 points for two variables  $X$  and  $Y$ , where  $X = np.random.rand(100)$  and  $Y = 2X + np.random.rand(100)$ .

2. Create a scatter plot of  $X$  vs  $Y$ .

3. Add labels to the x and y axes and give the plot a suitable title.

4. Experiment with changing marker size, color, and transparency.

- Expected Output: A scatter plot that shows the correlation between two variables.

## Task 7: Time Series Plot

- Objective: Work with time series data and plot it using Matplotlib.

- Instructions:

1. Create a time series plot showing stock prices over time.
2. Generate random data for 100 time points (e.g., `np.random.randn(100)` for stock prices).
3. Plot the data and format the x-axis to show time.
4. Label the axes and give the plot a title: "Random Time Series Plot."

- Expected Output: A time series line graph.

## Task 8: Heatmap

- Objective: Visualize data with a heatmap.

- Instructions:

1. Generate a 10x10 matrix of random values using `np.random.rand(10,10)`.
2. Plot a heatmap of this matrix using `plt.imshow()`.
3. Add a color bar and adjust the color map to 'coolwarm.'
4. Give the plot a title "Heatmap of Random Data."

- Expected Output: A heatmap plot of the random matrix with color variations.