Matplotlib - Task_7

`By Menna Jaheen`

⭐ **Matplotlib is a widely-used plotting library in Python that provides a flexible and powerful framework for creating a variety of static, interactive, and animated visualizations. It is especially popular in data analysis, scientific research, and engineering due to its extensive features and customization capabilities.**

Common Usage Patterns

- **Basic Plotting:** Using `plt.plot()`, `plt.scatter()`, or `plt.hist()` to create basic plots and visualize data distributions.
- **Customizing Plots:** Adjusting plot attributes such as titles, labels, colors, and styles using functions like `plt.title()`, `plt.xlabel()`, `plt.ylabel()`, and `plt.legend()`.
- **Subplots:** Creating complex figures with multiple subplots using `plt.subplots()` to organize different plots in a grid layout.
- **Interactive Plots:** Displaying plots within Jupyter notebooks or integrating with interactive environments for exploratory data analysis.

1. Basic Plotting

- **Line Plot:**

```
plt.plot(x, y)
```

Creates a simple line plot with x and y data.

- **Scatter Plot:**

  ```
  plt.scatter(x, y)
  ```

  Creates a scatter plot with x and y data.

- **Bar Plot:**

  ```
  plt.bar(x, height)
  ```

  Creates a vertical bar plot with categories in x and bar heights.

- **Histogram:**

  ```
  plt.hist(data, bins=10
  ```

  Plots a histogram with the specified number of bins.

- **Pie Chart:**

  ```
  plt.pie(sizes, labels=labels)
  ```

  Creates a pie chart with the given sizes and labels.

2. Customization

- **Title and Labels:**

  ```
  plt.title('Title')
  plt.xlabel('X-axis Label')
  plt.ylabel('Y-axis Label')
  ```

- **Legend:**

  ```
  plt.legend(['Label1', 'Label2'])
  ```

- **Grid:**

```
plt.grid(True)
```

- **Tick Marks:**

```
plt.xticks(rotation=45)
plt.yticks([0, 1, 2, 3])
```

- **Colors and Styles:**

```
plt.plot(x, y, color='red', linestyle='--', marker='o')
```

3. Subplots and Layout

- **Create Subplots:**

```
fig, axs = plt.subplots(nrows=2, ncols=2)
```

  Creates a grid of subplots (2×2 in this example).

- **Adjust Layout:**

```
plt.tight_layout()
```

4. Saving Figures

- **Save Plot:**

```
plt.savefig('filename.png')
```

5. Styles and Themes

- **Set Style:**

```
plt.style.use('seaborn-darkgrid')
```

6. Advanced Features

- **Annotate Plot:**

```
plt.annotate('Text', xy=(x, y), xytext=(x, y))
```

- **Error Bars:**

```
plt.errorbar(x, y, yerr=errors)
```

- **Histograms with Density:**

```
plt.hist(data, bins=30, density=True)
```

- **Customizing Colors and Markers:**

```
plt.plot(x, y, color='green', marker='x', markersize=10)
```

7. Interactive Plots

- **Show Plot:**

```
plt.show()
```