**Lab 1: Line Plot**

**Objective:** Learn to create a basic line chart using plt.plot().

**Theory:**

* Line charts show data trends over time or a sequence.
* Ideal for continuous data.

Code :

import matplotlib.pyplot as plt

x = [1, 2, 3, 4, 5]

y = [5, 7, 2, 4, 6]

plt.plot(x, y)

plt.title("Simple Line Plot")

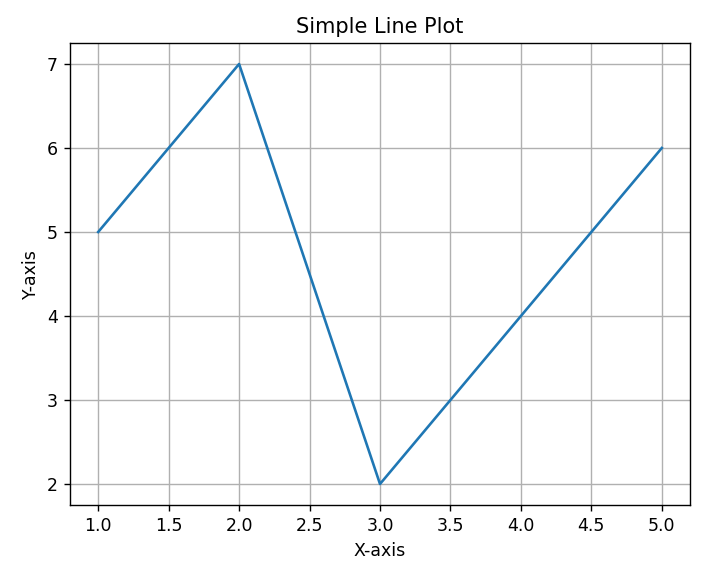
plt.xlabel("X-axis")

plt.ylabel("Y-axis")

plt.grid(True)

plt.show()

output:



**Lab 2: Bar Chart**

**Objective:** Create vertical and horizontal bar charts.

**Theory:**

* Compares values across categories.
* Useful in categorical data comparison.

Code :

import matplotlib.pyplot as plt

categories = ['A', 'B', 'C', 'D']

values = [10, 15, 7, 12]

colors = ['red', 'blue', 'green', 'purple']

plt.bar(categories, values, color=colors, label='Category Values')

plt.title("Bar Chart")

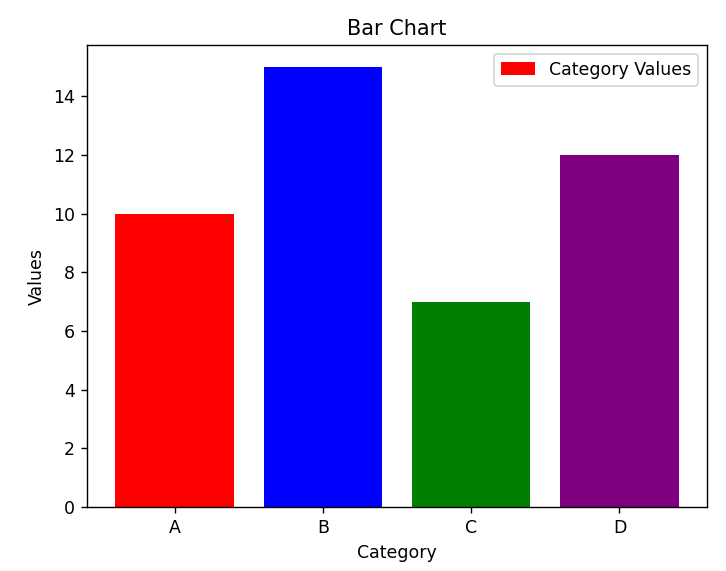
plt.xlabel("Category")

plt.ylabel("Values")

plt.legend()

plt.show()

Output :



**Lab 3: Scatter Plot**

**Objective:** Plot data points to find relationships.

**Theory:**

* Used to visualize correlation between two variables.

Code :

import numpy as np

import matplotlib.pyplot as plt

x = np.random.rand(50)

y = x + np.random.normal(0, 0.1, 50)

plt.scatter(x, y, color='orange', marker='\*', label='Data Points')

plt.title("Scatter Plot")

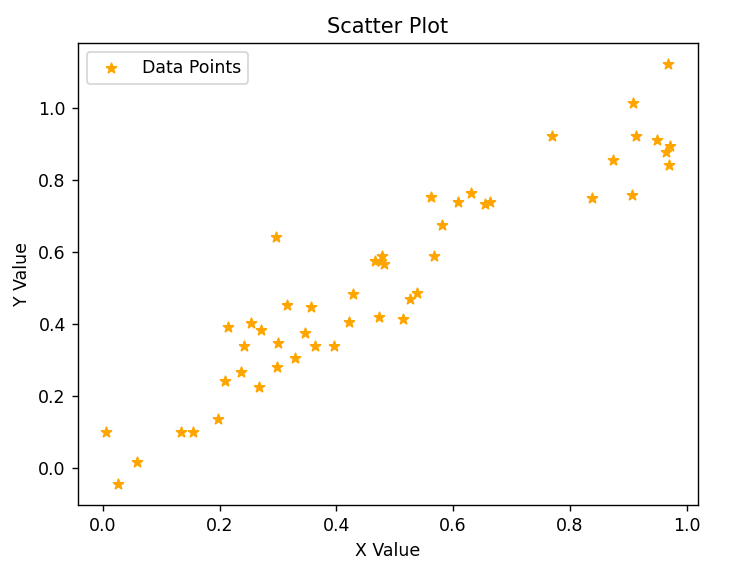
plt.xlabel("X Value")

plt.ylabel("Y Value")

plt.legend()

plt.show()

Output :



**Lab 4: Histogram**

**Objective:** Show data distribution.

**Theory:**

* Groups numeric data into bins.
* Shows frequency distribution.

Code :

import numpy as np

import matplotlib.pyplot as plt

data = np.random.randn(1000)

plt.hist(data, bins=20, color='skyblue', label='Data Distribution')

plt.title("Histogram")

plt.xlabel("Value Range")

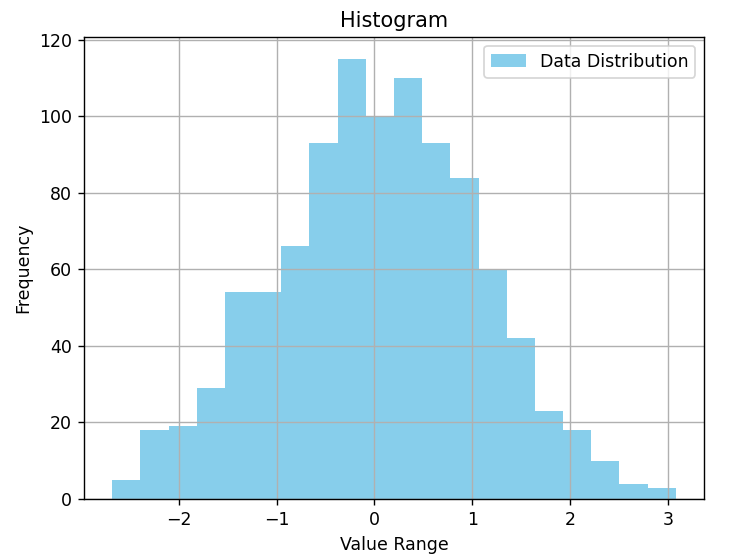
plt.ylabel("Frequency")

plt.grid(True)

plt.legend()

plt.show()

Output:



**Lab 5: Pie Chart**

**Objective:** Represent data as parts of a whole.

**Theory:**

* Good for percentage-based comparisons.

Code :

import matplotlib.pyplot as plt

labels = ['A', 'B', 'C', 'D']

sizes = [30, 25, 25, 20]

colors = ['gold', 'lightcoral', 'skyblue', 'lightgreen']

plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%', startangle=90)

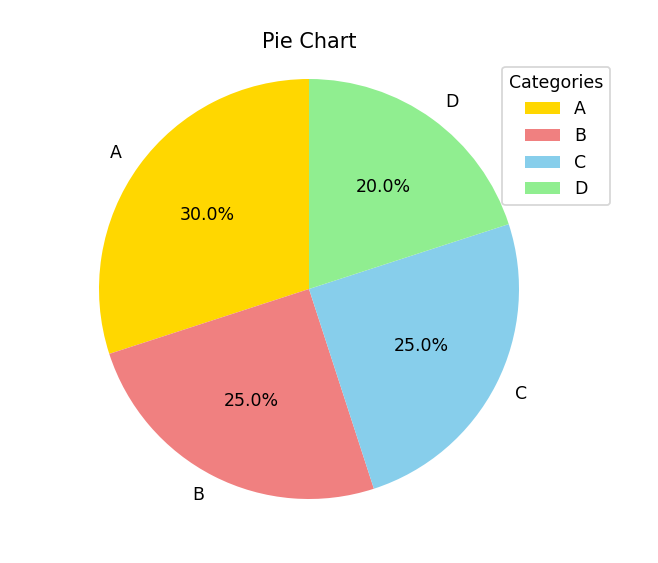
plt.title("Pie Chart")

plt.axis('equal')

plt.legend(labels, title="Categories")

plt.show()

Output:



**Lab 6: Subplots**

**Objective:** Create multiple plots in a single figure.

**Theory:**

* Allows comparison between multiple charts at once.

Code:

import matplotlib.pyplot as plt

import numpy as np

fig, axs = plt.subplots(2, 2)

axs[0, 0].plot([1, 2, 3], [1, 4, 9], color='orange')

axs[0, 0].set\_title("Line")

axs[0, 1].bar(['A', 'B'], [5, 7], color='green')

axs[0, 1].set\_title("Bar")

axs[1, 0].scatter([1, 2, 3], [3, 5, 2], color='red')

axs[1, 0].set\_title("Scatter")

axs[1, 1].hist(np.random.randn(100), color='purple')

axs[1, 1].set\_title("Histogram")

plt.tight\_layout()

plt.show()

Output:

