

Unit:4; Class:4

Matplotlib

Instructor: Musfique Ahmed

Introduction to Matplotlib

2

What is Matplotlib?

- Matplotlib is a powerful Python library for data visualization.
- Used to create a variety of static, animated, and interactive plots.
- Works well with NumPy and Pandas.

Why Use Matplotlib?

- Helps visualize complex data.
- Useful for exploratory data analysis (EDA).
- Customizable and supports different plot types.

Installation:

`pip install matplotlib`

Practice Problem:

- Install Matplotlib and create a basic line plot.



Basic Line Plot

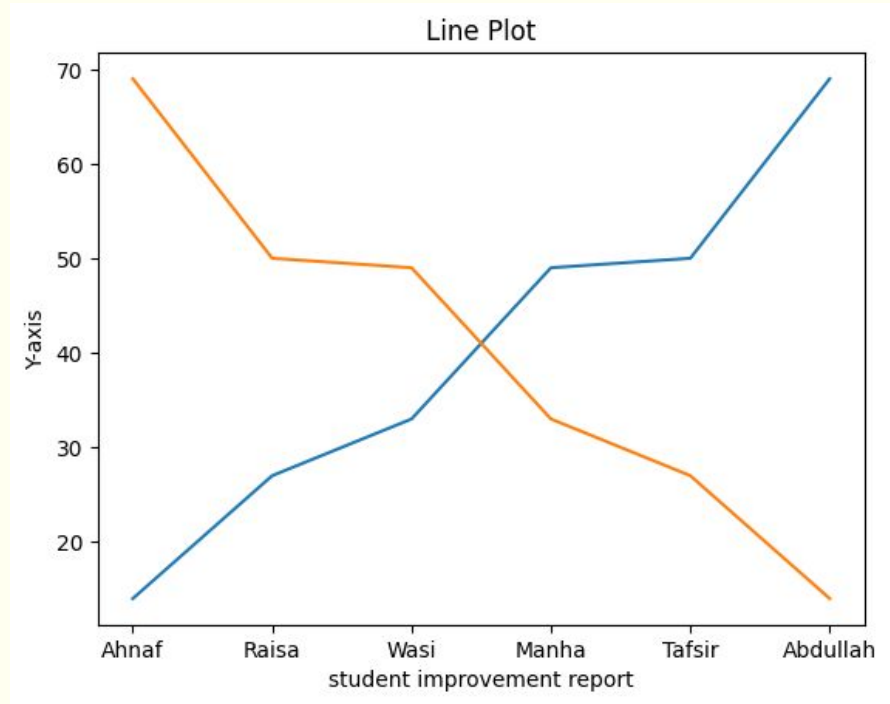
3

Creating a Simple Line Plot

```
import matplotlib.pyplot as plt

x = [1, 2, 3, 4, 5]
y = [10, 20, 25, 30, 50]

plt.plot(x, y)
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Basic Line Plot')
plt.show()
```



Practice Problem:

- Create a line plot showing the population growth of a city over 10 years.

Scatter Plot

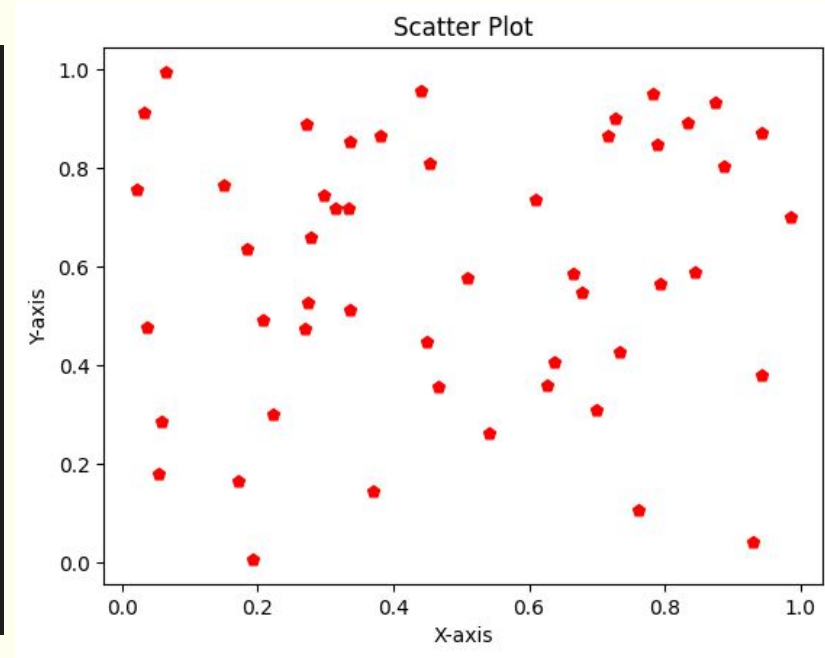
4

Creating a Scatter Plot

```
import numpy as np

x = np.random.rand(50)
y = np.random.rand(50)

plt.scatter(x, y, color='red', marker='o')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Basic Scatter Plot')
plt.show()
```



Practice Problem:

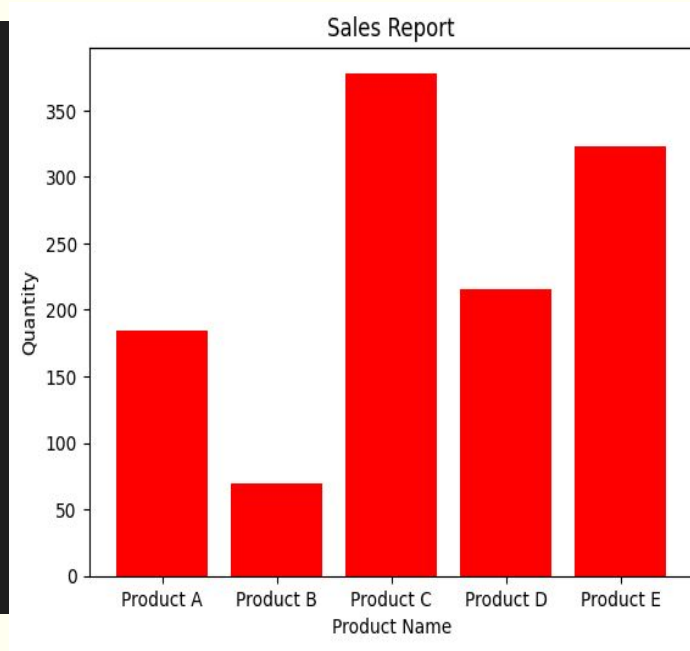
- Generate a scatter plot of students' height vs. weight.

Bar Chart

5

Creating a Bar Chart

```
categories = ['A', 'B', 'C', 'D']  
values = [23, 45, 56, 78]  
  
plt.bar(categories, values, color='blue')  
plt.xlabel('Categories')  
plt.ylabel('Values')  
plt.title('Basic Bar Chart')  
plt.show()
```



Practice Problem:

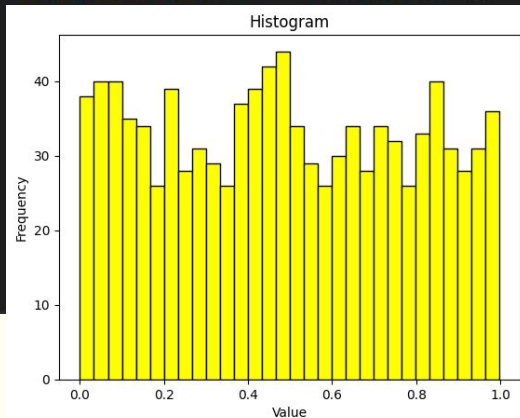
- Create a bar chart showing monthly sales of a store.

Histogram

6

Creating a Histogram

```
data = np.random.randn(1000)
plt.hist(data, bins=30, color='green', edgecolor='black')
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.title('Basic Histogram')
plt.show()
```



Practice Problem:

- Plot a histogram showing the distribution of students' test scores.

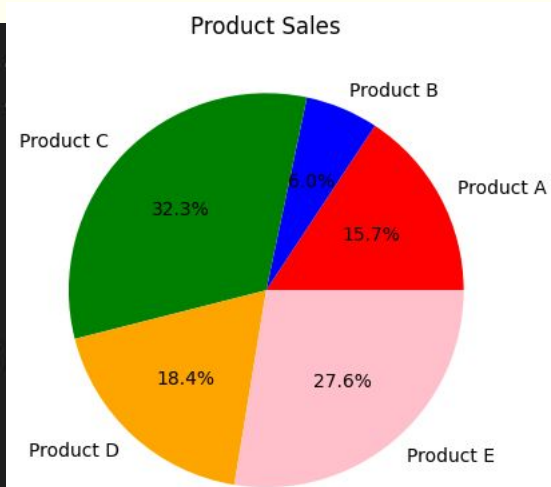
Pie Chart

7

Creating a Pie Chart

```
labels = ['Apples', 'Bananas', 'Cherries', 'Dates']
sizes = [20, 30, 25, 25]

plt.pie(sizes, labels=labels, autopct='%1.1f%%',
        colors=['red', 'yellow', 'pink', 'brown'])
plt.title('Fruit Distribution')
plt.show()
```



Practice Problem:

- Create a pie chart showing the percentage of students in different grade categories (A, B, C, D, F).

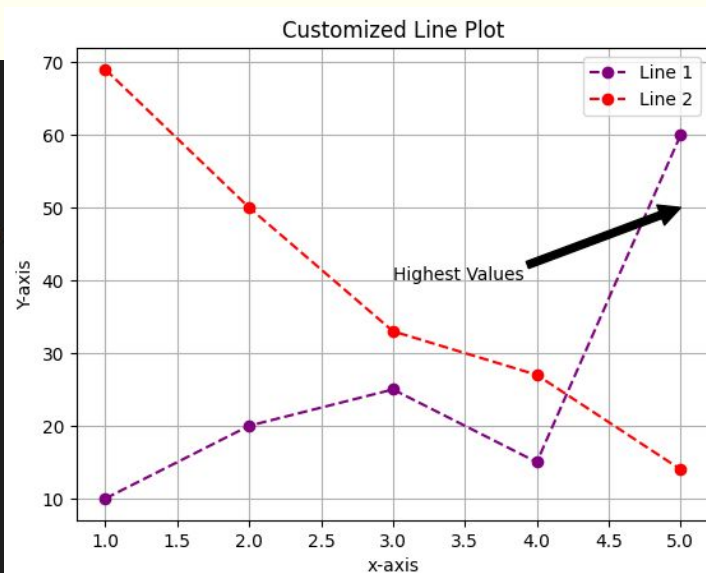
Customizing Plots

8

Adding Legends, Grid, and Annotations

```
x = [1, 2, 3, 4, 5]
y = [10, 20, 30, 40, 50]

plt.plot(x, y, label='Line 1', color='purple', linestyle='--',
         marker='o')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Customized Line Plot')
plt.legend()
plt.grid(True)
plt.annotate('Highest Value', xy=(5, 50), xytext=(3, 40),
            arrowprops=dict(facecolor='black'))
plt.show()
```



Practice Problem:

- Create a line plot with a legend, grid, and annotation highlighting the maximum value.

Subplots

Creating Multiple Subplots

```
fig, axs = plt.subplots(2, 2, figsize=(8, 8))
```

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

```
y = [10, 20, 30, 40, 50]
```

```
axs[0, 0].plot(x, y, color='blue')
```

```
axs[0, 0].set_title('Line Plot')
```

```
axs[0, 1].bar(x, y, color='red')
```

```
axs[0, 1].set_title('Bar Chart')
```

```
axs[1, 0].scatter(x, y, color='green')
```

```
axs[1, 0].set_title('Scatter Plot')
```

```
axs[1, 1].hist(y, bins=5, color='orange')
```

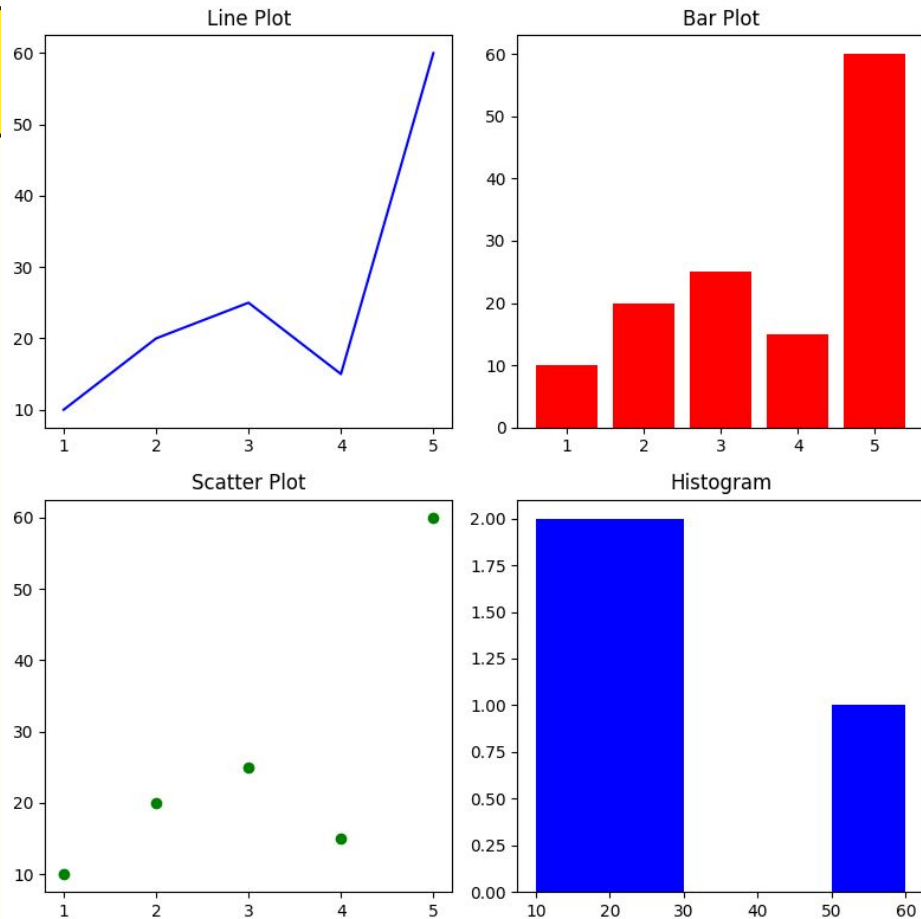
```
axs[1, 1].set_title('Histogram')
```

```
plt.tight_layout()
```

```
plt.show()
```

Practice Problem:

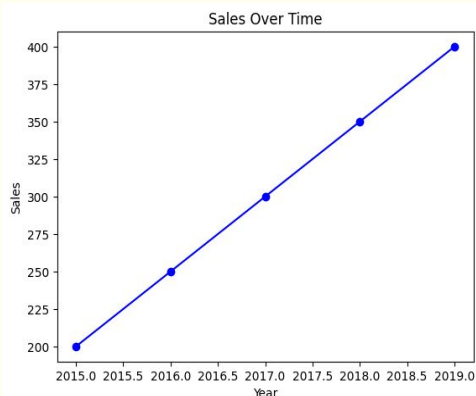
- Create a subplot with four different types of plots in one figure.



Real-World Data Visualization

10

Using Matplotlib with Pandas



```
import pandas as pd

df = pd.DataFrame({
    'Year': [2015, 2016, 2017, 2018, 2019],
    'Sales': [200, 250, 300, 350, 400]
})

plt.plot(df['Year'], df['Sales'], marker='o', linestyle='--',
         color='blue')
plt.xlabel('Year')
plt.ylabel('Sales')
plt.title('Sales Over Time')
plt.show()
```

Practice Problem:

- Load a dataset and visualize trends using Matplotlib.

Hands-on Project idea

11

Mini Project: Weather Data Visualization

1. Find a dataset containing daily temperatures of a city.
2. Create a line plot showing temperature trends.
3. Create a histogram of temperature distribution.
4. Use a bar chart to compare average monthly temperatures.
5. Add customization (title, labels, legends, etc.).

Practice Problem:

- Find a dataset and visualize its key trends using multiple Matplotlib plots.



Thank You

**Do the Quiz Please, you have
10 minutes to do that!**