

# Practice Session for Theory Activity

---

Name: Hariom Tripathi

PRN: 202401070068

Roll Number: ET1-06

Assignment Name: Significance of All Basic Graphs Required for Data Visualization in Python

## Introduction

Data visualization is a crucial technique in data analysis that helps in understanding, interpreting, and communicating patterns and trends in data. Python, with libraries like Matplotlib, Seaborn, and Plotly, enables the creation of effective visualizations using various types of basic graphs. Each graph type serves a unique purpose and fits different types of data.

## Line Graph

- Significance:
  - Shows trends over time.
  - Useful for time series data analysis.
- Example:

```
import matplotlib.pyplot as plt  
  
months = ['Jan', 'Feb', 'Mar', 'Apr']  
sales = [2500, 3000, 4000, 3500]  
  
plt.plot(months, sales)  
plt.title('Monthly Sales Trend')  
plt.xlabel('Month')  
plt.ylabel('Sales')  
plt.show()
```

---

## Bar Chart

- Significance:
  - Compares values across categories.
- Example:

```
categories = ['A', 'B', 'C']  
values = [10, 24, 36]
```

```
plt.bar(categories, values)  
plt.title('Category Comparison')  
plt.xlabel('Category')  
plt.ylabel('Value')  
plt.show()
```

---

## Histogram

- Significance:
  - Shows distribution of continuous data.
- Example:

```
import numpy as np  
  
data = np.random.normal(50, 10, 100)  
plt.hist(data, bins=10, color='skyblue')  
plt.title('Data Distribution')  
plt.xlabel('Value Range')  
plt.ylabel('Frequency')  
plt.show()
```

---

## Pie Chart

- Significance:
  - Displays proportion of categories.
- Example:

```
labels = ['Apple', 'Banana', 'Orange']  
sizes = [30, 45, 25]  
  
plt.pie(sizes, labels=labels, autopct='%1.1f%%')  
plt.title('Fruit Share')  
plt.show()
```

---

## Scatter Plot

- Significance:
  - Shows relationship between two numeric variables.

- Example:

```
x = [1, 2, 3, 4, 5]
y = [2, 4, 1, 8, 7]

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

---

## Box Plot

- Significance:
  - Displays data spread and outliers.
- Example:

```
import seaborn as sns

data = [7, 15, 13, 21, 22, 19, 30, 35]
sns.boxplot(data=data)
plt.title('Box Plot Example')
plt.show()
```

---

## Heatmap

- Significance:
  - Visualizes correlations or matrix data.
- Example:

```
import seaborn as sns
import numpy as np
import pandas as pd

data = np.random.rand(4, 4)
df = pd.DataFrame(data, columns=["A", "B", "C", "D"])

sns.heatmap(df, annot=True, cmap="YlGnBu")
plt.title('Heatmap Example')
plt.show()
```

---

## Area Chart

- Significance:
  - Emphasizes magnitude over time.
- Example:

```
x = [1, 2, 3, 4]  
y = [2, 3, 4, 6]
```

```
plt.fill_between(x, y, color="skyblue", alpha=0.4)  
plt.plot(x, y, color="Slateblue")  
plt.title('Area Chart')  
plt.xlabel('X-axis')  
plt.ylabel('Y-axis')  
plt.show()
```

---

## Violin Plot

- Significance:
  - Combines box plot and density plot.
- Example:

```
tips = sns.load_dataset("tips")  
sns.violinplot(x="day", y="total_bill", data=tips)  
  
plt.title('Violin Plot of Total Bill by Day')  
plt.show()
```

---

## Conclusion

Understanding the right graph to use for specific data types is essential in data analysis. Each type of visualization highlights different aspects of the data, from distribution and relationships to trends and comparisons. Python provides simple yet powerful tools for creating these visuals, which are critical for insightful analysis and presentation.