1. Introduction to Matplotlib

Matplotlib = Python's fundamental plotting library
Used for:

- Line plots
- Bar charts
- Scatter plots
- Histograms
- Custom visualizations

Basic Structure

```
import matplotlib.pyplot as plt

plt.plot(x, y)

plt.title("My Plot")

plt.xlabel("X-axis")

plt.ylabel("Y-axis")

plt.show()
```

2. LINE PLOTS

Used for:

- Trends
- Time series
- Continuous data

Example:

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

sales = [100, 150, 120, 180, 200]

plt.plot(days, sales)

plt.title("Daily Sales Trend")

plt.xlabel("Days")

plt.ylabel("Sales")

plt.grid(True)
```

```
plt.show()
```

Key insight:

Sales increased steadily, with a dip on Day 3.

3. BAR CHARTS

Used for comparison between categories.

Example:

```
products = ["A", "B", "C", "D"]
revenue = [5000, 3000, 7000, 4000]
plt.bar(products, revenue)
plt.title("Product Revenue Comparison")
plt.xlabel("Product")
plt.ylabel("Revenue")
plt.show()
```

Insight:

Product C generates the maximum revenue.

4. SCATTER PLOTS

Used for:

- Relationship between variables
- Trends
- Outliers

Example:

```
age = [20, 25, 30, 35, 40, 45]
salary = [25, 35, 50, 60, 75, 85]

plt.scatter(age, salary)
plt.title("Age vs Salary")
plt.xlabel("Age")
plt.ylabel("Salary")
```

```
plt.show()
```

Insight:

There is a positive correlation between age and salary.

5. Styling & Best Practices

- Always add **titles, labels, legends**
- Use **grid lines** when needed
- Don’t overload too much data
- Use color contrast appropriately
- Use consistent scales
- Include insights when presenting

PART 2 — Seaborn Visualizations

1. Introduction to Seaborn

Seaborn provides statistical visualization built on Matplotlib:

- Heatmaps
- Distribution plots
- Pairplots
- Boxplots
- Violin plots
- Category visualizations

```
import seaborn as sns

sns.set_style("whitegrid")
```

2. Distribution Plots (distplot / histplot / kdeplot)

Example:

```
sns.histplot(df["Age"], kde=True)

plt.title("Age Distribution")

plt.show()
```

Insight:

Age follows a normal distribution with a peak around 30.

3. Boxplot

Shows spread, median, and outliers.

```
sns.boxplot(x=df["Fare"])
plt.title("Fare Distribution - Titanic")
plt.show()
```

Insight:

Many passengers paid low fares, with few expensive outliers.

4. Category Plot (countplot)

```
sns.countplot(x=df["Survived"])
plt.title("Survival Count")
plt.show()
```

Insight:

More passengers did not survive than survived.

5. Heatmaps

Used for correlation analysis.

```
sns.heatmap(df.corr(), annot=True, cmap="coolwarm")
plt.title("Correlation Heatmap")
plt.show()
```

Insight:

On Titanic, Fare correlates positively with Survival.

6. Pairplot (BONUS)

Shows relationship between all numerical columns.

```
sns.pairplot(iris, hue="species")
plt.show()
```

Insight:

Setosa is clearly separated from the other species in petal measurements.

PART 3 — Guided Hands-On

Students work on **Iris** and **Titanic** datasets.

Dataset 1 — IRIS

Visualizations to create:

1. **Scatter Plot:**
   Sepal length vs Petal length (color by species)
2. **Histogram:**
   Distribution of petal length
3. **Boxplot:**
   Petal width for each species
4. **Pairplot:**
   Compare all numeric features across species

Expected Insights:

- Setosa shows smaller petal measurements
- Virginica tends to have the largest petals
- Species clusters are visually separable

Dataset 2 — TITANIC

Visualizations to create:

1. **Countplot:**
   Survivors vs Non-survivors
2. **Bar Chart:**
   Survival rate by gender
3. **Histogram:**
   Fare distribution
4. **Boxplot:**
   Age distribution among survivors
5. **Heatmap:**
   Correlation among Age, Fare, Survived, Pclass

Expected Insights:

- Higher female survival rate
- Passengers in 1st class survived more
- Older passengers were fewer than young
- Fare positively correlates with survival

PART 4 — Understanding Insights from Graphs

Teach students:

- What story does the graph tell?
- What patterns do you see?
- Are there any anomalies?
- Are two variables correlated?
- What decision would a business make from this?

PART 5 — Good Visualization Practices

✓ Choose appropriate chart type
✓ Use readable colors
✓ Don’t clutter labels
✓ Always label axes
✓ Keep titles meaningful
✓ Add insights — don’t let the graph speak alone

PART 6 — Fun Task

Draw Your Mood Graph

Students create a 7-day personal dataset:

Day,Mood

Mon,7

Tue,5

Wed,8

Thu,6

Fri,9

Sat,10

Sun,8

Step 1 — Line Plot

plt.plot(days, moods, marker='o')

plt.title("My Mood Over the Week")

plt.xlabel("Day")

```
plt.ylabel("Mood Level (1-10)")

plt.ylim(0, 10)

plt.grid(True)

plt.show()
```

Step 2 — Add Colors

Students choose colors representing emotions:

- Blue → Calm
- Red → Stress
- Yellow → Happy

Step 3 — Write 2–3 Insights

Example:

“My mood improves significantly on Friday and peaks on Saturday. Mid-week slump seen on Tuesday.”