SHADOWFOX DATASCIENCE INTERNSHIP:

Name: Srithesh

Email: sritheshp@gmail.com

Python Visualization Libraries:

Matplotlib & Seaborn – Beginner Guide

---

1. Introduction

`Matplotlib` – Overview & Purpose:

Matplotlib is a Python visualization library used to create different types of graphs and charts. It helps us convert numbers into visual form so that we can easily understand patterns, trends, and relationships in data. It basic and flexible graphs.

-It is used for:

- Show trends (temperature, stock prices)
- Compare values (sales, scores)
- Find relationships (height vs weight)
- Make simple charts (bar, line, scatter)

Seaborn – Overview & Purpose:

Seaborn is a Python data visualization library built on top of Matplotlib. It is used to create more attractive and informative statistical graphs with less code. Advanced and more attractive statistical graphs.

-It is used for:

- Make attractive statistical charts easily
- Compare categories (like tips by day, species of penguins)
- Show relationships between variables (scatterplots, regression lines)
- Visualize distributions (histograms, KDE plots, boxplots)

---

2. Graph Types in Matplotlib

2.1 Line Plot

Description:
A line plot shows data points connected by a straight line. It is used to visualize trends over time.

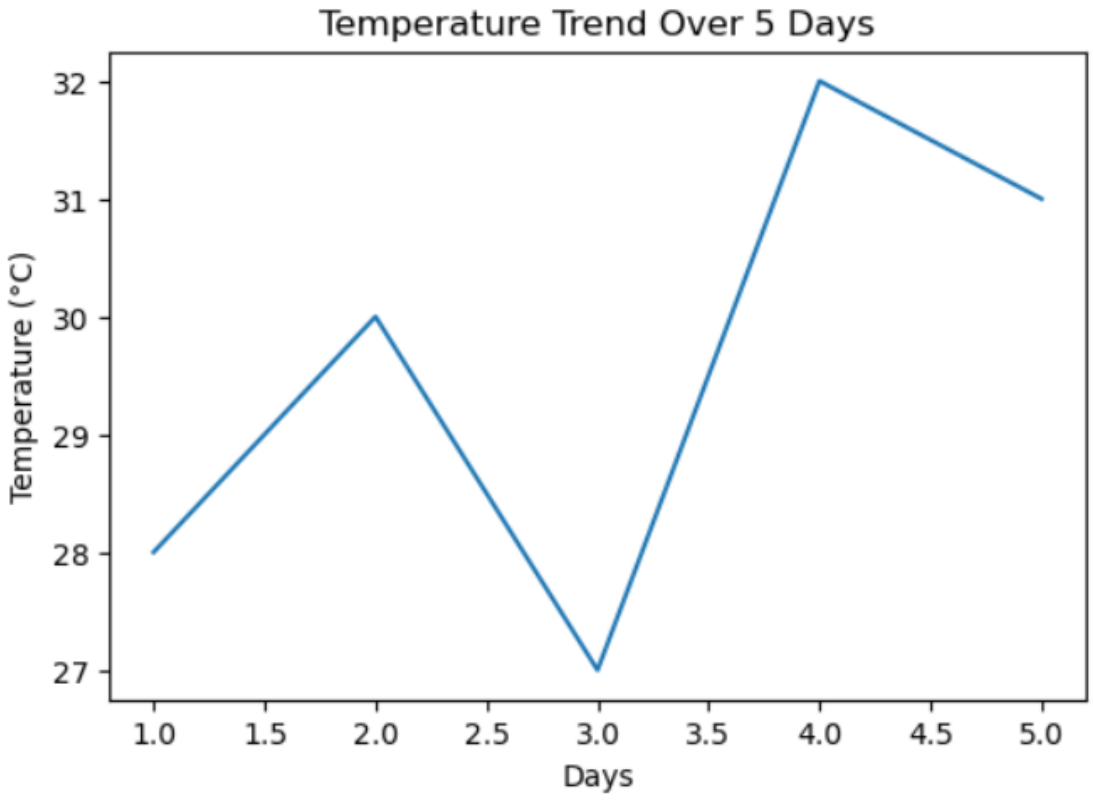
Use Case:
Stock prices, temperature changes, growth analysis.

Code Example:

```python
import numpy as np
import matplotlib.pyplot as plt

days = np.array([1, 2, 3, 4, 5])
temperature = np.array([28, 30, 27, 32, 31])

fig, ax = plt.subplots(figsize=(6, 4))
ax.plot(days, temperature)
ax.set_xlabel("Days")
ax.set_ylabel("Temperature (°C)")
ax.set_title("Temperature Trend Over 5 Days")
plt.show()
```



Description: fig, ax = plt.subplots() → creates the graph.

ax.plot(days, temperature) → plots the line.

ax.set_xlabel / set_ylabel / set_title → adds labels & title.

2.2 Bar Chart

Description:

A bar chart uses rectangular bars to represent value comparisons between categories.

Use Case:

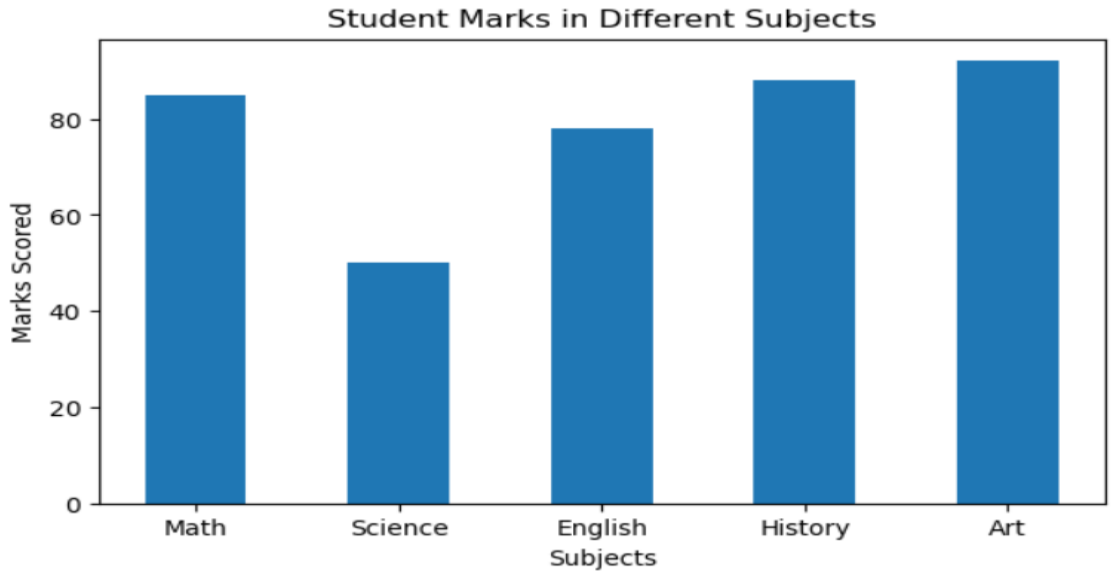
Comparing sales between different months or products.

Code Example:

```python
import numpy as np
import matplotlib.pyplot as plt

subjects = np.array(["Math", "Science", "English", "History", "Art"])
marks = np.array([85, 50, 78, 88, 92])

fig, ax = plt.subplots(figsize=(7, 4))
ax.bar(subjects, marks,width=0.5)
ax.set_xlabel("Subjects")
ax.set_ylabel("Marks Scored")
ax.set_title("Student Marks in Different Subjects")
plt.show()
```



Description: ax.bar(subjects, marks) → draws the bar chart.

fig, ax = plt.subplots() → creates the figure & axes.

2.3 Scatter Plot

Description:

A scatter plot displays points on a graph representing the relationship between two variables.

Use Case:

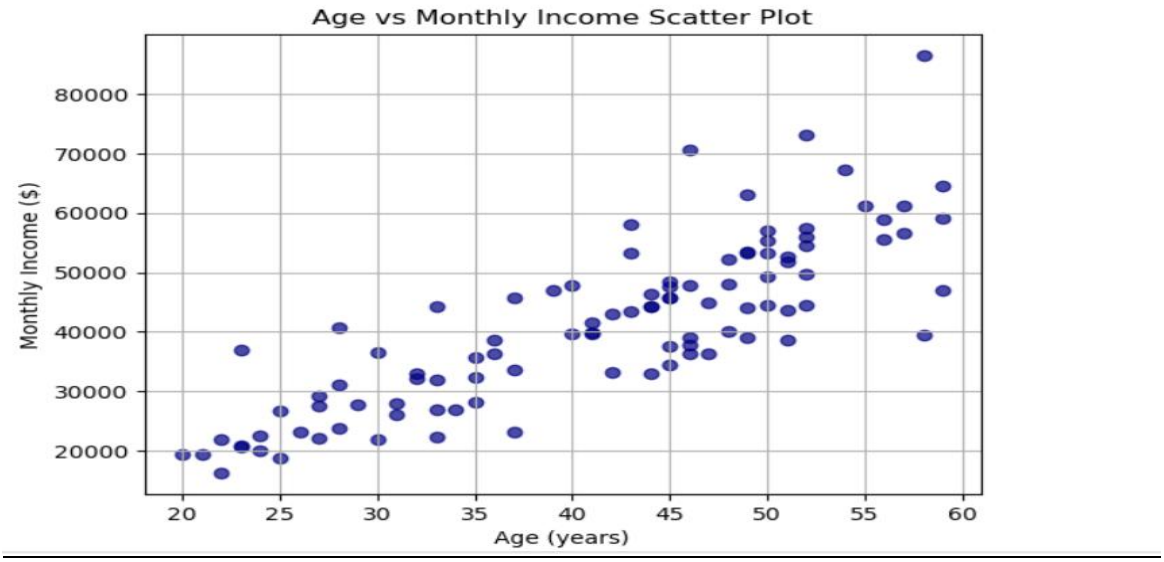
Showing correlation between height and weight.

Code Example:

```python
import matplotlib.pyplot as plt
import numpy as np

age = np.random.randint(20, 60, 100)
income = age * np.random.normal(1000, 200, 100)

plt.scatter(age, income, color='navy', alpha=0.7)
plt.xlabel("Age (years)")
plt.ylabel("Monthly Income ($)")
plt.title("Age vs Monthly Income Scatter Plot")
plt.grid(True)
plt.show()
```



Description: plt.scatter(age, income) → Draws a scatter plot.

age = Random ages between 20–60. income = Income values based on age plt.grid(True) → Adds grid lines.

2.4 Histogram

Description:

A histogram represents the distribution of numerical data.

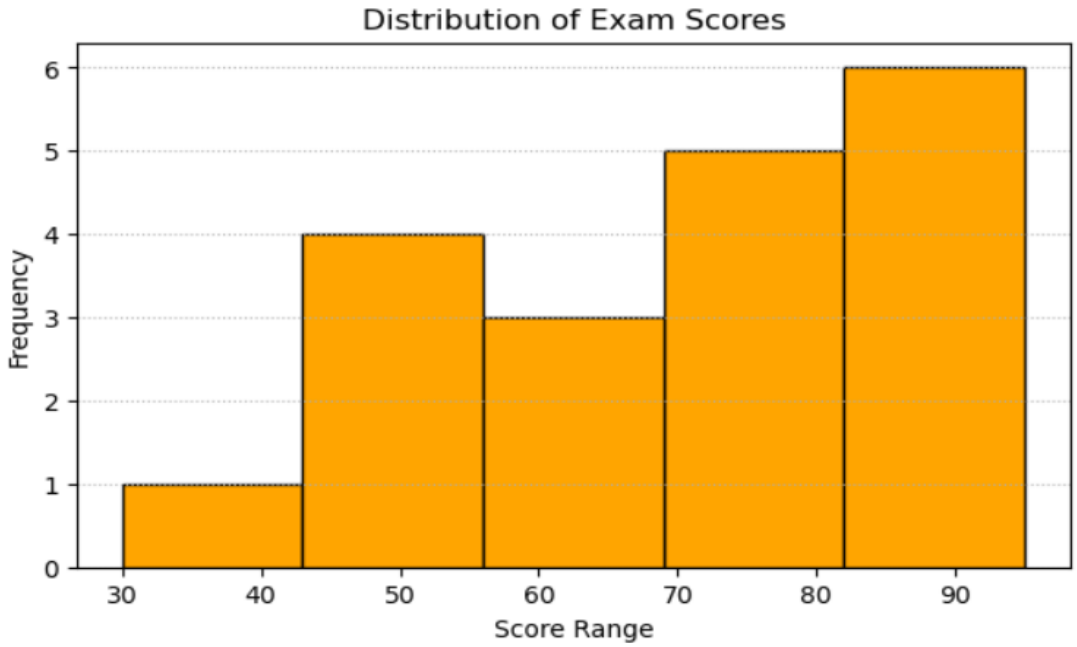
Use Case:

Age distribution, exam marks distribution.

Code Example:

```
import numpy as np
import matplotlib.pyplot as plt

marks = np.array([45, 55, 60, 62, 70, 72, 75, 30, 82, 85, 88, 90, 92, 95, 45, 52, 68, 74,80])
fig, ax = plt.subplots(figsize=(7, 4))
ax.hist(marks, bins=5, edgecolor='black', color='orange')
ax.set_xlabel("Score Range")
ax.set_ylabel("Frequency")
ax.set_title("Distribution of Exam Scores")
ax.grid(axis='y', linestyle=':')
plt.show()
```



Description: ax.hist() - draws histogram, bins=5 - groups data into 5 ranges. edgecolor='black' - cleaner bars.

2.5 Pie Chart

Description:

A pie chart represents data in a circular format as slices.

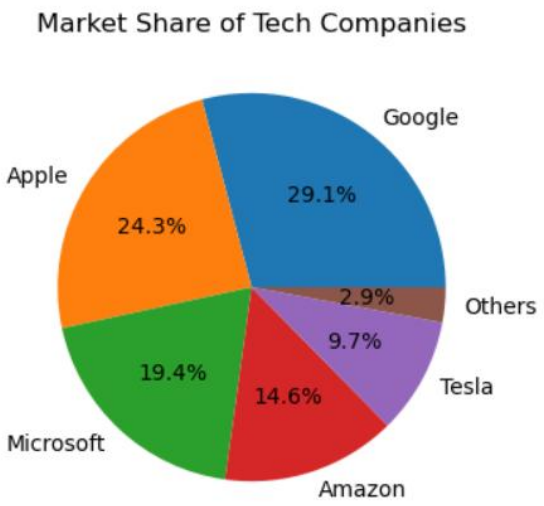
Use Case:

Market share, budget distribution.

Code Example:

```
import matplotlib.pyplot as plt
import numpy as np

companies = np.array(["Google", "Apple", "Microsoft", "Amazon", "Tesla","Others"])
market_share = np.array([30, 25, 20, 15, 10,3])
fig, ax = plt.subplots(figsize=(6, 4))
ax.pie(market_share, labels=companies, autopct="%1.1f%%")
ax.set_title("Market Share of Tech Companies")
plt.show()
```



Description: fig, ax = plt.subplots() → creates the figure & axes.

ax.pie() → draws pie chart.

autopct → shows percentage on each slice.

---

3. Graph Types in Seaborn

3.1 Line Plot

Description:

Seaborn line plot is used to show trends with built-in beauty and styling.

Use Case:

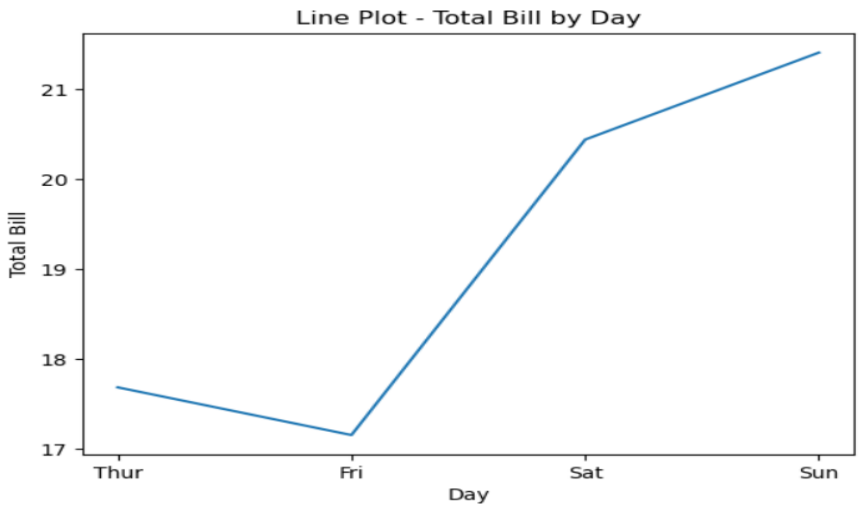
Sales growth per year.

Code Example:

```
import seaborn as sns
import matplotlib.pyplot as plt

tips = sns.load_dataset("tips")

sns.lineplot(x="day", y="total_bill", data=tips,errorbar=None)
plt.title("Line Plot - Total Bill by Day")
plt.xlabel("Day")
plt.ylabel("Total Bill")
plt.show()
```



Description: tips = **sns.load_dataset("tips")** → loads the built-in tips dataset. sns.lineplot(...) → creates a line plot.

3.2 Bar Plot

Description:

Creates a bar chart with better styling and automatic aggregation.

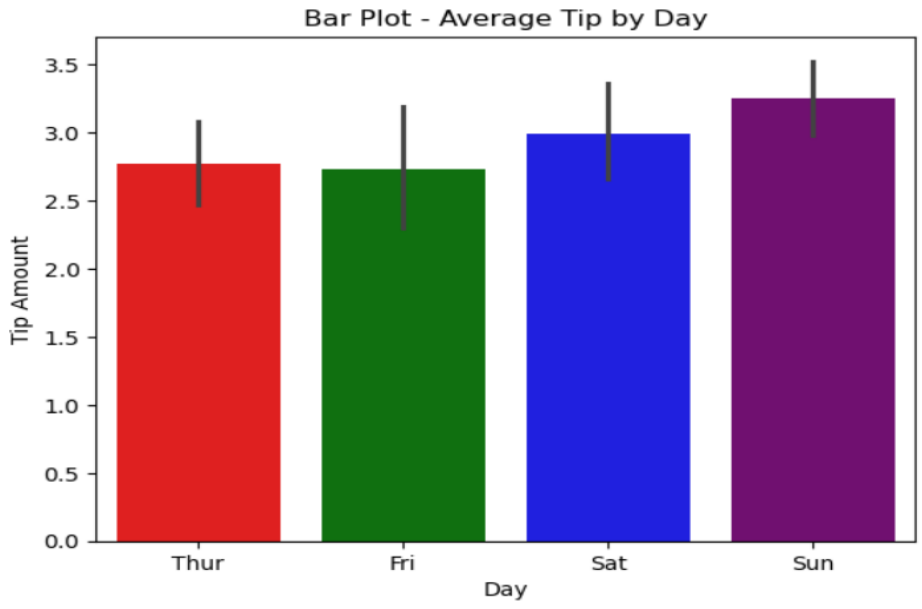
Use Case:

Comparing average values.

Code Example:

```
import seaborn as sns
import matplotlib.pyplot as plt

tips = sns.load_dataset("tips")
colors = ["red", "green", "blue", "purple"]
sns.barplot(x="day", y="tip", data=tips, hue="day", palette=colors, legend=False)
plt.title("Bar Plot - Average Tip by Day")
plt.xlabel("Day")
plt.ylabel("Tip Amount")
plt.show()
```



Description: sns.barplot() → automatically computes average of total bill per day.

3.3 Scatter Plot

Description:

Shows relationships between two variables with optional grouping.

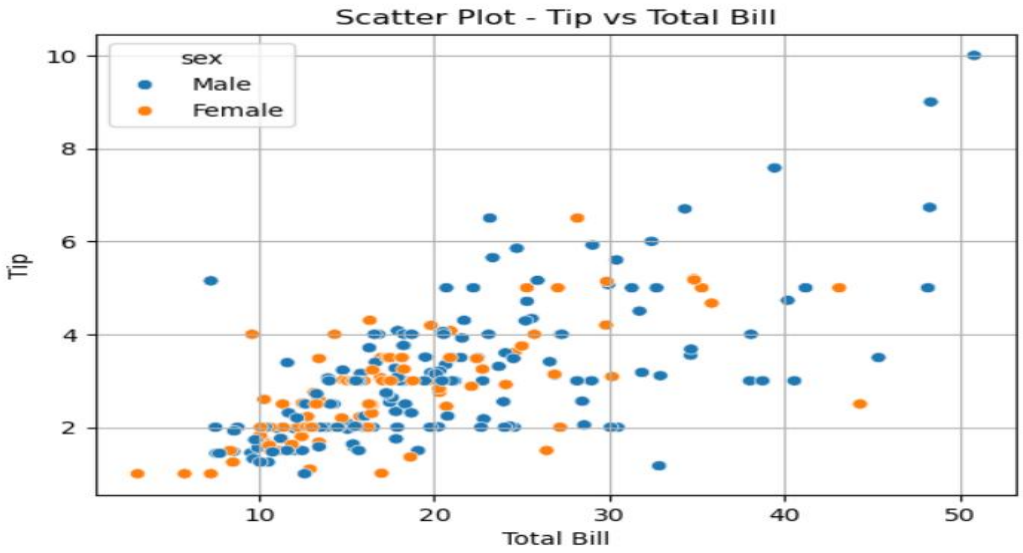
Use Case:

Relation between hours studied and marks.

Code Example:

```
import seaborn as sns
import matplotlib.pyplot as plt

tips = sns.load_dataset("tips")
sns.scatterplot(x="total_bill", y="tip", data=tips, hue="sex")
plt.title("Scatter Plot - Tip vs Total Bill")
plt.xlabel("Total Bill")
plt.ylabel("Tip")
plt.grid()
plt.show()
```



Description: Scatter plot shows relationship between total bill and tip.

hue='sex' → colors points by gender.

3.4 Histogram

Description:

A heatmap is a graph that shows numbers using colours.

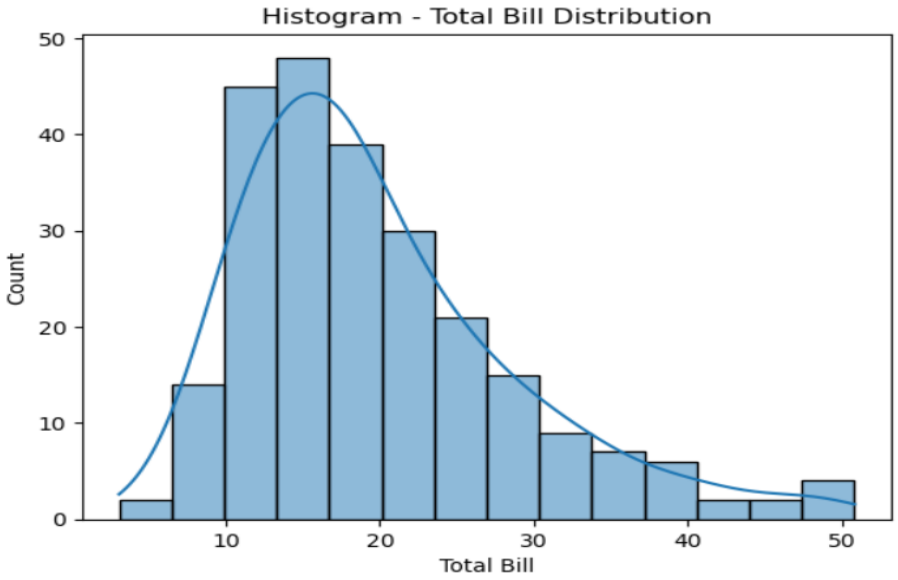
Use Case:

Salary, age, or data distribution.

Code Example:

```
import seaborn as sns
import matplotlib.pyplot as plt

tips = sns.load_dataset("tips")
sns.histplot(tips["total_bill"], kde=True)
plt.title("Histogram - Total Bill Distribution")
plt.xlabel("Total Bill")
plt.ylabel("Count")
plt.show()
```



Description: sns.histplot() → draws histogram.

kde=True → adds smooth line showing distribution.

3.5 Heatmap

Description:

Creates multiple plots to show relationships between variables.

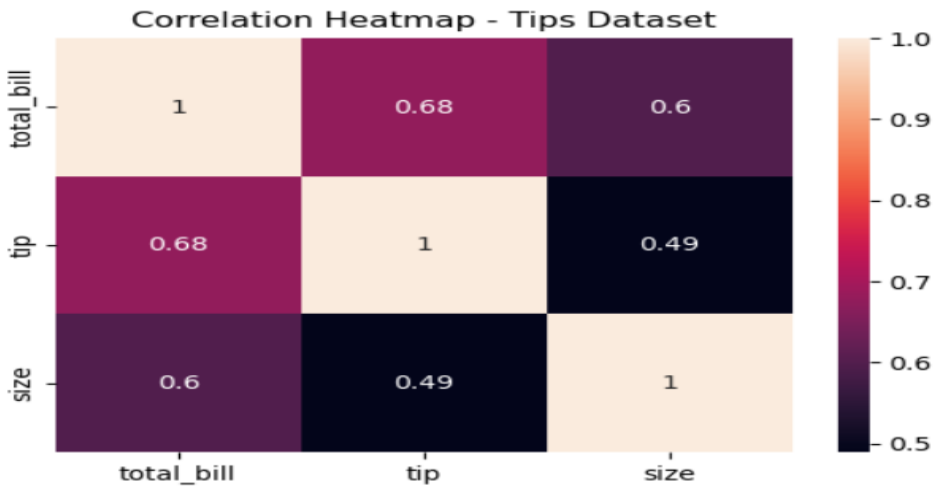
Use Case:

Correlation analysis, Category comparison.

Code Example:

```
import seaborn as sns
import matplotlib.pyplot as plt

tips = sns.load_dataset("tips")
corr = tips.corr(numeric_only=True)
plt.figure(figsize=(6,4))
sns.heatmap(corr, annot=True)
plt.title("Correlation Heatmap - Tips Dataset")
plt.show()
```



Description: tips.corr() → calculates correlation between numeric columns.

annot=True → shows correlation numbers inside each cell.

---

4. Comparison

Feature	Matplotlib	Seaborn
Type of Library	Basic visualization library	Advanced visualization library built on Matplotlib
Graph Style	Simple, manual styling	Beautiful, clean, automatic styling
Ease of Use	Requires more code	Easier, less code for good visuals
Best For	Custom plots, full control	Custom plots, full control
Plot Variety	Standard plots (line, bar, scatter)	Extra plots (heatmap, pairplot, violin plot)
Customization	Very high customization	Less manual work, auto handles many things
Default Look	Plain	Modern and colorful
Learning Curve	Moderate	Easy for beginners

---

5. Conclusion: Both Matplotlib and Seaborn are important tools in data science and analytics. Matplotlib is powerful for detailed customization, while Seaborn is ideal for creating beautiful statistical visualizations quickly. Learning both libraries provides a strong foundation for data visualization in Python.
