

Data Visualization in Python

◆ 1. What is Data Visualization?

Definition:

👉 The process of representing data in a **graphical format** (charts, plots, maps).

👉 Helps to **see patterns, relationships, and trends**.

Why it is important?

Easier to **understand large datasets**

Makes **comparison and trends** clear

Used in **data science, business, research**

◆ 2. Libraries for Visualization in Python

Matplotlib → foundation library (basic plots).

Seaborn → built on Matplotlib, gives **statistical & beautiful plots**.

Plotly → interactive visualizations.

We'll cover **Matplotlib + Seaborn** in detail.

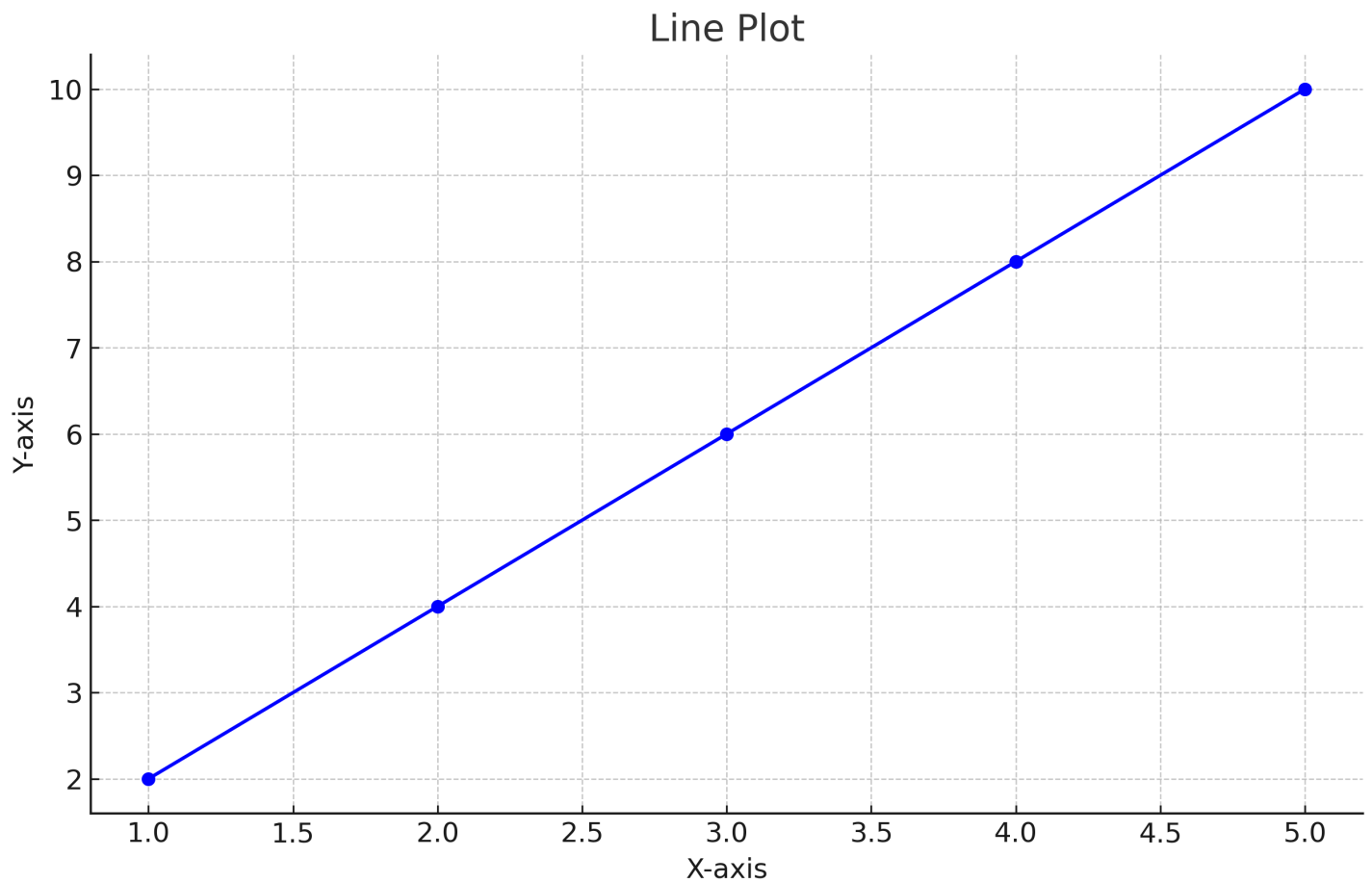
Part A: Matplotlib

◆ 3. Matplotlib Basics

```
import matplotlib.pyplot as plt
```

(a) Line Plot

👉 Used to show **trends** over time.



```
x = [1,2,3,4,5]
y = [2,4,6,8,10]

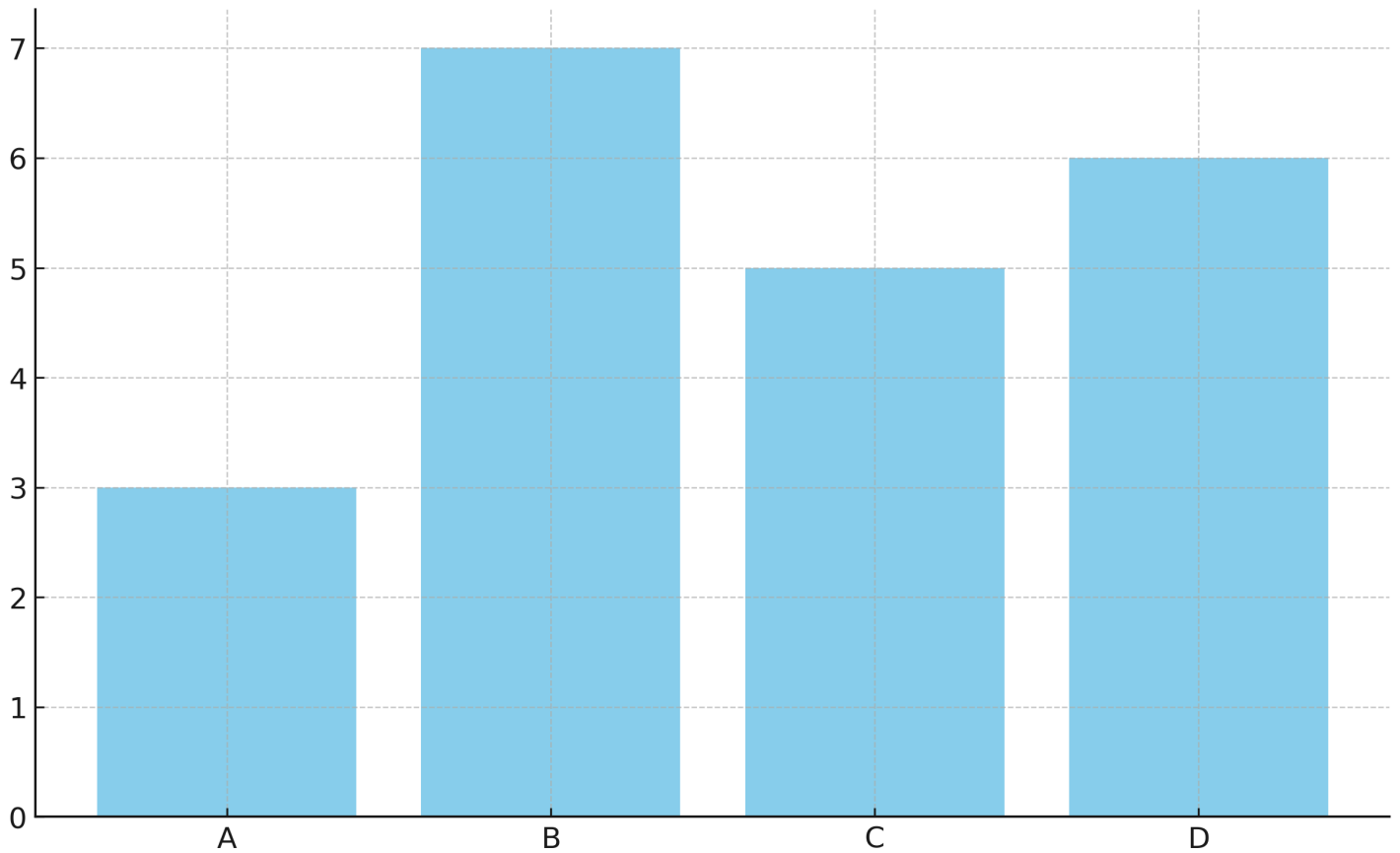
plt.plot(x,y, color="blue", marker="o")
plt.title("Line Plot")
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.show()
```

📌 **When to use?** → Growth, stock prices, time series

(b) Bar Chart

👉 Used to **compare categories**.

Bar Chart

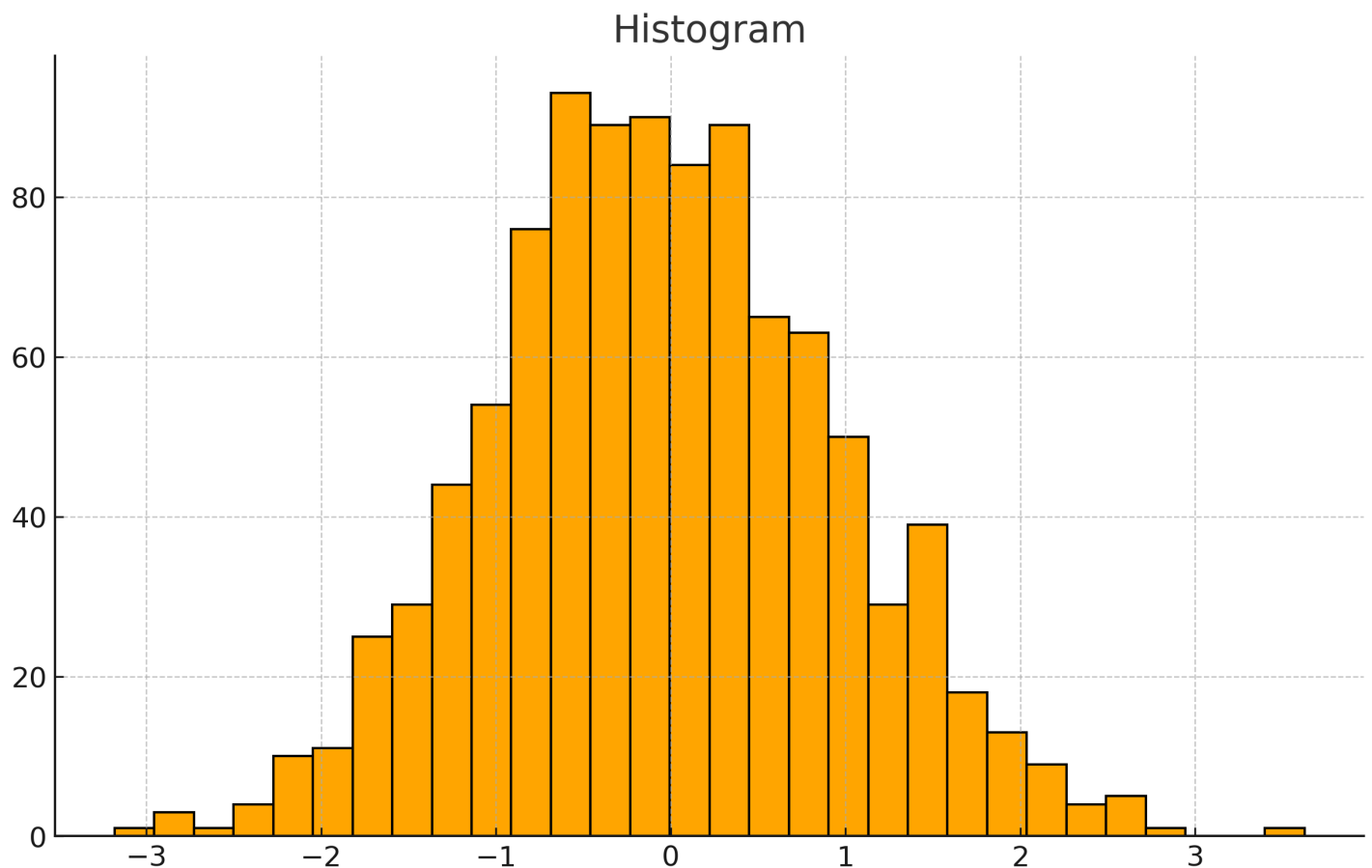


```
categories = ['A','B','C','D']  
values = [3,7,5,6]  
  
plt.bar(categories, values, color='skyblue')  
plt.title("Bar Chart")  
plt.show()
```

📌 **When to use?** → Compare sales by product, scores by group

(c) Histogram

👉 Shows **distribution of values**.



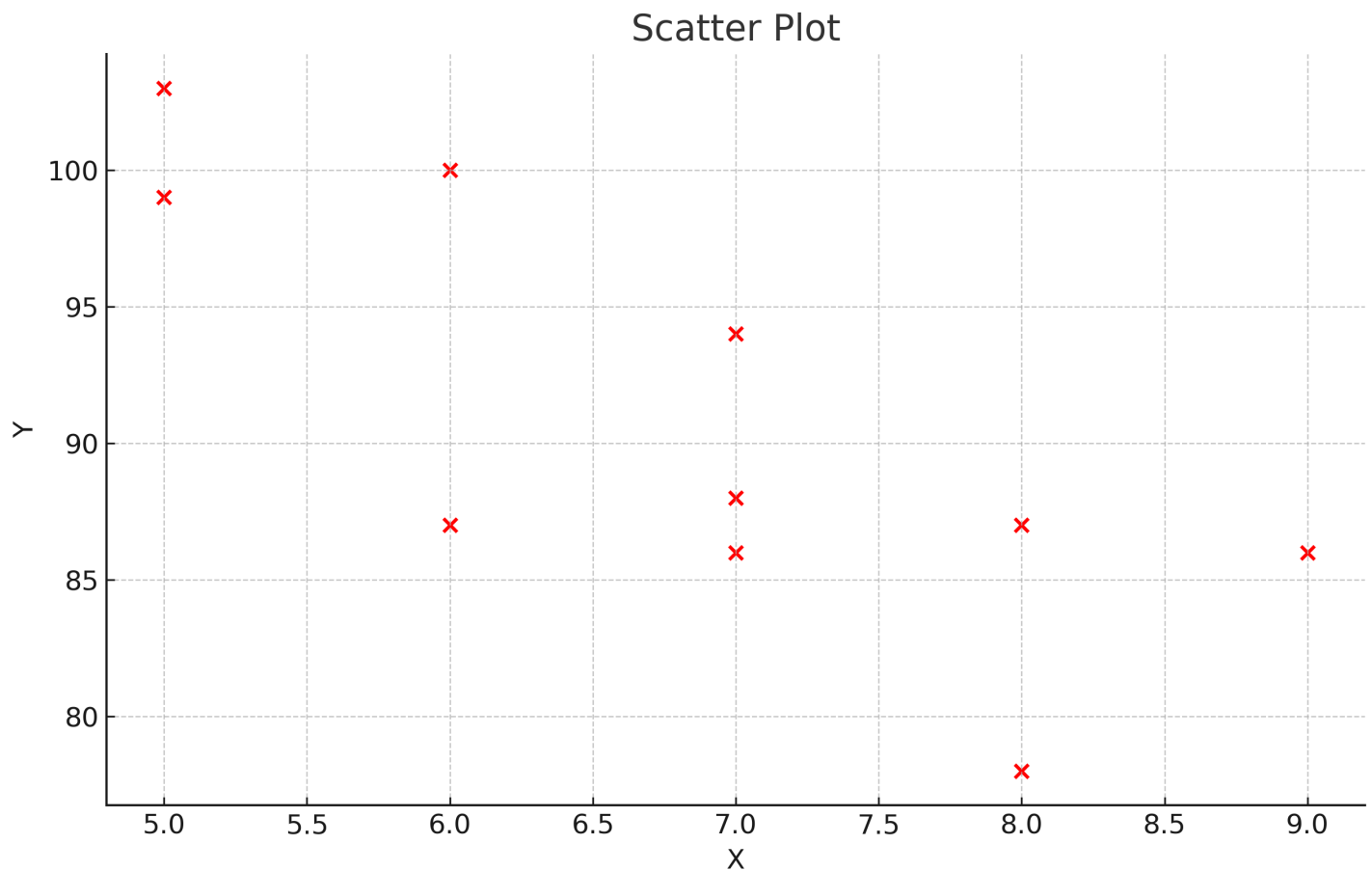
```
import numpy as np
data = np.random.randn(1000)

plt.hist(data, bins=30, color='orange', edgecolor='black')
plt.title("Histogram")
plt.show()
```

📌 **When to use?** → Age distribution, exam marks, frequency

(d) Scatter Plot

👉 Shows **relationship between two variables**.



```
x = [5,7,8,7,6,9,5,6,7,8]
y = [99,86,87,88,100,86,103,87,94,78]
```

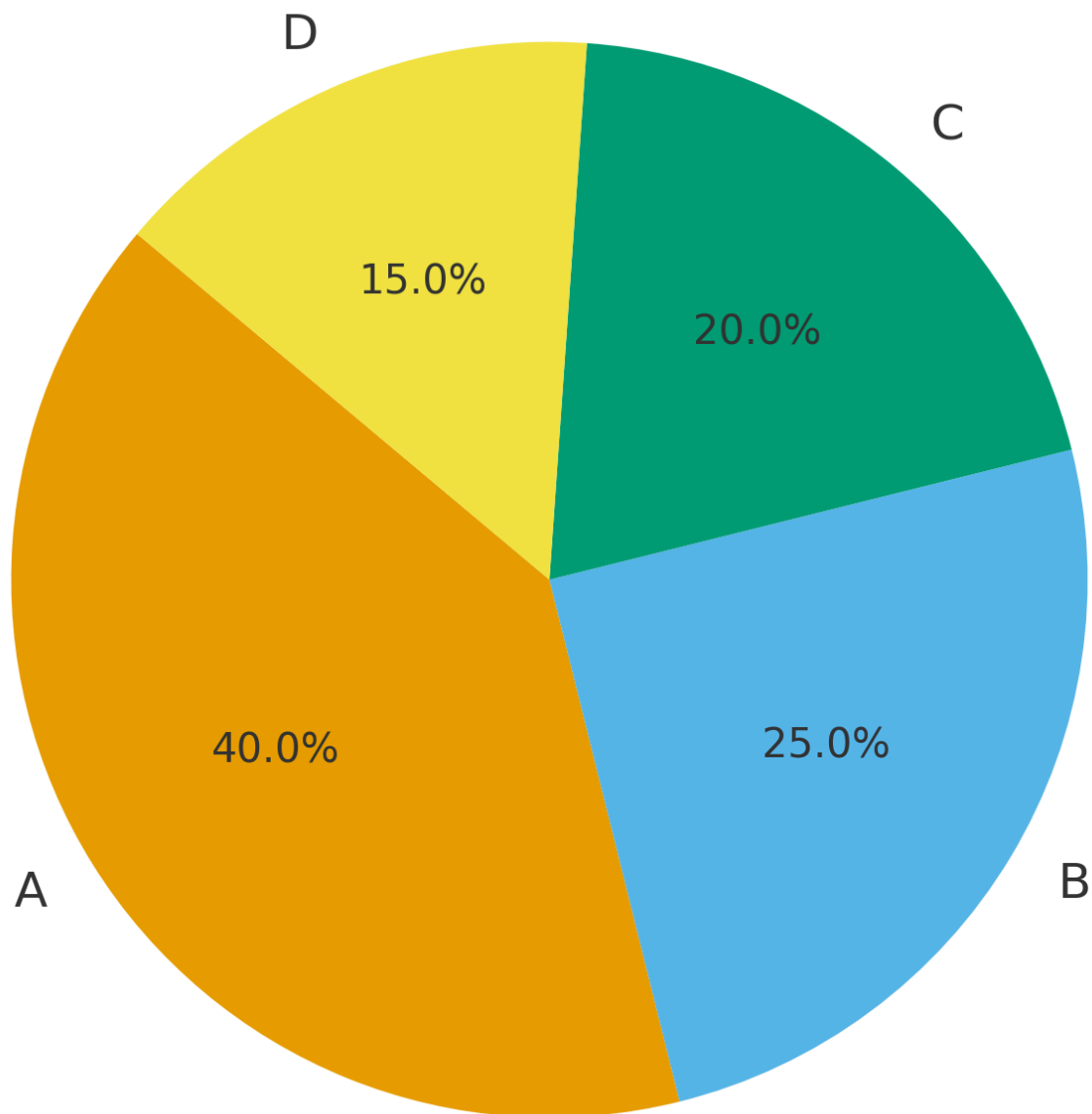
```
plt.scatter(x, y, color='red')
plt.title("Scatter Plot")
plt.xlabel("X")
plt.ylabel("Y")
plt.show()
```

📌 **When to use?** → Height vs Weight, Salary vs Experience

(e) Pie Chart

👉 Shows **proportions of categories**.

Pie Chart

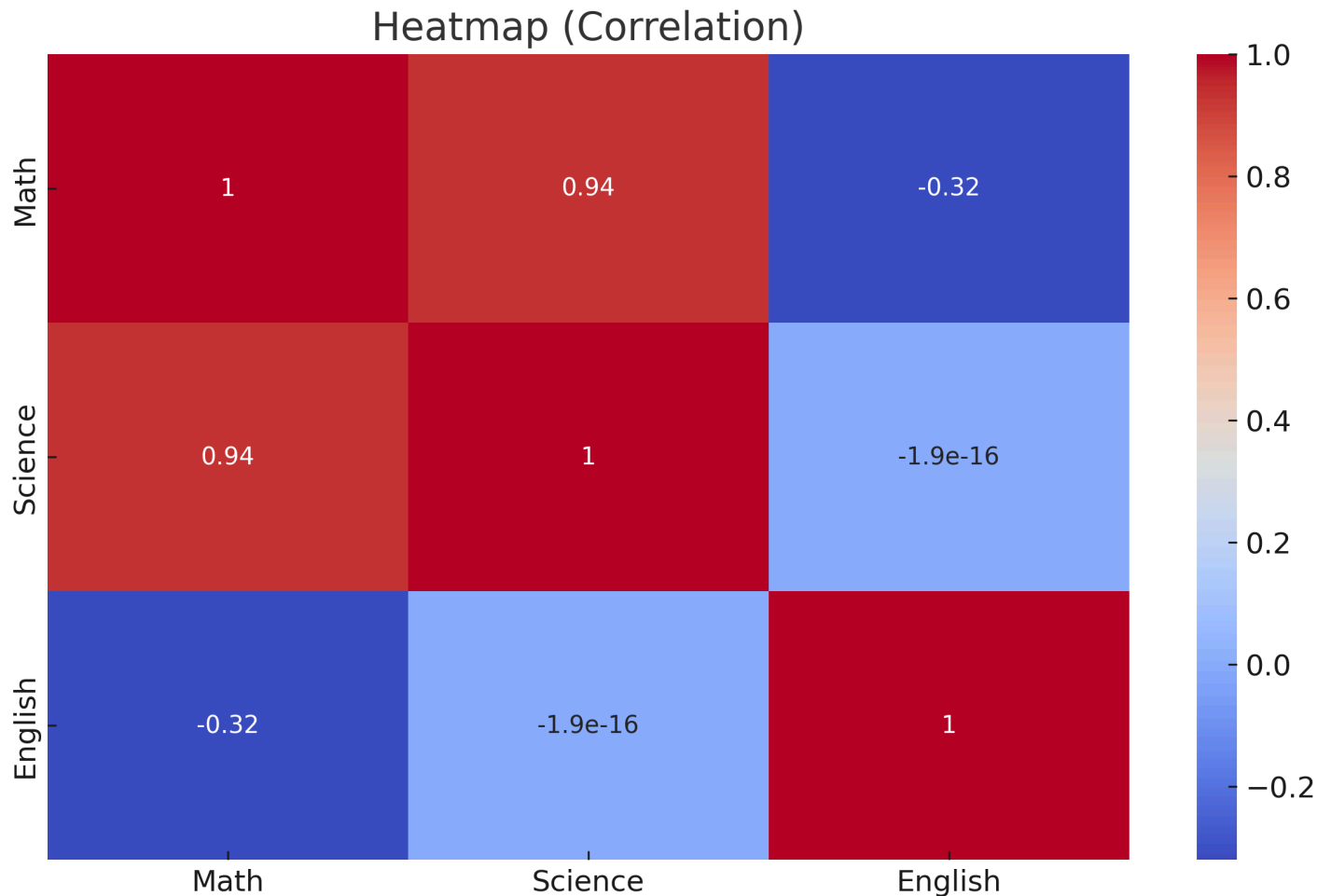


```
sizes = [40, 25, 20, 15]
labels = ["A","B","C","D"]

plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=140)
plt.title("Pie Chart")
plt.show()
```

📌 **When to use?** → Market share, percentage distribution

◆ 4. Matplotlib Customization



```
plt.plot(x, y, color='green', linestyle='--', marker='o')
plt.title("Customized Plot", fontsize=16, color="red")
plt.xlabel("X-axis", fontsize=12)
plt.ylabel("Y-axis", fontsize=12)
plt.grid(True)
plt.legend(["Line"])
plt.show()
```

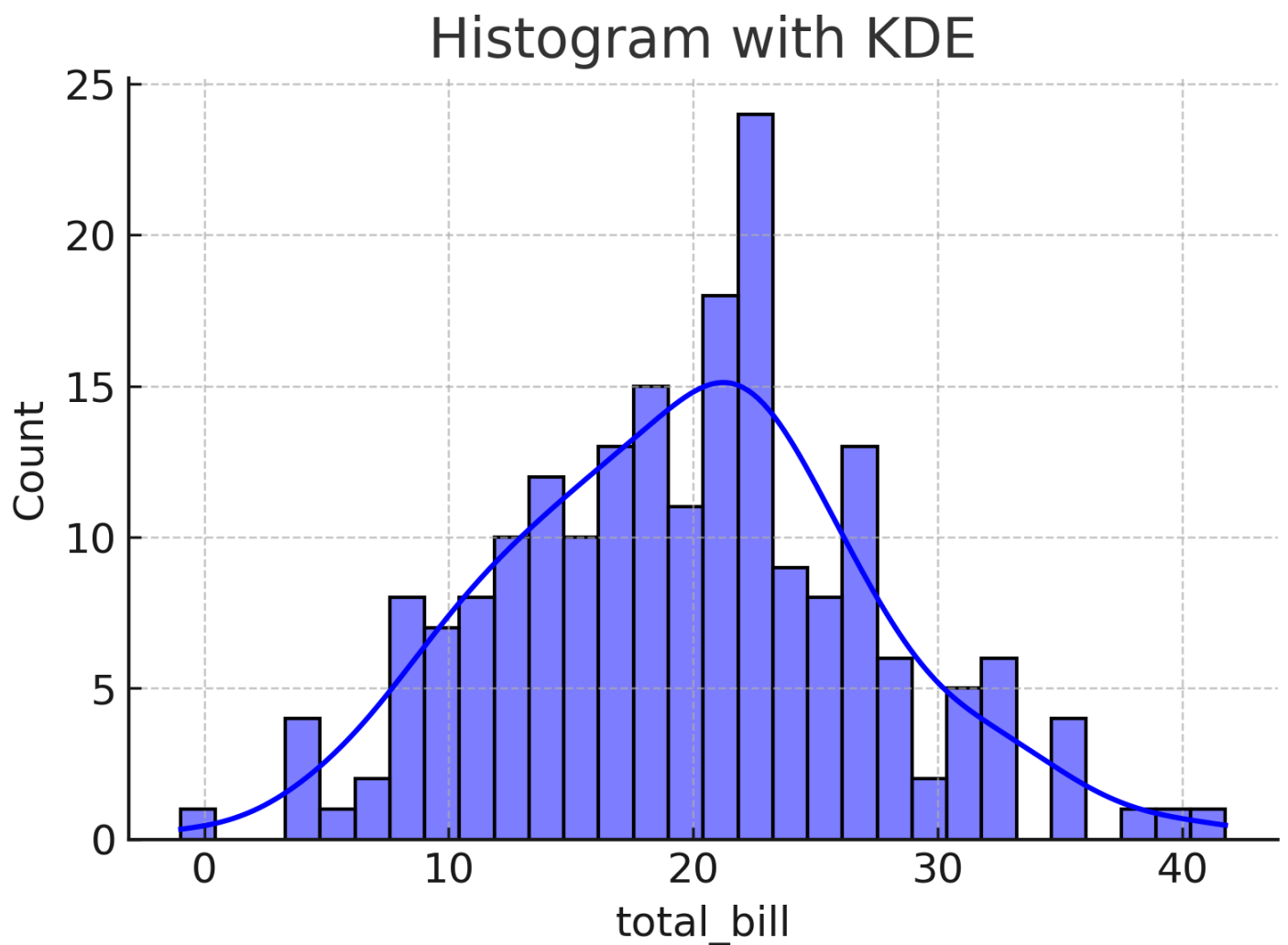
📌 Features: **color, labels, markers, legends, gridlines**

Part B: Seaborn

5. Seaborn Basics

```
import seaborn as sns
```

(a) Histogram with KDE

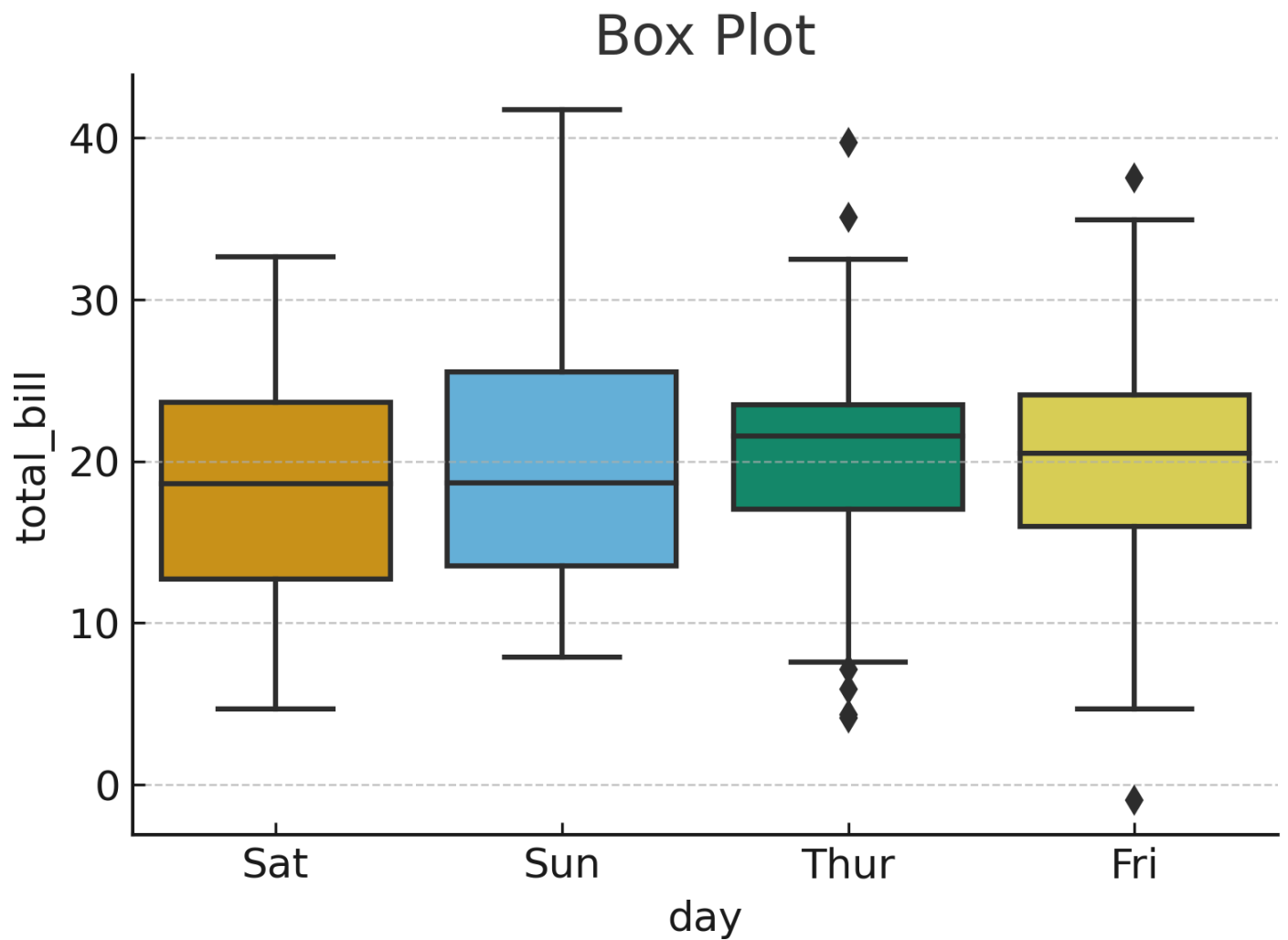


```
sns.histplot(data, kde=True, bins=30, color="blue")
```

📌 Adds **smooth curve** for distribution

(b) Box Plot (Whisker Plot)

👉 Shows **spread & outliers**

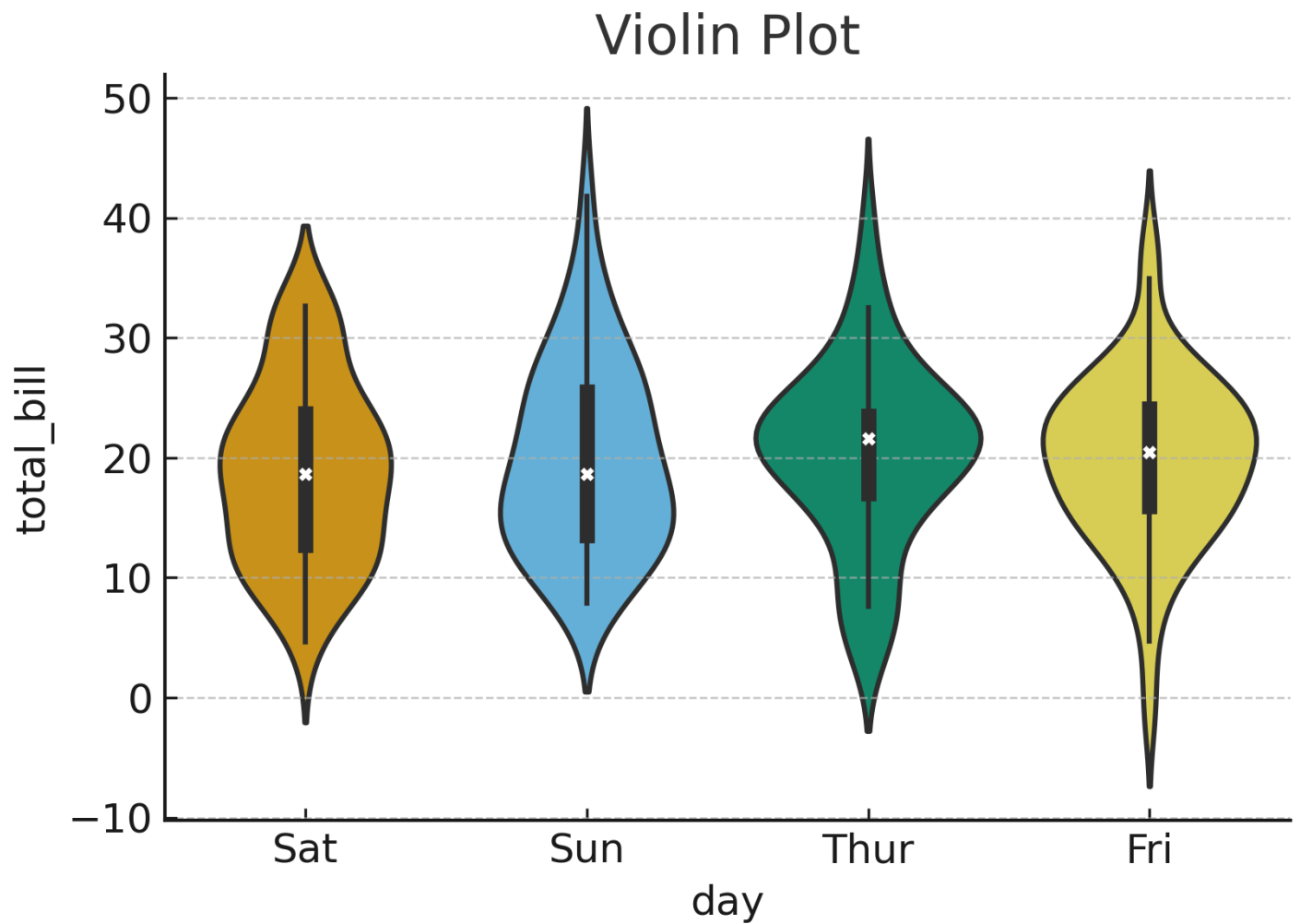


```
sns.boxplot(x=categories, y=values)
```

📌 Median, quartiles, and outliers visible

(c) Violin Plot

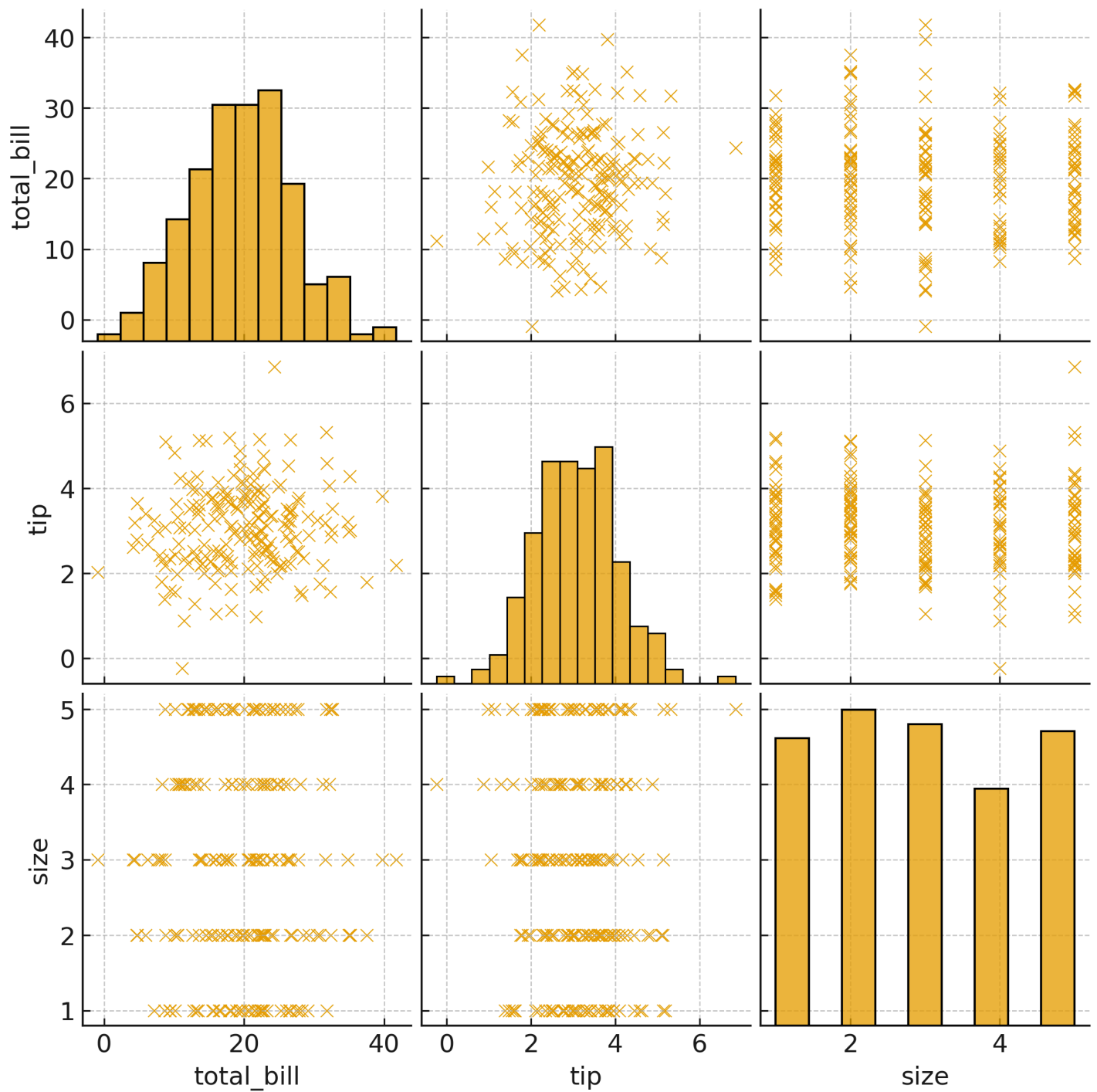
👉 Similar to boxplot but shows **distribution shape**



```
sns.violinplot(x=categories, y=values)
```

(d) Pairplot

👉 Multiple scatter plots for all features

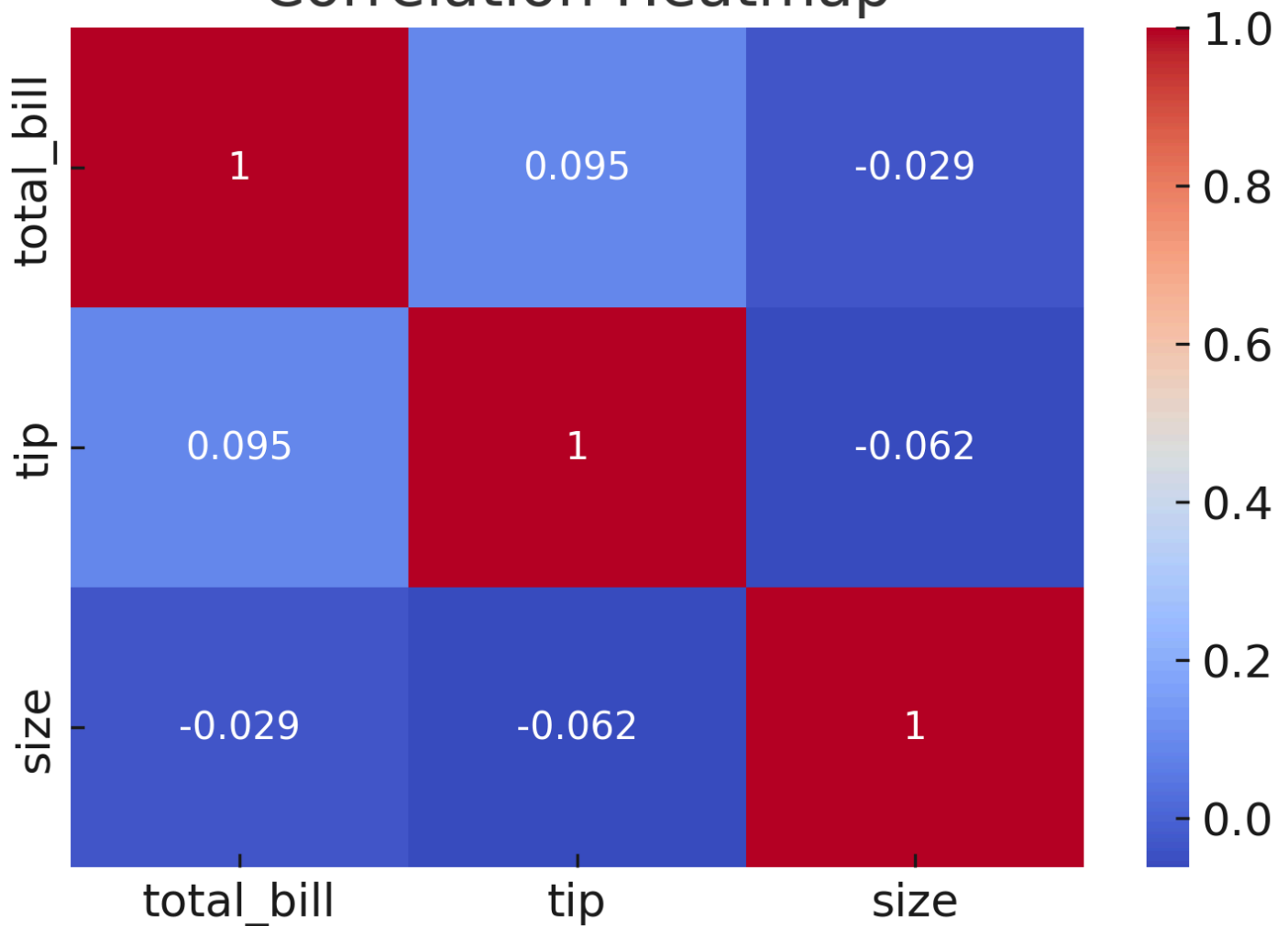


```
sns.pairplot(df)
```

(e) Heatmap

👉 Shows **correlation between variables**

Correlation Heatmap



```
corr = df.corr()  
sns.heatmap(corr, annot=True, cmap="coolwarm")
```

📌 Colors indicate strength of relationship

Part C: Choosing the Right Chart

Line Plot → Trend over time

Bar Chart → Compare categories

Histogram → Distribution of values

Scatter Plot → Relationship between variables

Pie Chart → Percentage/proportion

Box Plot → Outliers & spread

Heatmap → Correlation

✓ Quick Recall (Cheat Sheet)

Matplotlib → plot, bar, hist, scatter, pie

Customization → color, linestyle, marker, legend, grid

Seaborn → histplot, boxplot, violinplot, pairplot, heatmap

Use Case Guide → Trend = Line, Comparison = Bar, Distribution = Histogram,
Relation = Scatter