```
!pip install matplotlib seaborn pandas
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
```

```
Requirement already satisfied: matplotlib in /usr/local/lib/python3.11/dist-packages (3.10.0)
Requirement already satisfied: seaborn in /usr/local/lib/python3.11/dist-packages (0.13.2)
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packages (2.2.2)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (1.3.2)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (4.58.4)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (1.4.8)
Requirement already satisfied: numpy>=1.23 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (2.0.2)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (24.2)
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (11.2.1)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (3.2.3)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas) (2025.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.7->matplotlib) (1.17.0)
```

Start coding or generate with AI.

Library Overview

Matplotlib

Matplotlib is a popular plotting library for static graphs

Seaborn

Seaborn builds on Matplotlib and adds support for statistical plots

MATPLOTLIB GRAPHS

1.Line Chart

Description: A line chart connects data points with straight lines. It's ideal for showing trends over time or continuous data.

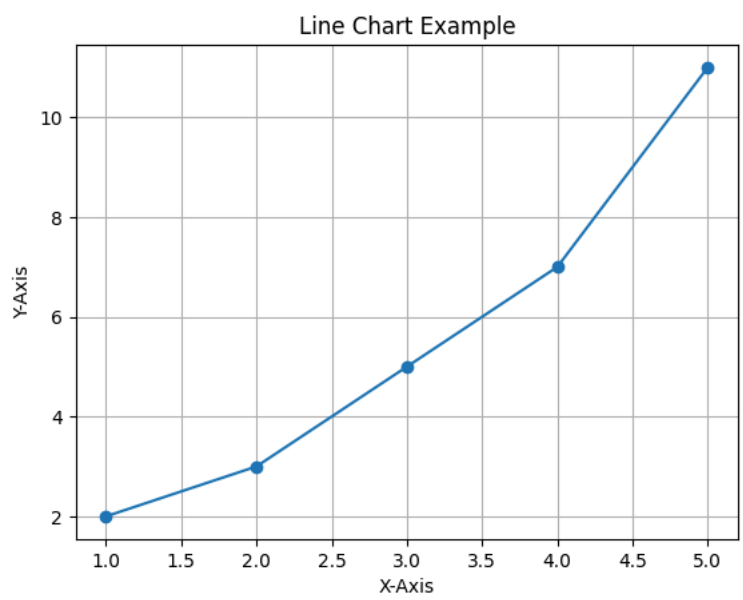
Use Case: Visualizing stock prices over months

Monitoring temperature changes during the day

```
import matplotlib.pyplot as plt

# Data
x = [1, 2, 3, 4, 5]
y = [2, 3, 5, 7, 11]

# Plot
plt.plot(x, y, marker='o')
plt.title("Line Chart Example")
plt.xlabel("X-Axis")
plt.ylabel("Y-Axis")
plt.grid(True)
plt.show()
```



2. Scatter Plot

Description: A scatter plot displays individual data points plotted by two variables, useful for showing relationships or distribution.

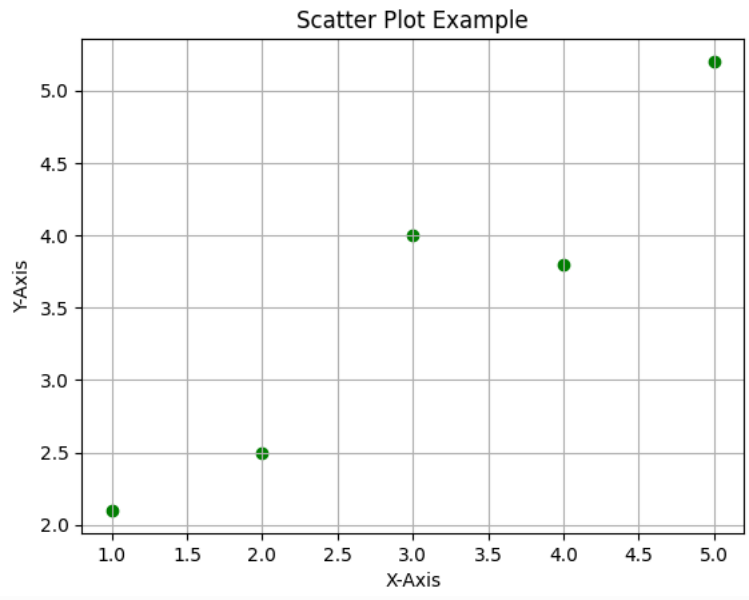
Use Case: Analyzing the correlation between height and weight

Detecting outliers in datasets

```
import matplotlib.pyplot as plt

# Data
x = [1, 2, 3, 4, 5]
y = [2.1, 2.5, 4.0, 3.8, 5.2]

# Plot
plt.scatter(x, y, color='green')
plt.title("Scatter Plot Example")
plt.xlabel("X-Axis")
plt.ylabel("Y-Axis")
plt.grid(True)
plt.show()
```



3. Bar Chart

Description: A bar chart uses rectangular bars to compare categories or groups.

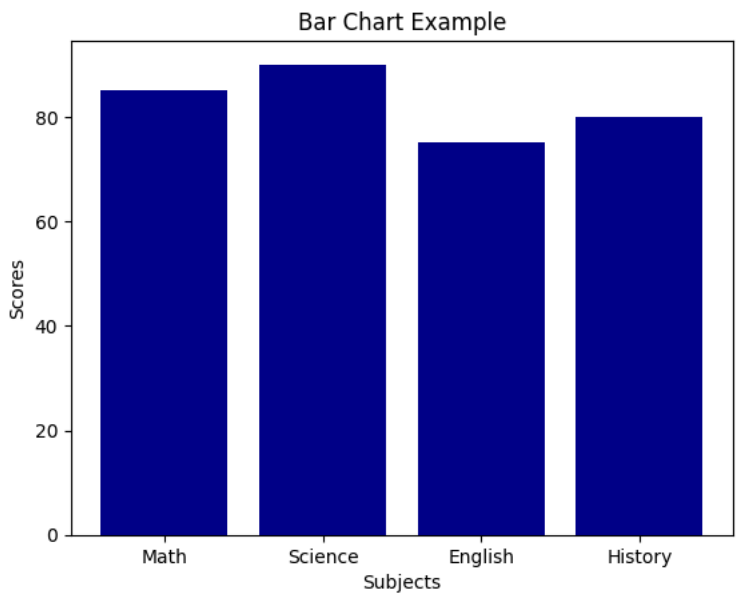
Use Case: Comparing sales across regions

Showing the number of students in each grade

```
import matplotlib.pyplot as plt

# Data
categories = ['Math', 'Science', 'English', 'History']
scores = [85, 90, 75, 80]

# Plot
plt.bar(categories, scores, color='darkblue')
plt.title("Bar Chart Example")
plt.xlabel("Subjects")
plt.ylabel("Scores")
plt.show()
```



4. Pie Chart

Description: A pie chart shows parts of a whole as slices of a circle, best for visualizing percentage or proportion data.

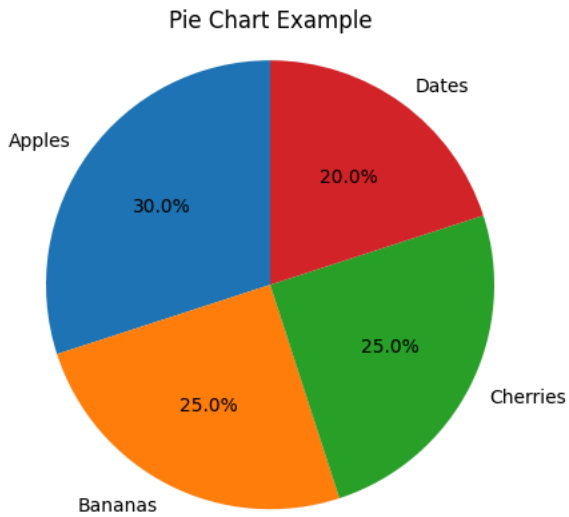
Use Case: Showing market share of companies

Representing population distribution by age group

```
import matplotlib.pyplot as plt

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

# Plot
plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=90)
plt.title("Pie Chart Example")
plt.axis('equal')  # Equal aspect ratio ensures a circular pie.
plt.show()
```



SEABORN GRAPHS

1. Line Chart

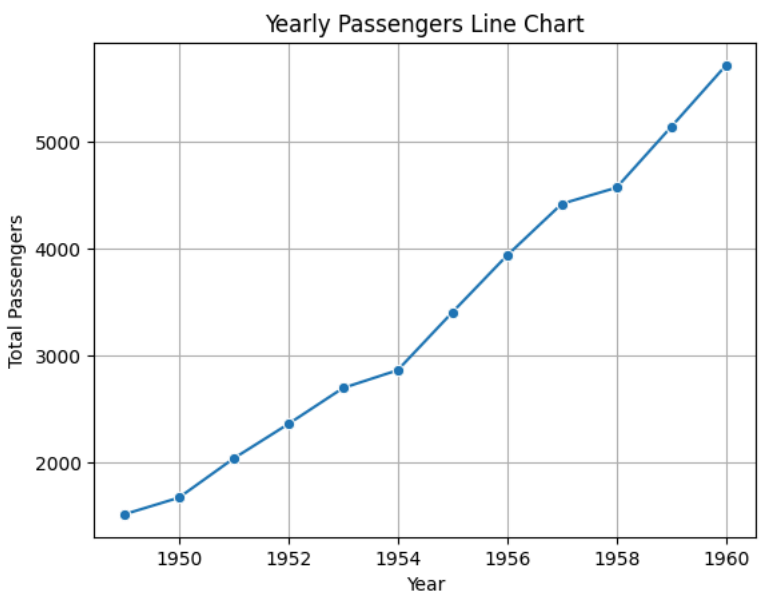
Description: Seaborn's line plot is used to visualize data trends with optional confidence intervals (error bands).

Use Case: Analyzing trends in revenue or number of users over time

Visualizing averages with confidence levels

```
# Example using the 'flights' dataset
flights = sns.load_dataset("flights")
yearly = flights.groupby("year")["passengers"].sum().reset_index()

sns.lineplot(x="year", y="passengers", data=yearly, marker='o')
plt.title("Yearly Passengers Line Chart")
plt.xlabel("Year")
plt.ylabel("Total Passengers")
plt.grid(True)
plt.show()
```



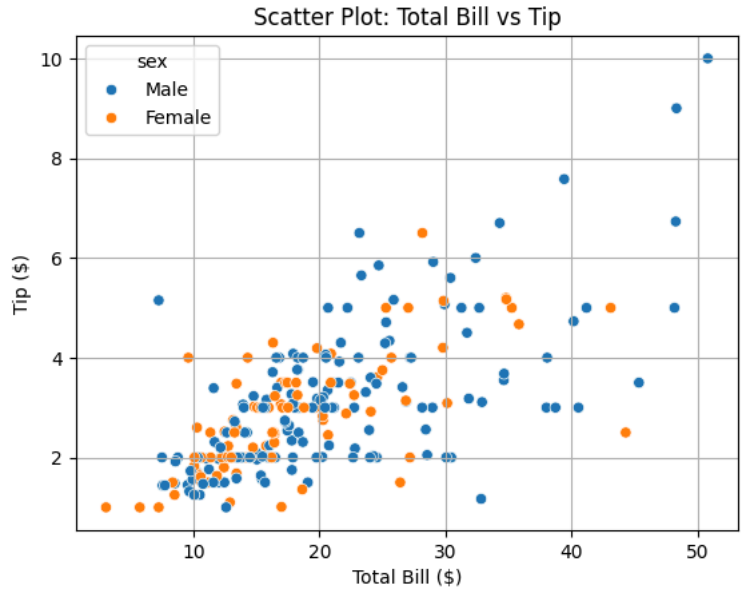
2. Scatter Plot

Description: Displays individual data points to observe the relationship between two continuous variables, with optional grouping by hue/style.

Use Case: Identifying correlation between bill amount and tip

Visualizing clustering of data points by category

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



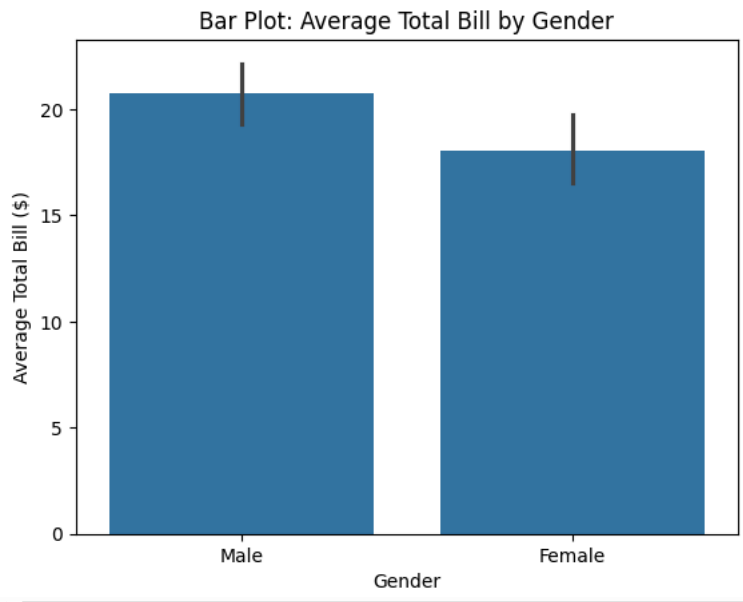
3. Bar Plot

Description: Shows the average value of a variable for different categories, with built-in error bars for confidence intervals.

Use Case: Comparing average total bills between male and female customers

Visualizing group-wise averages

```
sns.barplot(x="sex", y="total_bill", data=tips)
plt.title("Bar Plot: Average Total Bill by Gender")
plt.xlabel("Gender")
plt.ylabel("Average Total Bill ($)")
plt.show()
```



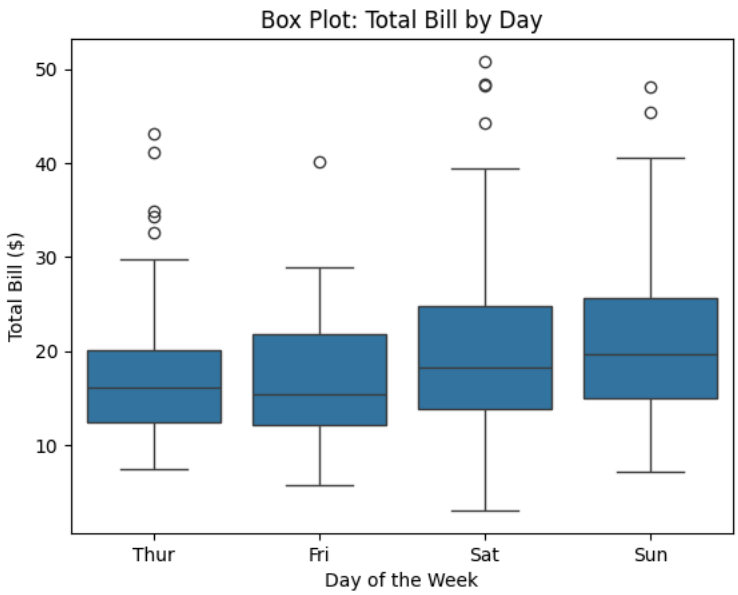
4. Box Plot

Description: Box plots display the distribution of data through quartiles, highlighting the median, interquartile range, and outliers.

Use Case: Comparing bill distributions across days

Detecting outliers and understanding spread

```
sns.boxplot(x="day", y="total_bill", data=tips)
plt.title("Box Plot: Total Bill by Day")
plt.xlabel("Day of the Week")
plt.ylabel("Total Bill ($)")
plt.show()
```



COMPARISON BETWEEN MATPLOTLIB AND SEABORN

Ease of Use

Matplotlib provides a lot of flexibility but can be a bit verbose and complex, especially for beginners. You often need several lines of code just to create a basic plot. Seaborn, on the other hand, simplifies plotting by offering a higher-level API. It requires less code and automatically handles many visual elements like color palettes and labels.

Verdict: Seaborn is easier to use for quick and clean plots.

Customization

Matplotlib is highly customizable. You can control every single detail of the plot—line styles, tick marks, grid lines, labels, font size, and more. Seaborn offers beautiful default styles and color themes, but customization is more limited unless you dive into Matplotlib-level tweaks.

Verdict: Use Matplotlib if you need fine-grained customization.

Graph Variety

Both libraries support essential plot types like line charts, bar plots, and scatter plots. However, Matplotlib supports a wider range of general-purpose plots, including pie charts, error bars, and 3D plots. Seaborn specializes in statistical and distribution plots like box plots, violin plots, histograms, and heatmaps.

Verdict: Use Matplotlib for general plotting needs, Seaborn for statistical data visualization.

Performance

Both libraries are efficient and can handle medium to large datasets fairly well. Since Seaborn is built on top of Matplotlib, there's no significant difference in speed for standard use cases.

Verdict: Both perform well; performance isn't a deciding factor for most users.

Interactivity

Neither Matplotlib nor Seaborn is designed for interactive charts by default. Their plots are typically static. For interactivity like zooming or