

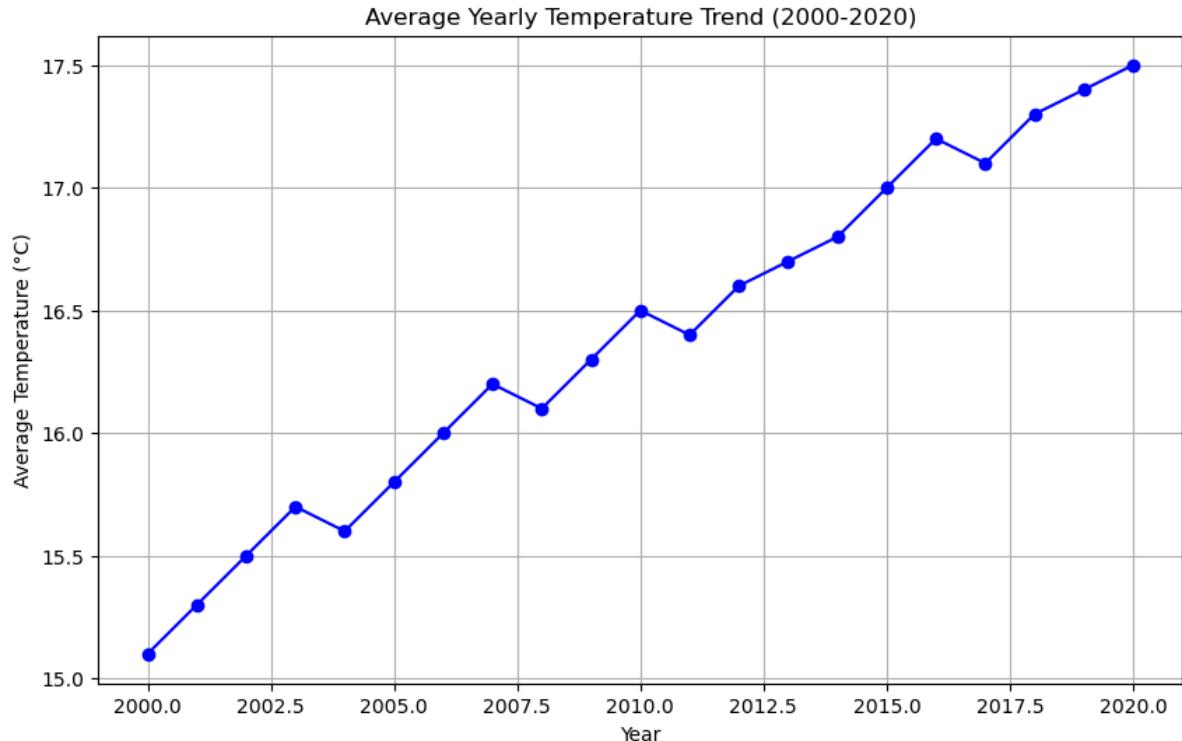
240701160 GURU PRASAD V FDS WEEK 05

Q1. Line Plot from CSV (Temperature Trends) Download or create a CSV file weather.csv with the following columns: Year, AvgTemperature. Write a Python program using Pandas and Matplotlib to plot a line chart showing the trend of average yearly temperature from 2000 to 2020. • Add markers on the line. • Label the axes (Year, Average Temperature (°C)) and add a title.

```
In [1]: import pandas as pd
data = {
    'Year': list(range(2000, 2021)),
    'AvgTemperature': [15.1, 15.3, 15.5, 15.7, 15.6, 15.8, 16.0, 16.2, 16.1, 16.4, 16.6, 16.7, 16.8, 17.0, 17.2, 17.1, 17.3, 17.4, 17.5]
}
df = pd.DataFrame(data)
df.to_csv('weather.csv', index=False)
print("weather.csv created successfully!")
```

weather.csv created successfully!

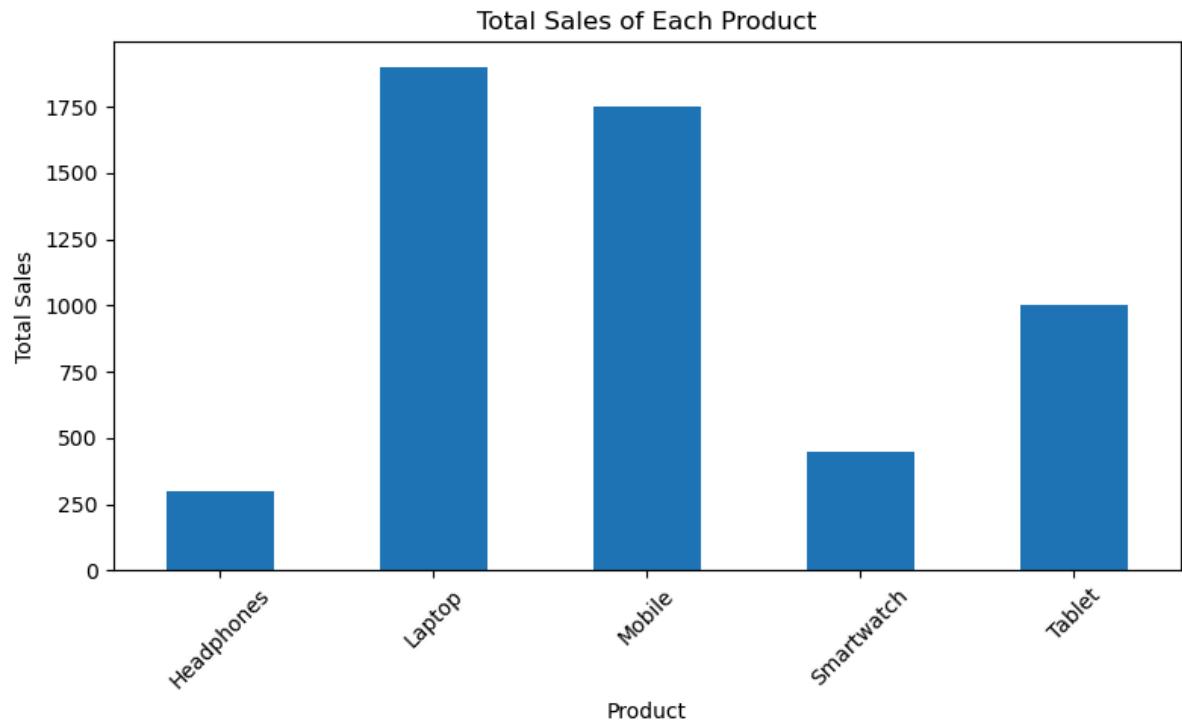
```
In [2]: import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv('weather.csv')
df = df[(df['Year'] >= 2000) & (df['Year'] <= 2020)]
plt.figure(figsize=(10, 6))
plt.plot(df['Year'], df['AvgTemperature'], marker='o', linestyle='-', color='blue')
plt.xlabel('Year')
plt.ylabel('Average Temperature (°C)')
plt.title('Average Yearly Temperature Trend (2000-2020)')
plt.grid(True)
plt.show()
```



Q2. Bar Chart from CSV (Sales Data) Given a CSV file sales.csv with columns: Product, Sales. Write a Python program to plot a bar chart showing the total sales of each product. • Add axis labels and a title. • Rotate product names if necessary for better visibility.

```
In [7]: import pandas as pd
import matplotlib.pyplot as plt
data = {
    "Product": ["Laptop", "Mobile", "Tablet", "Headphones", "Smartwatch", "Mobile", "Tablet"],
    "Sales": [1200, 800, 600, 300, 450, 950, 700, 400]
}
df = pd.DataFrame(data)
df.to_csv("sales.csv", index=False)
print("sales.csv file created successfully!")
sales_data = pd.read_csv("sales.csv")
grouped_sales = sales_data.groupby("Product")["Sales"].sum()
plt.figure(figsize=(8, 5))
grouped_sales.plot(kind="bar")
plt.xlabel("Product")
plt.ylabel("Total Sales")
plt.title("Total Sales of Each Product")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

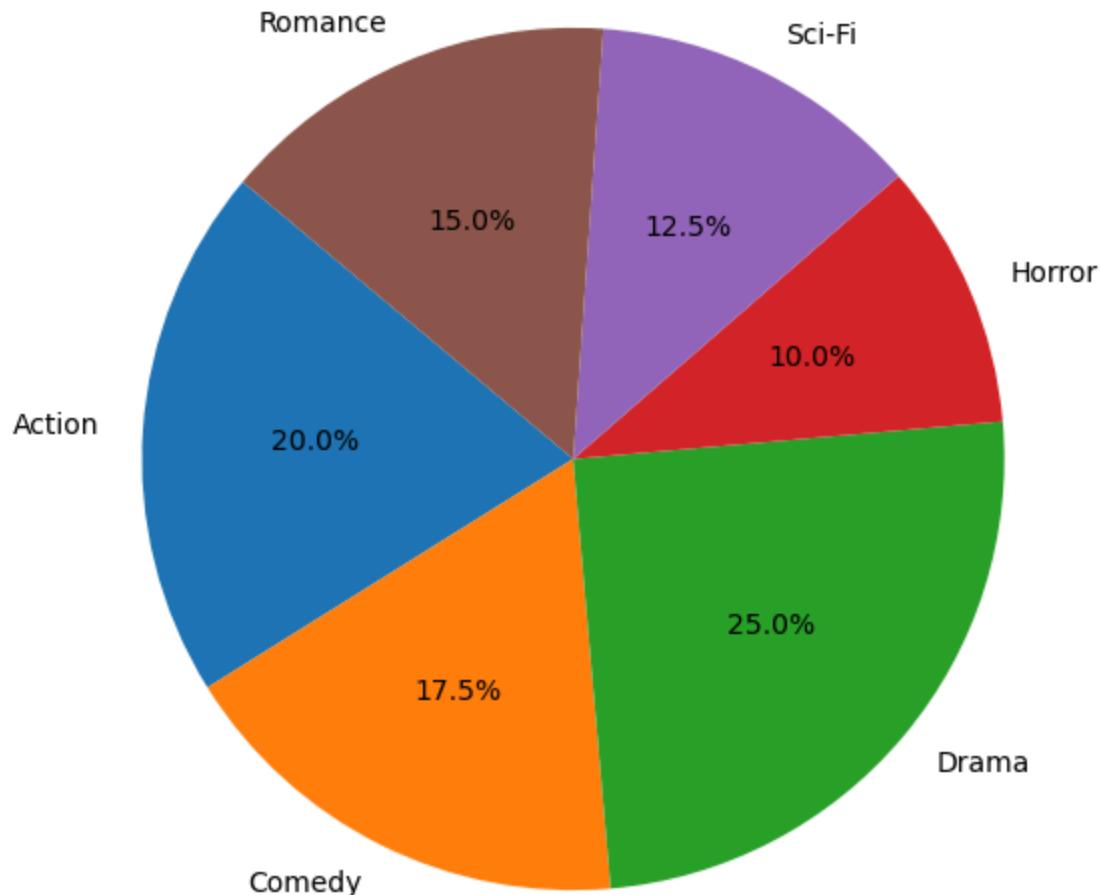
sales.csv file created successfully!



Q3. Pie Chart (Movie Genres Distribution) Create a CSV file movies.csv with columns: Genre, Count. Write a Python program to plot a pie chart showing the percentage distribution of movie genres. • Add labels and percentage values to each slice. • Add a title: "Distribution of Movie Genres".

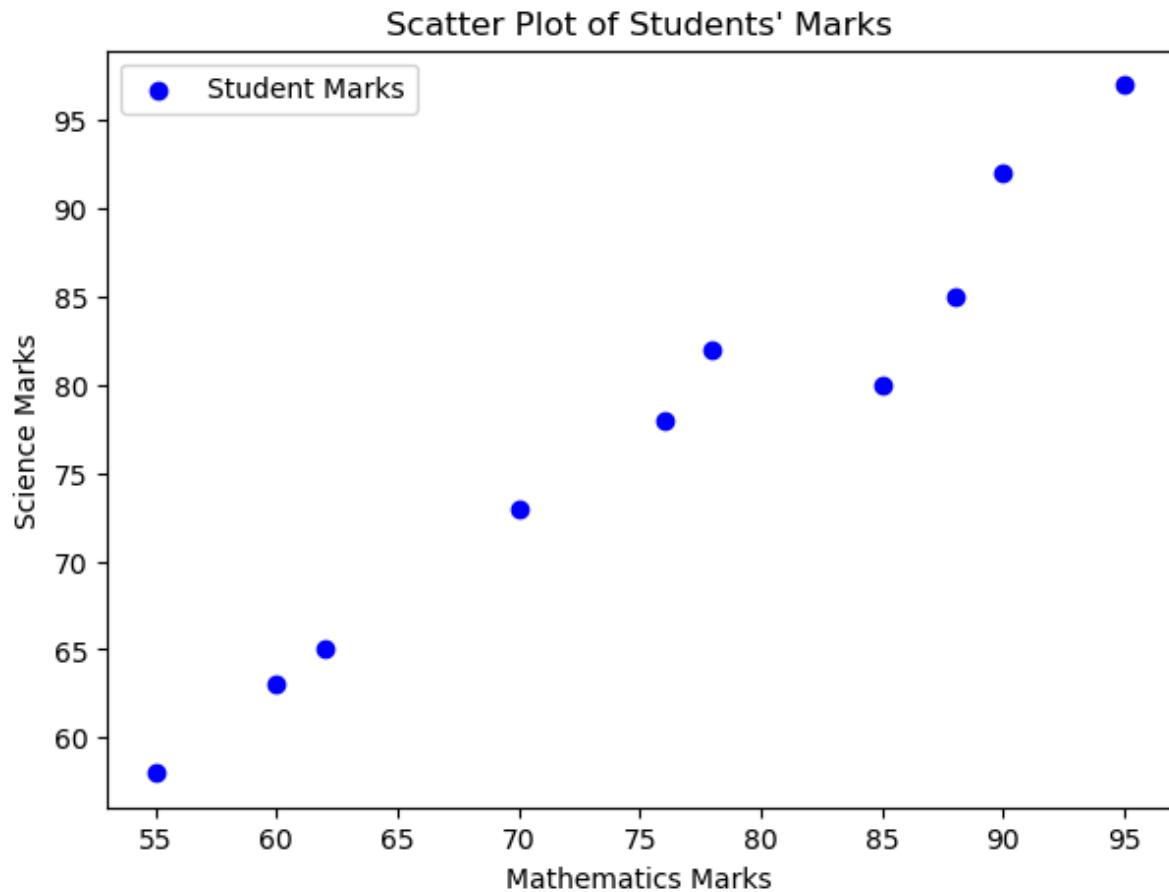
```
In [9]: import pandas as pd
import matplotlib.pyplot as plt
data = {
    "Genre": ["Action", "Comedy", "Drama", "Horror", "Sci-Fi", "Romance"],
    "Count": [40, 35, 50, 20, 25, 30]
}
df = pd.DataFrame(data)
df.to_csv("movies.csv", index=False)
movies_data = pd.read_csv("movies.csv")
plt.figure(figsize=(7, 7))
plt.pie(
    movies_data["Count"],
    labels=movies_data["Genre"],
    autopct="%1.1f%%",
    startangle=140
)
plt.title("Distribution of Movie Genres")
plt.show()
```

Distribution of Movie Genres



Q4. Scatter Plot (Students' Marks) Given a CSV file students.csv with columns: MathMarks, ScienceMarks. Write a Python program to plot a scatter plot showing the relationship between marks in Mathematics and Science. • Add axis labels and a title. • Use different colors/markers if possible.

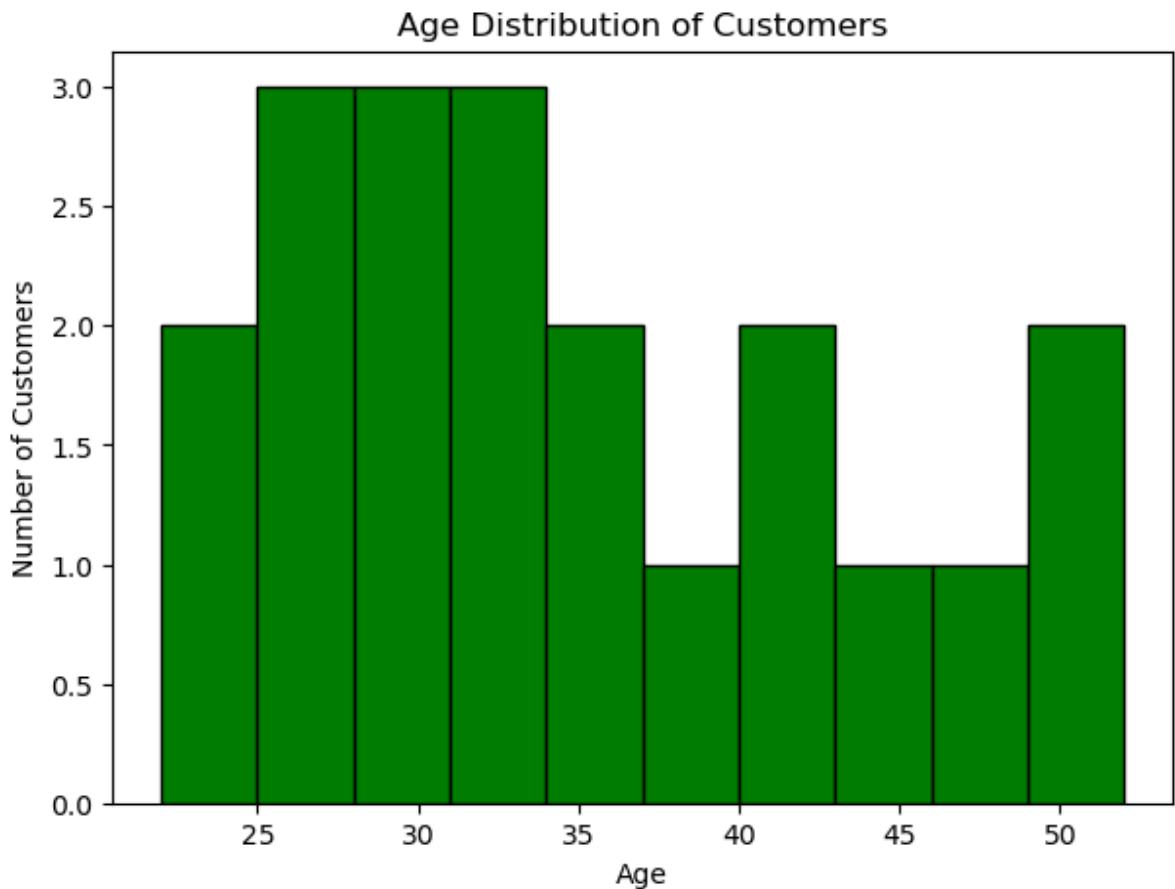
```
In [11]: import pandas as pd
import matplotlib.pyplot as plt
data = {
    "MathMarks": [78, 85, 62, 90, 55, 70, 88, 95, 60, 76],
    "ScienceMarks": [82, 80, 65, 92, 58, 73, 85, 97, 63, 78]
}
df = pd.DataFrame(data)
df.to_csv("students.csv", index=False)
students_data = pd.read_csv("students.csv")
plt.figure(figsize=(7, 5))
plt.scatter(students_data["MathMarks"], students_data["ScienceMarks"],
            color="blue", marker="o", label="Student Marks")
plt.xlabel("Mathematics Marks")
plt.ylabel("Science Marks")
plt.title("Scatter Plot of Students' Marks")
plt.legend()
plt.show()
```



Q5. Histogram (Customer Ages) Given a CSV file `customers.csv` with a column: Age. Write a Python program to plot a histogram showing the age distribution of customers.

- Use 10 bins.
- Add axis labels and a title.

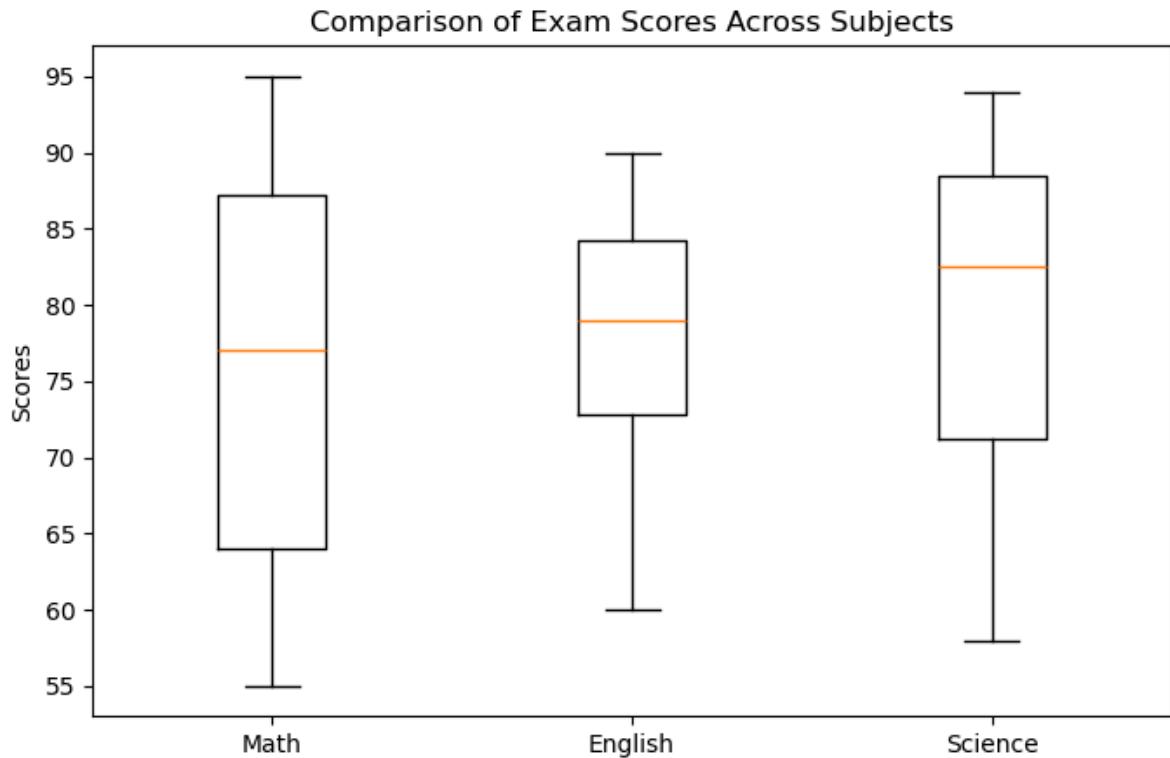
```
In [12]: import pandas as pd
import matplotlib.pyplot as plt
data = {
    "Age": [22, 25, 30, 45, 35, 40, 28, 50, 32, 29,
            41, 38, 27, 23, 36, 33, 48, 52, 31, 26]
}
df = pd.DataFrame(data)
df.to_csv("customers.csv", index=False)
customers_data = pd.read_csv("customers.csv")
plt.figure(figsize=(7, 5))
plt.hist(customers_data["Age"], bins=10, color="green", edgecolor="black")
plt.xlabel("Age")
plt.ylabel("Number of Customers")
plt.title("Age Distribution of Customers")
plt.show()
```



Q6. Boxplot (Exam Scores Comparison) Create a CSV file `exam_scores.csv` with columns: Math, English, Science. Write a Python program to plot a boxplot comparing score distributions across subjects.

- Add axis labels and a title.

```
In [15]: import pandas as pd
import matplotlib.pyplot as plt
data = {
    "Math": [78, 85, 62, 90, 55, 70, 88, 95, 60, 76],
    "English": [82, 80, 75, 88, 60, 72, 85, 90, 65, 78],
    "Science": [85, 87, 70, 92, 58, 75, 89, 94, 63, 80]
}
df = pd.DataFrame(data)
df.to_csv("exam_scores.csv", index=False)
scores_data = pd.read_csv("exam_scores.csv")
plt.figure(figsize=(8, 5))
plt.boxplot(
    [scores_data["Math"], scores_data["English"], scores_data["Science"]],
    tick_labels=["Math", "English", "Science"]
)
plt.ylabel("Scores")
plt.title("Comparison of Exam Scores Across Subjects")
plt.show()
```



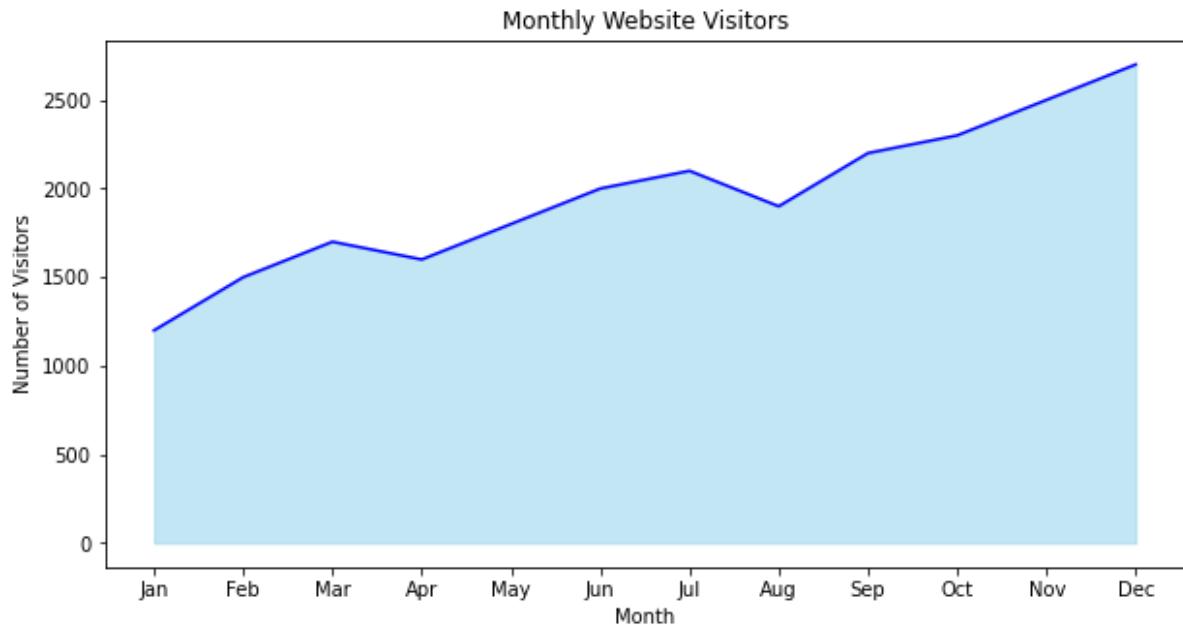
#### Q7. Area Chart (Website Traffic)

Create a CSV file traffic.csv with columns: Month, Visitors.

Write a Python program to plot an area chart showing monthly website visitors for a year.

- Add axis labels and a title.
- Use a different color for the filled area.

```
In [2]: import pandas as pd
import matplotlib.pyplot as plt
data = {
    "Month": ["Jan", "Feb", "Mar", "Apr", "May", "Jun",
              "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"],
    "Visitors": [1200, 1500, 1700, 1600, 1800, 2000,
                 2100, 1900, 2200, 2300, 2500, 2700]
}
df = pd.DataFrame(data)
df.to_csv("traffic.csv", index=False)
traffic_data = pd.read_csv("traffic.csv")
plt.figure(figsize=(10, 5))
plt.fill_between(traffic_data["Month"], traffic_data["Visitors"], color="skyblue")
plt.plot(traffic_data["Month"], traffic_data["Visitors"], color="blue")
plt.xlabel("Month")
plt.ylabel("Number of Visitors")
plt.title("Monthly Website Visitors")
plt.show()
```



#### Q8. Heatmap (Correlation Matrix)

Given a CSV file `students_scores.csv` with columns: Math, English, Science, History, Computer.

Write a Python program using Seaborn + Matplotlib to create a heatmap of the correlation between subjects.

- Add a title "Correlation Heatmap of Student Scores".

```
In [4]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

data = {
    "Math": [78, 85, 62, 90, 55, 70, 88, 95, 60, 76],
    "English": [82, 80, 75, 88, 60, 72, 85, 90, 65, 78],
    "Science": [85, 87, 70, 92, 58, 75, 89, 94, 63, 80],
    "History": [70, 75, 65, 80, 60, 68, 78, 85, 62, 74],
    "Computer": [90, 88, 75, 95, 65, 80, 92, 98, 70, 85]
}

df = pd.DataFrame(data)
df.to_csv("students_scores.csv", index=False)

scores_data = pd.read_csv("students_scores.csv")
correlation = scores_data.corr()

plt.figure(figsize=(8, 6))
sns.heatmap(correlation, annot=True, cmap="coolwarm", linewidths=0.5)
plt.title("Correlation Heatmap of Student Scores")
plt.show()
```

