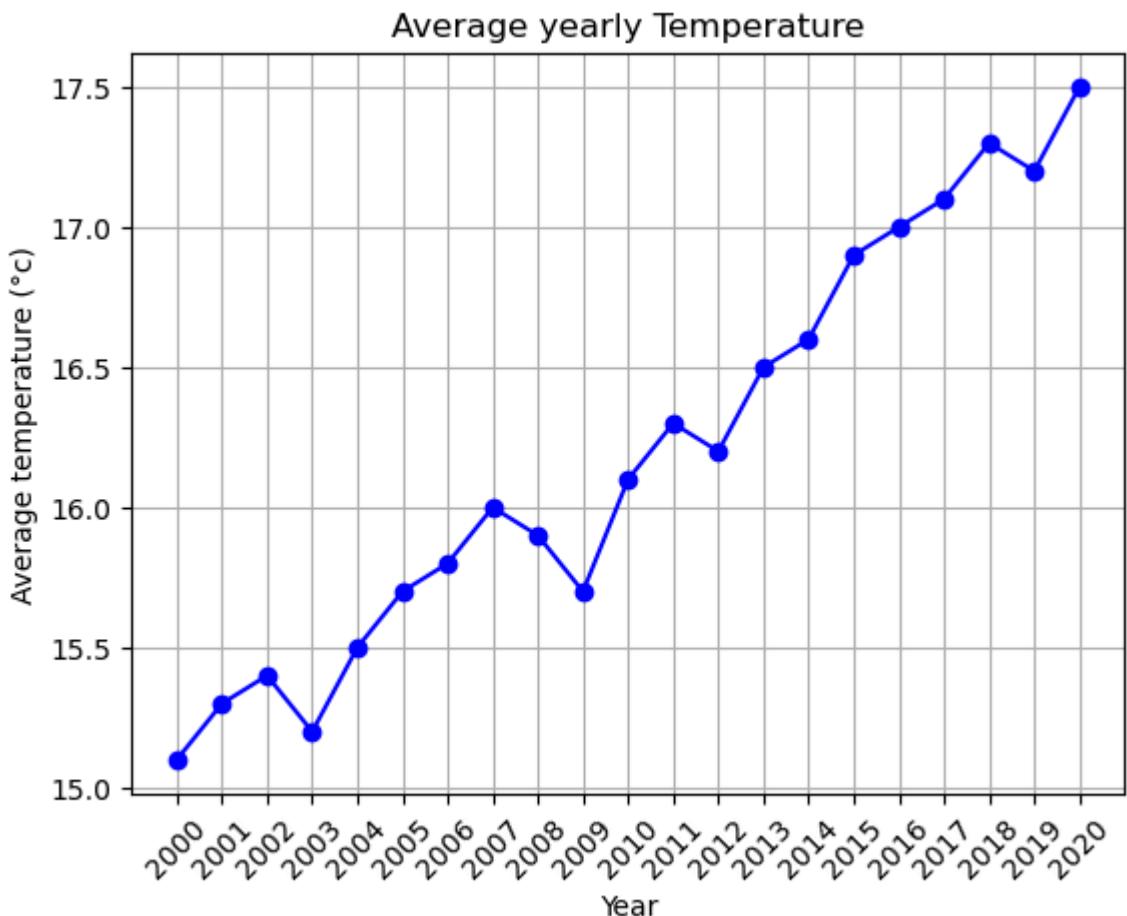


Q1. Line Plot from CSV (Temperature Trends) Download or create a CSV file weather.csv with the following columns: Year, AvgTemperature. Write a Python program using Pandas and Matplotlib to plot a line chart showing the trend of average yearly temperature from 2000 to 2020. • Add markers on the line.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

```
In [6]: import pandas as pd
import matplotlib.pyplot as plt
data = {
    'Year': list(range(2000,2021)),
    'AvgTemperature': [
        15.1, 15.3, 15.4, 15.2, 15.5,
        15.7, 15.8, 16.0, 15.9, 15.7,
        16.1, 16.3, 16.2, 16.5, 16.6,
        16.9, 17.0, 17.1, 17.3, 17.2,
        17.5
    ]
}
df=pd.DataFrame(data)
df.to_csv('weather.csv',index=False)
print("weather.csv file created using pandas.")
df=pd.read_csv('weather.csv')
plt.plot(df['Year'],df['AvgTemperature'],marker='o',linestyle='-',color='blue')
plt.xlabel('Year')
plt.ylabel('Average temperature (°c)')
plt.title('Average yearly Temperature')
plt.xticks(df['Year'], rotation=45)
plt.grid(True)
plt.show()
```

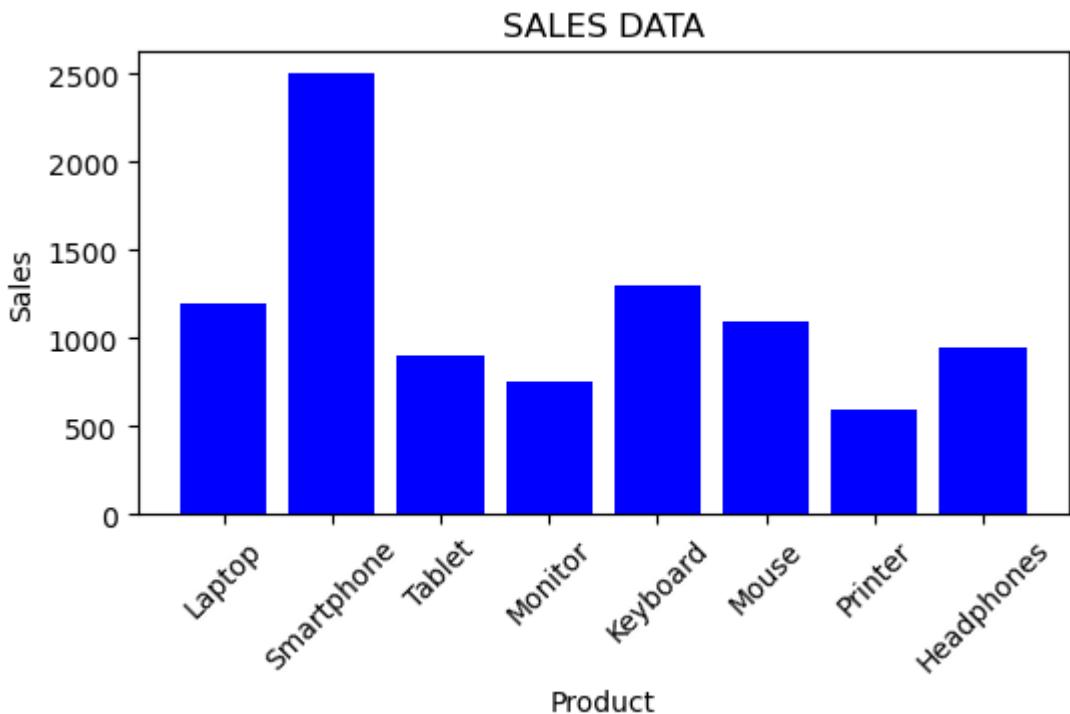
weather.csv file created using pandas.



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.

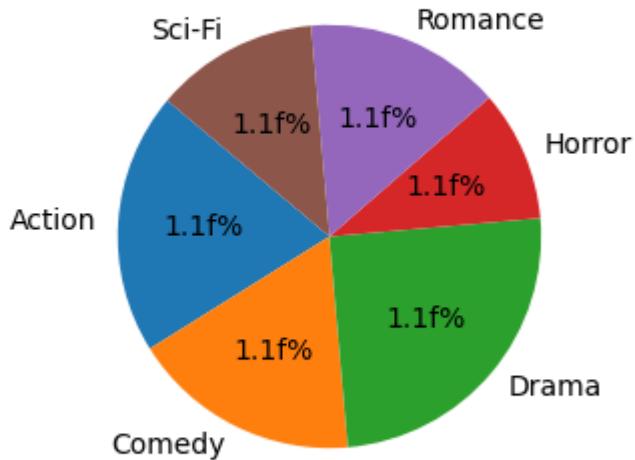
```
In [3]: import pandas as pd
import matplotlib.pyplot as plt
sales_data = {
    'Product': ['Laptop', 'Smartphone', 'Tablet', 'Monitor', 'Keyboard', 'Mouse'],
    'Sales': [1200, 2500, 900, 750, 1300, 1100, 600, 950]
}
df=pd.DataFrame(sales_data)
df.to_csv('sales.csv',index=False)
df=pd.read_csv('sales.csv')
plt.figure(figsize=(6,3))
plt.bar(df['Product'],df['Sales'],color='blue')
plt.xlabel('Product')
plt.ylabel('Sales')
plt.title('SALES DATA')
plt.xticks(df['Product'],rotation=45)
plt.show()
```



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 [6]: import pandas as pd
import matplotlib.pyplot as plt
movie_data = {
    'Genre': ['Action', 'Comedy', 'Drama', 'Horror', 'Romance', 'Sci-Fi'],
    'Count': [40, 35, 50, 20, 30, 25]
}
df=pd.DataFrame(movie_data)
df.to_csv('movie.csv',index=False)
df=pd.read_csv('movie.csv')
genre=df['Genre']
count=df['Count']
plt.figure(figsize=(3,5))
plt.pie(count,labels=genre,autopct='1.1f%%',startangle=140)
plt.title('Distribution of Movie Genres')
plt.axis('equal')
plt.show()
```

Distribution of Movie Genres

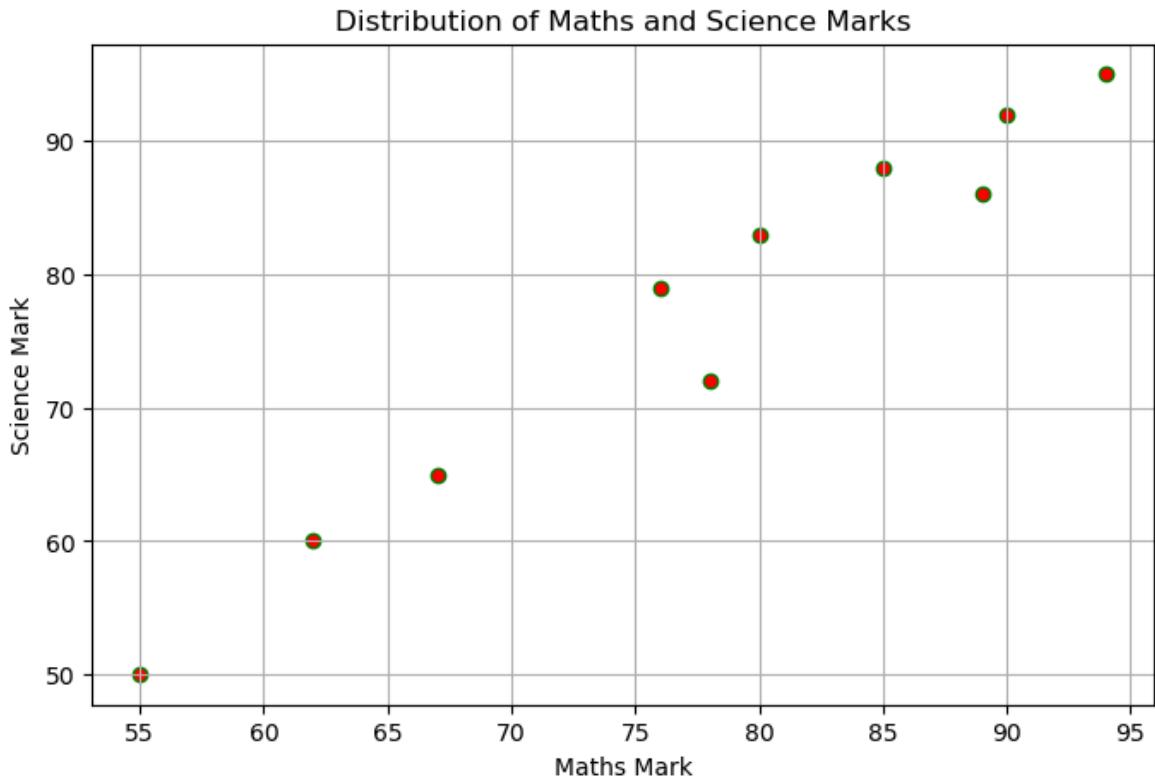


Q4. Scatter Plot (Students' Marks) Given a CSV file students.csv with columns: MathMarks, ScienceMarks. Write a Python program to plot a scatter plot showing the relationship between marks in Mathematics and Science.

- Add axis labels and a title.
- Use different colors/markers if possible.

```
In [12]: import pandas as pd
import matplotlib.pyplot as plt
student_data = {
    'MathMarks': [78, 85, 62, 90, 55, 89, 76, 94, 67, 80],
    'ScienceMarks': [72, 88, 60, 92, 50, 86, 79, 95, 65, 83]
}
df=pd.DataFrame(student_data)
df.to_csv('student.csv',index=False)
print("Student.csv file created")
df=pd.read_csv('student.csv')
plt.figure(figsize=(8,5))
plt.scatter(df['MathMarks'],df['ScienceMarks'],color='red',marker='o',edgecolors
plt.xlabel('Maths Mark')
plt.ylabel('Science Mark')
plt.title('Distribution of Maths and Science Marks')
plt.grid(True)
plt.show()
```

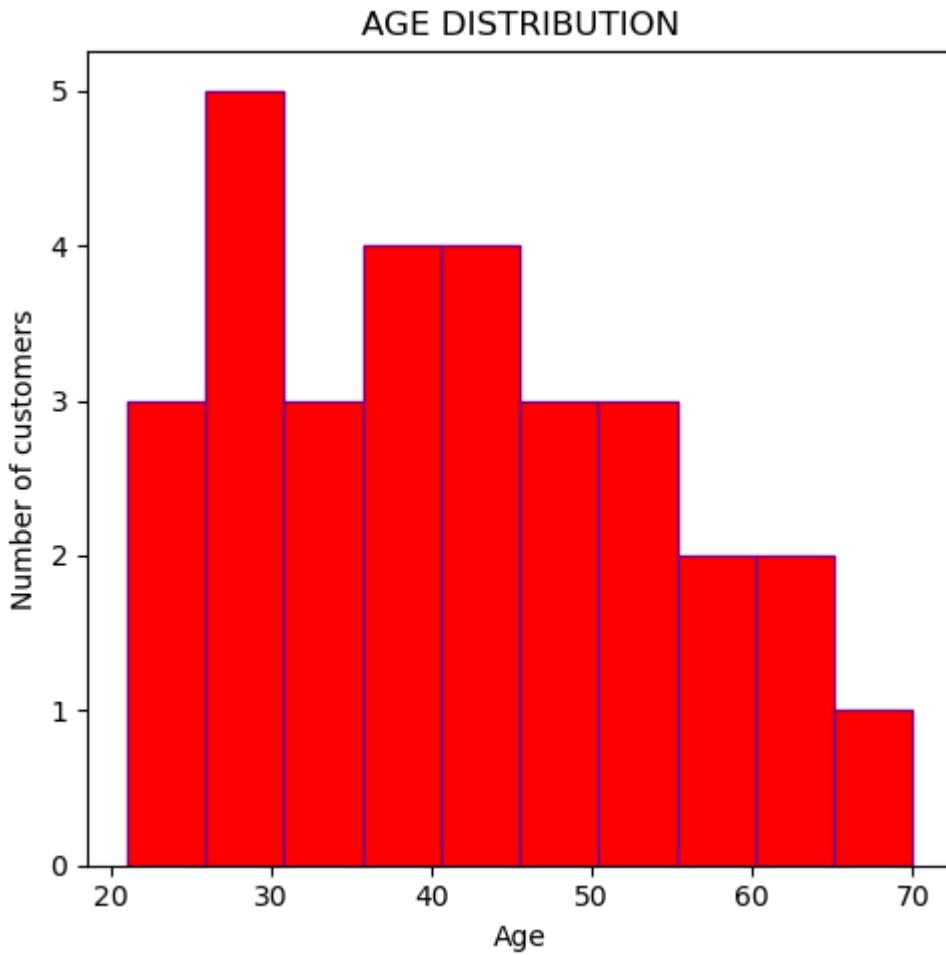
Student.csv file created



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.

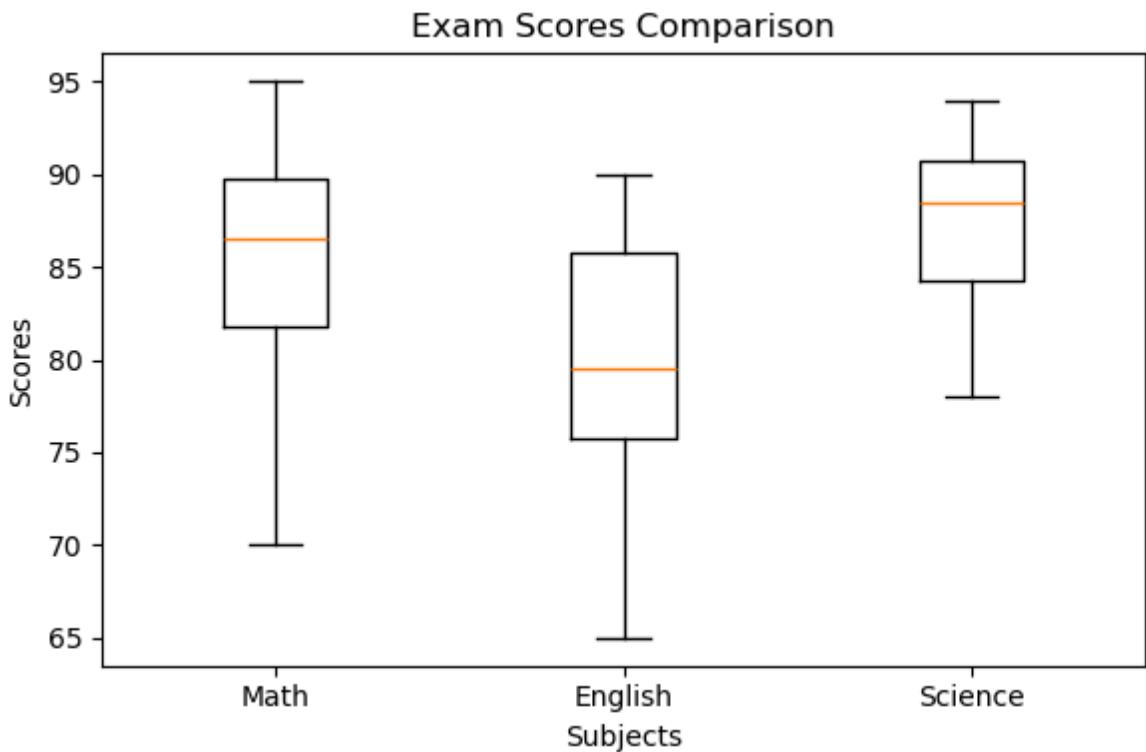
- Use 10 bins.
- Add axis labels and a title.

```
In [18]: import pandas as pd
import matplotlib.pyplot as plt
customer_data = {
    'Age': [22, 25, 29, 21, 34, 45, 52, 48, 37, 26, 30, 31, 33, 42, 41,
            55, 60, 28, 27, 36, 38, 40, 43, 46, 50, 53, 58, 61, 65, 70]
}
df=pd.DataFrame(customer_data)
df.to_csv('customer.csv',index=False)
df=pd.read_csv('customer.csv')
plt.figure(figsize=(5,5))
plt.hist(df['Age'],bins=10,color='red',edgecolor='purple')
plt.xlabel('Age')
plt.ylabel('Number of customers')
plt.title('AGE DISTRIBUTION')
plt.tight_layout()#to prevent overlaps
plt.show()
```



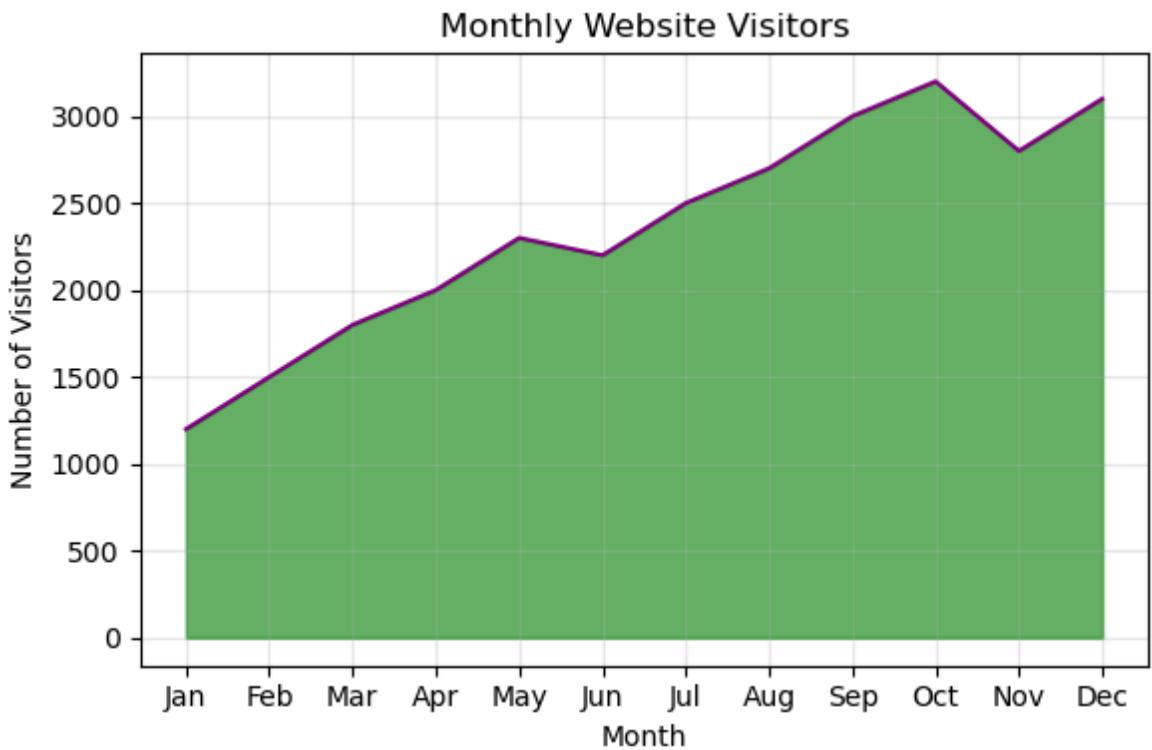
Q6. Boxplot (Exam Scores Comparison) Create a CSV file exam_scores.csv with columns: Math, English, Science. Write a Python program to plot a boxplot comparing score distributions across subjects. • Add axis labels and a title.

```
In [25]: import pandas as pd
import matplotlib.pyplot as plt
data = {
    'Math': [85, 88, 90, 76, 95, 89, 70, 84, 92, 81],
    'English': [78, 74, 80, 85, 90, 88, 65, 79, 86, 75],
    'Science': [92, 89, 94, 83, 91, 85, 78, 88, 90, 84]
}
df=pd.DataFrame(data)
df.to_csv('score.csv',index=False)
df=pd.read_csv('score.csv')
plt.figure(figsize=(6,4))
plt.boxplot([df['Math'],df['English'],df['Science']],tick_labels=['Math','English','Science'])
plt.title('Exam Scores Comparison')
plt.xlabel('Subjects')
plt.ylabel('Scores')
plt.tight_layout()
plt.show()
```



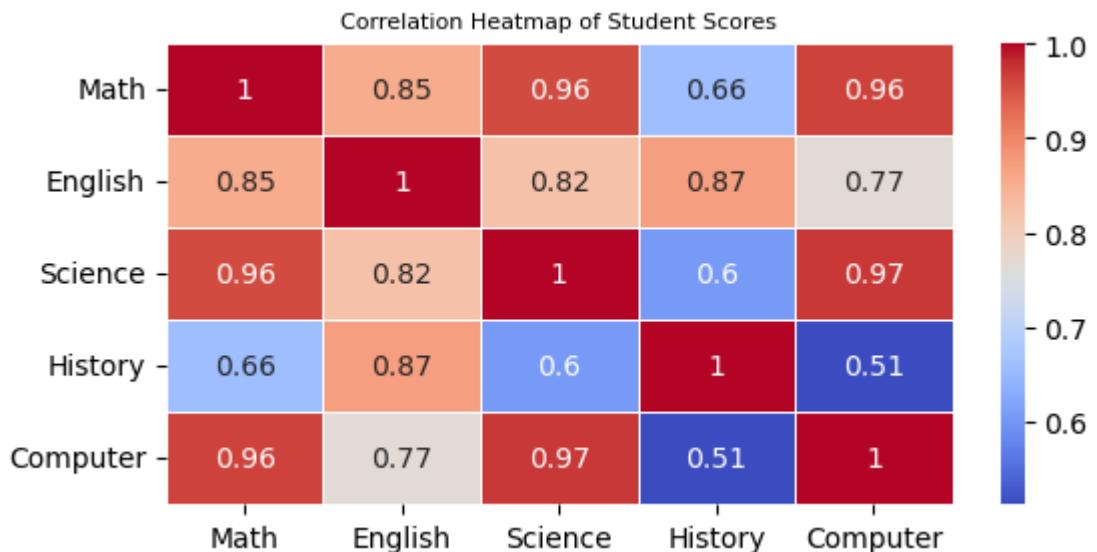
Q7. Area Chart (Website Traffic) Create a CSV file traffic.csv with columns: Month, Visitors.
Write a Python program to plot an area chart showing monthly website visitors for a year. • Add axis labels and a title. • Use a different color for the filled area.

```
In [30]: 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, 1800, 2000, 2300, 2200, 2500, 2700, 3000, 3200, 2800, 2600]
}
df = pd.DataFrame(data)
df.to_csv('traffic.csv', index=False)
df = pd.read_csv('traffic.csv')
plt.figure(figsize=(6, 4))
plt.fill_between(df['Month'], df['Visitors'], color='green', alpha=0.6)
plt.plot(df['Month'], df['Visitors'], color='purple')
plt.title('Monthly Website Visitors')
plt.xlabel('Month')
plt.ylabel('Number of Visitors')
plt.grid(alpha=0.3)# alpha is for visibility
plt.tight_layout()
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 [5]: import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
data = {
    'Math': [88, 92, 80, 89, 100, 67, 76, 94, 85, 91],
    'English': [78, 85, 82, 88, 90, 72, 79, 84, 81, 83],
    'Science': [90, 95, 85, 88, 98, 70, 77, 89, 84, 92],
    'History': [70, 75, 78, 80, 85, 68, 72, 76, 79, 74],
    'Computer': [95, 98, 90, 93, 100, 85, 88, 96, 91, 97]
}
df = pd.DataFrame(data)
df.to_csv('Student_scores.csv', index=False)
df = pd.read_csv('student_scores.csv')
corr_mat=df.corr()
plt.figure(figsize=(6,3))
sns.heatmap(corr_mat, annot=True, cmap='coolwarm', linewidths=0.5)
plt.title('Correlation Heatmap of Student Scores', fontsize=8)
plt.tight_layout()
plt.show()
```



In []: