Exercise 1

```
In [3]:   import numpy as np
```

```
In [5]:   array1 = np.arange(1, 11)
```

```
In [90]:  matrix1 = array1.reshape(2, 5)
          print("Matrix:\n", matrix1)
```

```
Matrix:
 [[ 1  2  3  4  5]
 [ 6  7  8  9 10]]
```

Exercise 2

```
In [11]:  array2 = np.arange(1, 21)
```

```
In [93]:  extracted_elements = array2[5:15]
          print("Elements[5:15]:\n", extracted_elements)
```

```
Elements[5:15]:
 [ 6  7  8  9 10 11 12 13 14 15]
```

Exercise 3

```
In [17]:  import pandas as pd
```

```
In [29]:  fruits = pd.Series({'apples': 3, 'bananas': 2, 'oranges': 1})
```

```
In [31]:  print("Fruits\n", fruits)
```

```
Fruits
 apples     3
bananas     2
oranges     1
dtype: int64
```

```
In [33]:  fruits['pears'] = 4
          print("Fruits\n", fruits)
```

```
Fruits
 apples     3
bananas     2
oranges     1
pears       4
dtype: int64
```

Exercise 4

```
In [39]: data = {
             'name': ['Annamma', 'Baby', 'Charles', 'David', 'Elizabeth', 'Ferin', 'Gemma
             'age': [28, 34, 23, 45, 30, 29, 40, 35, 22, 31],
             'gender': ['F', 'M', 'M', 'M', 'F', 'F', 'F', 'F', 'M', 'M']
         }
```

```
In [47]: df = pd.DataFrame(data)
         print("Data framework:\n", df)
```

```
Data framework:
        name  age gender
0    Annamma   28      F
1       Baby   34      M
2    Charles   23      M
3      David   45      M
4  Elizabeth   30      F
5      Ferin   29      F
6      Gemma   40      F
7       Hima   35      F
8       Ivan   22      M
9       Jose   31      M
```

Exercise 5

```
In [51]: occupations = ['Programmer', 'Librarian', 'Analyst', 'Manager', 'Nurse', 'Tester
         df['occupation'] = occupations
         print("Updated Data framework:\n", df)
```

```
Updated Data framework:
        name  age gender  occupation
0    Annamma   28      F  Programmer
1       Baby   34      M   Librarian
2    Charles   23      M     Analyst
3      David   45      M     Manager
4  Elizabeth   30      F       Nurse
5      Ferin   29      F      Tester
6      Gemma   40      F       Nurse
7       Hima   35      F    Lecturer
8       Ivan   22      M     Analyst
9       Jose   31      M     Teacher
```

Exercise 6

```
In [55]: age_filtered_df = df[df['age'] >= 30]
         print("Age>= 30: \n", age_filtered_df)
```

```
Age>= 30:
        name  age gender occupation
1       Baby   34      M  Librarian
3      David   45      M    Manager
4  Elizabeth   30      F      Nurse
6      Gemma   40      F      Nurse
7       Hima   35      F   Lecturer
9       Jose   31      M    Teacher
```

Exercise 7

In [59]:
```python
df.to_csv('data.csv', index=False)
```

In [100…
```python
df_from_csv = pd.read_csv('data.csv')
print("From csv file:\n", df_from_csv)
```

```
From csv file:
        name  age gender   occupation
0    Annamma   28      F   Programmer
1       Baby   34      M    Librarian
2    Charles   23      M      Analyst
3      David   45      M      Manager
4  Elizabeth   30      F        Nurse
5      Ferin   29      F       Tester
6      Gemma   40      F        Nurse
7       Hima   35      F     Lecturer
8       Ivan   22      M      Analyst
9       Jose   31      M      Teacher
```
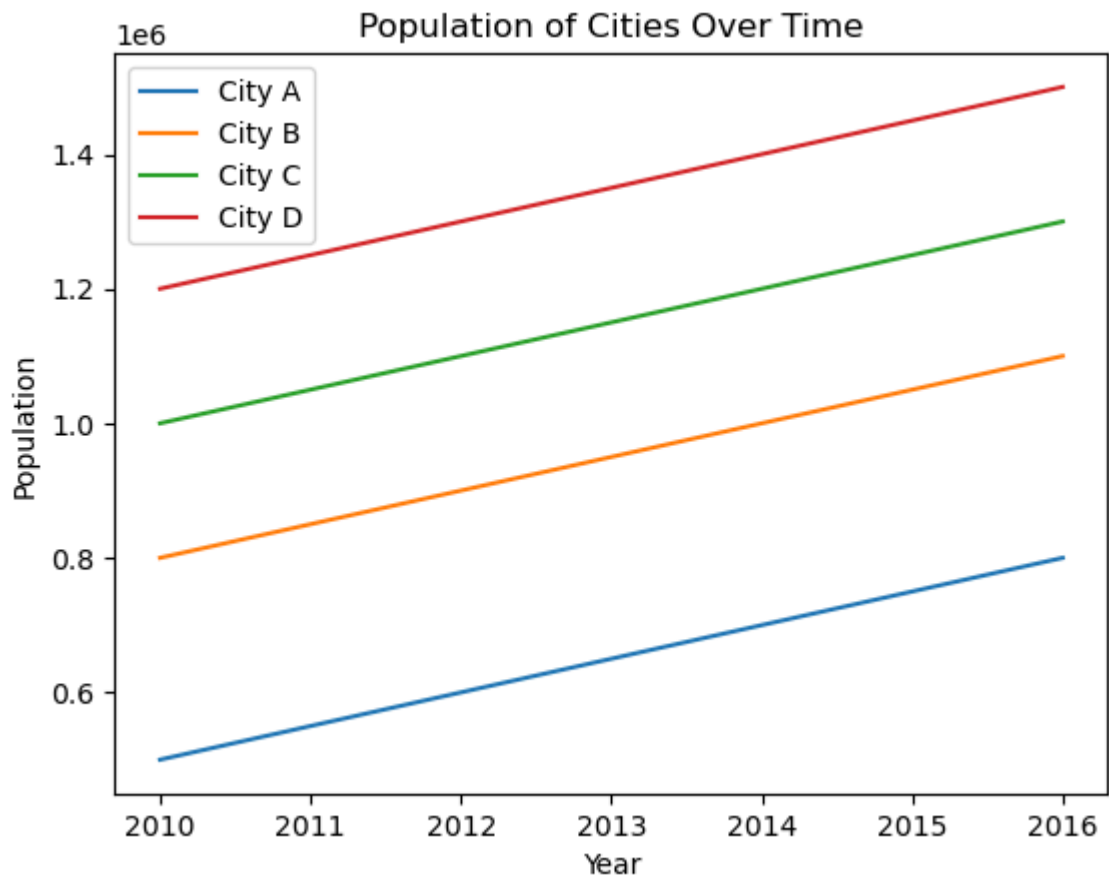
Exercise 8

In [65]:
```python
import matplotlib.pyplot as plt
```

In [116…
```python
years = [2010, 2011, 2012, 2013, 2014, 2015, 2016]
city_a = [500000, 550000, 600000, 650000, 700000, 750000, 800000]
city_b = [800000, 850000, 900000, 950000, 1000000, 1050000, 1100000]
city_c = [1000000, 1050000, 1100000, 1150000, 1200000, 1250000, 1300000]
city_d = [1200000, 1250000, 1300000, 1350000, 1400000, 1450000, 1500000]
```

In [118…
```python
plt.plot(years, city_a, label='City A')
plt.plot(years, city_b, label='City B')
plt.plot(years, city_c, label='City C')
plt.plot(years, city_d, label='City D')

plt.xlabel('Year')
plt.ylabel('Population')
plt.title('Population of Cities Over Time')
plt.legend()
plt.show()
```
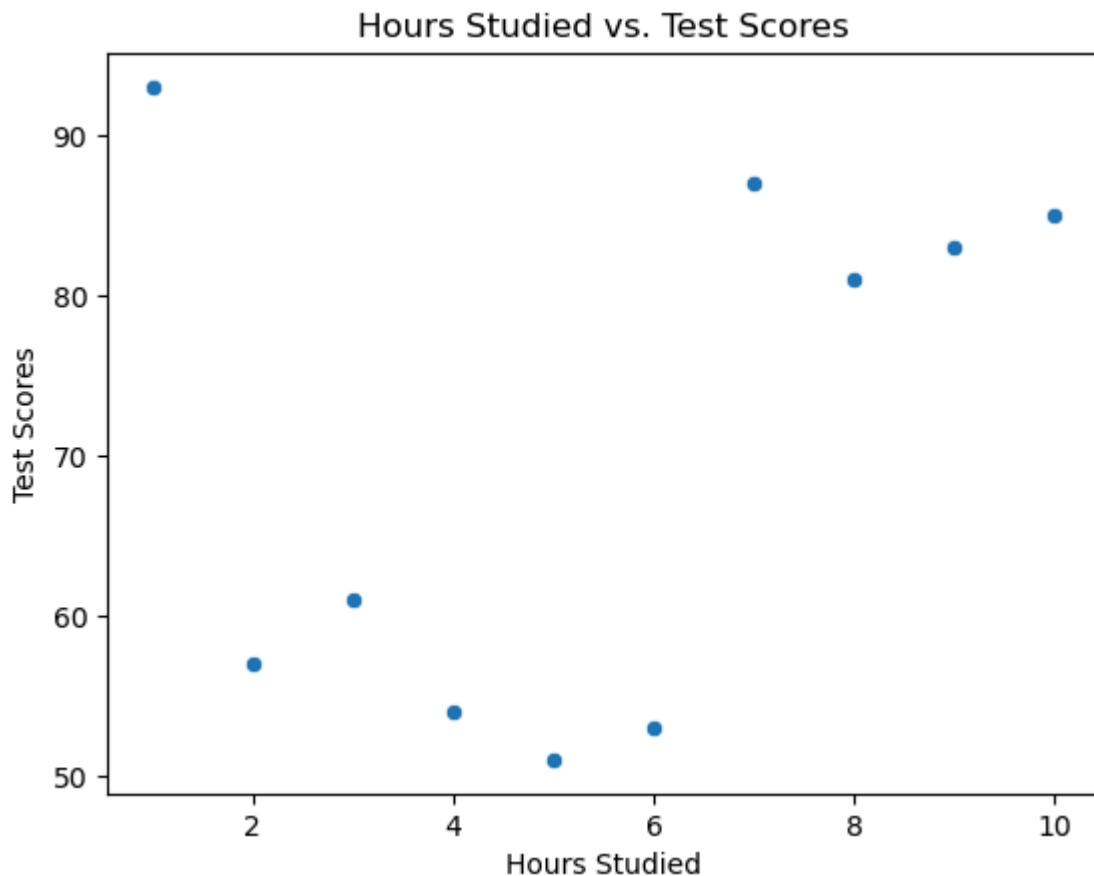
Exercise 9

```
In [77]: import seaborn as sns
```

```
In [79]: hours_studied = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
         test_scores = [93, 57, 61, 54, 51, 53, 87, 81, 83, 85]
```
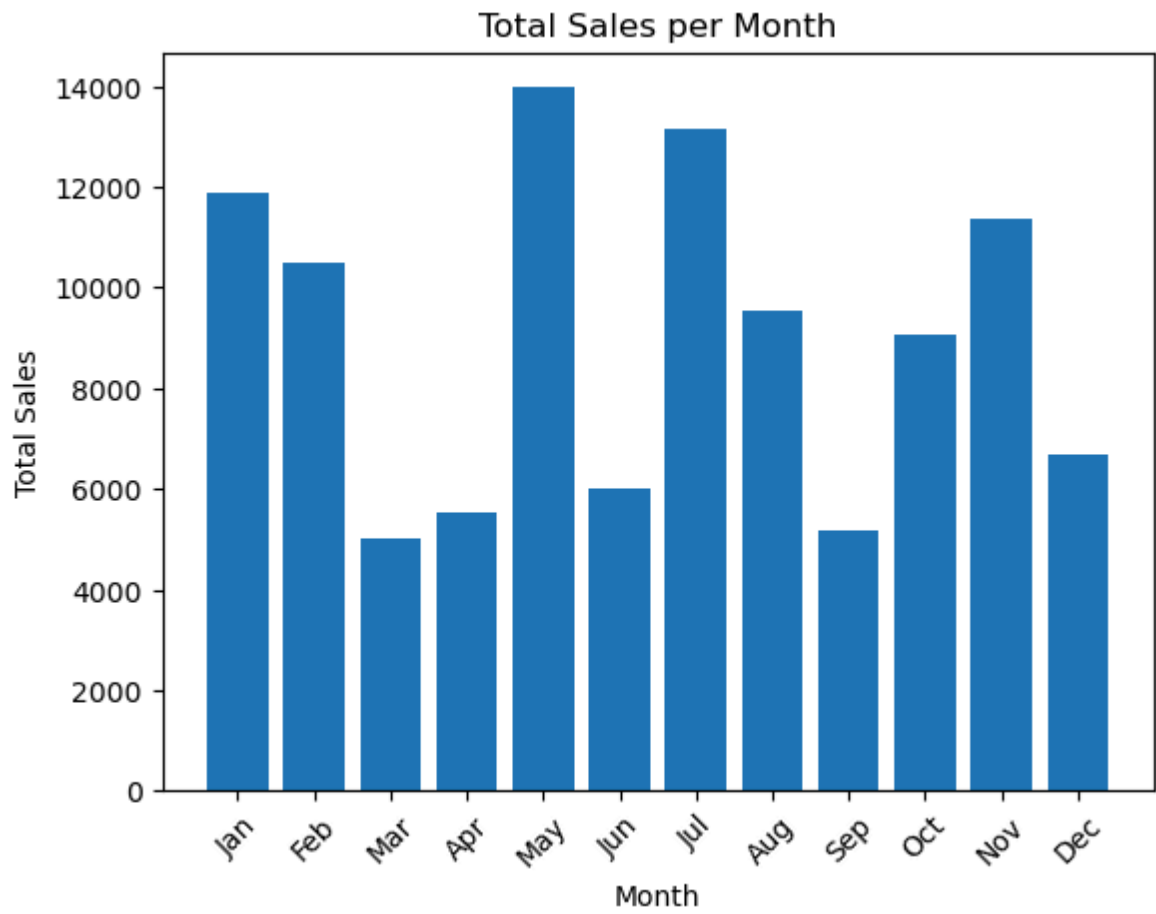
```
In [81]: sns.scatterplot(x=hours_studied, y=test_scores)
         plt.xlabel('Hours Studied')
         plt.ylabel('Test Scores')
         plt.title('Hours Studied vs. Test Scores')
         plt.show()
```

Hours Studied vs. Test Scores

Exercise 10

```
In [85]: months = ["Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct",
         sales = [11860, 10480, 4997, 5523, 13965, 6011, 13158, 9533, 5158, 9058, 11346,
```

```
In [87]: plt.bar(months, sales)
         plt.xlabel('Month')
         plt.ylabel('Total Sales')
         plt.title('Total Sales per Month')
         plt.xticks(rotation=45)
         plt.show()
```

Total Sales per Month

In [ ]: