

*# Exercise 1*

```
import numpy as np
```

```
arr = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
```

```
newarr = arr.reshape(2, 5)
```

```
print(newarr)
```

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

*# Exercise 2*

```
import numpy as np
```

*# Create a numpy array with numbers from 1 to 20*

```
array = np.arange(1, 21)
```

*# Extract elements between the 5th and 15th index (inclusive of 5 and exclusive of 15)*

```
extracted_elements = array[5:15]
```

```
print("Original array:", array)
```

```
print("Extracted elements:", extracted_elements)
```

```
Original array: [ 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18
 19 20]
```

```
Extracted elements: [ 6  7  8  9 10 11 12 13 14 15]
```

*# Exercise 3*

```
import pandas as pd
```

*# Create a Pandas Series with the given data*

```
data = {'apples': 3, 'bananas': 2, 'oranges': 1}
```

```
series = pd.Series(data)
```

*# Add a new item with the key 'pears' and the value 4*

```
series['pears'] = 4
```

```
print(series)
```

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

*# Exercise 4*

```
import pandas as pd

# Data for the DataFrame
data = {
    'name': ['Maya', 'Abhi', 'Ram', 'Das', 'Aleena', 'Ravi', 'Mary',
            'Hari', 'Rose', 'Jack'],
    'age': [25, 30, 35, 28, 22, 40, 31, 27, 29, 24],
    'gender': ['Female', 'Male', 'Male', 'Male', 'Female', 'Male',
              'Female', 'Male', 'Female', 'Male']
}

# Create the DataFrame
df = pd.DataFrame(data)

# Display the DataFrame
print(df)
```

	name	age	gender
0	Maya	25	Female
1	Abhi	30	Male
2	Ram	35	Male
3	Das	28	Male
4	Aleena	22	Female
5	Ravi	40	Male
6	Mary	31	Female
7	Hari	27	Male
8	Rose	29	Female
9	Jack	24	Male

*# Exercise 5*

```
# New column data
occupations = ['Programmer', 'Manager', 'Analyst', 'Programmer',
               'Manager',
               'Analyst', 'Programmer', 'Manager', 'Analyst',
               'Programmer']
```

```
# Adding the new column
df['occupation'] = occupations
print("Exercise 5:\n", df)
```

Exercise 5:

	name	age	gender	occupation
0	Maya	25	Female	Programmer
1	Abhi	30	Male	Manager
2	Ram	35	Male	Analyst
3	Das	28	Male	Programmer
4	Aleena	22	Female	Manager
5	Ravi	40	Male	Analyst
6	Mary	31	Female	Programmer

7	Hari	27	Male	Manager
8	Rose	29	Female	Analyst
9	Jack	24	Male	Programmer

# Exercise 6

```
# Select rows where age is >= 30
filtered_df = df[df['age'] >= 30]
print("Exercise 6:\n", filtered_df)
```

Exercise 6:

	name	age	gender	occupation
1	Abhi	30	Male	Manager
2	Ram	35	Male	Analyst
5	Ravi	40	Male	Analyst
6	Mary	31	Female	Programmer

# Exercise 7

```
# Save to CSV
df.to_csv('data.csv', index=False)
```

```
# Read the CSV file
read_df = pd.read_csv('data.csv')
print("Exercise 7:\n", read_df)
```

Exercise 7:

	name	age	gender	occupation
0	Maya	25	Female	Programmer
1	Abhi	30	Male	Manager
2	Ram	35	Male	Analyst
3	Das	28	Male	Programmer
4	Aleena	22	Female	Manager
5	Ravi	40	Male	Analyst
6	Mary	31	Female	Programmer
7	Hari	27	Male	Manager
8	Rose	29	Female	Analyst
9	Jack	24	Male	Programmer

# Exercise 8

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

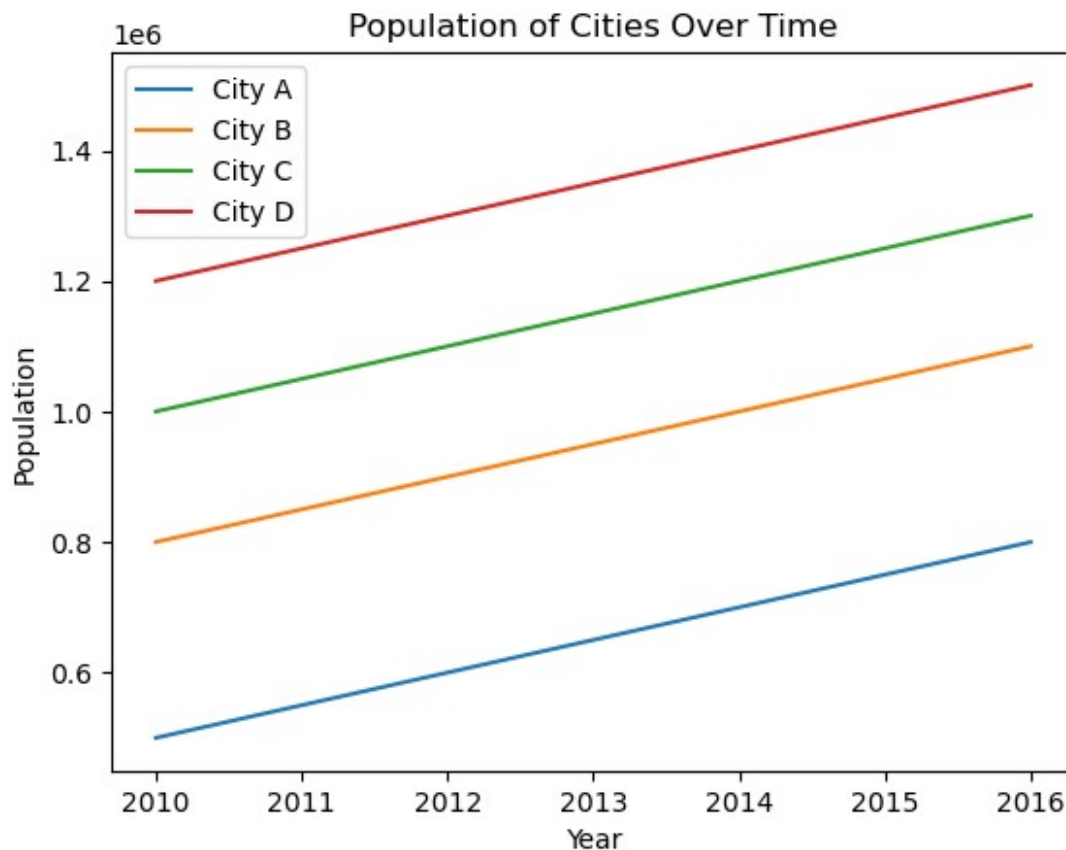
# Data for the cities

```
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]
```

```

# Plotting
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

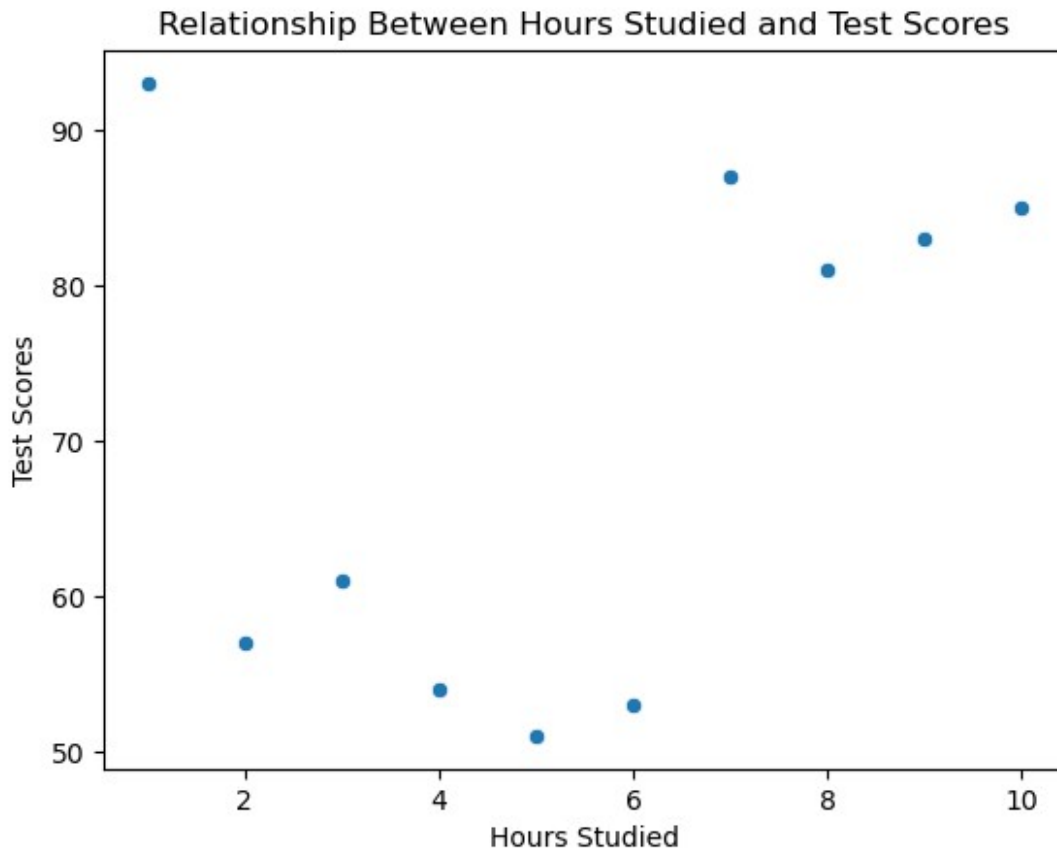
import seaborn as sns

# Data for hours studied and test scores
hours_studied = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
test_scores = [93, 57, 61, 54, 51, 53, 87, 81, 83, 85]

# Create a scatter plot
sns.scatterplot(x=hours_studied, y=test_scores)
plt.xlabel('Hours Studied')

```

```
plt.ylabel('Test Scores')
plt.title('Relationship Between Hours Studied and Test Scores')
plt.show()
```



*# Exercise 10*

*# Data for the bar chart*

```
months = ["Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug",  
          "Sep", "Oct", "Nov", "Dec"]  
sales = [11860, 10480, 4997, 5523, 13965, 6011, 13158, 9533, 5158,  
         9058, 11346, 6675]
```

*# Create a bar chart*

```
plt.bar(months, sales)  
plt.xlabel('Month')  
plt.ylabel('Sales')  
plt.title('Total Sales for Each Month of the Year')  
plt.xticks(rotation=45)  
plt.show()
```

