Assignment 2

Name: Rohitha Sai Alla

No: 700734780

Video link: https://drive.google.com/file/d/1kM939JO4BBH6JqnuMoQriAFlPf1eZI1U/view?usp=sharing

Git link: https://github.com/RohithaSaiML/Assignment2

1.Numpy

To create a random vector of 15 size a=with integers and in the range of 1-20.

- 1.Reshaping the array to 3 by 5
- 2.To print the array shape.
- 3.Replace the max value in each row with 0.

```
In [7]:  #1.Numpy
         import numpy as np
         #Creating random vector of size 15 between 1 to 20
         a = np.random.randint(1,high=20,size=15)

         #1.Reshaping array to 3 by 5
         a_reshape = a.reshape((3,5))
         print(a_reshape)

         #2.To print shape of the array
         print("Array shape: ",a_reshape.shape)

         #3.Replacing the max value of each row to 0
         a_reshape[np.isin(a_reshape,np.max(a_reshape, axis = 1))] = 0
         print("Max value replaced with zero in each row: ")
         print(a_reshape)
```

Output:

```
[[ 5 13 19 11 19]
 [ 4 19  1 16  7]
 [15 13  6  2  3]]
Array shape:  (3, 5)
Max value replaced with zero in each row:
[[ 5 13  0 11  0]
 [ 4  0  1 16  7]
 [ 0 13  6  2  3]]
```

2.Pandas

1.To read the csv data file.

```
In [20]: #2.pandas
         import pandas as pd

         #1.To read the csv file
         df = pd.read_csv("data.csv")
         df.head()
```

Output:

Out[20]:

	Duration	Pulse	Maxpulse	Calories
0	60	110	130	409.1
1	60	117	145	479.0
2	60	103	135	340.0
3	45	109	175	282.4
4	45	117	148	406.0

2.To show the basic statistical description about the data.

```
In [21]: #2.Description of data
         df.describe()
```

Output:

Out[21]:

	Duration	Pulse	Maxpulse	Calories
count	169.000000	169.000000	169.000000	164.000000
mean	63.846154	107.461538	134.047337	375.790244
std	42.299949	14.510259	16.450434	266.379919
min	15.000000	80.000000	100.000000	50.300000
25%	45.000000	100.000000	124.000000	250.925000
50%	60.000000	105.000000	131.000000	318.600000
75%	60.000000	111.000000	141.000000	387.600000
max	300.000000	159.000000	184.000000	1860.400000

3.To check if the data has null values.

```
In [22]:  #3.To check if data has null values
          df.isnull().any()
```

Output:

```
Out[22]:  Duration    False
          Pulse       False
          Maxpulse    False
          Calories     True
          dtype: bool
```

a. To replace null values with mean values.

```
In [23]:  #Replacing null vales with mean
          df.fillna(df.mean(), inplace=True)
          df.isnull().any()
```

Output:

```
Out[23]:  Duration    False
          Pulse       False
          Maxpulse    False
          Calories    False
          dtype: bool
```

4.To select at least two columns and perform the aggregate operations.

```
In [17]:  #4.To find aggregate of Duration and Calories
          df.agg({'Duration':['min','max','count','mean'],'Calories':['min','max','count','mean']})
```

Output:

Out[17]:

	Duration	Calories
min	15.000000	50.300000
max	300.000000	1860.400000
count	169.000000	169.000000
mean	63.846154	375.790244

5.To filter the dataframe to select rows with calories between 500 and 1000.

```
In [24]:  #5.Rows with calories value between 500 and 1000
          df.loc[(df['Calories'] > 500) & (df['Calories'] < 1000)]
```

Output:

Out[24]:

	Duration	Pulse	Maxpulse	Calories
51	80	123	146	643.1
62	160	109	135	853.0
65	180	90	130	800.4
66	150	105	135	873.4
67	150	107	130	816.0
72	90	100	127	700.0
73	150	97	127	953.2
75	90	98	125	563.2
78	120	100	130	500.4
90	180	101	127	600.1
99	90	93	124	604.1
103	90	90	100	500.4
106	180	90	120	800.3
108	90	90	120	500.3

6.To filter the dataframe to select the rows with calories values > 500 and pulse < 100.

```
In [25]: #6.Rows with calories > 500 and pulse < 100
         df.loc[(df['Calories'] > 500) & (df['Pulse'] < 100)]
```

Output:

Out[25]:

	Duration	Pulse	Maxpulse	Calories
65	180	90	130	800.4
70	150	97	129	1115.0
73	150	97	127	953.2
75	90	98	125	563.2
99	90	93	124	604.1
103	90	90	100	500.4
106	180	90	120	800.3
108	90	90	120	500.3

7.To create a modified data frame that contains all columns except "Maxpulse"

```
In [26]: #7.Modified dataframe to have all columns except "Maxpulse"
         df_modified = df[['Duration','Pulse','Calories']]
         df_modified.head()
```

Output:

Out[26]:

	Duration	Pulse	Calories
0	60	110	409.1
1	60	117	479.0
2	60	103	340.0
3	45	109	282.4
4	45	117	406.0

8. To delete maxpulse column from main dataframe.

```
In [27]: #8.To delete "Maxpulse" column from main df dataframe
         del df['Maxpulse']
         df.head()
```

Output:

Out[27]:

	Duration	Pulse	Calories
0	60	110	409.1
1	60	117	479.0
2	60	103	340.0
3	45	109	282.4
4	45	117	406.0

9. To convert calories column to int data type.

Data type of present columns

```
In [33]: df.dtypes
```

Output:

```
Out[33]: Duration      int64
         Pulse         int64
         Calories    float64
         dtype: object
```

Conversion of calories to int

```
In [34]:   #9.To convert Calories column to int data type.
           df['Calories'] = df['Calories'].astype(np.int64)
           df.dtypes
```

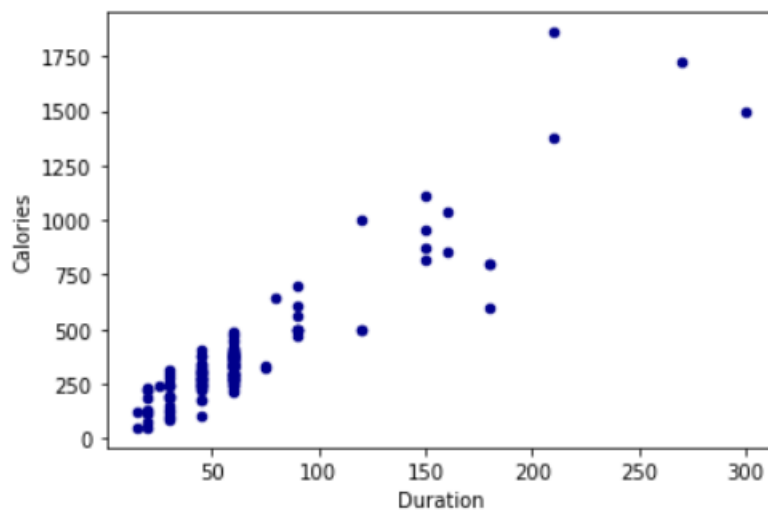Output:

```
Out[34]:  Duration      int64
          Pulse         int64
          Calories      int64
          dtype: object
```

10.Using pandas to create a scatter plot for duration and calories columns.

```
In [35]:  #10.To create scatter plot for Duration and calories with pandas
          df.plot.scatter(x='Duration',y='Calories',c='DarkBlue')
```

Output:

```
Out[35]:  <AxesSubplot:xlabel='Duration', ylabel='Calories'>
```


3.Matplotlib

1 and 2: To create a popularity chart for languages with given sample data.

Programming languages: Java, Python, PHP, JavaScript, C#, C++

Popularity: 22.2, 17.6, 8.8, 8, 7.7, 6.7

In [38]:
```python
#Matplotlib
#1.To create a chart of the popularity of programming Languages.
import matplotlib.pyplot as plt
#2.Given sample data to plot
languages = 'Java', 'Python', 'PHP', 'JavaScript', 'C#', 'C++'
popularity = [22.2, 17.6, 8.8, 8, 7.7, 6.7]
colors = ["#1f77b4", "#ff7f0e", "#2ca02c", "#d62728", "#9467bd", "#8c564b"]
# to explode 1st slice
explode = (0.1,0,0,0,0,0)
# to plot the figure
plt.pie(popularity, explode = explode, labels = languages, colors = colors, autopct = '%1.1f%%', shadow = True, startangle = 140)
plt.axis('equal')
plt.show()
```

Output: