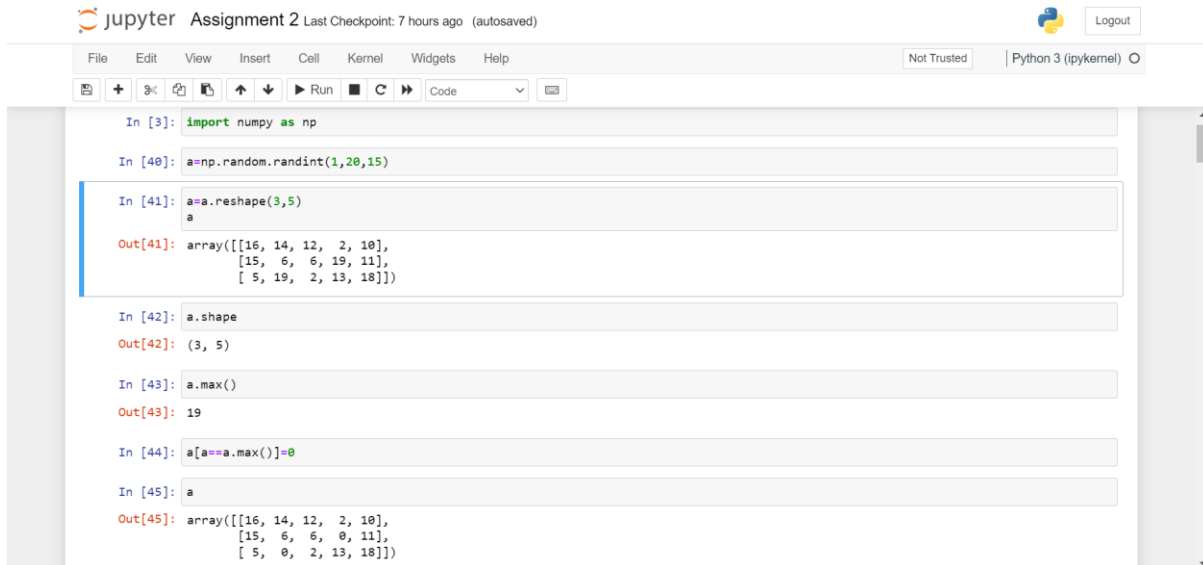


Question 1

`randint(1,20,15)` is used to generate array of size 15 and numbers within range (1,20) and reshape changes the array shape to 3,4. `a[a==a.max()]` replace every maximum value in each row with 0



Jupyter Assignment 2 Last Checkpoint: 7 hours ago (autosaved) Python 3 (ipykernel)

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

In [40]: a=np.random.randint(1,20,15)

In [41]: a=a.reshape(3,5)
a
Out[41]: array([[16, 14, 12,  2, 10],
               [15,  6,  6, 19, 11],
               [ 5, 19,  2, 13, 18]])

In [42]: a.shape
Out[42]: (3, 5)

In [43]: a.max()
Out[43]: 19

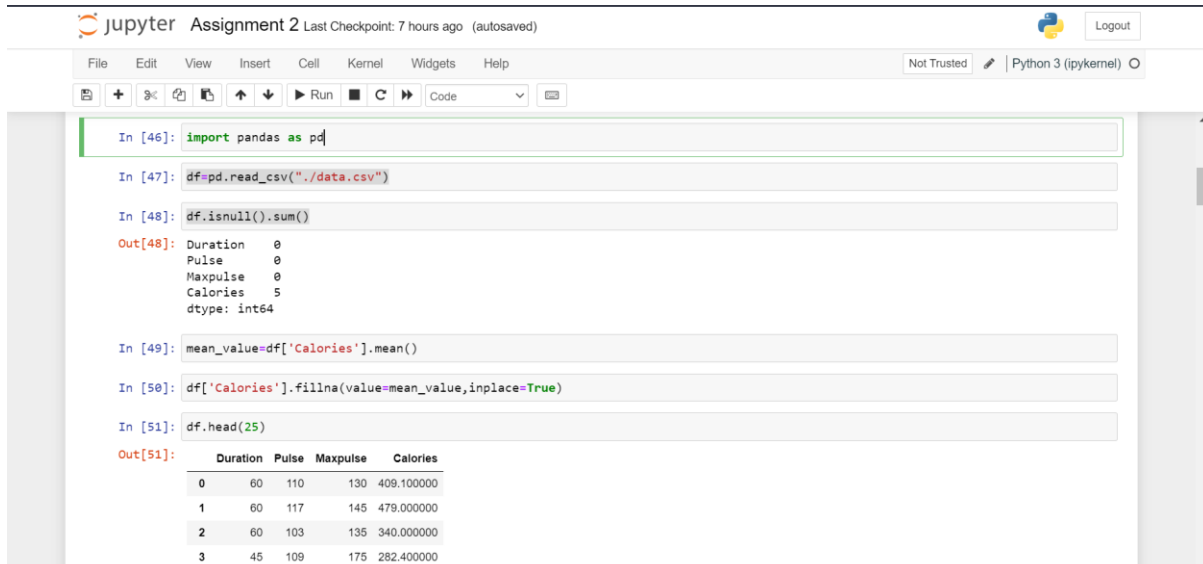
In [44]: a[a==a.max()]=0

In [45]: a
Out[45]: array([[16, 14, 12,  2, 10],
               [15,  6,  6,  0, 11],
               [ 5,  0,  2, 13, 18]])
```

Question 2

`isnull()` changes the dataframes to true and false based on the whether value is null or not and `sum` is used to find the number of false by this we can verify if the dataframe contains any missing values.

`Mean()` find the mean value of Calories the `fillna()` method replaces all the null values with mean value.



Jupyter Assignment 2 Last Checkpoint: 7 hours ago (autosaved) Python 3 (ipykernel)

```
In [46]: import pandas as pd

In [47]: df=pd.read_csv("./data.csv")

In [48]: df.isnull().sum()
Out[48]: Duration    0
Pulse             0
Maxpulse          0
Calories          5
dtype: int64

In [49]: mean_value=df['Calories'].mean()

In [50]: df['Calories'].fillna(value=mean_value,inplace=True)

In [51]: df.head(25)
Out[51]:
```

	Duration	Pulse	Maxpulse	Calories
0	60	110	130	409.100000
1	60	117	145	479.000000
2	60	103	135	340.000000
3	45	109	175	282.400000

Describe() generates all the useful statistical features such as count,mean,std and min

jupyter Assignment 2 Last Checkpoint: 7 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

```
19 45 97 125 243.000000
20 60 108 131 364.200000
21 45 100 119 282.000000
22 60 130 101 300.000000
23 45 105 132 246.000000
24 60 102 126 334.500000
```

```
In [52]: df.describe()
```

```
Out[52]:
```

	Duration	Pulse	Maxpulse	Calories
count	169.000000	169.000000	169.000000	169.000000
mean	63.846154	107.461538	134.047337	375.790244
std	42.299949	14.510259	16.450434	262.385991
min	15.000000	80.000000	100.000000	50.300000
25%	45.000000	100.000000	124.000000	253.300000
50%	60.000000	105.000000	131.000000	321.000000
75%	60.000000	111.000000	141.000000	384.000000
max	300.000000	159.000000	184.000000	1860.400000

```
In [53]: df.Duration.describe()
```

jupyter Assignment 2 Last Checkpoint: 7 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

```
In [53]: df.Duration.describe()
```

```
Out[53]:
```

count	169.000000
mean	63.846154
std	42.299949
min	15.000000
25%	45.000000
50%	60.000000
75%	60.000000
max	300.000000

Name: Duration, dtype: float64

```
In [54]: df.Pulse.describe()
```

```
Out[54]:
```

count	169.000000
mean	107.461538
std	14.510259
min	80.000000
25%	100.000000
50%	105.000000
75%	111.000000
max	159.000000

Name: Pulse, dtype: float64

```
In [55]: df[(df['Calories']>500) & (df['Calories']<1000)]
```

```
Out[55]:
```

	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

jupyter Assignment 2 Last Checkpoint: 7 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

```
In [55]: df[(df['Calories']>500) & (df['Calories']<1000)]
```

```
Out[55]:
```

	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

```
In [56]: df[(df['Calories']>500) & (df['Pulse']<100)]
```

```
In [56]: df[(df['Calories']>500 & (df['Pulse']<100))]
```

```
Out[56]:
```

	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
...
164	60	105	140	290.8
165	60	110	145	300.0
166	60	115	145	310.2
167	75	120	150	320.4
168	75	125	150	330.4

169 rows x 4 columns

Drop() drops the Maxpulse column from the dataframe

```
In [57]: df_modified=df.drop("Maxpulse",axis=1)
```

```
In [58]: df_modified
```

```
Out[58]:
```

	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
...
164	60	105	290.8
165	60	110	300.0
166	60	115	310.2
167	75	120	320.4
168	75	125	330.4

169 rows x 3 columns

```
In [59]: df=df.drop("Maxpulse",axis=1)
```

Astype(int) changes the column element type to integer and scatter() generates the scatter plot with x axis having duration and y axis having calories

```
In [61]: df['Calories']=df['Calories'].astype(int)
```

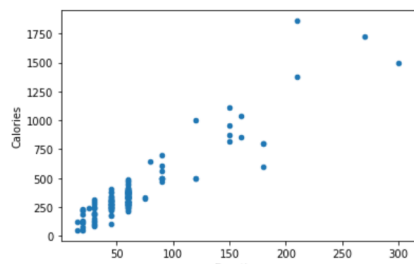
```
In [62]: df.dtypes
```

```
Out[62]:
```

Duration	int64
Pulse	int64
Calories	int32
dtype:	object

```
In [63]: df.plot.scatter( x = 'Duration', y = 'Calories')
```

```
Out[63]: <AxesSubplot:xlabel='Duration', ylabel='Calories'>
```



Question 3

DataFrame() generates the dataframes with index having programming languages.

Pie() generates the pie chart this generates pie and takes startangle as argument and explode is used to expand the pie chart.

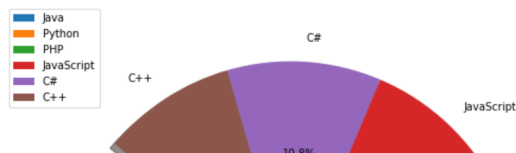
```
In [64]: prgmng_df=pd.DataFrame({"popularity": [22.2, 17.6, 8.8, 8, 7.7, 6.7]}, index=['Java', 'Python', 'PHP', 'JavaScript', 'C#', 'C++'])
prgmng_df
```

```
Out[64]:
```

	popularity
Java	22.2
Python	17.6
PHP	8.8
JavaScript	8.0
C#	7.7
C++	6.7

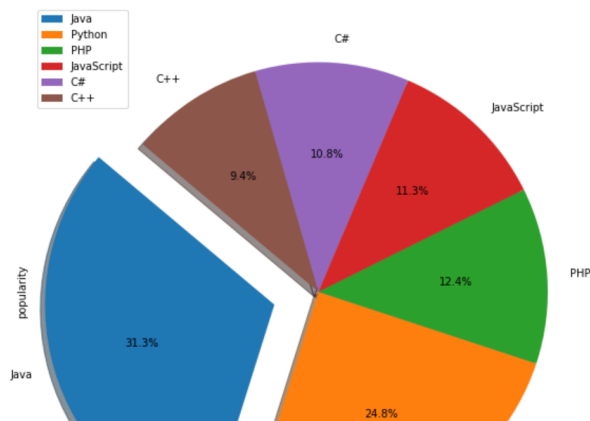
```
In [65]: myexplode = [0.2, 0, 0, 0, 0, 0]
prgmng_df.plot.pie(subplots=True, autopct='%1.1f%%', explode=myexplode, figsize=[10,10], startangle=140, shadow=True)
```

```
Out[65]: array([[<AxesSubplot:ylabel='popularity'>]], dtype=object)
```



```
In [65]: myexplode = [0.2, 0, 0, 0, 0, 0]
prgmng_df.plot.pie(subplots=True, autopct='%1.1f%%', explode=myexplode, figsize=[10,10], startangle=140, shadow=True)
```

```
Out[65]: array([[<AxesSubplot:ylabel='popularity'>]], dtype=object)
```



Github Link: <https://github.com/sainikhil3280/ML-Assignment-2>