Course : B. Sc. (h) Computer Science

Year : III

Semester : V

Name : Raman

College Rollno : CSC/20/26

University Rollno : 20059570019

Data Analysis and Visualisation Practical File

Submitted to :- Mr. Rajeev Rai

INDEX

<table>
<tr><td></td><td>students who had attended the workshop on both days.
b. Find names of all students who have attended workshop on either of the days.
c. Merge two data frames row-wise and find the total number of records in the data frame
d. Merge two data frames and use two columns names and duration as multi-row indexes. Generate descriptive statistics for this multi-index.</td><td></td></tr>
<tr><td>5</td><td>Ques5. Taking Iris data, plot the following with proper legend and axis labels: (Download IRIS data from: https://archive.ics.uci.edu/ml/datasets/iris or import it from sklearn.datasets
a. Plot bar chart to show the frequency of each class label in the data.
b. Draw a scatter plot for Petal width vs sepal width.
c. Plot density distribution for feature petal length.
d. Use a pair plot to show pairwise bivariate distribution in the Iris Dataset.</td><td>25</td></tr>
<tr><td>6</td><td>Ques6. Consider any sales training/ weather forecasting dataset
a. Compute mean of a series grouped by another series
b. Fill an intermittent time series to replace all missing dates with values of previous non-missing date.
c. Perform appropriate year-month string to dates conversion.
d. Split a dataset to group by two columns and then sort the aggregated results within the groups.
e. Split a given dataframe into groups with bin counts</td><td>31</td></tr>
<tr><td>7</td><td>Ques7. Consider a data frame containing data about students i.e. name, gender and passing division:
<table>
<tr><th></th><th>Name</th><th>Birth_Month</th><th>Gender</th><th>Pass_Division</th></tr>
<tr><td>0</td><td>Mudit Chauhan</td><td>December</td><td>M</td><td>III</td></tr>
<tr><td>1</td><td>Seema Chopra</td><td>January</td><td>F</td><td>II</td></tr>
<tr><td>2</td><td>Rani Gupta</td><td>March</td><td>F</td><td>I</td></tr>
<tr><td>3</td><td>Aditya Narayan</td><td>October</td><td>M</td><td>I</td></tr>
<tr><td>4</td><td>Sanjeev Sahni</td><td>February</td><td>M</td><td>II</td></tr>
<tr><td>5</td><td>Prakash Kumar</td><td>December</td><td>M</td><td>III</td></tr>
<tr><td>6</td><td>Ritu Agarwal</td><td>September</td><td>F</td><td>I</td></tr>
<tr><td>7</td><td>Akshay Goel</td><td>August</td><td>M</td><td>I</td></tr>
<tr><td>8</td><td>Meeta Kulkarni</td><td>July</td><td>F</td><td>II</td></tr>
<tr><td>9</td><td>Preeti Ahuja</td><td>November</td><td>F</td><td>II</td></tr>
<tr><td>10</td><td>Sunil Das Gupta</td><td>April</td><td>M</td><td>III</td></tr>
<tr><td>11</td><td>Sonali Sapre</td><td>January</td><td>F</td><td>I</td></tr>
<tr><td>12</td><td>Rashmi Talwar</td><td>June</td><td>F</td><td>III</td></tr>
<tr><td>13</td><td>Ashish Dubey</td><td>May</td><td>M</td><td>II</td></tr>
<tr><td>14</td><td>Kiran Sharma</td><td>February</td><td>F</td><td>II</td></tr>
<tr><td>15</td><td>Sameer Bansal</td><td>October</td><td>M</td><td>I</td></tr>
</table>
a. Perform one hot encoding of the last two columns of categorical data using the get_dummies() function.
b. Sort this data frame on the “Birth Month” column (i.e. January to December). Hint: Convert Month to Categorical.</td><td>35</td></tr>
</table>

<table>
<tr>
<td>8</td>
<td>

Ques8. Consider the following data frame containing a family name, gender of the family member and her/his monthly income in each record.

Name	Gender	MonthlyIncome (Rs.)
Shah	Male	114000.00
Vats	Male	65000.00
Vats	Female	43150.00
Kumar	Female	69500.00
Vats	Female	155000.00
Kumar	Male	103000.00
Shah	Male	55000.00
Shah	Female	112400.00
Kumar	Female	81030.00
Vats	Male	71900.00

Write a program in Python using Pandas to perform the following:
a. Calculate and display familywise gross monthly income.
b. Calculate and display the member with the highest monthly income in a family.
c. Calculate and display monthly income of all members with income greater than Rs. 60000.00.
d. Calculate and display the average monthly income of the female members in the Shah family.

</td>
<td>38</td>
</tr>
</table>

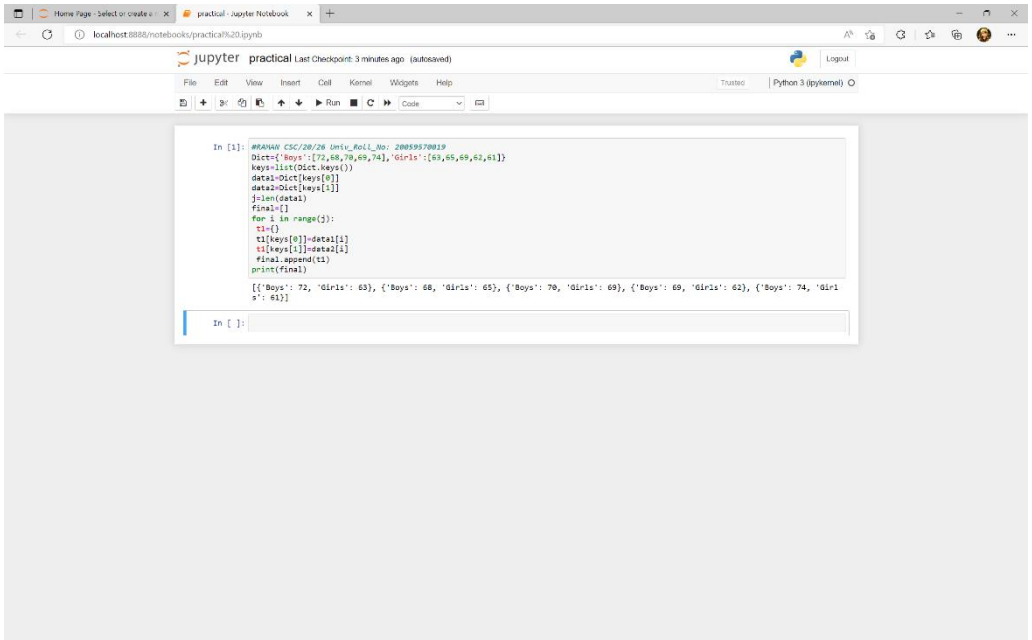
Ques 1.

Given below is a dictionary having two keys ‘Boys’ and ‘Girls’ and having two lists of heights of five Boys and Five Girls respectively as values associated with these keys
Original dictionary of lists: {'Boys': [72, 68, 70, 69, 74], 'Girls': [63, 65, 69, 62, 61]}
From the given dictionary of lists create the following list of dictionaries: [{'Boys': 72, 'Girls': 63}, {'Boys': 68, 'Girls': 65}, {'Boys': 70, 'Girls': 69}, {'Boys': 69, 'Girls': 62}, {‘Boys’:74, ‘Girls’:61].

Code :

```
#RAMAN CSC/20/26 Univ_Roll_No: 20059570019

Dict={'Boys':[72,68,70,69,74],'Girls':[63,65,69,62,61]}

keys=list(Dict.keys())

data1=Dict[keys[0]]

data2=Dict[keys[1]]

j=len(data1)

final=[]

for i in range(j):

    t1={}

    t1[keys[0]]=data1[i]

    t1[keys[1]]=data2[i]

    final.append(t1)

print(final)
```

OUTPUT :-



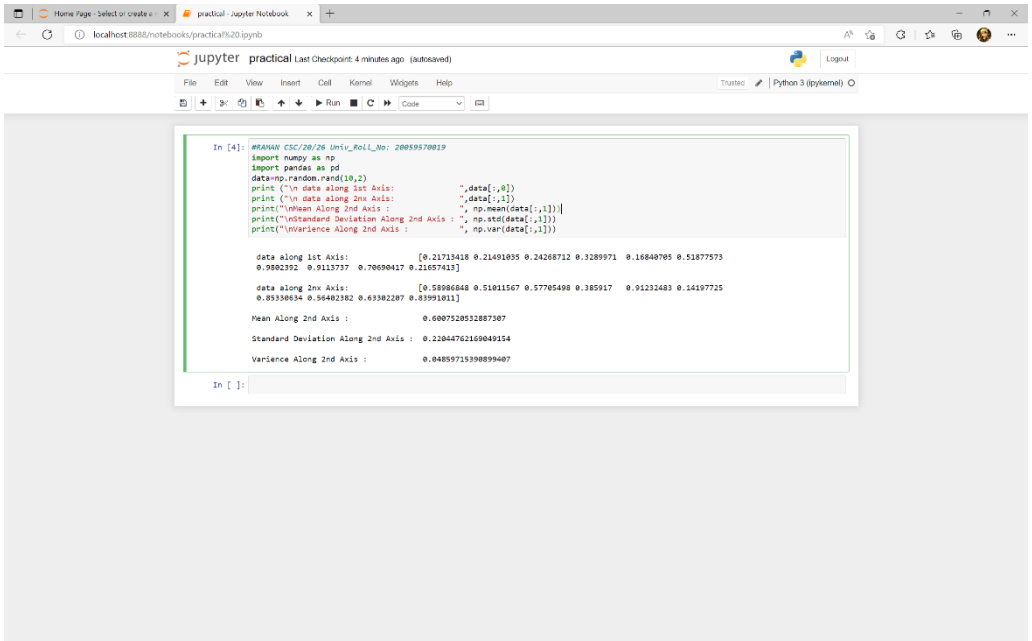
Ques 2.

Write programs in Python using NumPy library to do the following:
a) Compute the mean, standard deviation, and variance of a two dimensional random integer array along the second axis.
b) Get the indices of the sorted elements of a given array.
a. B = [56, 48, 22, 41, 78, 91, 24, 46, 8, 33]
c) Create a 2-dimensional array of size m x n integer elements, also print the shape, type and data type of the array and then reshape it into nx m array, n and m are user inputs given at the run time.
d) Test whether the elements of a given array are zero, non-zero and NaN. Record the indices of these elements in three separate arrays.

Code a) :

```
#RAMAN CSC/20/26 Univ_Roll_No: 20059570019

import numpy as np

import pandas as pd

data=np.random.rand(10,2)

print ("\n data  along 1st Axis: ",data[:,0])

print ("\n data  along 2nx Axis: ",data[:,1])

print("\nMean Along 2nd Axis : ", np.mean(data[:,1]))

print("\nStandard Deviation Along 2nd Axis : ", np.std(data[:,1]))

print("\nVarience Along 2nd Axis : ", np.var(data[:,1]))
```

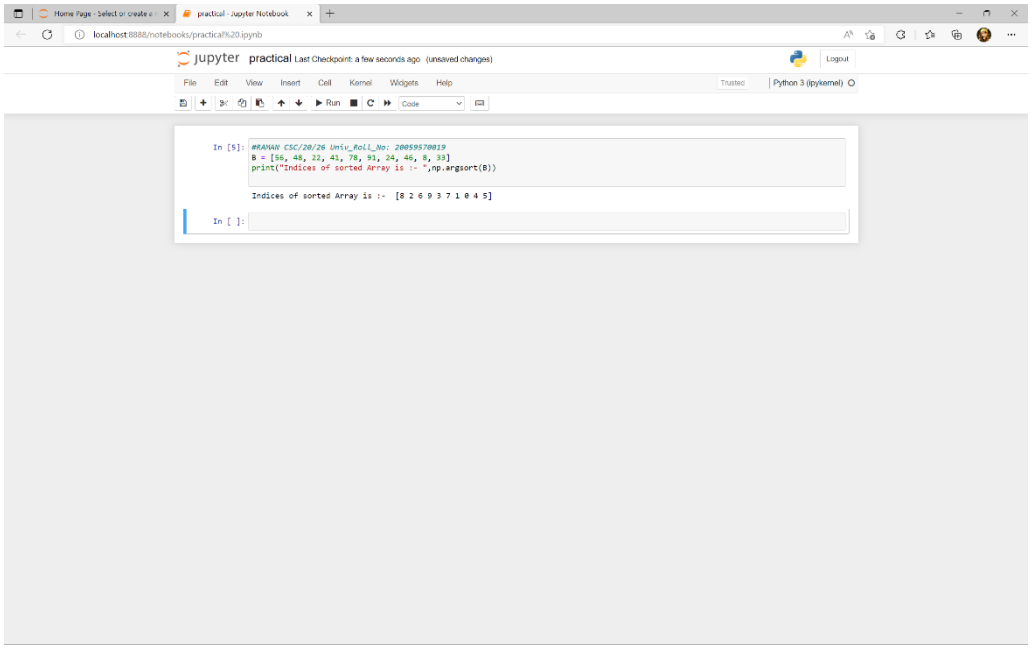
OUTPUT :-



Code b) :

```
#RAMAN CSC/20/26 Univ_Roll_No: 20059570019

B = [56, 48, 22, 41, 78, 91, 24, 46, 8, 33]

print("Indices of sorted Array is :- ",np.argsort(B))
```

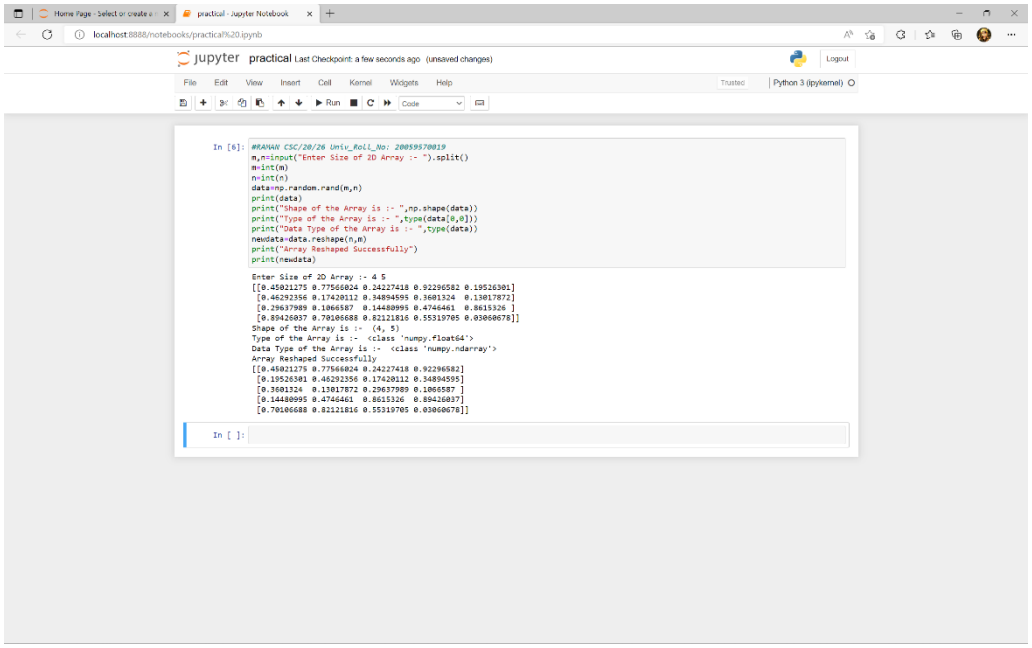
OUTPUT :-



Code c) :

```
#RAMAN CSC/20/26 Univ_Roll_No: 20059570019
m,n=input("Enter Size of 2D Array :- ").split()
m=int(m)
n=int(n)
data=np.random.rand(m,n)
print(data)
print("Shape of the Array is :- ",np.shape(data))
print("Type of the Array is :- ",type(data[0,0]))
print("Data Type of the Array is :- ",type(data))
newdata=data.reshape(n,m)
print("Array Reshaped Successfully")
print(newdata)
```

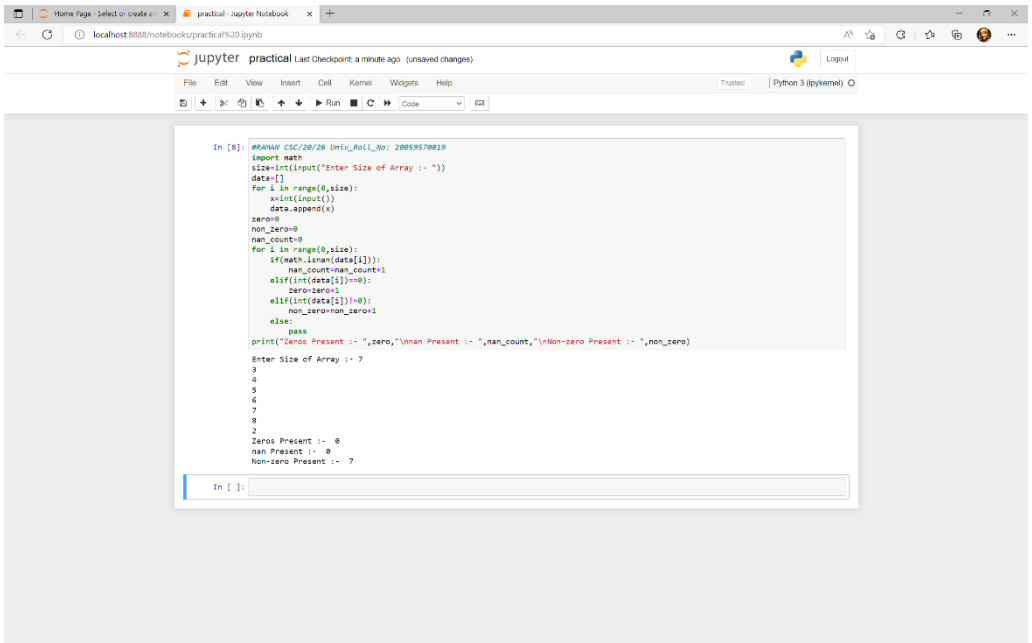
OUTPUT :-



Code d) :

```
#RAMAN CSC/20/26 Univ_Roll_No: 20059570019
import math
size=int(input("Enter Size of Array :- "))
data=[]
for i in range(0,size):
    x=int(input())
    data.append(x)
zero=0
non_zero=0
nan_count=0
for i in range(0,size):
    if(math.isnan(data[i])):
        nan_count=nan_count+1
    elif(int(data[i])==0):
        zero=zero+1
    elif(int(data[i])!=0):
        non_zero=non_zero+1
    else:
        pass
print("Zeros Present :- ",zero,"\nnan Present :- ",nan_count,"\nNon-zero Present :- ",non_zero)
```

OUTPUT :-



Ques 3.

Create a dataframe having at least 3 columns and 50 rows to store numeric data generated using a random function. Replace 10% of the values by null values whose index positions are generated using random function.Do the following:

a. Identify and count missing values in a dataframe.
b. Drop the column having more than 5 null values.
c. Identify the row label having maximum of the sum of all values in a row and drop that row.
d. Sort the dataframe on the basis of the first column.
e. Remove all duplicates from the first column.
f. Find the correlation between first and second column and covariance between 2nd & 3rd column.
g. Detect the outliers and remove the rows having outliers.
h. Discretize second column and create 5 bins

Code a) :

```
#RAMAN CSC/20/26 Univ_Roll_No: 20059570019

import numpy as np

import pandas as pd

import math

firstdata=np.random.randint(1,100,size=(50,3))

df=pd.DataFrame(firstdata,columns=['x','Y','Z'])

size2=len(df)*0.1

size2=int(size2)

for i in range(0,(size2*3)):

        df.iat[np.random.randint(1,50,),1]=float("nan")

count_nan=0

for i in range(0,len(df.columns)):

    for j in range(0,len(df)):

        if(math.isnan(df.iat[j,i])):

            count_nan=count_nan+1

print("Number  of Missing Values :- ",count_nan)
```

OUTPUT :-

In [9]:
```
#RAMAN CSC/20/26 Univ_Roll_No: 20059570019
import numpy as np
import pandas as pd
import math
firstdata=np.random.randint(1,100,size=(50,3))
df=pd.DataFrame(firstdata,columns=['X','Y','Z'])
size2=len(df)*0.1
size2=int(size2)
for i in range(0,(size2*3)):
        df.iat[np.random.randint(1,50,),1]=float("nan")
count_nan=0
for i in range(0,len(df.columns)):
    for j in range(0,len(df)):
        if(math.isnan(df.iat[j,i])):
            count_nan=count_nan+1
print("Number  of Missing Values :- ",count_nan)
```

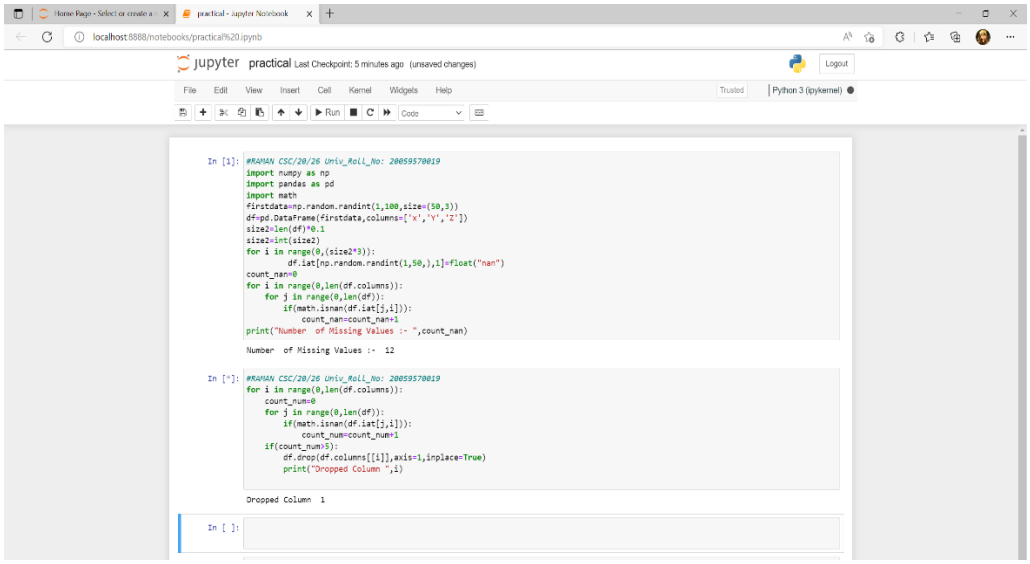
```
Number  of Missing Values :-  14
```

In [ ]:

Code b) :

```
#RAMAN CSC/20/26 Univ_Roll_No: 20059570019
 for i in range(0,len(df.columns)):
   count_num=0
   for j in range(0,len(df)):
      if(math.isnan(df.iat[j,i])):
         count_num=count_num+1
   if(count_num>5):
      df.drop(df.columns[[i]],axis=1,inplace=True)
      print("Dropped Column ",i)
```

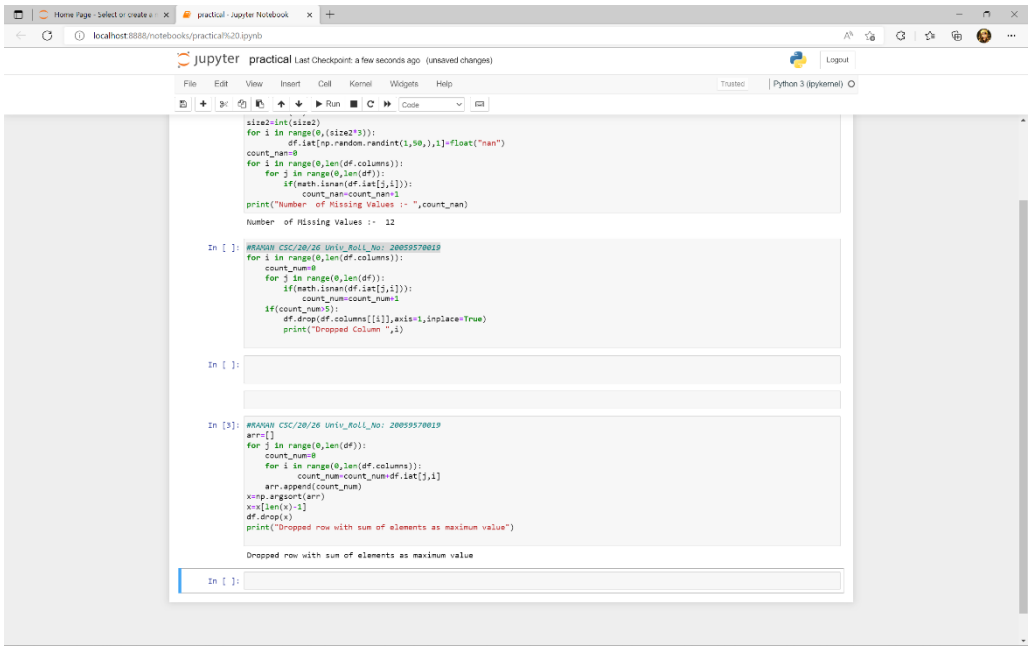
OUTPUT :-



Code c) :

```
#RAMAN CSC/20/26 Univ_Roll_No: 20059570019
arr=[]
for j in range(0,len(df)):
    count_num=0
    for i in range(0,len(df.columns)):
        count_num=count_num+df.iat[j,i]
    arr.append(count_num)
x=np.argsort(arr)
x=x[len(x)-1]
df.drop(x)
print("Dropped row with sum of elements as maximum value")
```

OUTPUT :-

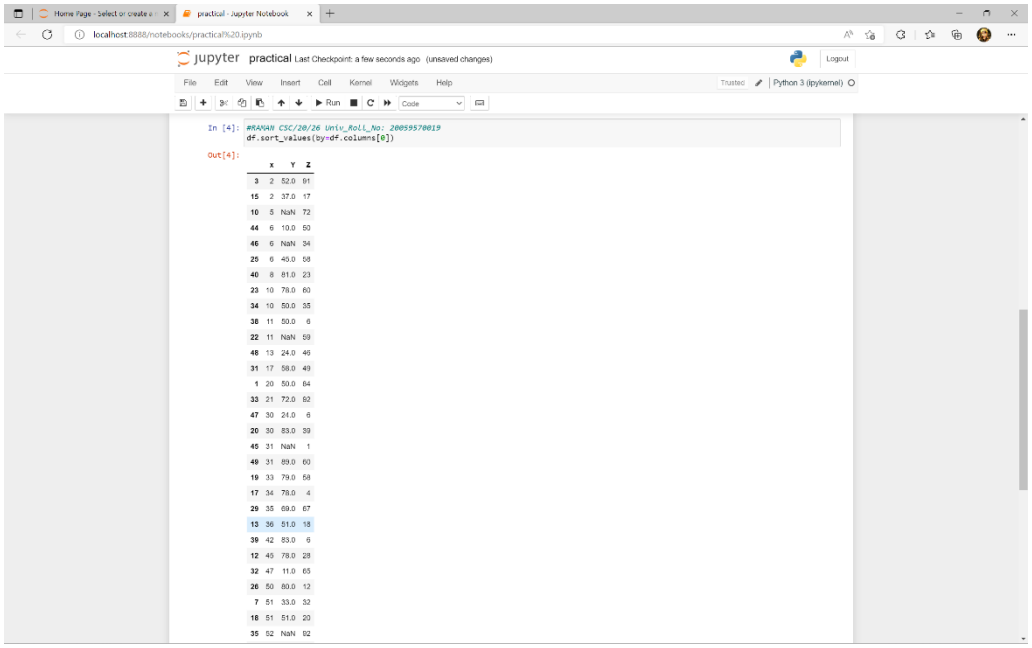


Code d) :

#RAMAN CSC/20/26 Univ_Roll_No: 20059570019

df.sort_values(by=df.columns[0])

OUTPUT :-

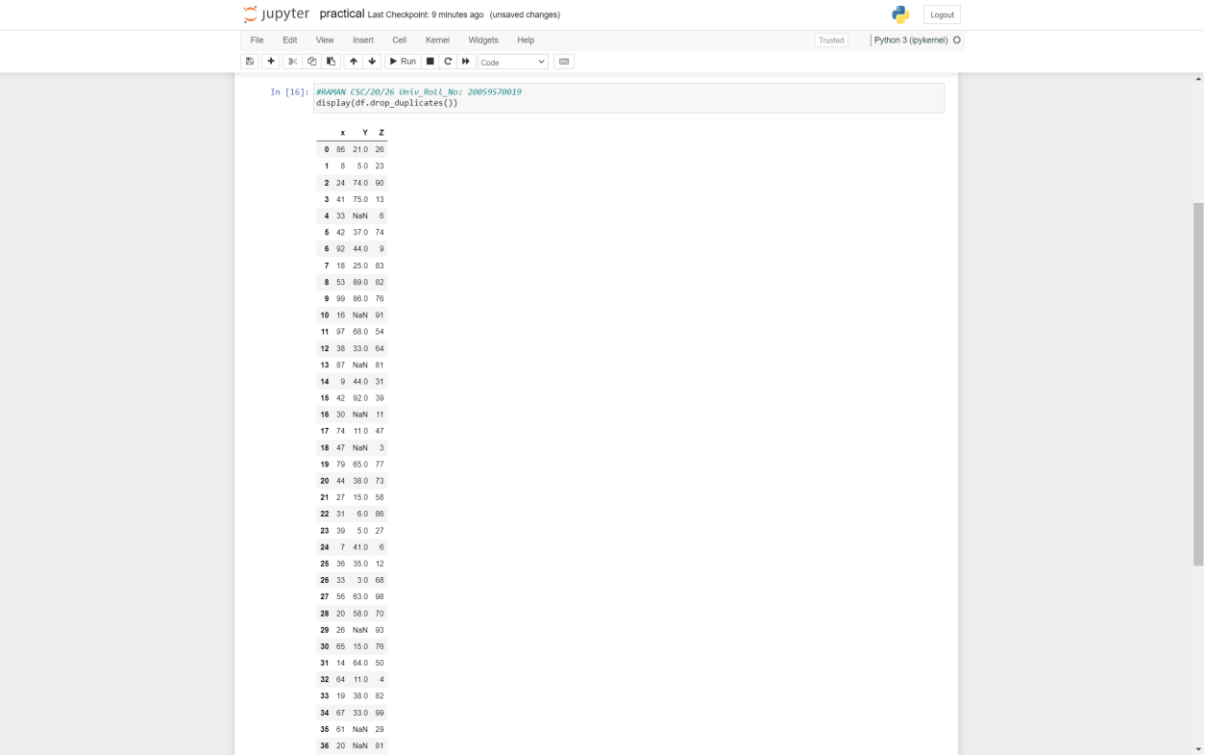


	x	Y	Z
3	2	52.0	91
15	2	37.0	17
10	5	NaN	72
44	6	10.0	50
46	6	NaN	34
25	6	45.0	58
40	8	81.0	23
23	10	78.0	60
34	10	50.0	35
38	11	50.0	6
22	11	NaN	59
48	13	24.0	46
31	17	58.0	49
1	20	50.0	84
33	21	72.0	82
47	30	24.0	6
20	30	83.0	39
45	31	NaN	1
49	31	89.0	60
19	33	79.0	58
17	34	78.0	4
29	35	69.0	67
13	36	51.0	18
39	42	83.0	6
12	45	78.0	28
32	47	11.0	65
26	50	80.0	12
7	51	33.0	32
18	51	51.0	20
35	52	NaN	92

Code e) :

```
#RAMAN CSC/20/26 Univ_Roll_No: 20059570019

display(df.drop_duplicates())
```

OUTPUT :-

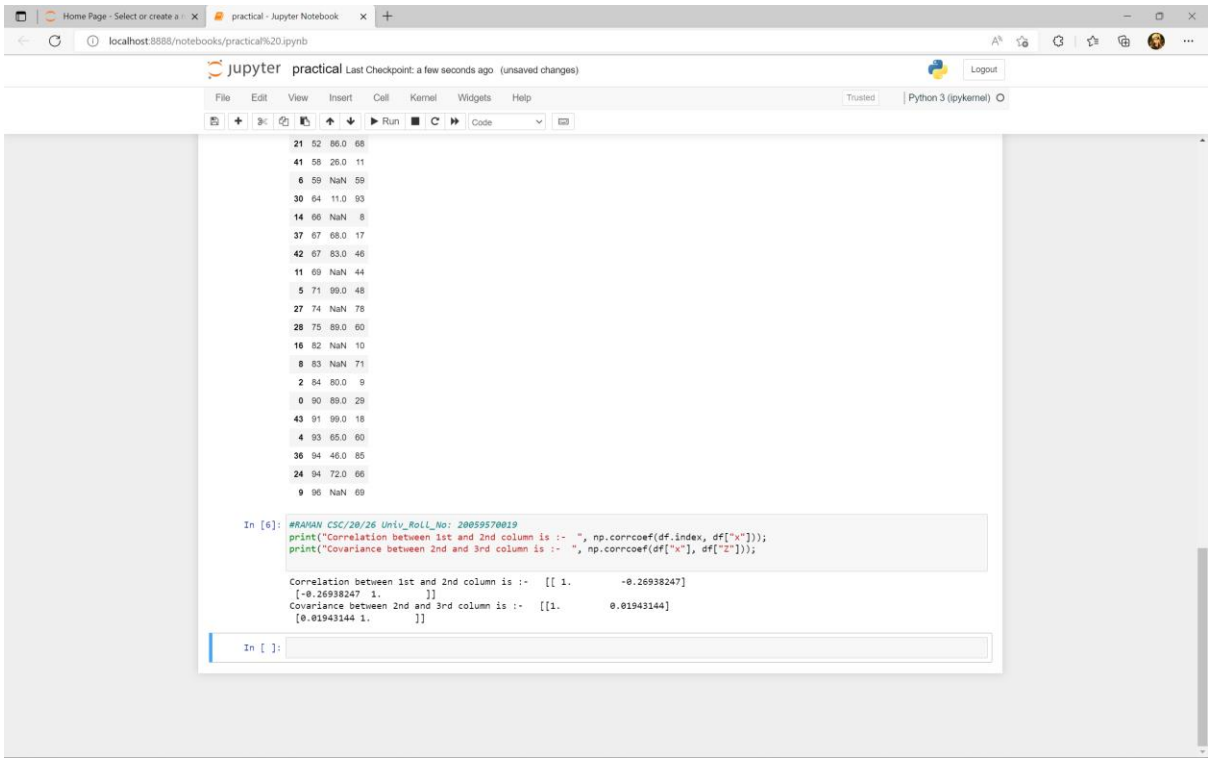


	x	Y	Z
0	86	21.0	26
1	8	5.0	23
2	24	74.0	90
3	41	75.0	13
4	33	NaN	6
5	42	37.0	74
6	92	44.0	9
7	18	25.0	83
8	53	89.0	82
9	99	86.0	76
10	16	NaN	91
11	97	68.0	54
12	38	33.0	64
13	87	NaN	81
14	9	44.0	31
15	42	92.0	39
16	30	NaN	11
17	74	11.0	47
18	47	NaN	3
19	79	65.0	77
20	44	38.0	73
21	27	15.0	58
22	31	6.0	86
23	39	5.0	27
24	7	41.0	6
25	36	35.0	12
26	33	3.0	68
27	56	63.0	98
28	20	58.0	70
29	26	NaN	93
30	65	15.0	76
31	14	64.0	50
32	64	11.0	4
33	19	38.0	82
34	67	33.0	99
35	61	NaN	29
36	20	NaN	81

Code f) :

```
#RAMAN CSC/20/26 Univ_Roll_No: 20059570019
print("Correlation between 1st and 2nd column is :-  ", np.corrcoef(df.index, df["x"]));
print("Covariance between 2nd and 3rd column is :-  ", np.corrcoef(df["x"], df["Z"]));
```

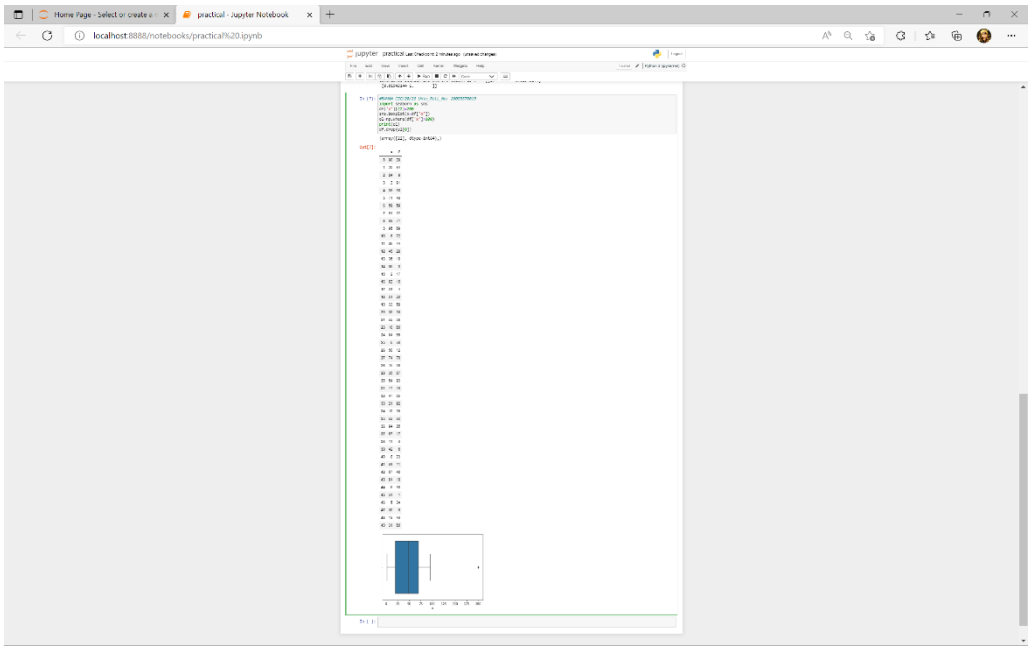
OUTPUT :-



Code g) :

```
#RAMAN CSC/20/26 Univ_Roll_No: 20059570019
 import seaborn as sns
df['x'][22]=200
sns.boxplot(x=df['x'])
ol=np.where(df['x']>100)
print(ol)
df.drop(ol[0])
```

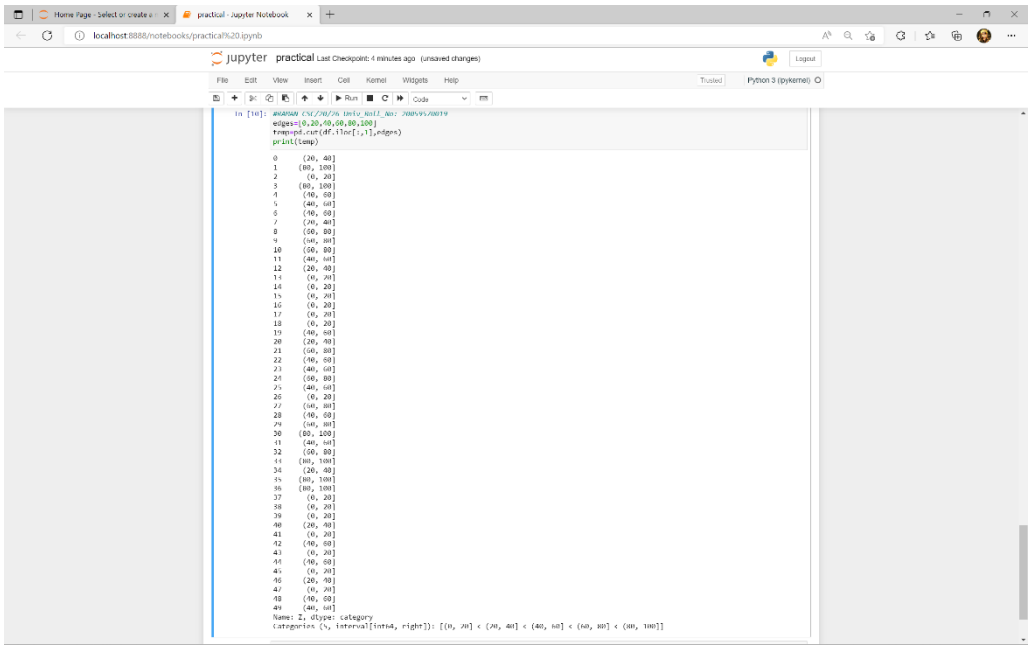
OUTPUT :-



Code h) :

```
#RAMAN CSC/20/26 Univ_Roll_No: 20059570019

edges=[0,20,40,60,80,100]

temp=pd.cut(df.iloc[:,2],edges)

print(temp)
```

OUTPUT :-



Ques 4.

Consider two excel files having attendance of a workshop's participants for two days. Each file has three fields 'Name', 'Time of joining', duration (in minutes) where names are unique within a file. Note that duration may take one of three values (30, 40, 50) only. Import the data into two dataframes and do the following:

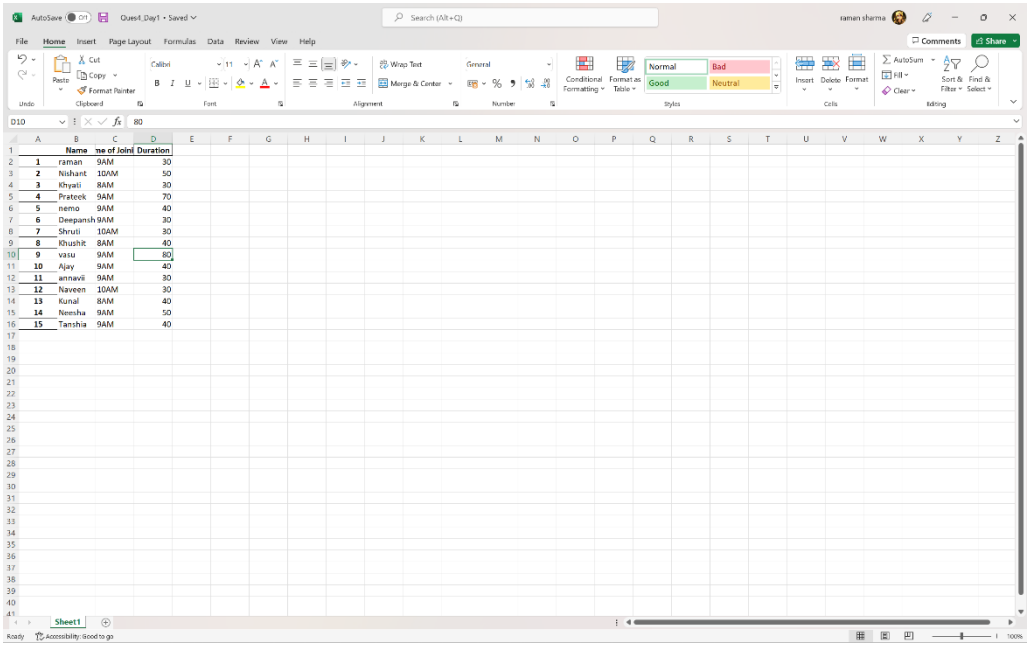
a. Perform merging of the two dataframes to find the names of students who had attended the workshop on both days.

b. Find names of all students who have attended workshop on either of the days.

c. Merge two data frames row-wise and find the total number of records in the data frame.

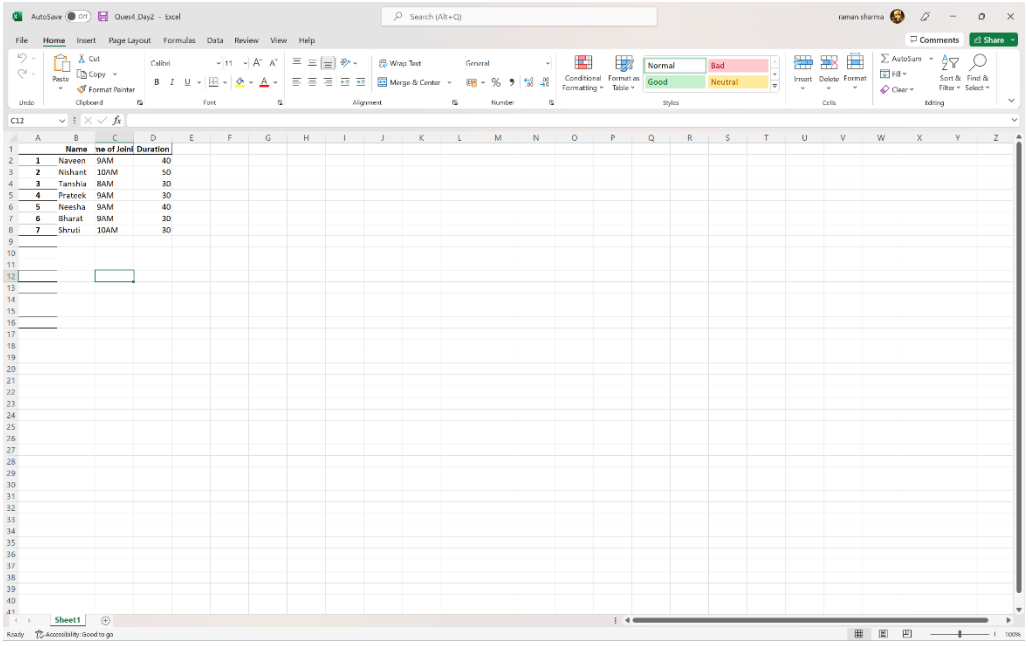
d. Merge two data frames and use two columns names and duration as multi-row indexes. Generate descriptive statistics for this multi-index.

1ST File



	Name	ne of Joini	Duration
1	raman	9AM	30
2	Nishant	10AM	50
3	Khyati	8AM	30
4	Prateek	9AM	70
5	nemo	9AM	40
6	Deepansh	9AM	30
7	Shruti	10AM	30
8	Khushit	8AM	40
9	vasu	9AM	80
10	Ajay	9AM	40
11	annavii	9AM	30
12	Naveen	10AM	30
13	Kunal	8AM	40
14	Neesha	9AM	50
15	Tanshia	9AM	40

2ND File

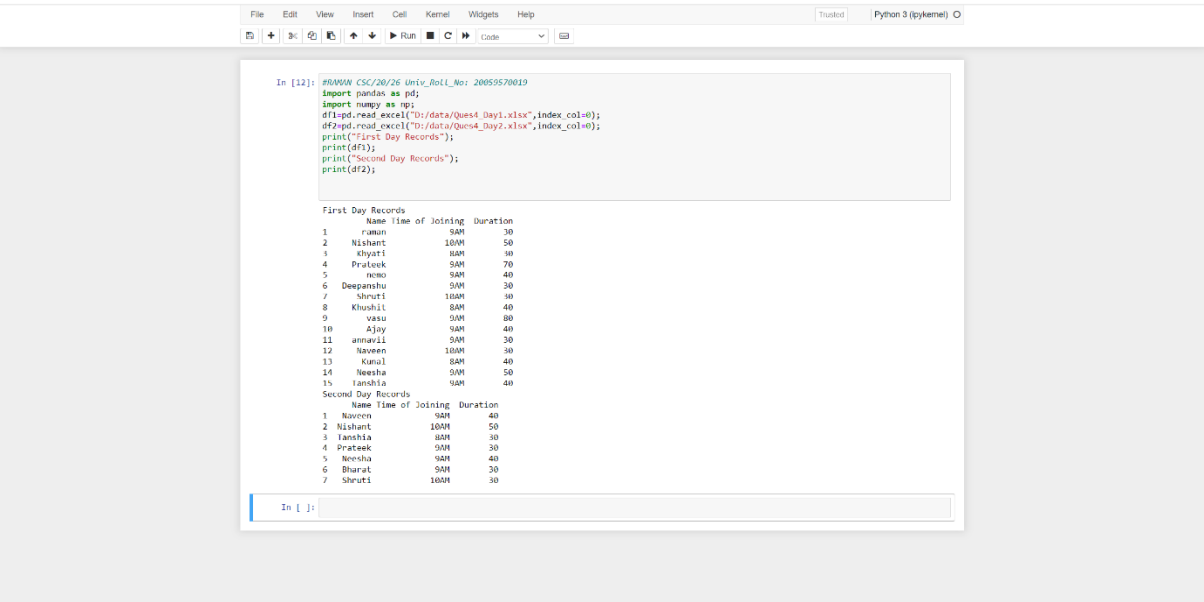


	Name	me of Joinl	Duration
1	Naveen	9AM	40
2	Nishant	10AM	50
3	Tanshia	8AM	30
4	Prateek	9AM	30
5	Neesha	9AM	40
6	Bharat	9AM	30
7	Shruti	10AM	30

Code :

```
#RAMAN CSC/20/26 Univ_Roll_No: 20059570019
import pandas as pd;
import numpy as np;
df1=pd.read_excel("D:/data/Ques4_Day1.xlsx",index_col=0);
df2=pd.read_excel("D:/data/Ques4_Day2.xlsx",index_col=0);
print("First Day Records");
print(df1);
print("Second Day Records");
print(df2);
```

OUTPUT :-



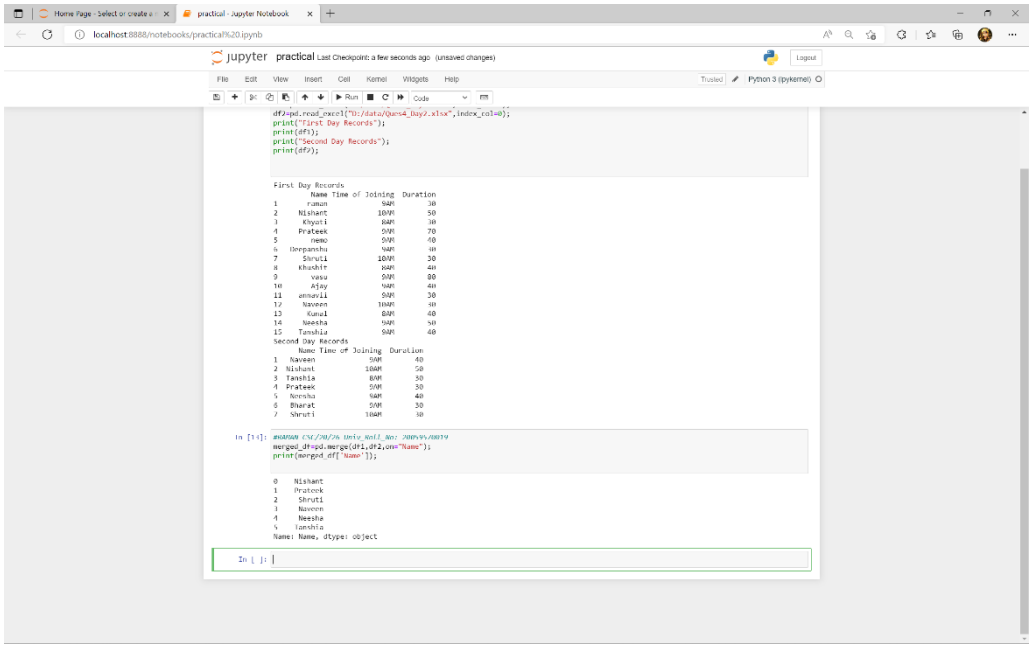
Code a) :

#RAMAN CSC/20/26 Univ_Roll_No: 20059570019

merged_df=pd.merge(df1,df2,on="Name");

print(merged_df['Name']);

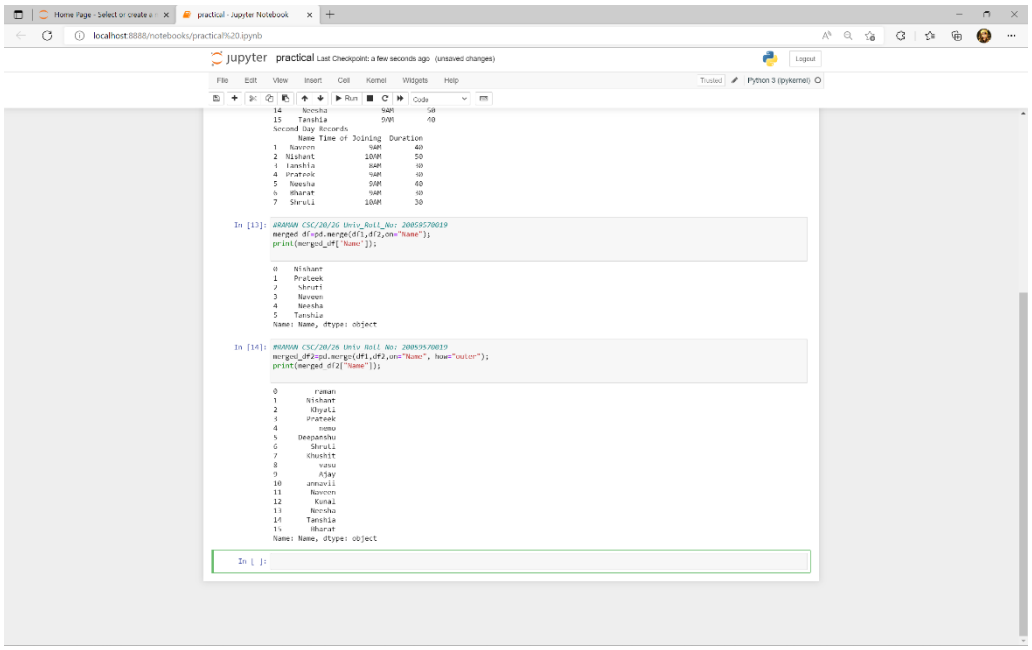
OUTPUT :-



Code b) :

```
#RAMAN CSC/20/26 Univ_Roll_No: 20059570019
merged_df2=pd.merge(df1,df2,on="Name", how="outer");
print(merged_df2["Name"]);
```

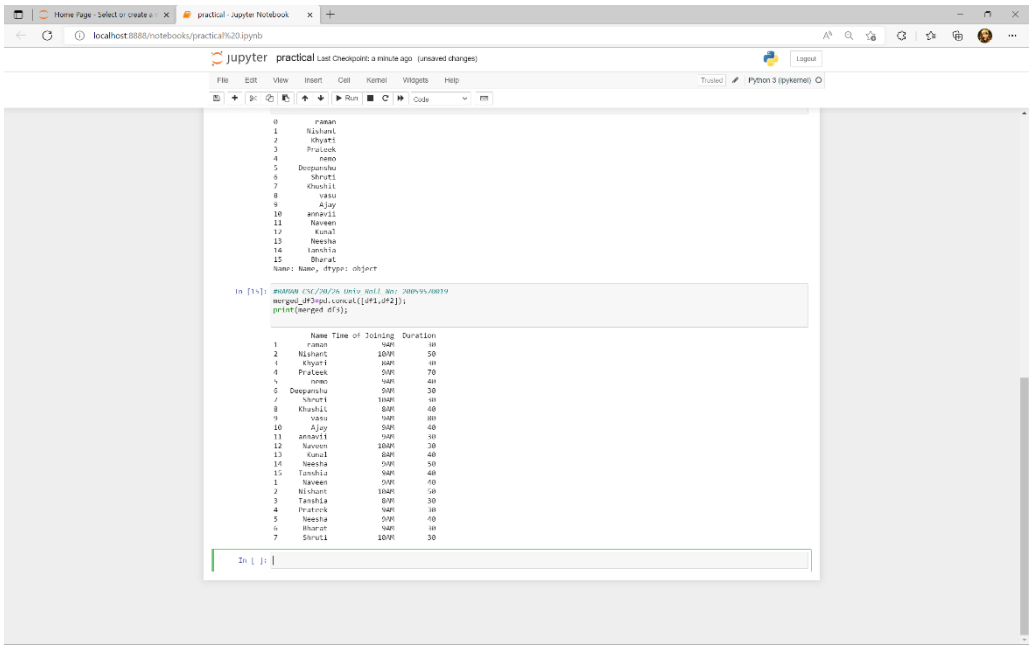
OUTPUT :-



Code c) :

```
#RAMAN CSC/20/26 Univ_Roll_No: 20059570019
merged_df3=pd.concat([df1,df2]);
print(merged_df3);
```

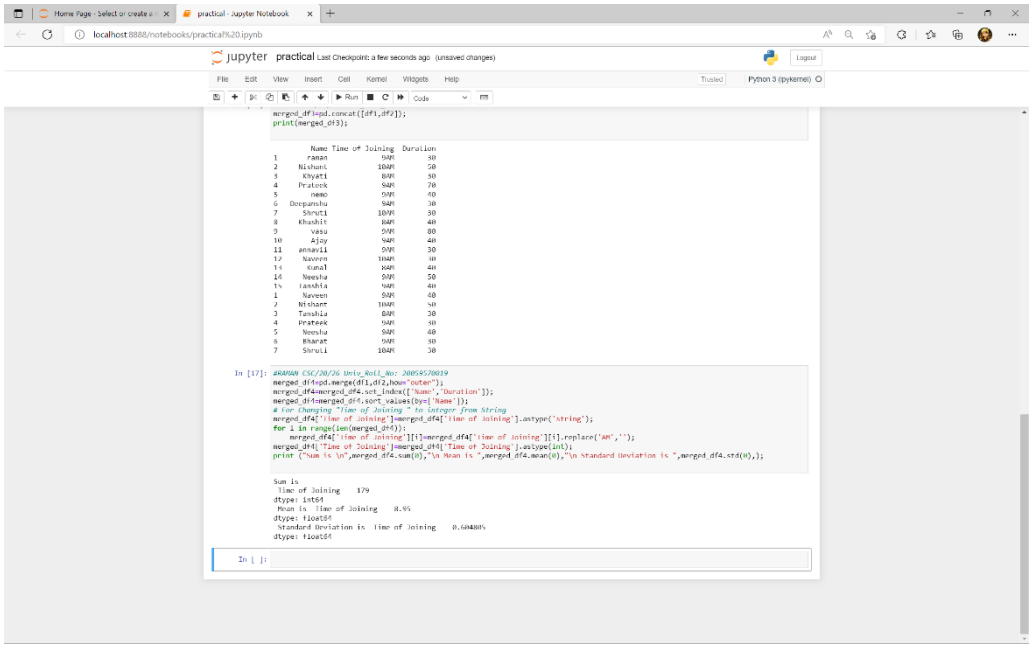
OUTPUT :-



Code d) :

```
#RAMAN CSC/20/26 Univ_Roll_No: 20059570019
merged_df4=pd.merge(df1,df2,how="outer");
merged_df4=merged_df4.set_index(['Name','Duration']);
merged_df4=merged_df4.sort_values(by=['Name']);
# For Changing "Time of Joining " to integer from String
merged_df4['Time of Joining']=merged_df4['Time of Joining'].astype('string');
for i in range(len(x)):
    merged_df4['Time of Joining'][i]=merged_df4['Time of Joining'][i].replace('AM','');
merged_df4['Time of Joining']=merged_df4['Time of Joining'].astype(int);
print ("Sum is \n",merged_df4.sum(0),"\n Mean is ",merged_df4.mean(0),"\n Standard Deviation is ",merged_df4.std(0),);
```

OUTPUT :-



Ques 5.

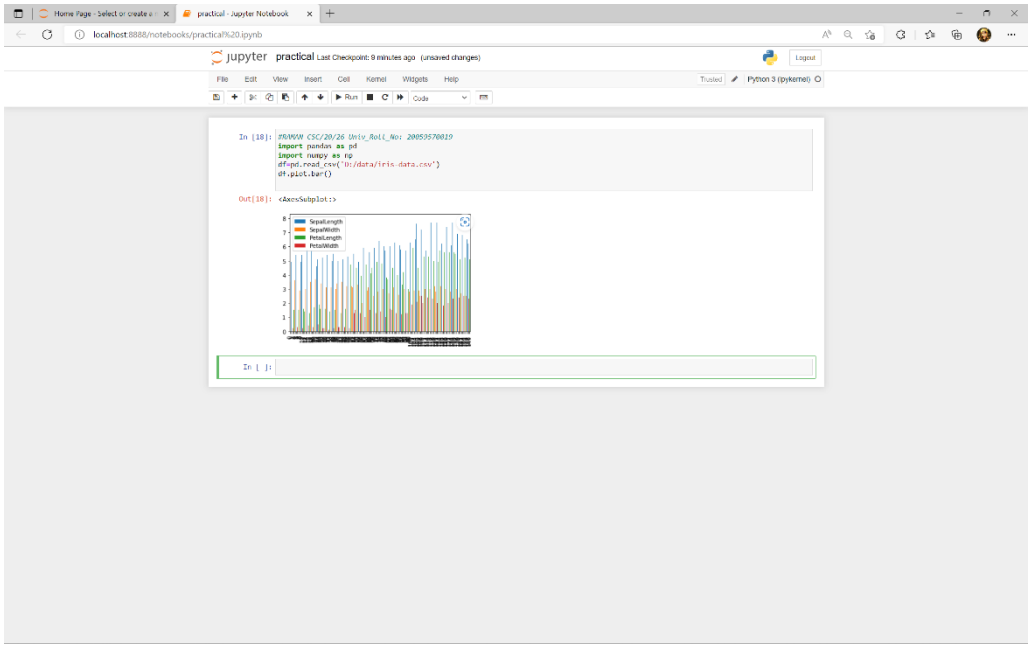
Taking Iris data, plot the following with proper legend and axis labels: (Download IRIS data from: https://archive.ics.uci.edu/ml/datasets/iris or import it from sklearn.datasets

a. Plot bar chart to show the frequency of each class label in the data.
b. Draw a scatter plot for Petal width vs sepal width.
c. Plot density distribution for feature petal length.
d. Use a pair plot to show pairwise bivariate distribution in the Iris Dataset.

Code a):

```
#RAMAN CSC/20/26 Univ_Roll_No: 20059570019

import pandas as pd

import numpy as np

df=pd.read_csv('D:/data/iris-data.csv')

df.plot.bar()
```

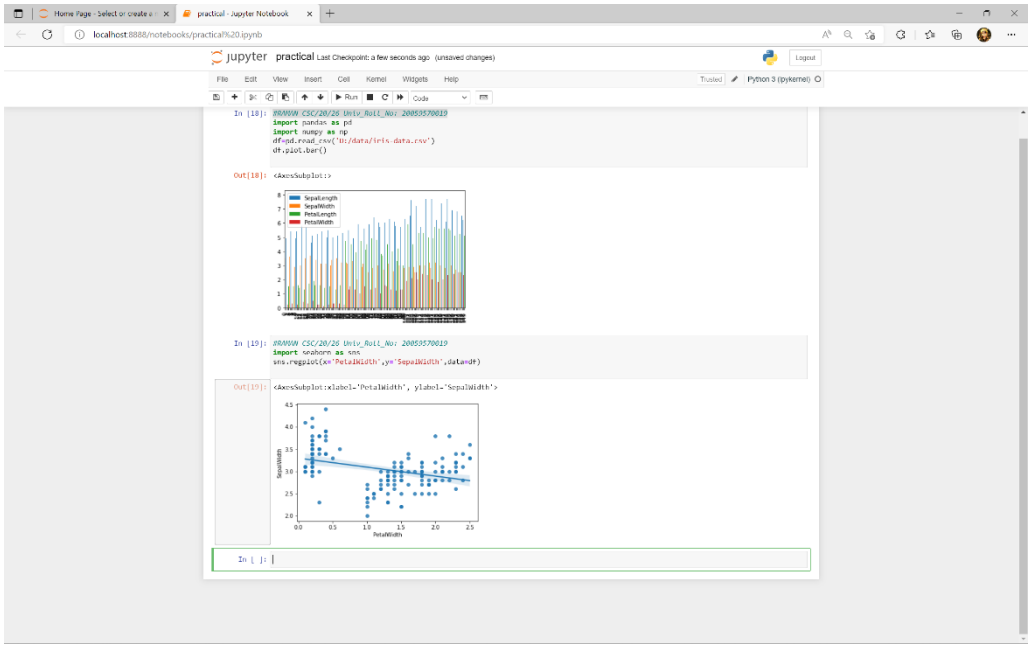
OUTPUT :-



Code b):

```
#RAMAN CSC/20/26 Univ_Roll_No: 20059570019
import seaborn as sns
sns.regplot(x='PetalWidth',y='SepalWidth',data=df)
```

OUTPUT :-

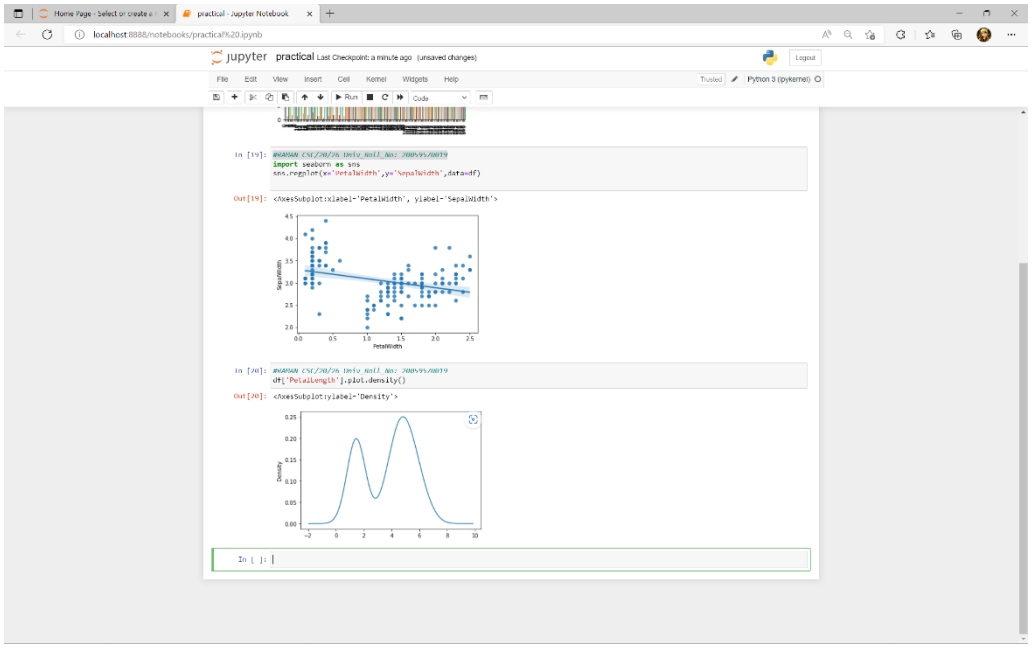


Code c):

#RAMAN CSC/20/26 Univ_Roll_No: 20059570019

df['PetalLength'].plot.density()

OUTPUT :-

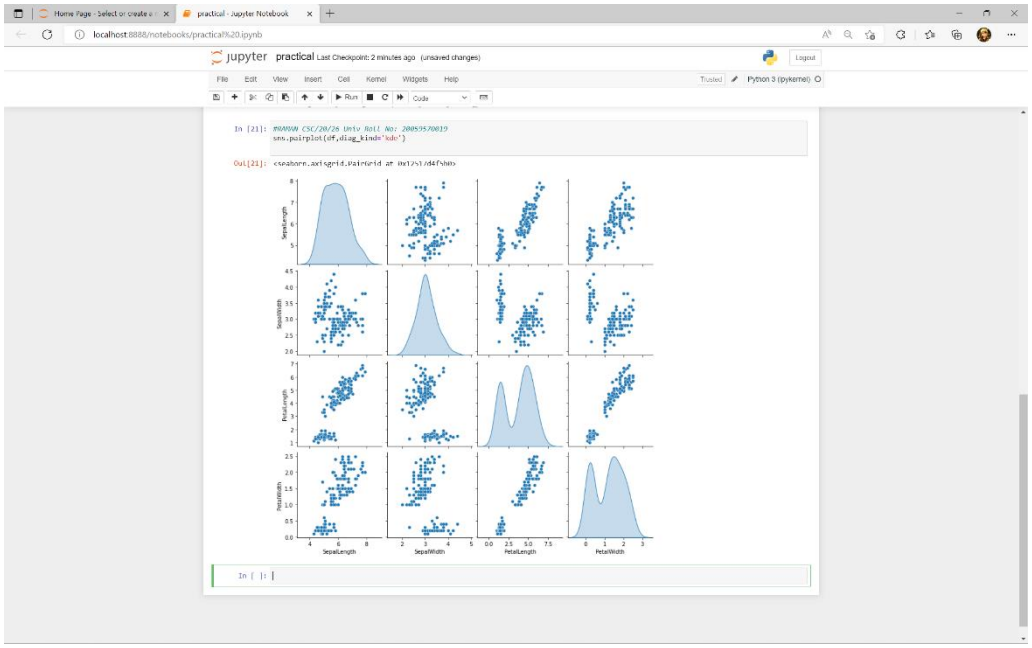


Code d):

#RAMAN CSC/20/26 Univ_Roll_No: 20059570019

sns.pairplot(df,diag_kind='kde')

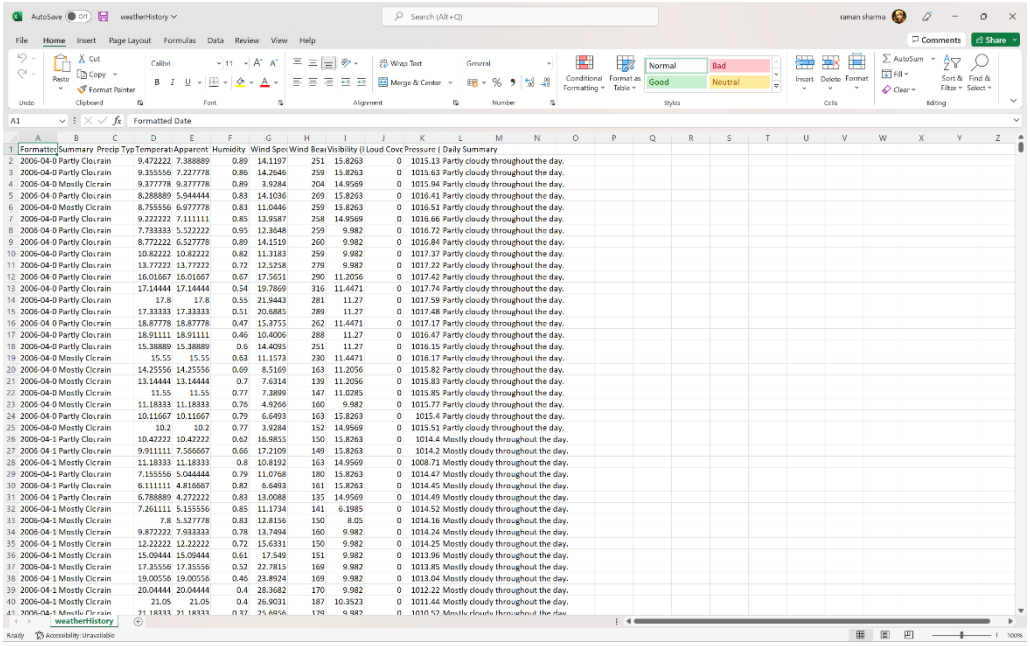
OUTPUT :-



Ques 6.

Consider any sales training/ weather forecasting dataset
a. Compute mean of a series grouped by another series
b. Fill an intermittent time series to replace all missing dates with values of previous non-missing date.
c. Perform appropriate year-month string to dates conversion.
d. Split a dataset to group by two columns and then sort the aggregated results within the groups.
e. Split a given dataframe into groups with bin counts

File taken :- weatherHistory.csv



Formatted	Summary	Precip Typ	Temperati	Apparent	Humidity	Wind Spe	Wind Bea	Visibility (I	Loud Cove	Pressure (	Daily Summary
2006-04-0	Partly Clo	rain	9.472222	7.388889	0.89	14.1197	251	15.8263	0	1015.13	Partly cloudy throughout the day.
2006-04-0	Partly Clo	rain	9.355556	7.227778	0.86	14.2646	259	15.8263	0	1015.63	Partly cloudy throughout the day.
2006-04-0	Mostly Clo	rain	9.377778	9.377778	0.89	3.9284	204	14.9569	0	1015.94	Partly cloudy throughout the day.
2006-04-0	Partly Clo	rain	8.288889	5.944444	0.83	14.1036	269	15.8263	0	1016.41	Partly cloudy throughout the day.
2006-04-0	Mostly Clo	rain	8.755556	6.977778	0.83	11.0446	259	15.8263	0	1016.51	Partly cloudy throughout the day.
2006-04-0	Partly Clo	rain	9.222222	7.111111	0.85	13.9587	258	14.9569	0	1016.66	Partly cloudy throughout the day.
2006-04-0	Partly Clo	rain	7.733333	5.522222	0.95	12.3648	259	9.982	0	1016.72	Partly cloudy throughout the day.
2006-04-0	Partly Clo	rain	8.772222	6.527778	0.89	14.1519	260	9.982	0	1016.84	Partly cloudy throughout the day.
2006-04-0	Partly Clo	rain	10.82222	10.82222	0.82	11.3183	259	9.982	0	1017.37	Partly cloudy throughout the day.
2006-04-0	Partly Clo	rain	13.77222	13.77222	0.72	12.5258	279	9.982	0	1017.22	Partly cloudy throughout the day.
2006-04-0	Partly Clo	rain	16.01667	16.01667	0.67	17.5651	290	11.2056	0	1017.42	Partly cloudy throughout the day.
2006-04-0	Partly Clo	rain	17.14444	17.14444	0.54	19.7869	316	11.4471	0	1017.74	Partly cloudy throughout the day.
2006-04-0	Partly Clo	rain	17.8	17.8	0.55	21.9443	281	11.27	0	1017.59	Partly cloudy throughout the day.
2006-04-0	Partly Clo	rain	17.33333	17.33333	0.51	20.6885	289	11.27	0	1017.48	Partly cloudy throughout the day.
2006-04-0	Partly Clo	rain	18.87778	18.87778	0.47	15.3755	262	11.4471	0	1017.17	Partly cloudy throughout the day.
2006-04-0	Partly Clo	rain	18.91111	18.91111	0.46	10.4006	288	11.27	0	1016.47	Partly cloudy throughout the day.
2006-04-0	Partly Clo	rain	15.38889	15.38889	0.6	14.4095	251	11.27	0	1016.15	Partly cloudy throughout the day.
2006-04-0	Mostly Clo	rain	15.55	15.55	0.63	11.1573	230	11.4471	0	1016.17	Partly cloudy throughout the day.
2006-04-0	Mostly Clo	rain	14.25556	14.25556	0.69	8.5169	163	11.2056	0	1015.82	Partly cloudy throughout the day.
2006-04-0	Mostly Clo	rain	13.14444	13.14444	0.7	7.6314	139	11.2056	0	1015.83	Partly cloudy throughout the day.
2006-04-0	Mostly Clo	rain	11.55	11.55	0.77	7.3899	147	11.0285	0	1015.85	Partly cloudy throughout the day.
2006-04-0	Mostly Clo	rain	11.18333	11.18333	0.76	4.9266	160	9.982	0	1015.77	Partly cloudy throughout the day.
2006-04-0	Partly Clo	rain	10.11667	10.11667	0.79	6.6493	163	15.8263	0	1015.4	Partly cloudy throughout the day.
2006-04-0	Mostly Clo	rain	10.2	10.2	0.77	3.9284	152	14.9569	0	1015.51	Partly cloudy throughout the day.
2006-04-1	Partly Clo	rain	10.42222	10.42222	0.62	16.9855	150	15.8263	0	1014.4	Mostly cloudy throughout the day.
2006-04-1	Partly Clo	rain	9.911111	7.566667	0.66	17.2109	149	15.8263	0	1014.2	Mostly cloudy throughout the day.
2006-04-1	Mostly Clo	rain	11.18333	11.18333	0.8	10.8192	163	14.9569	0	1008.71	Mostly cloudy throughout the day.
2006-04-1	Partly Clo	rain	7.155556	5.044444	0.79	11.0768	180	15.8263	0	1014.47	Mostly cloudy throughout the day.
2006-04-1	Partly Clo	rain	6.111111	4.816667	0.82	6.6493	161	15.8263	0	1014.45	Mostly cloudy throughout the day.
2006-04-1	Partly Clo	rain	6.788889	4.272222	0.83	13.0088	135	14.9569	0	1014.49	Mostly cloudy throughout the day.
2006-04-1	Mostly Clo	rain	7.261111	5.155556	0.85	11.1734	141	6.1985	0	1014.52	Mostly cloudy throughout the day.
2006-04-1	Mostly Clo	rain	7.8	5.527778	0.83	12.8136	150	8.05	0	1014.16	Mostly cloudy throughout the day.
2006-04-1	Mostly Clo	rain	9.872222	7.933333	0.78	13.7494	160	9.982	0	1014.24	Mostly cloudy throughout the day.
2006-04-1	Mostly Clo	rain	12.22222	12.22222	0.72	15.6331	150	9.982	0	1014.25	Mostly cloudy throughout the day.
2006-04-1	Mostly Clo	rain	15.09444	15.09444	0.61	17.549	151	9.982	0	1013.96	Mostly cloudy throughout the day.
2006-04-1	Mostly Clo	rain	17.35556	17.35556	0.52	22.7815	169	9.982	0	1013.85	Mostly cloudy throughout the day.
2006-04-1	Mostly Clo	rain	19.00556	19.00556	0.46	23.8924	169	9.982	0	1013.04	Mostly cloudy throughout the day.
2006-04-1	Mostly Clo	rain	20.04444	20.04444	0.4	28.3682	170	9.982	0	1012.22	Mostly cloudy throughout the day.
2006-04-1	Mostly Clo	rain	21.05	21.05	0.4	26.9031	187	10.3523	0	1011.44	Mostly cloudy throughout the day.
2006-04-1	Mostly Clo	rain	21.18333	21.18333	0.37	25.6856	179	9.982	0	1010.52	Mostly cloudy throughout the day.

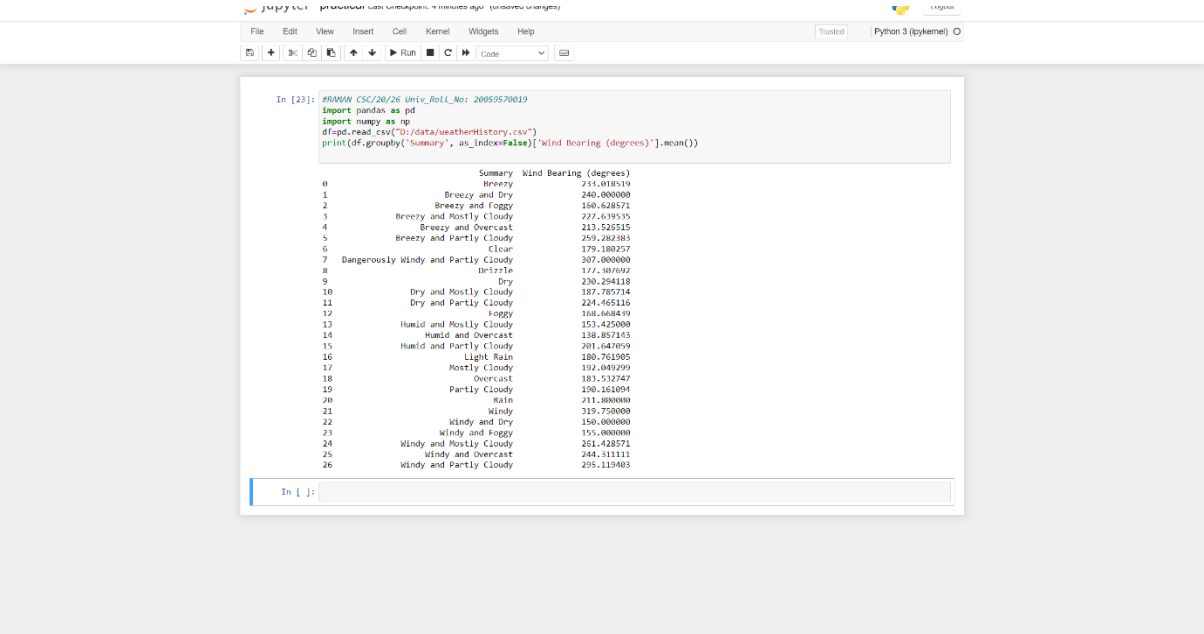
Code a):

Note:- I have calculated Mean value of "Wind Bearing (degrees)" grouped by "Summary"

```
#RAMAN CSC/20/26 Univ_Roll_No: 20059570019

import pandas as pd

import numpy as np

df=pd.read_csv("D:/data/weatherHistory.csv")
```

print(df.groupby('Summary', as_index=False)['Wind Bearing (degrees)'].mean())

OUTPUT :-



```
In [23]: #RAMAN CSC/20/26 Univ_Roll_No: 20059570019
         import pandas as pd
         import numpy as np
         df=pd.read_csv("D:/data/weatherHistory.csv")
         print(df.groupby('Summary', as_index=False)['Wind Bearing (degrees)'].mean())

                                 Summary  Wind Bearing (degrees)
0                                 Breezy              233.018519
1                        Breezy and Dry              240.000000
2                      Breezy and Foggy              160.628571
3                Breezy and Mostly Cloudy              227.639535
4                    Breezy and Overcast              213.526515
5                Breezy and Partly Cloudy              259.282383
6                                  Clear              179.180257
7   Dangerously Windy and Partly Cloudy              307.000000
8                               Drizzle              177.307692
9                                    Dry              230.294118
10                 Dry and Mostly Cloudy              187.785714
11                 Dry and Partly Cloudy              224.465116
12                                 Foggy              168.648439
13                Humid and Mostly Cloudy              153.425000
14                    Humid and Overcast              138.857143
15                Humid and Partly Cloudy              201.647059
16                            Light Rain              180.761905
17                        Mostly Cloudy              192.049299
18                              Overcast              183.532747
19                        Partly Cloudy              190.161094
20                                 Rain              211.000000
21                                Windy              319.750000
22                          Windy and Dry              150.000000
23                        Windy and Foggy              155.000000
24                  Windy and Mostly Cloudy              261.428571
25                      Windy and Overcast              244.311111
26                  Windy and Partly Cloudy              295.119403

In [ ]:
```

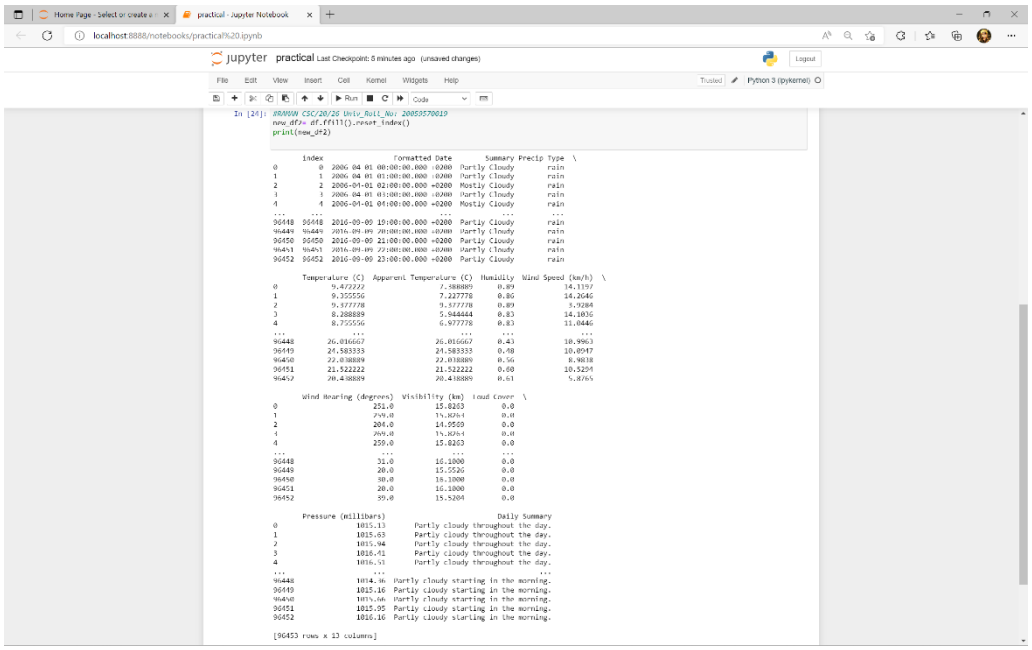
Code b):

#RAMAN CSC/20/26 Univ_Roll_No: 20059570019

new_df2= df.ffill().reset_index()

print(new_df2)

OUTPUT :-

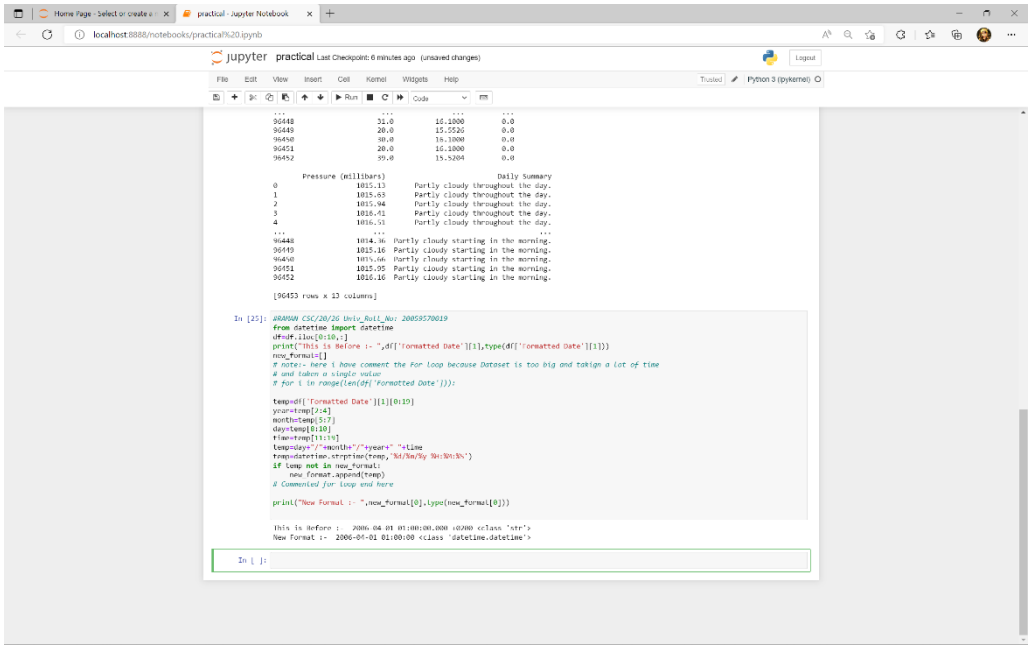


Code c):

#RAMAN CSC/20/26 Univ_Roll_No: 20059570019

```
from datetime import datetime
df=df.iloc[0:10,:]
print("This is Before :- ",df['Formatted Date'][1],type(df['Formatted Date'][1]))
new_format=[]
# note:- here i have comment the For loop because Dataset is too big and takign a lot of time
# and taken a single value
# for i in range(len(df['Formatted Date'])):

temp=df['Formatted Date'][1][0:19]
year=temp[2:4]
month=temp[5:7]
day=temp[8:10]
time=temp[11:19]
temp=day+"/"+month+"/"+year+" "+time
temp=datetime.strptime(temp,'%d/%m/%y %H:%M:%S')
if temp not in new_format:
    new_format.append(temp)
# Commented for loop end here

print("New Format :- ",new_format[0],type(new_format[0]))
```

OUTPUT :-

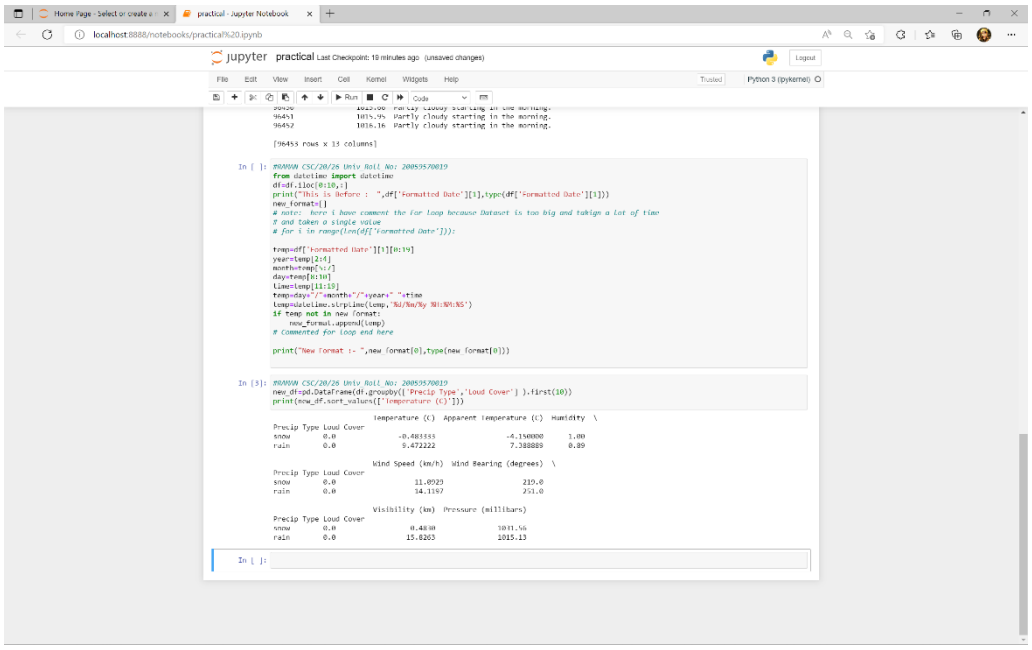


Code d):

#RAMAN CSC/20/26 Univ_Roll_No: 20059570019

```
new_df=pd.DataFrame(df.groupby(['Precip Type','Loud Cover'] ).first(10))

print(new_df.sort_values(['Temperature (C)']))
```

OUTPUT :-

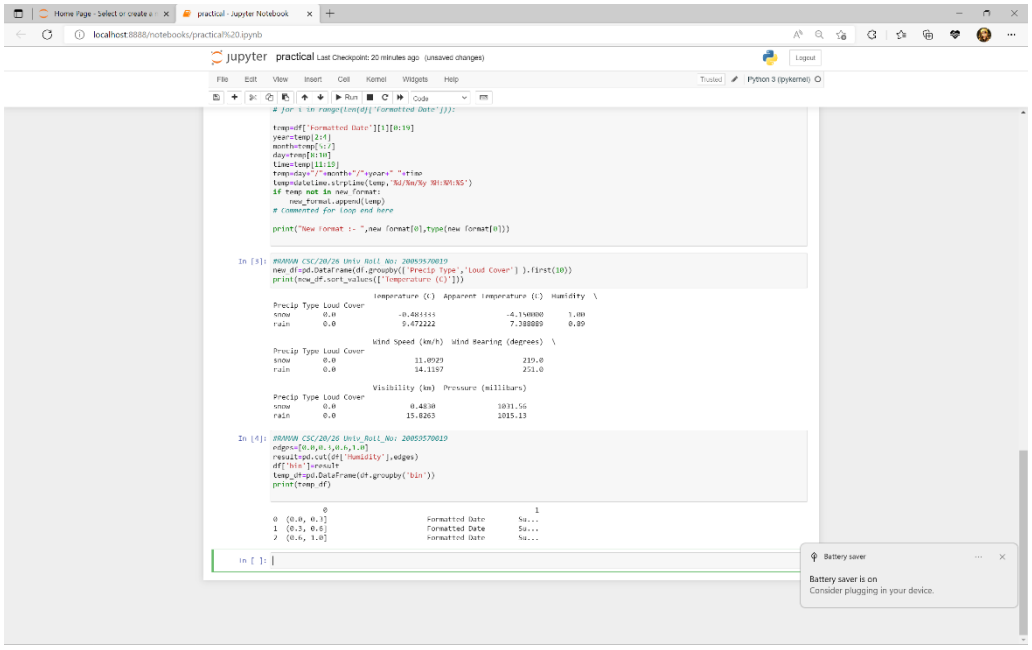


Code e):

#RAMAN CSC/20/26 Univ_Roll_No: 20059570019

```
edges=[0.0,0.3,0.6,1.0]
result=pd.cut(df['Humidity'],edges)
df['bin']=result
temp_df=pd.DataFrame(df.groupby('bin'))
print(temp_df)
```

OUTPUT :-



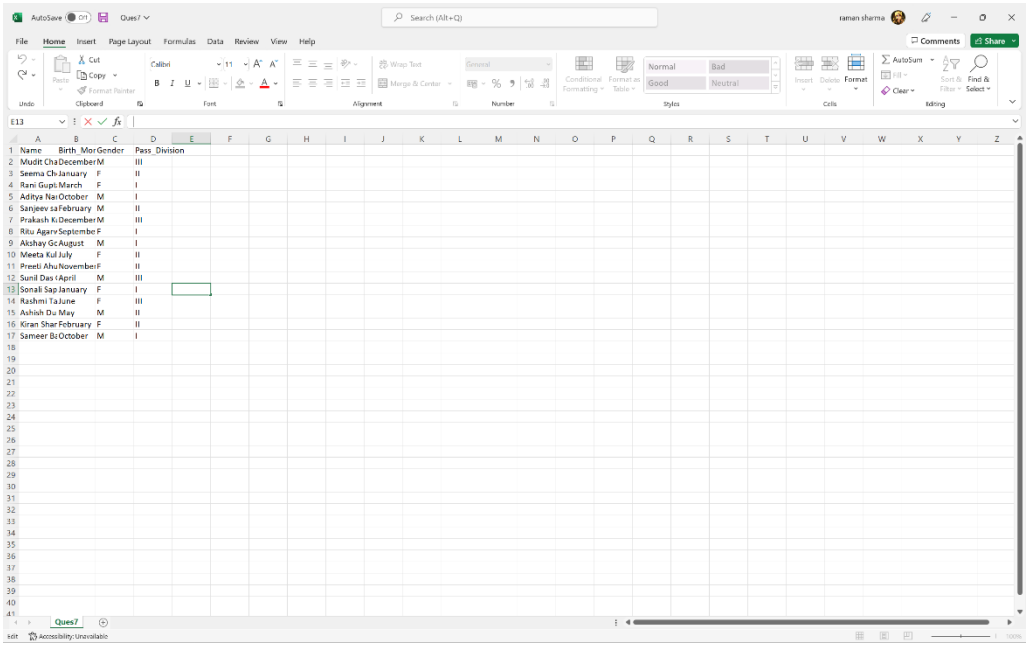
Ques 7.

Consider a data frame containing data about students i.e. name, gender and passing division:

	Name	Birth_Month	Gender	Pass_Division
0	Mudit Chauhan	December	M	III
1	Seema Chopra	January	F	II
2	Rani Gupta	March	F	I
3	Aditya Narayan	October	M	I
4	Sanjeev Sahni	February	M	II
5	Prakash Kumar	December	M	III
6	Ritu Agarwal	September	F	I
7	Akshay Goel	August	M	I
8	Meeta Kulkarni	July	F	II
9	Preeti Ahuja	November	F	II
10	Sunil Das Gupta	April	M	III
11	Sonali Sapre	January	F	I
12	Rashmi Talwar	June	F	III
13	Ashish Dubey	May	M	II
14	Kiran Sharma	February	F	II
15	Sameer Bansal	October	M	I

a. Perform one hot encoding of the last two columns of categorical data using the get_dummies() function.
b. Sort this data frame on the "Birth Month" column (i.e. January to December). Hint: Convert Month to Categorical.

Note: I have used a CSV File for storing data



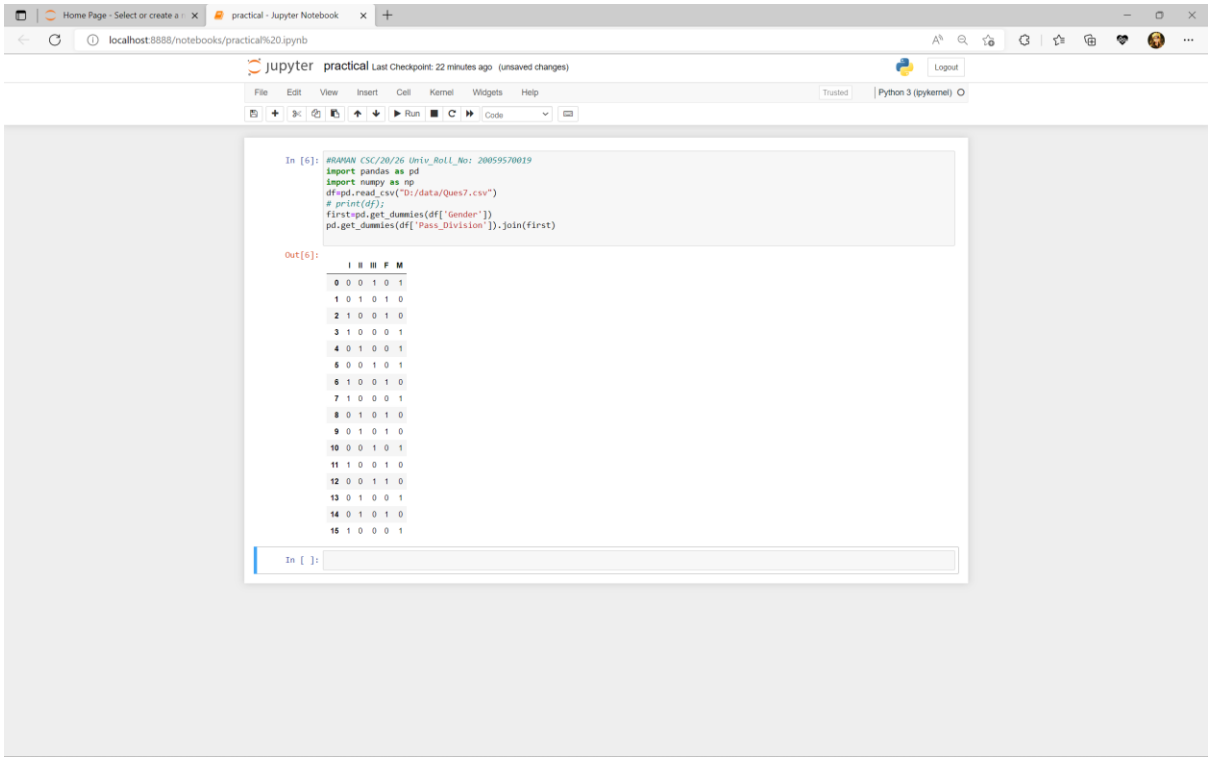
Code a):

#RAMAN CSC/20/26 Univ_Roll_No: 20059570019

import pandas as pd

```
import numpy as np

df=pd.read_csv("D:/data/Ques7.csv")

# print(df);

first=pd.get_dummies(df['Gender'])

pd.get_dummies(df['Pass_Division']).join(first)
```

OUTPUT :-



Code b):

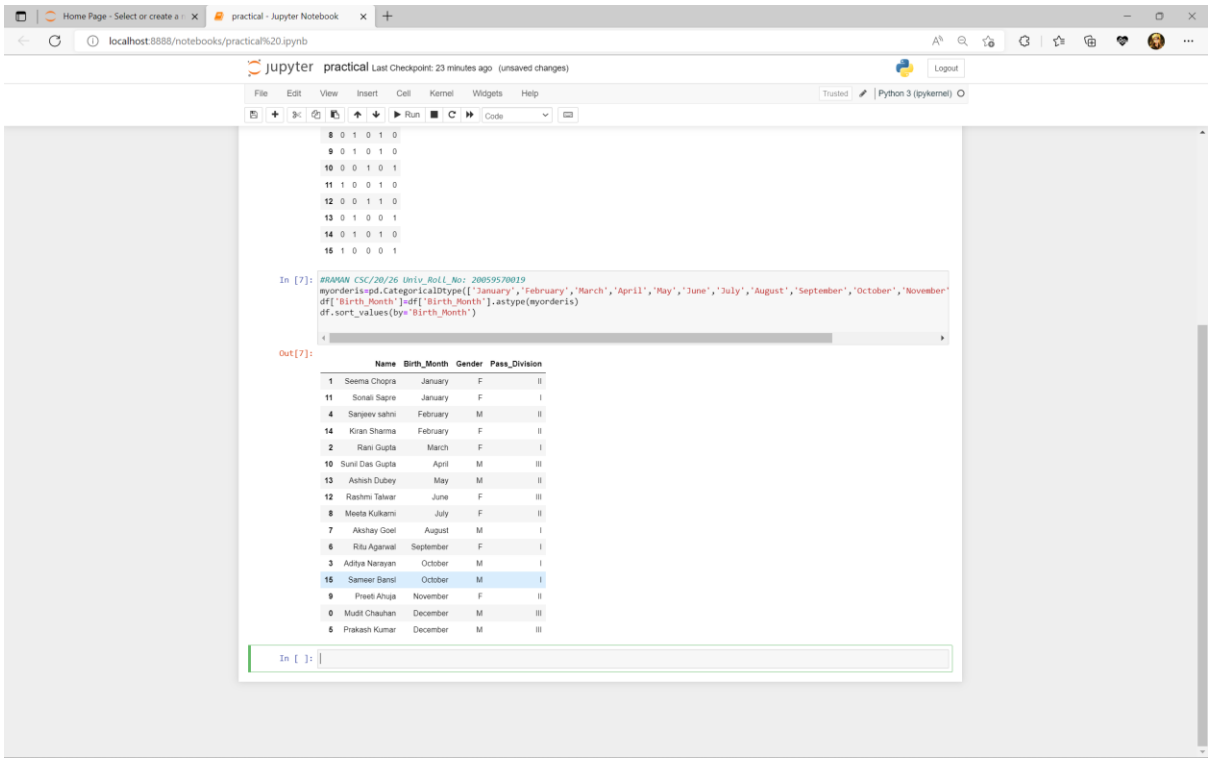
#RAMAN CSC/20/26 Univ_Roll_No: 20059570019

myorderis=pd.CategoricalDtype(['January','February','March','April','May','June','July','August','September','October','November', 'December'],ordered=True)

df['Birth_Month']=df['Birth_Month'].astype(myorderis)

df.sort_values(by='Birth_Month')

OUTPUT :-



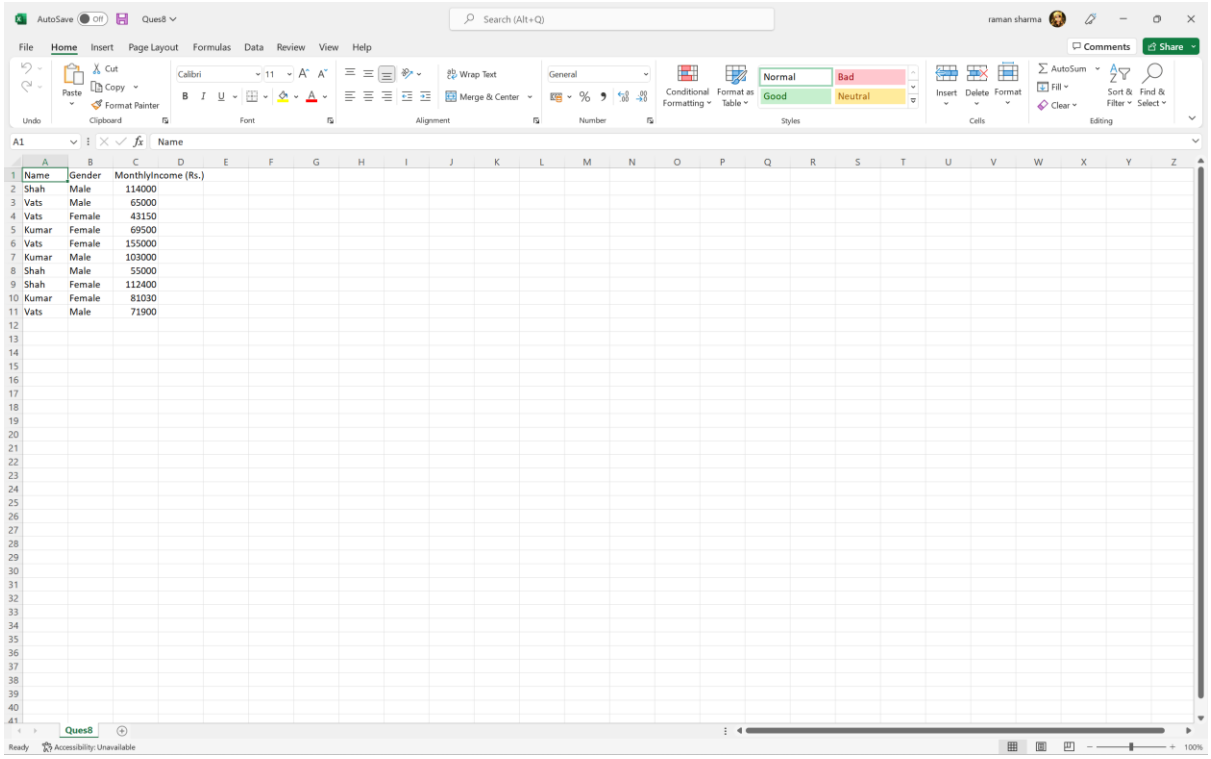
Ques 8.

Consider the following data frame containing a family name, gender of the family member and her/his monthly income in each record.

Name	Gender	MonthlyIncome (Rs.)
Shah	Male	114000.00
Vats	Male	65000.00
Vats	Female	43150.00
Kumar	Female	69500.00
Vats	Female	155000.00
Kumar	Male	103000.00
Shah	Male	55000.00
Shah	Female	112400.00
Kumar	Female	81030.00
Vats	Male	71900.00

Write a program in Python using Pandas to perform the following:
a. Calculate and display familywise gross monthly income.
b. Calculate and display the member with the highest monthly income in a family.
c. Calculate and display monthly income of all members with income greater than Rs. 60000.00.
d. Calculate and display the average monthly income of the female members in the Shah family.

Note:- I have created a CSV File using the Provided Dataset and load the Data into a Dataframe from the Created CSV File.



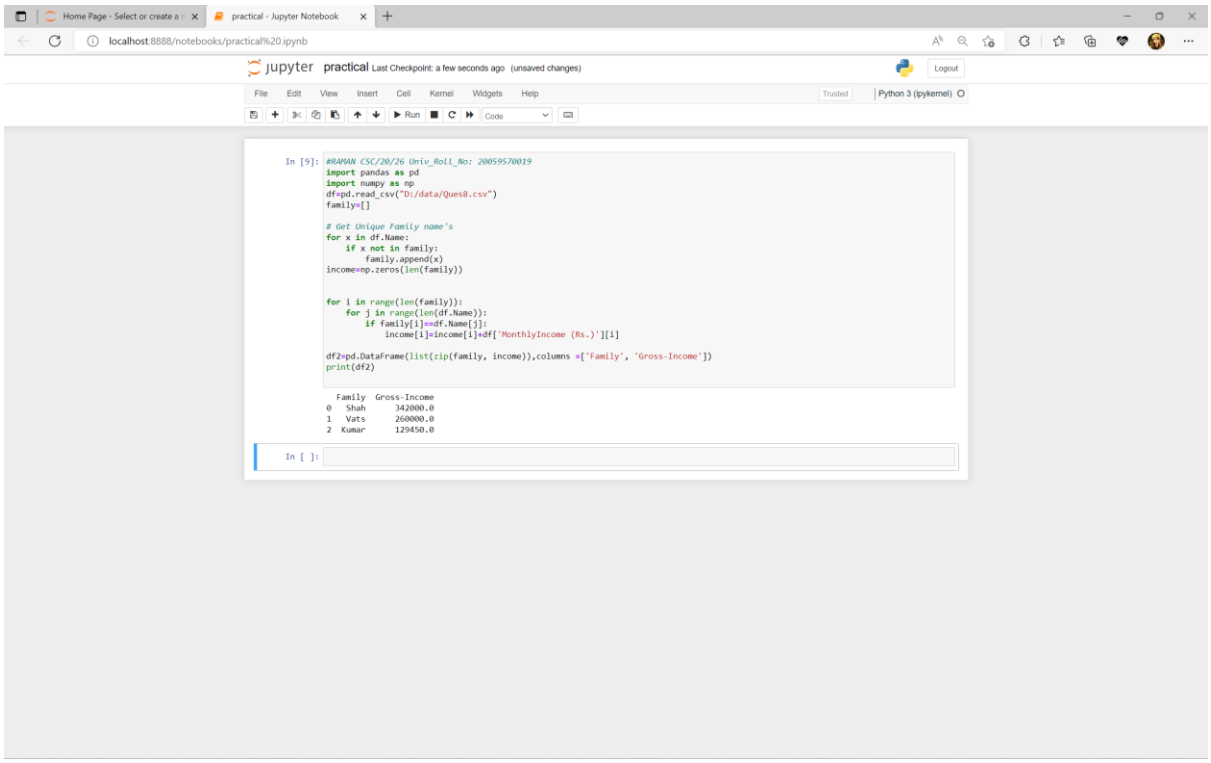
Name	Gender	MonthlyIncome (Rs.)
Shah	Male	114000
Vats	Male	65000
Vats	Female	43150
Kumar	Female	69500
Vats	Female	155000
Kumar	Male	103000
Shah	Male	55000
Shah	Female	112400
Kumar	Female	81030
Vats	Male	71900

Code a):

```
#RAMAN CSC/20/26 Univ_Roll_No: 20059570019

import pandas as pd

import numpy as np

df=pd.read_csv("D:/data/Ques8.csv")

family=[]


# Get Unique Family name's

for x in df.Name:

    if x not in family:

        family.append(x)

income=np.zeros(len(family))



for i in range(len(family)):

    for j in range(len(df.Name)):

        if family[i]==df.Name[j]:

            income[i]=income[i]+df['MonthlyIncome (Rs.)'][i]
```

```
df2=pd.DataFrame(list(zip(family, income)),columns =['Family', 'Gross-Income'])
print(df2)
```

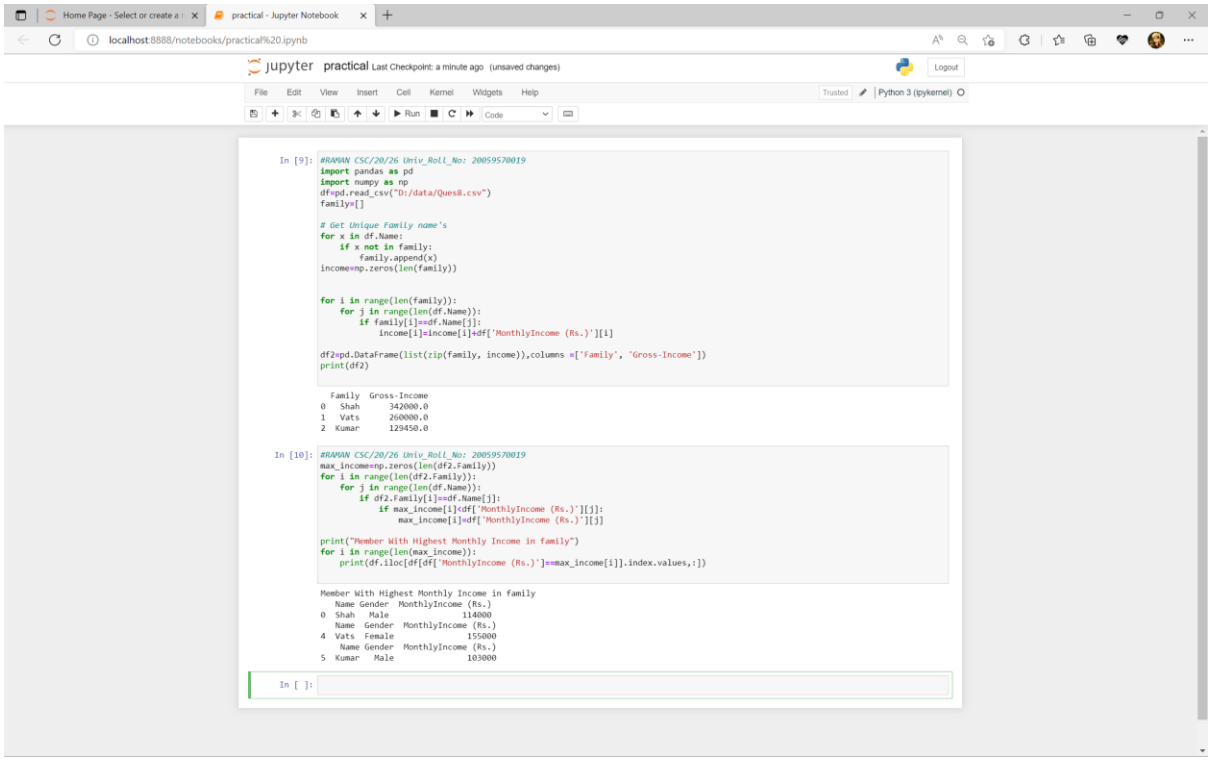
OUTPUT :-



Code b):

```
#RAMAN CSC/20/26 Univ_Roll_No: 20059570019
max_income=np.zeros(len(df2.Family))
for i in range(len(df2.Family)):
    for j in range(len(df.Name)):
        if df2.Family[i]==df.Name[j]:
            if max_income[i]<df['MonthlyIncome (Rs.)'][j]:
                max_income[i]=df['MonthlyIncome (Rs.)'][j]

print("Member With Highest Monthly Income in family")
for i in range(len(max_income)):
    print(df.iloc[df[df['MonthlyIncome (Rs.)']==max_income[i]].index.values,:])
```

OUTPUT :-

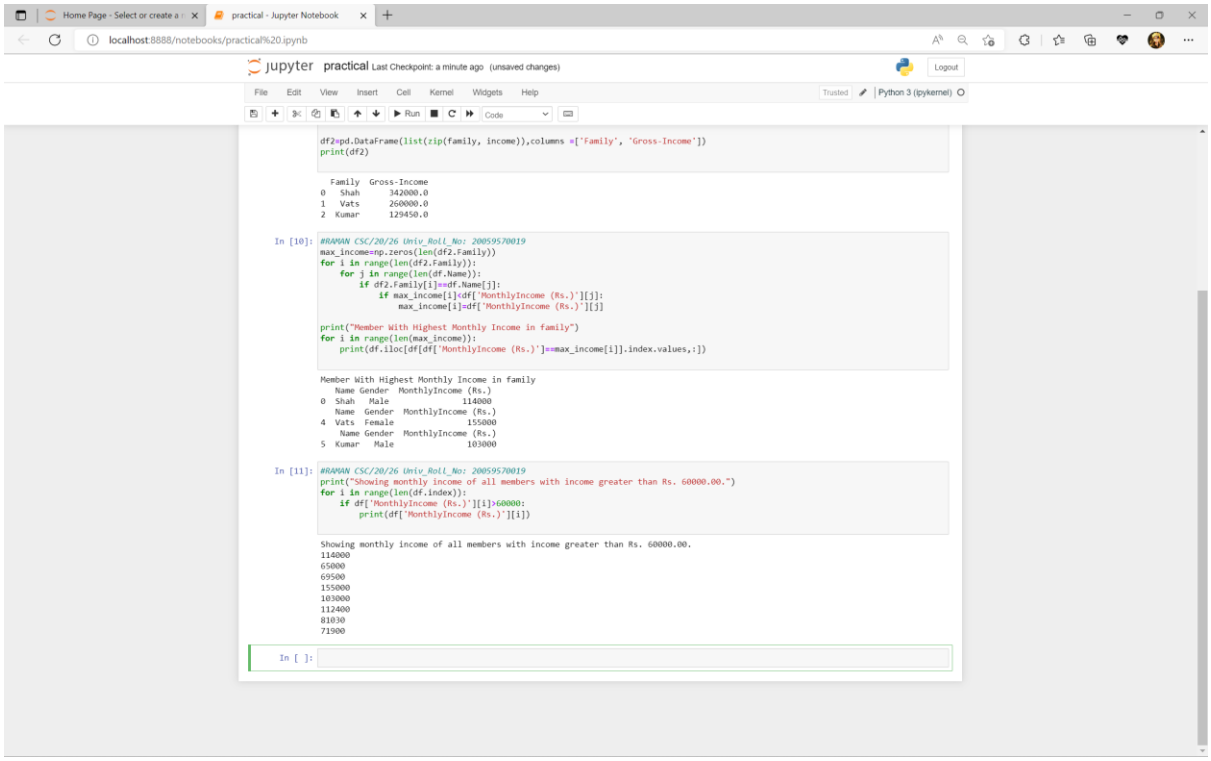


```
In [9]: #RAMAN CSC/20/26 Univ_Roll_No: 20059570019
        import pandas as pd
        import numpy as np
        df=pd.read_csv("D:/data/Ques8.csv")
        family=[]

        # Get Unique Family name's
        for x in df.Name:
            if x not in family:
                family.append(x)
        income=np.zeros(len(family))


        for i in range(len(family)):
            for j in range(len(df.Name)):
                if family[i]==df.Name[j]:
                    income[i]=income[i]+df['MonthlyIncome (Rs.)'][i]

        df2=pd.DataFrame(list(zip(family, income)),columns =['Family', 'Gross-Income'])
        print(df2)

          Family  Gross-Income
        0   Shah      342000.0
        1   Vats      260000.0
        2  Kumar      129450.0

In [10]: #RAMAN CSC/20/26 Univ_Roll_No: 20059570019
         max_income=np.zeros(len(df2.Family))
         for i in range(len(df2.Family)):
             for j in range(len(df.Name)):
                 if df2.Family[i]==df.Name[j]:
                     if max_income[i]<df['MonthlyIncome (Rs.)'][j]:
                         max_income[i]=df['MonthlyIncome (Rs.)'][j]

         print("Member With Highest Monthly Income in family")
         for i in range(len(max_income)):
             print(df.iloc[df[df['MonthlyIncome (Rs.)']==max_income[i]].index.values,:])

         Member With Highest Monthly Income in family
            Name Gender  MonthlyIncome (Rs.)
         0  Shah   Male               114000
            Name  Gender  MonthlyIncome (Rs.)
         4  Vats  Female               155000
             Name Gender  MonthlyIncome (Rs.)
         5  Kumar   Male               103000

In [ ]:
```

Code c):

```
#RAMAN CSC/20/26 Univ_Roll_No: 20059570019
print("Showing monthly income of all members with income greater than Rs. 60000.00.")
for i in range(len(df.index)):
    if df['MonthlyIncome (Rs.)'][i]>60000:
        print(df['MonthlyIncome (Rs.)'][i])
```

OUTPUT :-



Code d):

```
#RAMAN CSC/20/26 Univ_Roll_No: 20059570019

import statistics as stats

female_Shah=[]

for i  in range(len(df.index)):

    if (df.Name[i]=='Shah') and (df.Gender[i]=='Female'):
```

```
    female_Shah.append(df['MonthlyIncome (Rs.)'][i])
print("The average monthly income of the female members in the Shah family :- ",stats.mean(female_Shah))
```

OUTPUT :-