Assignment No. 01

Set A

1) Write a python program to create a dataframe containing columns name, age and percentage. Add 10 rows to the dataframe. View the dataframe.

```
>>> import pandas as pd
>>> df = pd.DataFrame(columns =
['Name','Age','percentage'])
>>> df.loc[1] = ['Maya',18,65.25]
>>> df.loc[2] = ['Kajal',19,70.00]
>>> df.loc[3] = ['Mayur',21,85.00]
>>> df.loc[4] = ['Mayuri',20,75.85]
>>> df.loc[5] = ['Sayara',21,75.00]
>>> df.loc[6] = ['Sania',18,65.45]
>>> df.loc[7] = ['Sapana',10,70.00]
>>> df.loc[8] = ['Balasaheb',11,80.00]
>>> df.loc[9] = ['Sanjay',15,75.45]
>>> df.loc[10] = ['Madhuri',16,55.13]
>>> df
```

output :-

	Name	Age	percentage
1	Maya	18	65.25
2	Kajal	19	70.00
3	Mayur	21	85.00
4	Mayuri	20	75.85
5	Sayara	21	75.00
6	Sania	18	65.45
7	Sapana	10	70.00
8	Balasaheb	11	80.00
9	Sanjay	15	75.45
10	Madhuri	16	55.13

2) Write a Python program to print the shape, number of rows-columns, data types, feature names and the description of the data.

```
>>> df = pd.DataFrame(columns =
['Name','Age','percentage'])
>>> df.loc[1] = ['Maya',18,65.25]
>>> df.loc[2] = ['Kajal',19,70.00]
>>> df.loc[3] = ['Mayuri',20,75.85]
>>> df.loc[4] = ['Sayara',21,75.00]
>>> df.loc[5] = ['Sania',18,65.45]
>>> df
     Name  Age  percentage
1    Maya   18       65.25
2   Kajal   19       70.00
3  Mayuri   20       75.85
4  Sayara   21       75.00
5   Sania   18       65.45

>>> df.info()
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5 entries, 1 to 5
Data columns (total 3 columns):
Name          5 non-null object
Age           5 non-null int64
percentage    5 non-null float64
dtypes: float64(1), int64(1), object(1)
memory usage: 160.0+ bytes
>>> df.shape
(5, 3)
>>> len(df)
5
>>> len(df.columns)
3
```

4) Write a Python program to Add 5 rows with duplicate values and missing values. Add a column 'remarks' with empty values. Display the data.

```
>>> import pandas as pd
>>>
df=pd.DataFrame(columns=['company','model','year'])
>>>df.loc[4]=['Honda','Jazz','None']
>>> df.loc[5]=['toyoto','None',2018]
>>> df.loc[6]=['None','None','None']
>>> df.loc[7]=['Tata','Nexon',2017]
>>> df.loc[8]=['Hyundai','Creata',2015]
>>> #df["remarks"]=None
...
>>> df["remarks"]=None
>>> df
   company   model  year remarks
4    Honda    Jazz  None    None
5   toyoto    None  2018    None
6     None    None  None    None
7     Tata   Nexon  2017    None
8  Hyundai  Creata  2015    None
```

5) Write a Python program to get the number of observations, missing values and duplicate values.

```
>>> import pandas as pd
>>> df=pd.DataFrame(columns=['name','college','year'])
>>> df.loc[0]=['dau','kkw',2022]
>>> df.loc[0]=['gau','kkw',2022]
>>> df.loc[1]=['gau','kkw',2022]
>>> df.loc[2]=['mau','kkw',2022]
>>> df.loc[3]=['mau','kkw',2022]
>>> df
  name college  year
0  gau     kkw  2022
1  gau     kkw  2022
2  mau     kkw  2022
3  mau     kkw  2022
>>> df.isnull()
    name  college   year
0  False    False  False
1  False    False  False
2  False    False  False
3  False    False  False
>>> df.duplicated()
0    False
1     True
2    False
3     True
dtype: bool
```

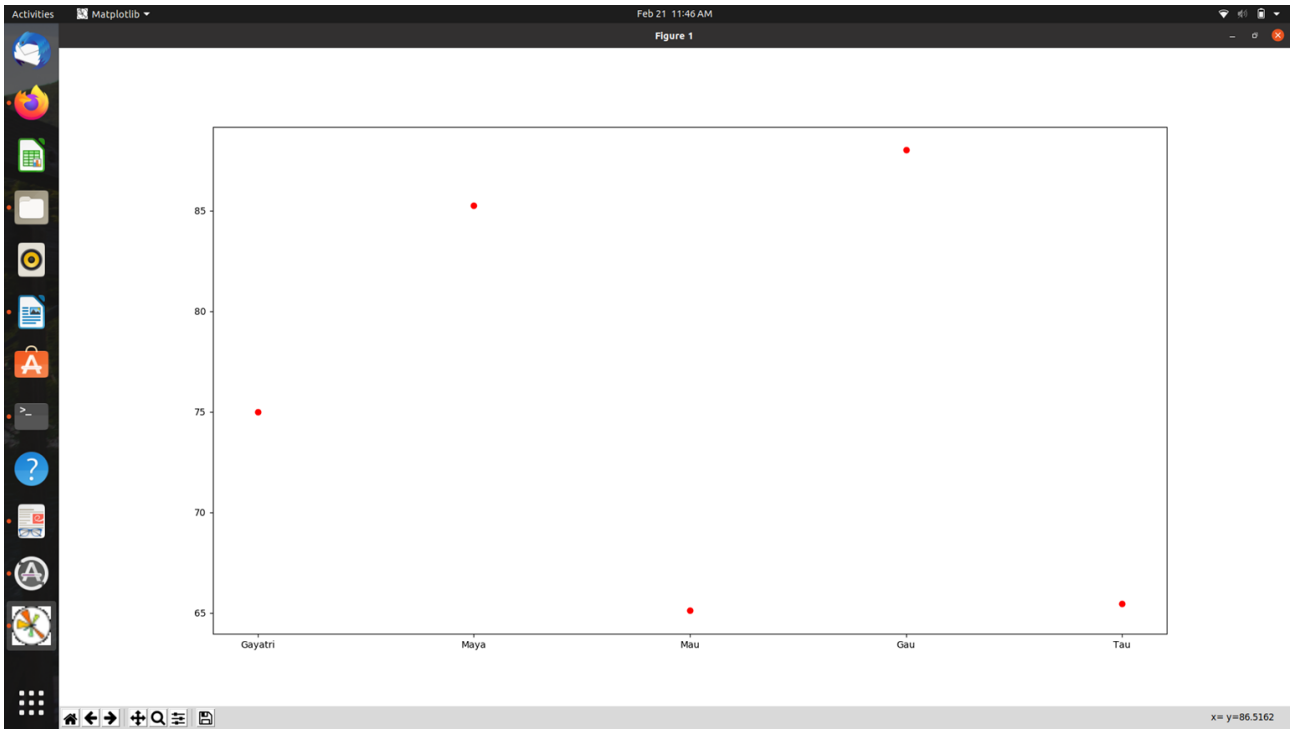
6) Write a Python program to drop 'remarks' column from the dataframe. Also drop all null and empty values. Print the modified data.

```
>>> import pandas as pd
>>>
df=pd.DataFrame(columns=['company','model','year'])
>>> df.loc[1]=['Honda','Jazz','None']
>>> df.loc[2]=['toyoto','None',2018]
>>> df.loc[3]=['None','None','None']
>>> df.loc[4]=['Tata','Nexon',2017]
>>> df.loc[5]=['Hyundai','Creata',2015]
>>> df["remarks"]=None
>>> df
  company   model  year remarks
1   Honda    Jazz  None    None
2  toyoto    None  2018    None
3    None    None  None    None
4    Tata   Nexon  2017    None
5 Hyundai  Creata  2015    None
>>> df.drop(columns='remarks',axis=1,inplace=True)
>>> df
  company   model  year
1   Honda    Jazz  None
2  toyoto    None  2018
3    None    None  None
4    Tata   Nexon  2017
5 Hyundai  Creata  2015
>>> df.dropna()
  company   model  year
1   Honda    Jazz  None
2  toyoto    None  2018
3    None    None  None
4    Tata   Nexon  2017
5 Hyundai  Creata  2015
```

8) Write a Python program to generate a scatter plot of name vs percentage.

```
>>> import matplotlib.pyplot as plt
>>> name = ['Gayatri','Maya','Mau','Gau','Tau']
>>> percentage = [75.00,85.25,65.12,88.00,65.45]
>>> plt.scatter(name,percentage,color="red")
>>> plt.show()
```

===> Output



Set B

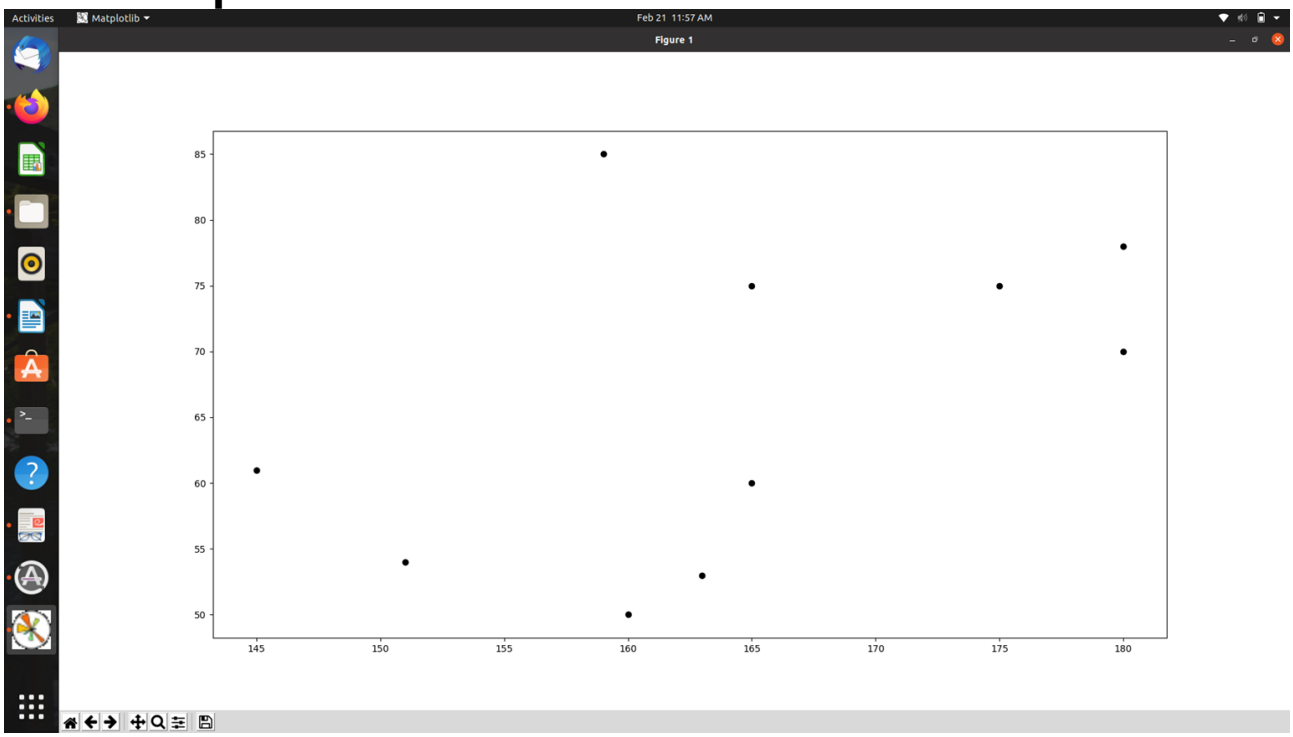
5) Write a Python program to add a column to the dataframe "BMI" which is calculated as : weight/height2

```
>>> import pandas as pd
>>> data = [
...   { 'h': 1.80, 'w': 80 },
...   { 'h': 1.70, 'w': 90 },
...   { 'h': 1.60, 'w': 60 },
... ]
>>> df = pd.DataFrame(data)
>>> df
     h   w
0  1.8  80
1  1.7  90
2  1.6  60
>>> def BMI(data):
...  return data['w'] / data['h']**2
...
>>> df['bmi'] = df.apply(BMI, axis=1)
>>> df
     h   w        bmi
0  1.8  80  24.691358
1  1.7  90  31.141869
2  1.6  60  23.437500
```

7) Write a Python program to generate a scatter plot of height vs weight.

```
>>> import matplotlib.pyplot as plt
>>> height = [160,165,180,175,159,165,145,163,151,180]
>>> weight = [50,60,70,75,85,75,61,53,54,78]
>>> plt.scatter(height,weight,color="black")
>>> plt.show()
```

===> Output



Assignment No.02

Set A

1) Write a Python program to find the maximum and minimum value of a given flattened array.

Expected Output:

Original flattened array:
[[0 1]
[2 3]]
Maximum value of the above flattened array:
3
Minimum value of the above flattened array:
0

```
>>>import numpy as np
>>>a = np.arange(4).reshape((2,2))
>>>print("Original flattened array:")
>>>print(a)
>>>print("Maximum value of the above flattened array:")
>>>print(np.amax(a))
>>>print("Minimum value of the above flattened array:")
>>>print(np.amin(a))
```

==> output

```
Original flattened array:
[[0 1]
 [2 3]]
Maximum value of the above flattened array:
3
Minimum value of the above flattened array:
0
```

2) Write a python program to compute Euclidian Distance between two data points in a dataset. [Hint: Use linalgo.norm function from NumPy]

```
import numpy as np
p1 = np.array((4,7))
p2 = np.array((2,3))
dist = np.linalg.norm(p1 - p2)
print(dist)
```

==>output:4.472135954999579

4) Write a python program to compute sum of Manhattan distance between all pairs of points.

[10 Marks]

```
def distancesum (x, y, n):
    sum = 0

    # for each point, finding distance
    # to rest of the point
    for i in range(n):
        for j in range(i+1,n):
            sum += (abs(x[i] - x[j]) +
                    abs(y[i] - y[j]))

    return sum

# Driven Code
x = [ -1, 1, 3, 2 ]
y = [ 5, 6, 5, 3 ]
n = len(x)
print(distancesum(x, y, n) )
 output:
22
```

Set C

3) Write a NumPy program to compute cross-correlation of two given arrays.

```
>>> import numpy as np
>>> x = np.array([0,1,3])
>>> y = np.array([2,4,5])
>>> print("\nOriginal array1:")

Original array1:
>>> print(x)
[0 1 3]
>>> print("\nOriginal array1:")

Original array1:
>>> print(y)
[2 4 5]
>>> print("\nCross-correlation of the said
arrays:\n",np.cov(x,y))

Cross-correlation of array x and y is:
 [[2.33333333 2.16666667]
 [2.16666667 2.33333333]]
```

6) Create one dataframe with 5 subjects and marks of 10 students for each subject. Find arithmetic mean, geometric mean, and harmonic mean.

```
>>> #Arithmetic mean
>>> import pandas as pd
>>> import numpy as np
>>> data = {'Name':
pd.Series(['Arun','Adi','Aayush','Arjun','Abhi','Lata','shre
e','Maya','Pallavi','Kajal']),'Python':pd.Series([70,80,70,60
,65,55,65,75,85,70]),'Electronics':pd.Series([70,85,75,65,5
5,65,60,70,80,70]),'Statistics':pd.Series([65,40,70,75,65,4
5,85,64,46,70]),'FODS':pd.Series([60,70,80,90,65,85,75,65
,60,75]),'TCS':pd.Series([75,60,70,65,45,85,90,75,60,85])}
>>> df=pd.DataFrame(data)
>>> df
```

	Name	Python	Electronics	Statistics	FODS	TCS
0	Arun	70	70	65	60	75
1	Adi	80	85	40	70	60
2	Aayush	70	75	70	80	70
3	Arjun	60	65	75	90	65
4	Abhi	65	55	65	65	45
5	Lata	55	65	45	85	85
6	shree	65	60	85	75	90
7	Maya	75	70	64	65	75
8	Pallavi	85	80	46	60	60
9	Kajal	70	70	70	75	85

```
>>>#Row-wise geometric mean
>>> from numpy import mean
>>> result = mean(df)
>>> print('Arithmetic mean is:\n',result)
Arithmetic mean is:
 Python        69.5
```

```
Electronics    69.5
Statistics     62.5
FODS           72.5
TCS           71.0
dtype: float64

>>> import scipy
>>> from scipy import stats
>>> df['Geometric
mean']=stats.gmean(df.iloc[:,1:3],axis=1)
>>> df

     Name  Python  Electronics  ...  FODS  TCS
Geometric mean
0     Arun      70           70  ...    60   75       70.000000
1      Adi      80           85  ...    70   60       82.462113
2   Aayush      70           75  ...    80   70       72.456884
3    Arjun      60           65  ...    90   65       62.449980
4     Abhi      65           55  ...    65   45       59.791304
5     Lata      55           65  ...    85   85       59.791304
6    shree      65           60  ...    75   90       62.449980
7     Maya      75           70  ...    65   75       72.456884
8  Pallavi      85           80  ...    60   60       82.462113
9    Kajal      70           70  ...    75   85       70.000000

[10 rows x 7 columns]

>>> #Column-wise geometric mean
>>> Python = scipy.stats.gmean(df.loc[:,"Python"])
>>> print("Geometric mean of Python = ",Python)
Geometric mean of Python =  68.97957193219382
>>> Electronics =
scipy.stats.gmean(df.loc[:,"Electronics"])
>>> print("Geometric mean of Electronics =
",Electronics)
```

```
Geometric mean of Electronics =  68.97957193219382
>>> Statistics = scipy.stats.gmean(df.loc[:,"Statistics"])
>>> print("Geometric mean of Statistics = ",Statistics)
Geometric mean of Statistics =  60.879535378491134
>>> FODS = scipy.stats.gmean(df.loc[:,"FODS"])
>>> print("Geometric mean of FODS = ",FODS)
Geometric mean of FODS =  71.84724510540313
>>> TCS = scipy.stats.gmean(df.loc[:,"TCS"])
>>> print("Geometric mean of TCS = ",TCS)
Geometric mean of TCS =  69.67435065197321

>>>#Row-wise Harmonic mean
>>> scipy.stats.hmean(df.iloc[:,1:3],axis=1)
array([70 . , 82.42424242, 72.4137931 , 62.4, 59.58333333
59.58333333, 62.4, 72.4137931 , 82.42424242, 70. ])

>>># column-wise Harmonic mean
>>> Python = scipy.stats.hmean(df.loc[:,"Python"])
>>> print("Harminic mean of Python = ",Python)
Harminic mean of Python =  68.4589691729467
>>> Electronics =
scipy.stats.hmean(df.loc[:,"Electronics"])
>>> print("Harmonic mean of Electronics =
",Electronics)
Harmonic mean of Electronics =  68.4589691729467
>>> Statistics = scipy.stats.hmean(df.loc[:,"Statistics"])
>>> print("Harmonic mean of Statistics = ",Statistics)
Harmonic mean of Statistics =  59.162827806971556
>>> FODS = scipy.stats.hmean(df.loc[:,"FODS"])
>>> print("Harmonic mean of FODS = ",FODS)
Harmonic mean of FODS =  71.20946902326722
>>> TCS = scipy.stats.hmean(df.loc[:,"TCS"])
>>> print("Harmonic mean of TCS = ",TCS)
Harmonic mean of TCS =  68.24397847248771
```

Assignment No.03

Set A

Dataset Name: Data.CSV (save following data in Excel and save it with .CSV extension)

Country, Age, Salary, Purchased

France, 44, 72000, No

Spain, 27, 48000, Yes

Germany, 30, 54000, No

Spain, 38, 61000, No

Germany, 40, , Yes

France, 35, 58000, Yes

Spain, 52000, No

France, 48, 79000, Yes

Germany, 50, 83000, No

France, 37, 67000, Yes

Import Dataset and do the followings:

a) Describing the dataset

b) Shape of the dataset

c) Display first 3 rows from dataset

```
>>> import pandas as pd
>>> import numpy as np
>>>
md={'Country':pd.Series(['France','spain','Germany','spa
```

```
in','Germany','France','spain','France','Germany','France'
]),'Age':pd.Series([44,27,30,38,40,35,_,48,50,37]),'Salary':
pd.Series([72000,48000,54000,61000,_,58000,52000,7900
0,83000,67000]),'Purchased':pd.Series(['No','Yes','No','N
o','yes','Yes','No','Yes','No','Yes'])}
>>> df.describe()
       Country  ... Purchased
count       10  ...        10
unique       3  ...         3
top     France  ...        No
freq         4  ...         5

[4 rows x 4 columns]
>>> df.shape
(10, 4)
>>> df.head(3)
   Country Age Salary Purchased
0   France  44  72000        No
1    spain  27  48000       Yes
2  Germany  30  54000        No
```

Assignment No.04

Set A

5) Write a Python program to create a Pie plot to get the frequency of the three species of the Iris data.python

```
>>> import matplotlib.pyplot as plt
>>> IrisData = [33,40,27]
>>> Data = ['A','B','C']
>>> exp = [0,0,0]
>>> plt.pie(IrisData, labels=Data, explode = exp, autopct = '%0.2f%%')
>>> plt.show( )
```

output :-