

Numpy

1. Perform addition of two numpy array

```
import numpy as np
study_duration1=np.array([4,5,2,3,5,4,6])
study_duration2=np.array([1,2,5,3,2,2,1])
total_duration1=study_duration1+study_duration2
print("total study duration using + operator", total_duration1)
total_duration2=np.add(study_duration1,study_duration2)
print("total study duration using add() function", total_duration2)
```

In the similar manner execute: subtract(), multiply(), divide ()

2. Finding exponential

```
import numpy as np
number_array=np.array([4,5,6])
number_exponent1=number_array **2
print("exponent using ** operator", number_exponent1)
number_exponent2=np.power(number_array,2)
print("exponent using power() function", number_exponent2)
```

3. Finding mean

```
import numpy as np
score_english=[92,98,85,54,76,87,95,87,77,58]
mean_english=np.mean(score_english)
print(mean_english)
```

4. Finding median

```
import numpy as np
score_english=[92,98,85,54,76,87,95,87,77,58,79]
mean_english=np.median(score_english)
print(mean_english)
////////////////////////////////////
import numpy as np
score_english=np.array([92,98,85,54,76,87,95,87,77,58,79])
mean_english=np.median(score_english)
print(mean_english)
```

In the same manner execute min(), max()

5. Reshape a numpy array

```
import numpy as np
number_array=np.array([4,5,2,3,5,6])
reshaped_number_array=np.reshape(number_array,(2,3))
print("\n actual array \n", number_array)
print("\n reshaped array \n", reshaped_number_array)
```

6. Save numpy array in different format and load it

```
import numpy as np
path="C:/Users/UEM/Desktop/ML_LAB/"
arr=np.array([1,2,3,4,5,6])
np.savetxt(path+"data.txt",arr)

arr1=np.loadtxt(path+"data.txt")
print(arr1)
```

Pandas

1. Creating a panda Series

```
import pandas as pd
st_dataset={'key_1':1, 'key_2':2}
dtf=pd.Series(st_dataset)
print(dtf)
```

2. Creating a panda DataFrame

```
import pandas as pd
st_dataset={'st_name':['Varun','Aftab','Dipika'],
'email':['varun@gmail.com','aftab@rediff.com','dipika@yahoo.com']}
dtf=pd.DataFrame(st_dataset)
print(dtf)
```

3. Counting the number of rows (length of any key) of a dataframe

```
import pandas as pd
st_dataset={'st_name':['Varun','Aftab','Dipika'],
'email':['varun@gmail.com','aftab@rediff.com','dipika@yahoo.com']}
dtf=pd.DataFrame(st_dataset)
dtf_length=len(dtf["st_name"])
print(dtf)
print(dtf_length)
//////////using loop//////////
import pandas as pd
```

```

st_dataset={'st_name':['Varun','Aftab','Dipika'],
'email':['varun@gmail.com','aftab@rediff.com','dipika@yahoo.com']}
dtf=pd.DataFrame(st_dataset)
dtf_length=0
for i in dtf["st_name"]:
    dtf_length=dtf_length+1
print(dtf)
print(dtf_length)

```

4. Assign own index in a dataframe

import pandas as pd

```

data = {
    'st_name':['Varun','Aftab','Dipika'],
    'email':['varun@gmail.com','aftab@rediff.com','dipika@yahoo.com']
}

```

```
df = pd.DataFrame(data, index = ["student1", "student2", "student3"])
```

```
print(df)
```

5. Reading a csv file using pandas

```

import pandas as pd
path="C:/Users/anayg/Desktop/junk/ml_lab/"
data=pd.read_csv(path+"employees.csv")
print(data)

```

6. Reading a csv file and add one extra column by calculating from one existing column

```

import pandas as pd
path="C:/Users/anayg/Desktop/junk/ml_lab/"
data=pd.read_csv(path+"employees.csv")
data1=data
data1=data1.assign(commission_yearly=data1['salary']*0.2)
print(data)

```

7. Removing the header of a dataframe

```

import pandas as pd
path="C:/Users/anayg/Desktop/junk/ml_lab/"
data=pd.read_csv(path+"employees.csv")
data1=pd.read_csv(path+"employees.csv",header=None)
data1=data1.iloc[1:]
print(data)

```

```
print(data1)
```

8. Removing one column by its name

```
import pandas as pd
path="C:/Users/anayg/Desktop/junk/ml_lab/"
data=pd.read_csv(path+"employees.csv")
data1=data.drop('job_id',axis=1)
print(data1)
```

9. Delete multiple columns

```
import pandas as pd
path="C:/Users/anayg/Desktop/junk/ml_lab/"
data=pd.read_csv(path+"employees.csv")
data1=data
data1.drop(data1.columns[[0,1,3]], axis=1, inplace=True)
//////////using ilocation//////////
import pandas as pd
path="C:/Users/anayg/Desktop/junk/ml_lab/"
data=pd.read_csv(path+"employees.csv")
data.drop(data.iloc[:,0:3], axis=1, inplace=True)
```

MatPlotLib

1. Plot 2d graph using MatPlotLib

```
import matplotlib.pyplot as plt
import numpy as np
xpoints = np.array([0, 6])
ypoints = np.array([0, 250])
plt.plot(xpoints, ypoints)
plt.title("Sample Plots")
plt.xlabel("X Axis")
plt.ylabel("Y Axis")
plt.show()
```

2. Plot 2d graph using MatPlotLib with point and solid line

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
xpoints = np.array([0, 6])
ypoints = np.array([0, 250])
plt.plot(xpoints, ypoints, marker='o')
plt.title("Sample Plots")
plt.xlabel("X Axis")
plt.ylabel("Y Axis")
```

```
plt.show()
```

3. Plot 2d graph using Matplotlib with point only

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
xpoints = np.array([0, 6])
ypoints = np.array([0, 250])
plt.plot(xpoints, ypoints, 'o')
plt.title("Sample Plots")
plt.xlabel("X Axis")
plt.ylabel("Y Axis")
plt.show()
```

4. Plot 2d graph using Matplotlib with point and dotted line

```
import matplotlib.pyplot as plt
import numpy as np
xpoints = np.array([0, 6])
ypoints = np.array([0, 250])
plt.plot(xpoints, ypoints, linestyle='dotted', marker='o')
plt.title("Sample Plots")
plt.xlabel("X Axis")
plt.ylabel("Y Axis")
plt.show()
```

Similarly try with `linestyle='dashed'`

5. Plot 2d graph using Matplotlib with point, dotted line and grid

```
import matplotlib.pyplot as plt
import numpy as np
xpoints = np.array([0, 6])
ypoints = np.array([0, 250])
plt.plot(xpoints, ypoints, linestyle='dotted', marker='o')
plt.title("Sample Plots")
plt.xlabel("X Axis")
plt.ylabel("Y Axis")
plt.grid()
plt.show()
```

6. Plot 2D graph using subplot

```
import matplotlib.pyplot as plt
import numpy as np
x1 = np.array([1, 5, 8, 11])
y1 = np.array([17, 5, 21, 16])
```

```
plt.subplot(1,2,1)
plt.plot(x1, y1)
x2 = np.array([1,6,12,15])
y2 = np.array([17,5,21,16])
plt.subplot(1,2,2)
plt.plot(x2, y2)
plt.title("Sample Plots")
plt.grid()
plt.show()
```

7. Plot scatter plot

```
import matplotlib.pyplot as plt
import numpy as np
x1 = np.array([1,5,8,11])
y1 = np.array([17,5,21,16])
plt.scatter(x1, y1)
plt.title("Sample Plots")
plt.grid()
plt.show()
```

8. Plot bar chart

```
import matplotlib.pyplot as plt
import numpy as np
x1 = np.array([1,5,8,11])
y1 = np.array([17,5,21,16])
plt.bar(x1, y1)
plt.title("Sample Plots")
plt.grid()
plt.show()
```

9. Plot pie chart

```
import matplotlib.pyplot as plt
import numpy as np
x1 = np.array([10,8,6.5,5.2])
data_labels=["Mumbai", "Delhi", "Kolkata", "Chennai"]
plt.pie(x1, labels=data_labels)
plt.title("Sample Pie Plots")
plt.legend(title="Cities with population")
plt.show()
```

10. Plot pie chart

```
import matplotlib.pyplot as plt
import numpy as np
x1 = np.array([11,8,6.5,5.2])
```

```

data_labels=["Mumbai","Delhi", "Kolkata", "Chennai"]
myexplode = [0.2, 0, 0, 0]
plt.pie(x1, labels=data_labels,explode = myexplode, shadow = True)
plt.title("Sample Pie Plots")
plt.legend(title="Cities with population")
plt.show()

```

11. Plot Histogram

```

import matplotlib.pyplot as plt
import numpy as np
x1 = np.array([1,1,2,0,0,0,2,2,2,2,4,4,4,5,5,4,5,8,4,8,8,2,5,
               1,4,5,7,6,3,5,6,7,8,9,2,5,4,8,8,5])
plt.hist(x1)
plt.title("Sample histogram")
plt.show()

```

12. Plot 2D graph based on CSV file

```

import pandas as pd
import matplotlib.pyplot as plt
path="C:/Users/anayg/Desktop/junk/ml_lab/"
data=pd.read_csv(path+"data1.csv")
plt.plot(data)
plt.title("Plots of CSV data")
plt.show()

```

13. Plot 2D graph based on CSV file column vs column

```

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
path="C:/Users/anayg/Desktop/junk/ml_lab/"
data=pd.read_csv(path+"data1.csv")
x=data.drop('y',axis=1)
x1=np.array(x)
print(x1)
y=data.drop('x',axis=1)
y1=np.array(y)
plt.plot(x1,y1)
plt.title("Plots of CSV data")
plt.show()

```