Practical No. 6

Aim: To understand and implement basic Functionalities of Numpy & Mathplotlib libraries and the Pandas library in python.

Practical No. 6

Aim: To understand and implement basic functionalities of NumPy and MathPlotlib Libraries in Python. & Pandas Library in python.

Theory: NumPy is a Fundamental library for numerical computing in Python. It provides support for large multi-dimenssional arrays & matrices along with a collection of mathematical Functions to operate on them efficiently.

Mathplotlib is a plotting library for Python. It is used for visulazing data in the form of graphs & plots. This library allows users to customize visual style & layout, export to many File formats & environment.

Codes:- Numpy Implementation :

```
#Creating & Manipulating Arrays
#Creating an Array
import numpy as np
#create an 1D Array
array_1D = np.Array([1, 2, 3, 4, 5])
print("1D Array :", array_1D)

#Create a 2D Array
array_2D = np.array([[1, 2, 3], [4, 5, 6]])
print(" 2D Array :", array_2D)
```

Pandas is a powerful & open source python Library. The Pandas Library is used for data manipulation & analysis. Pandas consists of data structures & Functions to perform efficient operations on data.

Pandas is well suited for working for tabular data, such as spreadsheets or SQL Tables. The Pandas Library is an essential tool for data analyst, scientist & engineers working with the structured data in python.

code :> Creating a dataframe using Pandas Library

```
import pandas as pd
# Calling Dataframe constructor
df = pd.DataFrame()
print (df)

#lists of strings
lst = ['Geeks', 'For', 'Rushi', 'is', 'Portal', 'to', 'SVPCET']

#Calling DataFrame constructor on list.
df = pd.DataFrame(lst)
print (df)
```

Numpy Codes:

```
#Array Operations
arr1 = np.array([1, 2, 3])
arr2 = np.array([4, 5, 6])
print ("Addition :", arr1 + arr2)

# element-wise Multiplication
print(" Multiplication:", arr1 * arr2)

#Mean & Standard Deviation
print ("Mean :", np.mean(array_1D))
print("Standard deviation :", np.std (array_1D))

# Array Reshaping & Slicing
#Reshaping an array : Reshape 1D array into the
#2x3 Matrix
reshaped_Array = np.reshape(array_1D, (2,3))
print ("Reshaped Array :", reshaped_Array)

#Slicing Arrays : Slicing to get 1st & 2nd elements of
# an Array (1D)
print("Sliced Array :", array_1D[0:2])
```

Conclusion: This practical demonstrated using numpy for array manipulation & Mathplotlib for data visualization, thus covering key concepts, operations & plotting techniques.

Codes : Mathplotlib Implementation :

```
#Plotting Basics :
import mathplotlib.pyplot as plt
# X & Y values for plotting.
x = np.array ([0, 1, 2, 3, 4, 5])
y = np.array ([0, 1, 4, 9, 16, 25])

#plotting the Graph
plt.plot(x, y, label = "y=x^2", color = "blue",
              marker = "o")

#Adding Titles & labels :
plt.title ("Basic line graph plotting :")
plt.title ("X-axis")
plt.title ("Y-axis")
plt.legend ()

#Display the Graph
plt.show()
```

Conclusion: This practical demonstrated using Numpy for array manipulation & Mathplotlib for data visualization, thus covering key concepts, operations and plotting techniques.