**Experiment 3**

**Basic of numpy, pandas and matplotlib**

**NumPy**

* + NumPy is a Python library used for numerical computing.
  + It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays.

**Key Concepts:**

* + Arrays: The core data structure in NumPy is the ndarray, a multi-dimensional array of elements of the same type.
  + Vectorization: NumPy provides vectorized operations, allowing efficient element-wise operations on arrays without the need for explicit loops.
  + Broadcasting: Broadcasting is a powerful mechanism that allows NumPy to work with arrays of different shapes when performing arithmetic operations.

import numpy as np

# Creating an array

arr = np.array([1, 2, 3, 4, 5])

# Performing operations

print(arr \* 2) # Output: [2 4 6 8 10]

**Pandas**

* + Pandas is a Python library for data manipulation and analysis.
  + It provides data structures like DataFrame and Series, which are designed for working with structured or labeled data easily and intuitively.

**Key Concepts:**

* + DataFrame: A DataFrame is a 2-dimensional labeled data structure with columns of potentially different types. It is similar to a spreadsheet or SQL table.
  + Series: A Series is a one-dimensional labeled array capable of holding data of any type (integer, string, float, Python objects, etc.).
  + Indexing and Selecting Data: Pandas provides powerful methods for selecting and manipulating data within DataFrames.

import pandas as pd

# Creating a DataFrame

data = {'Name': ['Alice', 'Bob', 'Charlie'], 'Age': [25, 30, 35]}

df = pd.DataFrame(data)

# Selecting data

print(df['Name']) # Output: 0 Alice \n 1 Bob \n 2 Charlie \n Name: Name, dtype: object

**Matplotlib**

* + Matplotlib is a plotting library for Python.
  + It provides a MATLAB-like interface for creating static, animated, and interactive visualizations in Python.

**Key Concepts:**

* + Figure and Axes: A figure is the window or page where plots appear, while axes are the actual plotting area.
  + Plot Types: Matplotlib supports various types of plots, including line plots, scatter plots, bar plots, histograms, etc.
  + Customization: Matplotlib allows extensive customization of plots, including colors, labels, titles, legends, etc.

import matplotlib.pyplot as plt

# Creating a simple plot

x = [1, 2, 3, 4, 5]

y = [2, 4, 6, 8, 10]

plt.plot(x, y)

plt.xlabel('X-axis')

plt.ylabel('Y-axis')

plt.title('Simple Plot')

plt.show()