Experiment 2.2

Student Name: Rohan Singh Samant	**UID: 21BCS9687**
Branch: CSE	**Section/Group: 646/B**
Semester: 5	**Date of Performance:05/09/23**
Subject Name: AIML	**Subject Code:21CSH-316**

Aim: Implementation of Python basic Libraries such as Math, Numpy and Scipy and other libraries.

Objective: The objective of this experiment is to implement Python basic libraries such as Math, NumPy, and SciPy.

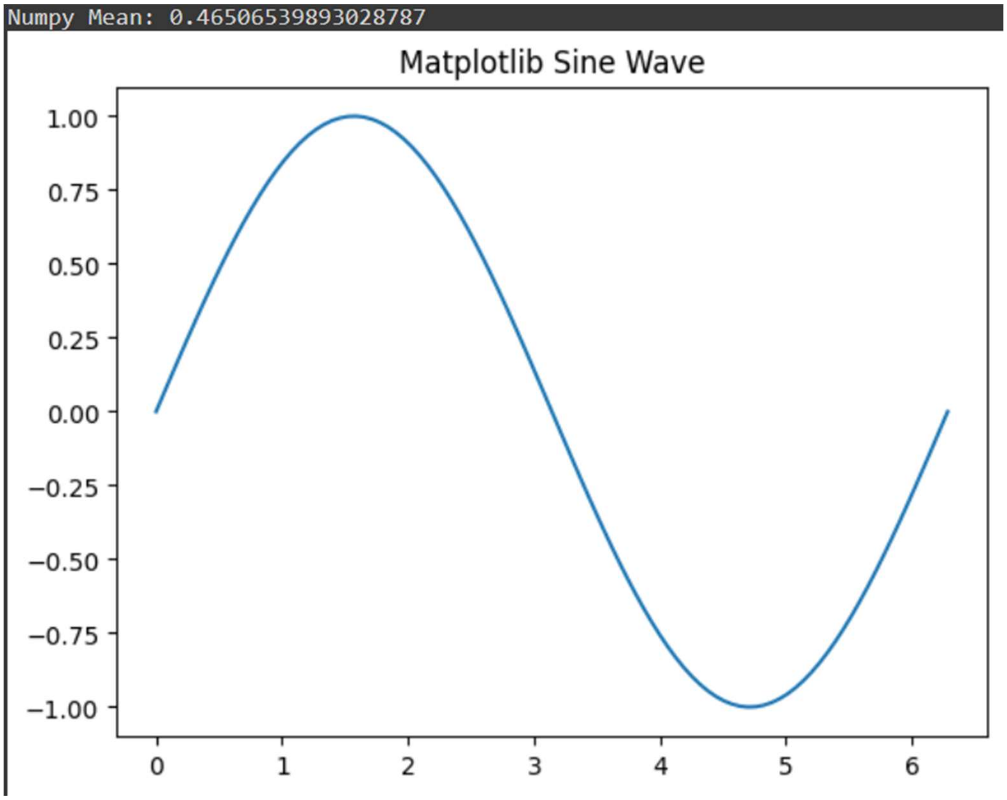
Program and output:

```
# Importing libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from scipy.stats import norm
import tensorflow as tf
from tensorflow import keras
import seaborn as sns

# Numpy example: Creating a random array and calculating its mean
numpy_array = np.random.rand(10)
mean = np.mean(numpy_array)
```

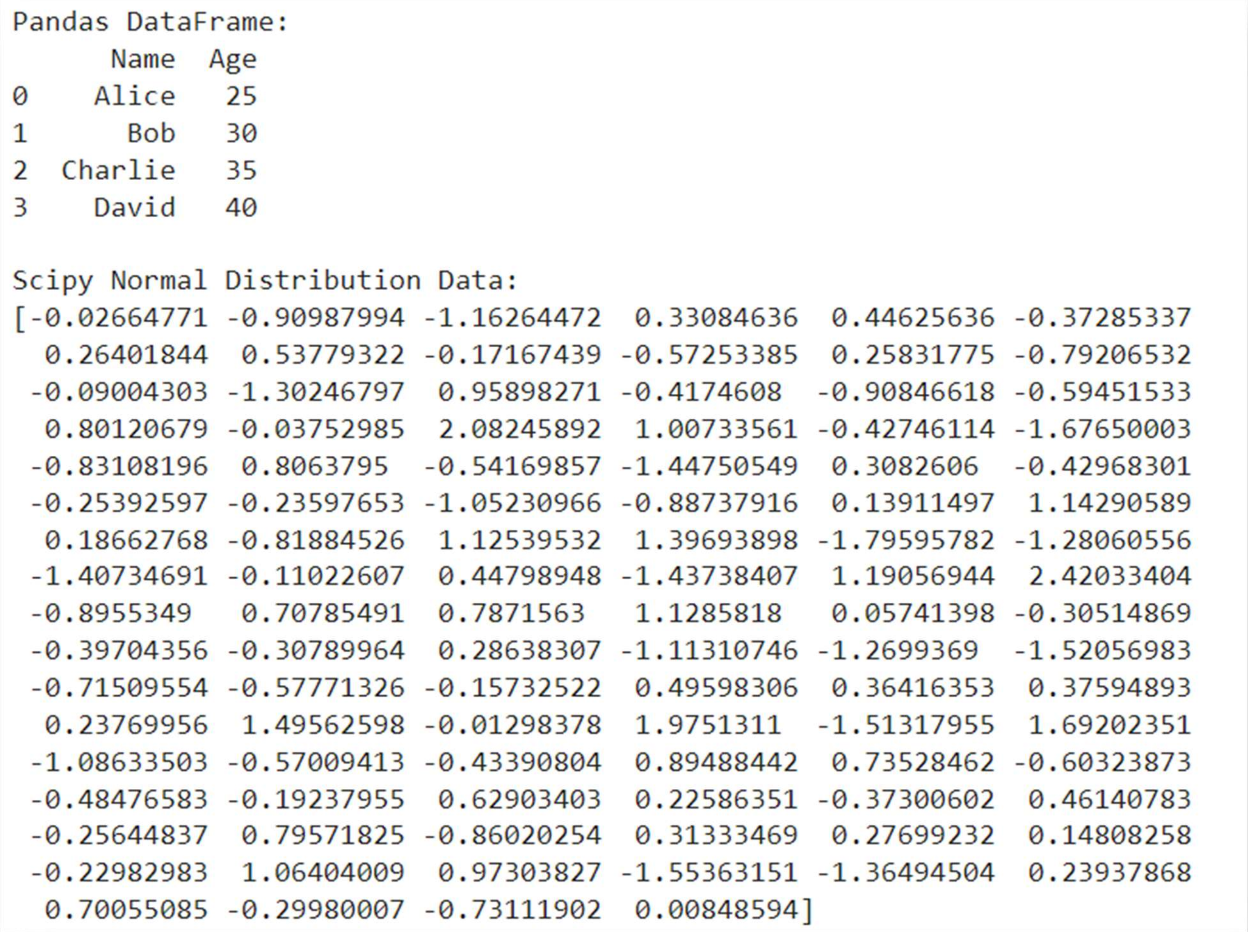
```
print("Numpy Mean:", mean)

# Matplotlib example: Plotting a sine wave
x = np.linspace(0, 2 * np.pi, 100)
y = np.sin(x)
plt.plot(x, y)
plt.title("Matplotlib Sine Wave")
plt.show()

# Pandas example: Creating a DataFrame
data = {'Name': ['Alice', 'Bob', 'Charlie', 'David'],
        'Age': [25, 30, 35, 40]}
df = pd.DataFrame(data)
print("\nPandas DataFrame:")
print(df)

# Scipy example: Generating random data with a normal distribution
data = norm.rvs(loc=0, scale=1, size=100)
print("\nScipy Normal Distribution Data:")
print(data)

# Keras and TensorFlow example: Creating a simple neural network
```

```
model = keras.Sequential([
    keras.layers.Dense(10, activation='relu', input_shape=(5,)),
    keras.layers.Dense(1, activation='linear')
])
print("\nKeras Model Summary:")
model.summary()

# Seaborn example: Creating a pairplot
iris = sns.load_dataset("iris")
sns.pairplot(iris, hue="species")
plt.title("Seaborn Pairplot")
plt.show()
```

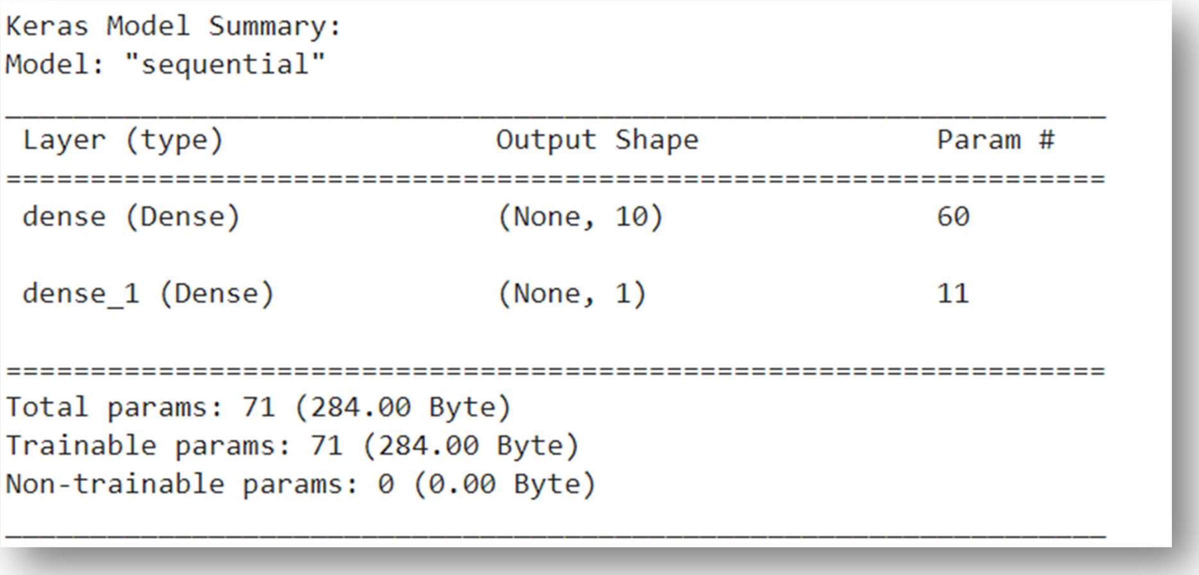
OUTPUT:



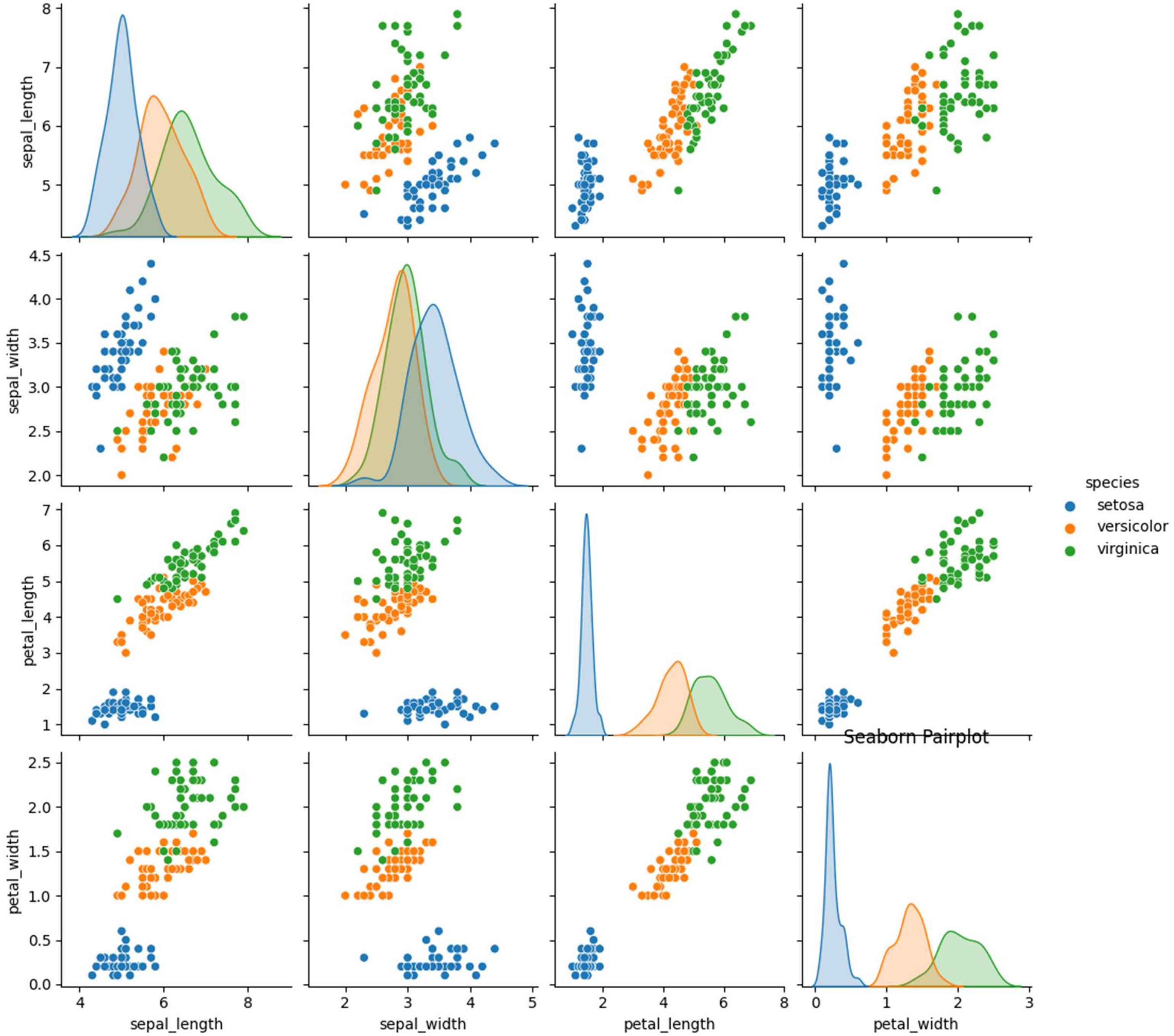
```
Numpy Mean: 0.46506539893028787
```



```
Pandas DataFrame:
      Name  Age
0    Alice   25
1      Bob   30
2  Charlie   35
3    David   40

Scipy Normal Distribution Data:
[-0.02664771 -0.90987994 -1.16264472  0.33084636  0.44625636 -0.37285337
  0.26401844  0.53779322 -0.17167439 -0.57253385  0.25831775 -0.79206532
 -0.09004303 -1.30246797  0.95898271 -0.4174608  -0.90846618 -0.59451533
  0.80120679 -0.03752985  2.08245892  1.00733561 -0.42746114 -1.67650003
 -0.83108196  0.8063795  -0.54169857 -1.44750549  0.3082606  -0.42968301
 -0.25392597 -0.23597653 -1.05230966 -0.88737916  0.13911497  1.14290589
  0.18662768 -0.81884526  1.12539532  1.39693898 -1.79595782 -1.28060556
 -1.40734691 -0.11022607  0.44798948 -1.43738407  1.19056944  2.42033404
 -0.8955349   0.70785491  0.7871563   1.1285818   0.05741398 -0.30514869
 -0.39704356 -0.30789964  0.28638307 -1.11310746 -1.2699369  -1.52056983
 -0.71509554 -0.57771326 -0.15732522  0.49598306  0.36416353  0.37594893
  0.23769956  1.49562598 -0.01298378  1.9751311  -1.51317955  1.69202351
 -1.08633503 -0.57009413 -0.43390804  0.89488442  0.73528462 -0.60323873
 -0.48476583 -0.19237955  0.62903403  0.22586351 -0.37300602  0.46140783
 -0.25644837  0.79571825 -0.86020254  0.31333469  0.27699232  0.14808258
 -0.22982983  1.06404009  0.97303827 -1.55363151 -1.36494504  0.23937868
  0.70055085 -0.29980007 -0.73111902  0.00848594]
```



```
Keras Model Summary:
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 10)                60

 dense_1 (Dense)             (None, 1)                 11

=================================================================
Total params: 71 (284.00 Byte)
Trainable params: 71 (284.00 Byte)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```


sepal_length
sepal_width
petal_length
petal_width
8
7
6
5
4.5
4.0
3.5
3.0
2.5
2.0
7
6
5
4
3
2
1
2.5
2.0
1.5
1.0
0.5
0.0
4
6
8
2
3
4
5
2
4
6
8
0
1
2
3
sepal_length
sepal_width
petal_length
petal_width
species
setosa
versicolor
virginica
Seaborn Pairplot