Dr. Vishwanath Karad
MIT WORLD PEACE UNIVERSITY | PUNE
TECHNOLOGY, RESEARCH, SOCIAL INNOVATION & PARTNERSHIPS

Third Year B. Tech (EL & CE)

Semester: VI **Subject:** Data Science for Engineering

Name: Shreerang Mhatre **Class: TY**

Roll No: 52 **Batch: A2**

Experiment No: 02

Name of the Experiment: **Data Science Fundamentals**

Performed on: 5/2/2024

Submitted on: 5/2/2024

Problem Statement:

Question - 1

Write a python program to output a 3 by 3 array of random numbers following normal distribution

stack these arrays vertically

```
a = np.arrange(10).reshape(2,-1)
b = np.repeat(1, 10).reshape(2,-1)
```

Question - 2

Get the common items between two numpy arrays

```
a =np.array([1,2,3,2,3,4,3,4,5,6])
b =np.repeat([7,2,10,2,7,4,9,4,9,8])
```

Question - 3

Create a series from list, numpy array and dictionary

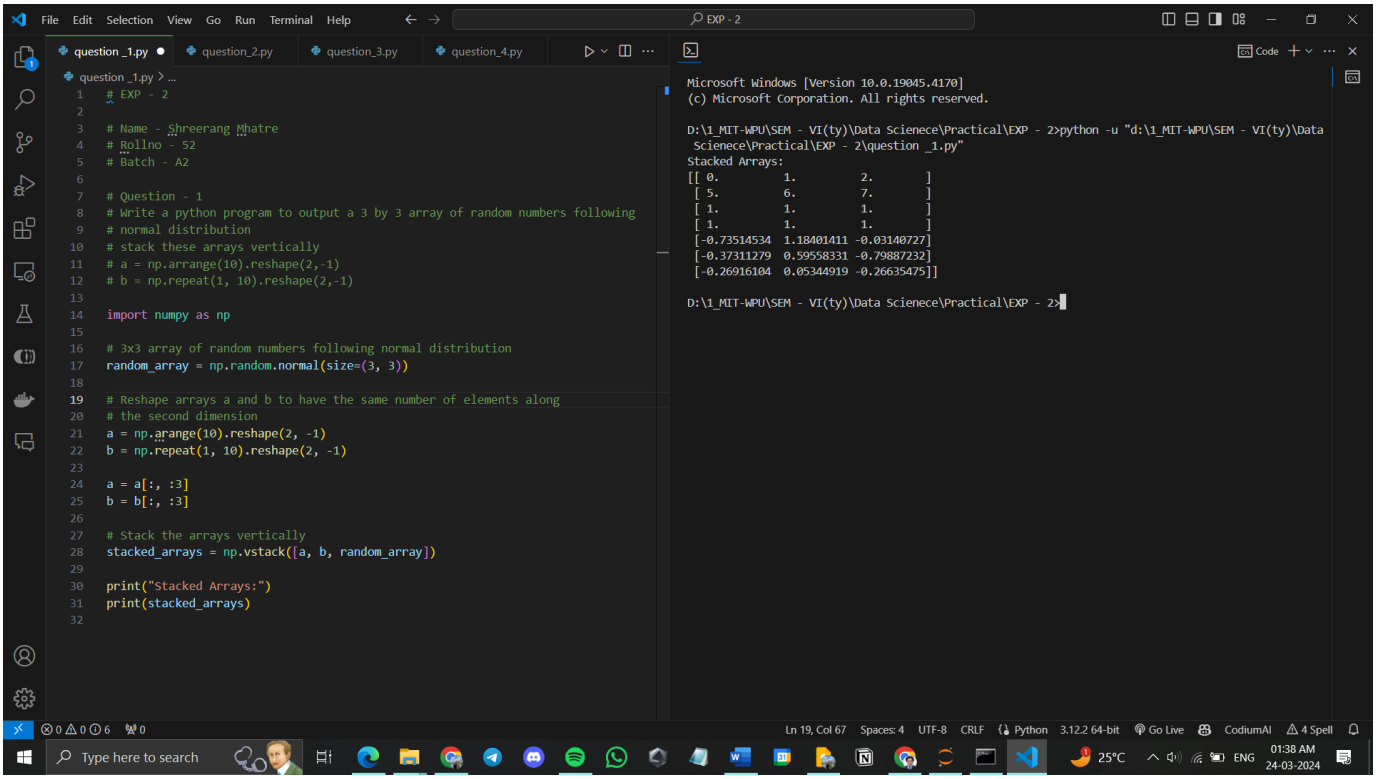
Combine many series to make a data frame

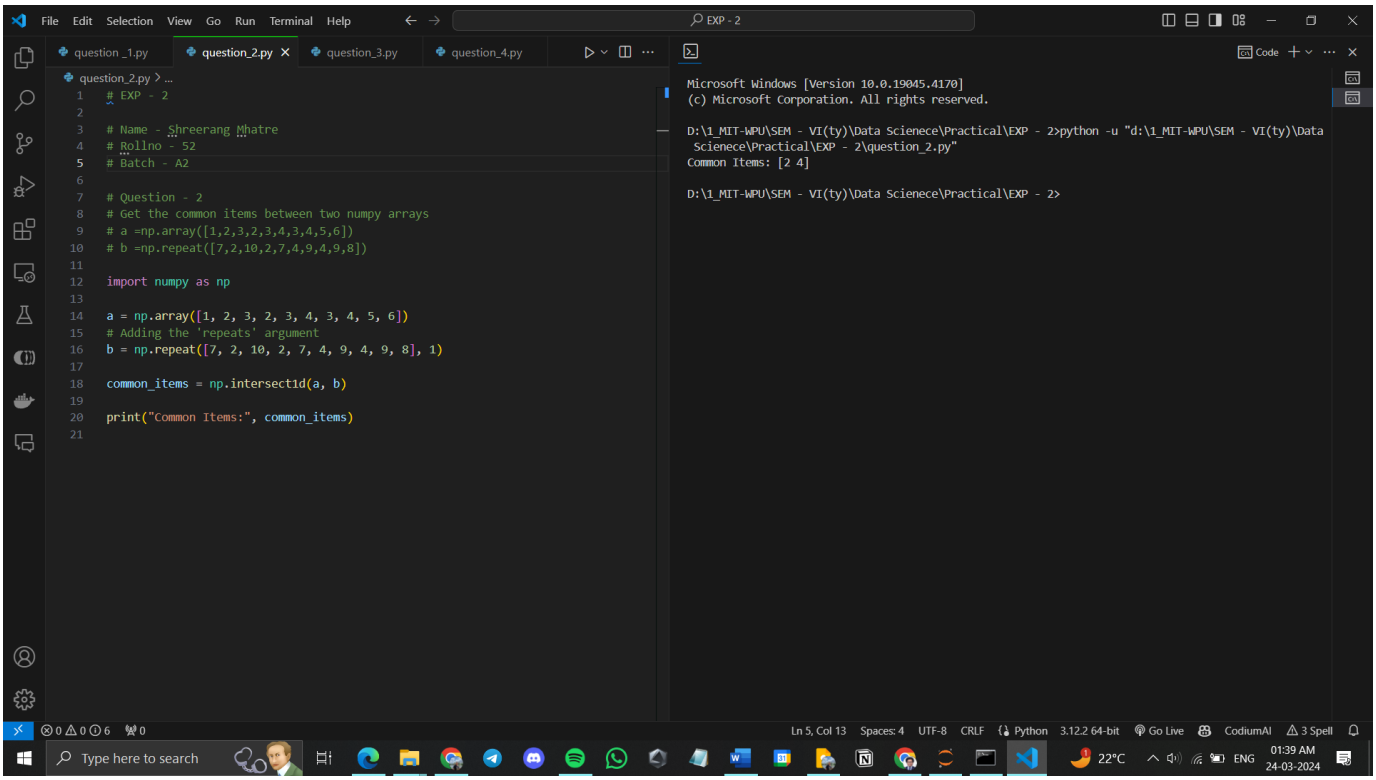
Question - 4

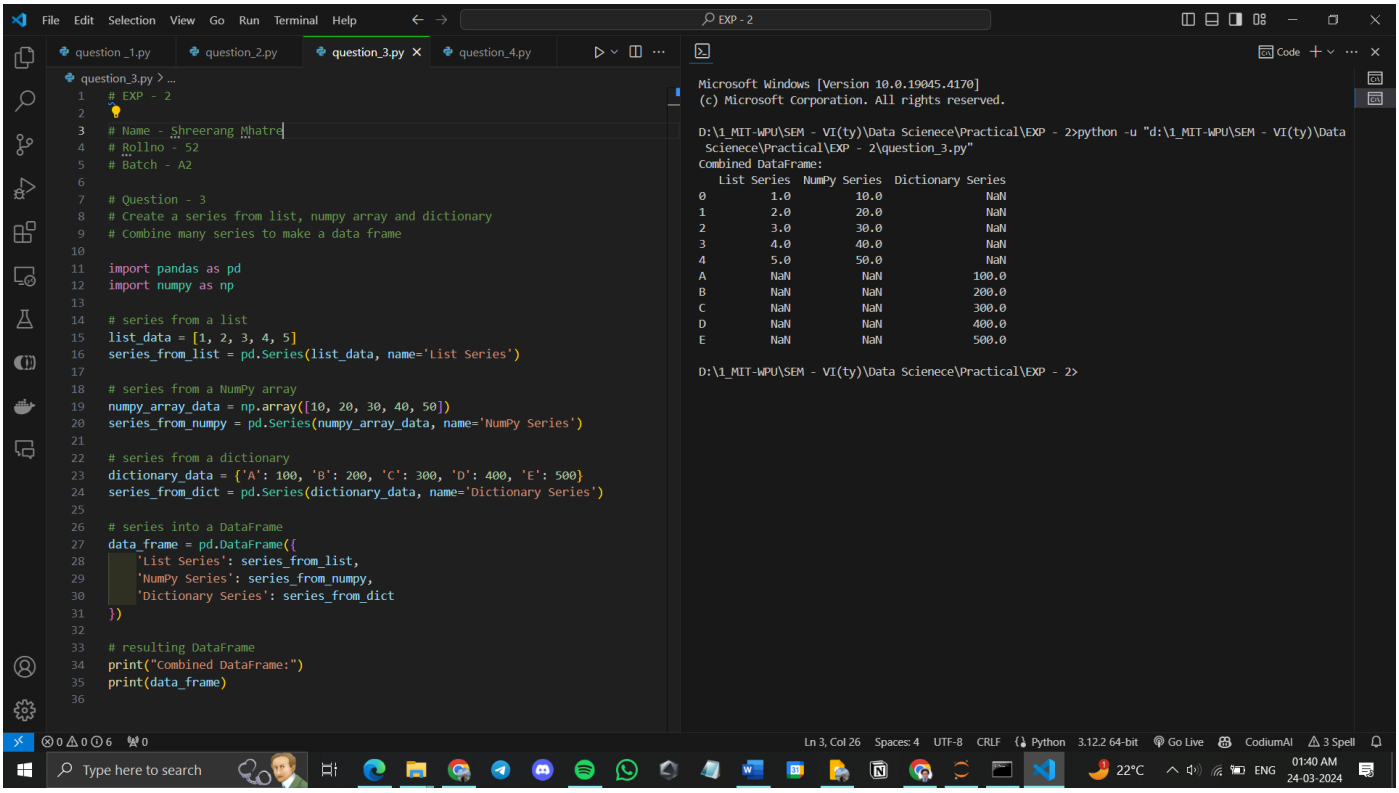
Create a normalized form of iris's sepallength whose values range exactly between 0 and 1 so that minimum has value 0 and maximum has value 1.

Input:

```
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data"
spelllength = np.genfromtxt(url delimiter = '', dtype='float', usecols=[0])
```

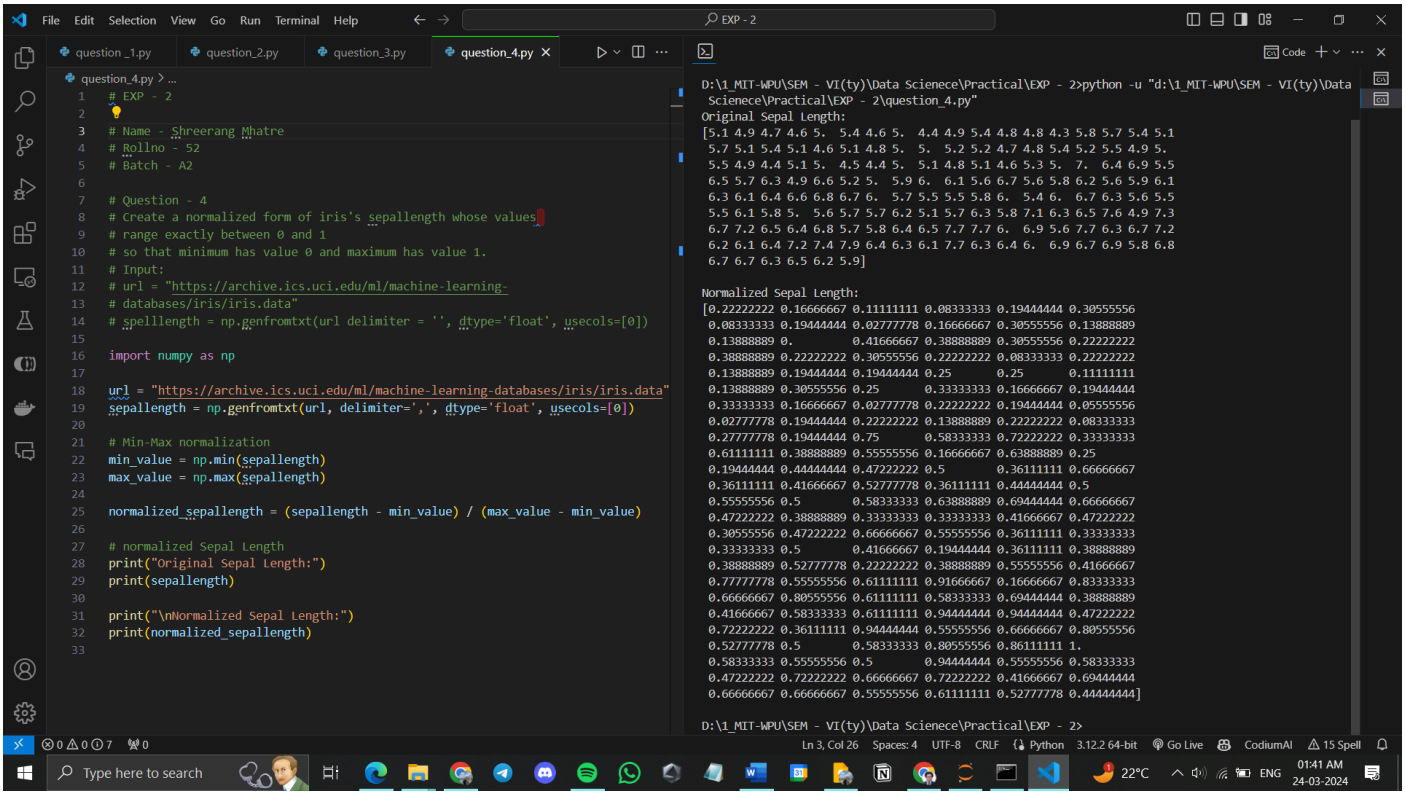

File Edit Selection View Go Run Terminal Help
EXP - 2
question _1.py
question_2.py
question_3.py
question_4.py
# EXP - 2
# Name - Shreerang Mhatre
# Rollno - 52
# Batch - A2
# Question - 1
# Write a python program to output a 3 by 3 array of random numbers following
# normal distribution
# stack these arrays vertically
# a = np.arrange(10).reshape(2,-1)
# b = np.repeat(1, 10).reshape(2,-1)
import numpy as np
# 3x3 array of random numbers following normal distribution
random_array = np.random.normal(size=(3, 3))
# Reshape arrays a and b to have the same number of elements along
# the second dimension
a = np.arange(10).reshape(2, -1)
b = np.repeat(1, 10).reshape(2, -1)
a = a[:, :3]
b = b[:, :3]
# Stack the arrays vertically
stacked_arrays = np.vstack([a, b, random_array])
print("Stacked Arrays:")
print(stacked_arrays)
Microsoft Windows [Version 10.0.19045.4170]
(c) Microsoft Corporation. All rights reserved.
D:\1_MIT-WPU\SEM - VI(ty)\Data Scienece\Practical\EXP - 2>python -u "d:\1_MIT-WPU\SEM - VI(ty)\Data Scienece\Practical\EXP - 2\question _1.py"
Stacked Arrays:
[[0. 1. 2.]
[5. 6. 7.]
[1. 1. 1.]
[1. 1. 1.]
[-0.73514534 1.18401411 -0.03140727]
[-0.37311279 0.59558331 -0.79887232]
[-0.26916104 0.05344919 -0.26635475]]
D:\1_MIT-WPU\SEM - VI(ty)\Data Scienece\Practical\EXP - 2>
Ln 19, Col 67 Spaces: 4 UTF-8 CRLF Python 3.12.2 64-bit Go Live CodiumAI 4 Spell
Type here to search
25°C
ENG
01:38 AM
24-03-2024


File Edit Selection View Go Run Terminal Help
EXP - 2
question _1.py
question_2.py
question_3.py
question_4.py
# EXP - 2
# Name - Shreerang Mhatre
# Rollno - 52
# Batch - A2
# Question - 2
# Get the common items between two numpy arrays
# a =np.array([1,2,3,2,3,4,3,4,5,6])
# b =np.repeat([7,2,10,2,7,4,9,4,9,8])
import numpy as np
a = np.array([1, 2, 3, 2, 3, 4, 3, 4, 5, 6])
# Adding the 'repeats' argument
b = np.repeat([7, 2, 10, 2, 7, 4, 9, 4, 9, 8], 1)
common_items = np.intersect1d(a, b)
print("Common Items:", common_items)
Microsoft Windows [Version 10.0.19045.4170]
(c) Microsoft Corporation. All rights reserved.
D:\1_MIT-WPU\SEM - VI(ty)\Data Scienece\Practical\EXP - 2>python -u "d:\1_MIT-WPU\SEM - VI(ty)\Data Scienece\Practical\EXP - 2\question_2.py"
Common Items: [2 4]
D:\1_MIT-WPU\SEM - VI(ty)\Data Scienece\Practical\EXP - 2>
Ln 5, Col 13 Spaces: 4 UTF-8 CRLF Python 3.12.2 64-bit Go Live CodiumAI 3 Spell
Type here to search
22°C
ENG
01:39 AM
24-03-2024



```python
# EXP - 2

# Name - Shreerang Mhatre
# Rollno - 52
# Batch - A2

# Question - 3
# Create a series from list, numpy array and dictionary
# Combine many series to make a data frame

import pandas as pd
import numpy as np

# series from a list
list_data = [1, 2, 3, 4, 5]
series_from_list = pd.Series(list_data, name='List Series')

# series from a NumPy array
numpy_array_data = np.array([10, 20, 30, 40, 50])
series_from_numpy = pd.Series(numpy_array_data, name='NumPy Series')

# series from a dictionary
dictionary_data = {'A': 100, 'B': 200, 'C': 300, 'D': 400, 'E': 500}
series_from_dict = pd.Series(dictionary_data, name='Dictionary Series')

# series into a DataFrame
data_frame = pd.DataFrame({
    'List Series': series_from_list,
    'NumPy Series': series_from_numpy,
    'Dictionary Series': series_from_dict
})

# resulting DataFrame
print("Combined DataFrame:")
print(data_frame)
```

```
Microsoft Windows [Version 10.0.19045.4170]
(c) Microsoft Corporation. All rights reserved.

D:\1_MIT-WPU\SEM - VI(ty)\Data Scienece\Practical\EXP - 2>python -u "d:\1_MIT-WPU\SEM - VI(ty)\Data
 Scienece\Practical\EXP - 2\question_3.py"
Combined DataFrame:
   List Series  NumPy Series  Dictionary Series
0          1.0          10.0                NaN
1          2.0          20.0                NaN
2          3.0          30.0                NaN
3          4.0          40.0                NaN
4          5.0          50.0                NaN
A          NaN           NaN              100.0
B          NaN           NaN              200.0
C          NaN           NaN              300.0
D          NaN           NaN              400.0
E          NaN           NaN              500.0

D:\1_MIT-WPU\SEM - VI(ty)\Data Scienece\Practical\EXP - 2>
```



```python
# EXP - 2

# Name - Shreerang Mhatre
# Rollno - 52
# Batch - A2

# Question - 4
# Create a normalized form of iris's sepallength whose values
# range exactly between 0 and 1
# so that minimum has value 0 and maximum has value 1.
# Input:
# url = "https://archive.ics.uci.edu/ml/machine-learning-
# databases/iris/iris.data"
# spelllength = np.genfromtxt(url delimiter = '', dtype='float', usecols=[0])

import numpy as np

url = "https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data"
sepallength = np.genfromtxt(url, delimiter=',', dtype='float', usecols=[0])

# Min-Max normalization
min_value = np.min(sepallength)
max_value = np.max(sepallength)

normalized_sepallength = (sepallength - min_value) / (max_value - min_value)

# normalized Sepal Length
print("Original Sepal Length:")
print(sepallength)

print("\nNormalized Sepal Length:")
print(normalized_sepallength)
```

```
D:\1_MIT-WPU\SEM - VI(ty)\Data Scienece\Practical\EXP - 2>python -u "d:\1_MIT-WPU\SEM - VI(ty)\Data
 Scienece\Practical\EXP - 2\question_4.py"
Original Sepal Length:
[5.1 4.9 4.7 4.6 5.  5.4 4.6 5.  4.4 4.9 5.4 4.8 4.8 4.3 5.8 5.7 5.4 5.1
 5.7 5.1 5.4 5.1 4.6 5.1 4.8 5.  5.  5.2 5.2 4.7 4.8 5.4 5.2 5.5 4.9 5.
 5.5 4.9 4.4 5.1 5.  4.5 4.4 5.  5.1 4.8 5.1 4.6 5.3 5.  7.  6.4 6.9 5.5
 6.5 5.7 6.3 4.9 6.6 5.2 5.  5.9 6.  6.1 5.6 6.7 5.6 5.8 6.2 5.6 5.9 6.1
 6.3 6.1 6.4 6.6 6.8 6.7 6.  5.7 5.5 5.5 5.8 6.  5.4 6.  6.7 6.3 5.6 5.5
 5.5 6.1 5.8 5.  5.6 5.7 5.7 6.2 5.1 5.7 6.3 5.8 7.1 6.3 6.5 7.6 4.9 7.3
 6.7 7.2 6.5 6.4 6.8 5.7 5.8 6.4 6.5 7.7 7.7 6.  6.9 5.6 7.7 6.3 6.7 7.2
 6.2 6.1 6.4 7.2 7.4 7.9 6.4 6.3 6.1 7.7 6.3 6.4 6.  6.9 6.7 6.9 5.8 6.8
 6.7 6.7 6.3 6.5 6.2 5.9]

Normalized Sepal Length:
[0.22222222 0.16666667 0.11111111 0.08333333 0.19444444 0.30555556
 0.08333333 0.19444444 0.02777778 0.16666667 0.30555556 0.13888889
 0.13888889 0.         0.41666667 0.38888889 0.30555556 0.22222222
 0.38888889 0.22222222 0.30555556 0.22222222 0.08333333 0.22222222
 0.13888889 0.19444444 0.19444444 0.25       0.25       0.11111111
 0.13888889 0.30555556 0.25       0.33333333 0.16666667 0.19444444
 0.33333333 0.16666667 0.02777778 0.22222222 0.19444444 0.05555556
 0.02777778 0.19444444 0.22222222 0.13888889 0.22222222 0.08333333
 0.27777778 0.19444444 0.75       0.58333333 0.72222222 0.33333333
 0.61111111 0.38888889 0.55555556 0.16666667 0.63888889 0.25
 0.19444444 0.44444444 0.47222222 0.5        0.36111111 0.66666667
 0.36111111 0.41666667 0.52777778 0.36111111 0.44444444 0.5
 0.55555556 0.5        0.58333333 0.63888889 0.69444444 0.66666667
 0.47222222 0.38888889 0.33333333 0.33333333 0.41666667 0.47222222
 0.30555556 0.47222222 0.66666667 0.55555556 0.36111111 0.33333333
 0.33333333 0.5        0.41666667 0.19444444 0.36111111 0.38888889
 0.38888889 0.52777778 0.22222222 0.38888889 0.55555556 0.41666667
 0.77777778 0.55555556 0.61111111 0.91666667 0.16666667 0.83333333
 0.66666667 0.80555556 0.61111111 0.58333333 0.69444444 0.38888889
 0.41666667 0.58333333 0.61111111 0.94444444 0.94444444 0.47222222
 0.72222222 0.36111111 0.94444444 0.55555556 0.66666667 0.80555556
 0.52777778 0.5        0.58333333 0.80555556 0.86111111 1.
 0.58333333 0.55555556 0.5        0.94444444 0.55555556 0.58333333
 0.47222222 0.72222222 0.66666667 0.72222222 0.41666667 0.69444444
 0.66666667 0.66666667 0.55555556 0.61111111 0.52777778 0.44444444]

D:\1_MIT-WPU\SEM - VI(ty)\Data Scienece\Practical\EXP - 2>
```

Exp 2

Shreerang Mhatre_52_A2

* Post Lab Questions

Q1) Discuss details features of the numpy library used in Python.

-> Numpy is a fundamental library for numerical computing in Python. It provides support for large, multi-dimensional arrays & matrices along with a collection of high-level mathematical functions to operate on these arrays efficiently.

Key features of NumPy:

1. Multi-dimensional Arrays
2. Element-wise operations
3. Broadcasting
4. Mathematical Functions
5. Array Indexing and Slicing
6. Integration with other Libraries.

www.mitwpu.edu.in

Q2) Why does normalization is one of the important steps in data cleaning?

→ ① Scaling Data - Normalization scales the data to a standard range, typically between 0 & 1 or -1 & 1.

② Improving Model Performance -
Many machine learning algorithms, such as K-means clustering, support vector machines, and neural networks, perform better when the input data is normalized.

③ Handling Outliners -
Normalization can also help mitigate the impact of outliners in the data.

Q3) Which are different popular languages that can be used in data science instead of Python?

→ ① R → R is a statistical programming language specifically designed for data Analysis

② SQL → structured Query Language.

③ Julia → high-performance language for technical computing, including data science

④ MATLAB → MATLAB is a programming language & environment primarily used in engineering & scientific disciplines.