



An Assignment On “Data Visualization using Python”

Submitted in partial fulfilment of the requirement for the award of

POST GRADUATE DIPLOMA IN MANAGEMENT

From

NARAYANA BUSINESS SCHOOL, AHMEDABAD

Subject: PGDM DATA SCIENCE AND ANALYTICS

**DSA2023 - Data Visualization Techniques using
Python**

Component : CEC

Submitted By

NAME : DEEP PRATEEK

BATCH : PGDMDSA 2023-25

ROLL NO : 004

SECTION : Data Science

DATE OF SUBMISSION: 24-1-2024

Under The Guidance Of

NAME : Miss. Archana Singh and Miss. Sejal Bhavsar

DEPARTMENT : Data Visualization Techniques using Python

Assignment- 1

import numpy as np

Q-1: Create a 1D NumPy array with elements from 1 to 10.

Input:-

```
ar1 = np.arange(1,11)
```

```
print(ar1)
```

Output:-

```
[ 1  2  3  4  5  6  7  8  9 10]
```

Q-2: Create a 2D NumPy array with a shape of (3, 4) containing random integers.

Input:-

```
arr4 = np.random.randint(1,11,size=(3,4))
```

```
print(arr4)
```

Output:-

```
[[6 2 1 6]
 [5 8 8 2]
 [3 3 4 9]]
```

Q-3: Perform element-wise addition, subtraction, multiplication, and division on two arrays.

Input:-

```
arr1 = np.array([[2,3,4],[4,5,6]])
```

```
arr2 = np.array([[4,5,7],[7,8,9]])
```

```
Add = arr1+arr2
```

```
Sub = arr1-arr2
```

```
mul = arr1*arr2
```

```
div = arr1/arr2
```

```
print('Addition',Add)
```

```
print('Subtraction',Sub)
```

```
print('Multiply',mul)
```

```
print('Divide',div)
```

Output:-

```
Addition [[ 6  8 11]
 [11 13 15]]
Subtraction [[-2 -2 -3]
 [-3 -3 -3]]
Multiply [[ 8 15 28]
 [28 40 54]]
Divide [[0.5      0.6      0.57142857]
 [0.57142857 0.625      0.66666667]]
```

Q-4: Extract and print the elements at even indices from a 1D array

import numpy as np

Input:-

```
arr = np.arange(10)
even_elements = arr[::2]
print(even_elements)
```

Output:-

```
[0 2 4 6 8]
```

Q-5: Create a 2-D array

1 3 5

7 9 2

4 6 8

access the second row of the array

access the third column of the array

Input:-

```
array_2d = [
    [1, 3, 5, 7, 9],
    [2, 4, 6, 8]
]
```

```
second_row = array_2d[1]
```

```
third_column = [row[2] for row in array_2d]
```

```
print("Original 2-D array:")
```

```
for row in array_2d:
```

```
    print(row)
```

```
print("\nSecond row:", second_row)
```

```
print("Third column:",third_column)
```

Output:-

```
Original 2-D array:
```

```
[1, 3, 5, 7, 9]
```

```
[2, 4, 6, 8]
```

```
Second row: [2, 4, 6, 8]
```

```
Third column: [5, 6]
```

Q-6: Create a 3D array with shape (2, 3, 4) and access a specific element of the array [1,2,1].

Input:-

```
array_3d = np.random.random((2, 3, 4))
```

```
element_value = array_3d[1, 2, 1]
```

```
print("3D Array:")
```

```
print(array_3d)
```

```
print("\nValue at [1, 2, 1]:",element_value)
```

Output:-

```
3D Array:
```

```
[[[0.7743173  0.34894067 0.98486191 0.84535539]
```

```
  [0.13854772 0.7561444 0.33293945 0.59697983]
```

```
  [0.24171431 0.59486047 0.0376608 0.02183642]]]
```

```
[[[0.36852581 0.33566108 0.43802765 0.51308869]
```

```
  [0.79938992 0.00427 0.21954334 0.36040645]
```

```
  [0.58267188 0.36870673 0.81443473 0.14044989]]]
```

```
Value at [1, 2, 1]: 0.3687067271515929
```

Q-7: Save one of your created arrays to a text file.

Input:-

```
array1 = np.array([1, 2, 3])
array2 = np.array([4, 5, 6])
array3 = np.array([7, 8, 9])
arr = np.stack((array1, array2, array3))
```

Q-8: Load the saved array back into a NumPy array

Input:-

```
np.savetxt('2d_array.txt', arr, delimiter=',', fmt='%d')
arr = np.loadtxt('2d_array.txt', delimiter=',', dtype=int)
print(arr)
```

Output:-

```
[[1 2 3]
 [4 5 6]
 [7 8 9]]
```

Q-9: Create a mark sheet for your class .

Find the percentage of scored by each student

Find the student with highest percentage and hence scored position 1

Input:-

```
import numpy as np
```

```
subjects = ["Python", "Java", "Hadoop", "Data Warehouse", "SQL", "MongoDB"]
```

```
students = ["Amit", "Ayush", "Darpan", "Deep", "Dharm", "Dhir", "Kartik",
"Khushi", "Manan", "Mansi Sahu",
```

```
    "Mansi", "Sapna", "Shobit", "SHRIYANSH", "Smit", "Sonika", "Vijay"]
```

```
marks = [
```

```
[85, 90, 78, 92, 88, 75], # Amit
[92, 88, 80, 85, 90, 78], # Ayush
[78, 82, 70, 88, 75, 80], # Darpan
[90, 85, 88, 98, 99, 85], # Deep
[82, 75, 90, 70, 88, 92], # Dharmi
[88, 78, 85, 82, 90, 70], # Dhir
[75, 80, 82, 85, 78, 88], # Kartik
[80, 92, 75, 90, 85, 82], # Khushi
[85, 85, 90, 75, 92, 78], # Manan
[90, 78, 82, 88, 75, 85], # Mansi Sahu
[78, 80, 75, 82, 88, 90], # Mansi
[92, 85, 88, 78, 82, 75], # Sapna
[88, 82, 78, 85, 90, 80], # Shobit
[85, 90, 75, 78, 88, 92], # SHRIYANSH
[90, 88, 82, 92, 75, 78], # Smit
[78, 85, 80, 88, 82, 90], # Sonika
[92, 90, 85, 78, 80, 75], # Vijay
]
```

```
marks_array = np.array(marks)
```

```
percentage = np.mean(marks_array, axis=1)
```

```
print("\nAll students' percentages:")
```

```
for student, percent in zip(students, percentage):
```

```
    print(f"{student}: {percent}")
```

```
highest_percentage_index = np.argmax(percentage)
```

```
highest_percentage_student = students[highest_percentage_index]
highest_percentage = percentage[highest_percentage_index]

print("Student with the highest percentage:", highest_percentage_student)
print("Highest Percentage:", highest_percentage)
```

Output:-

```
All students' percentages:
Amit: 84.66666666666667
Ayush: 85.5
Darpan: 78.83333333333333
Deep: 90.83333333333333
Dharm: 82.83333333333333
Dhir: 82.16666666666667
Kartik: 81.33333333333333
Khushi: 84.0
Manan: 84.16666666666667
Mansi Sahu: 83.0
Mansi: 82.16666666666667
Sapna: 83.33333333333333
Shobit: 83.83333333333333
SHRIYANSH: 84.66666666666667
Smit: 84.16666666666667
Sonika: 83.83333333333333
Vijay: 83.33333333333333
Student with the highest percentage: Deep
Highest Percentage: 90.83333333333333
```
