Experiment No. 3

Title: Exploratory data analysis using NUMPY

Batch:A2 Roll No:16010421063 Experiment

No.:3 Aim: To perform exploratory data analysis using python NUMPY

Resources needed: Python IDE

Theory:

- Data Analysis is basically where you use statistics and probability to figure out trends in the data set. It helps you to sort out the "real" trends from the statistical noise
- [illegible] ta Analysis (EDA) in Python is the first step in your data analysis process [illegible] John Tukey" in the 1970s.
- [illegible] ploratory data analysis is an approach to analyzing data sets to summarize [illegible] acteristics, often with visual methods.
- [illegible] of exploratory data analysis is to obtain confidence in your data to an [illegible] ou're ready to engage a machine learning algorithm.

I [illegible] ing things in EDA.

1 [illegible] dataset; number of rows/columns, missing data, data types, preview.

2 [illegible] a; handle missing data, invalid data types, incorrect values.

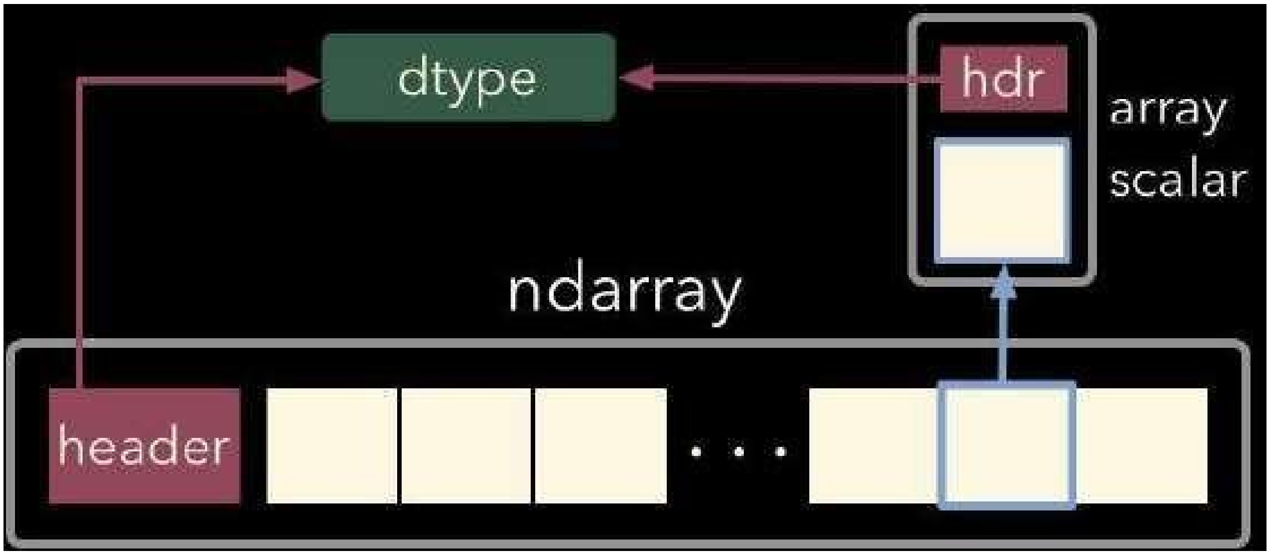
3) Visualize data distributions; bar charts, histograms, box plots.

4) Calculate and visualize correlations (relationships) between variables;

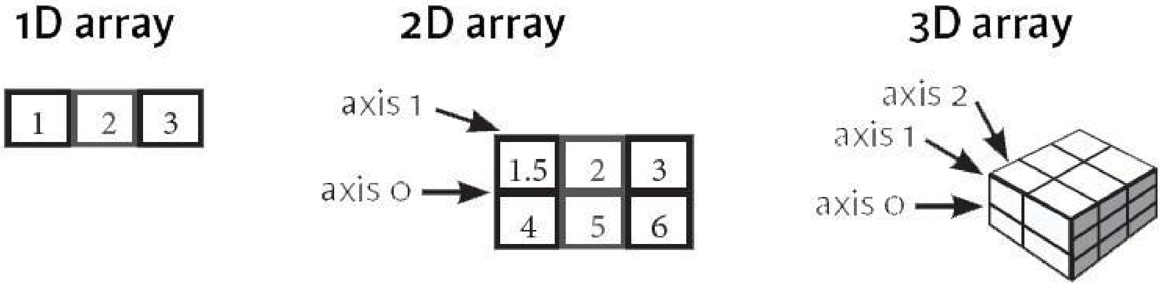
NUMPY(Numeric or Numerical Python):NumPy is a Python library that is the core library for scientific computing in Python.

It contains a collection of tools and techniques that can be used to solve on a computer mathematical models of problems in Science and Engineering.

One of these tools is a high-performance multidimensional array object, ndarray, that is a powerful data structure for efficient computation of arrays and matrices. Memory layout of ndarray is shown in figure below.



Memory layout of ndarrary of python



1D, 2D and 3D arrays in numpy

To work with these arrays, there's a vast amount of high-level mathematical functions operate on these matrices and arrays.

NumPy's main object is the homogeneous multidimensional array. It is a table of elements (usually numbers), all of the same type, indexed by a tuple of positive integers. In NumPy dimensions are called *axes*.

For example, the coordinates of a point in 3D space [1, 2, 1] has one axis. That axis has 3 elements in it, so we say it has a length of 3. In the example pictured below, the array has 2 axes. The first axis has a length of 2, the second axis has a length of 3.

[[1., 0., 0.], [0., 1., 2.]]

NumPy's array class is called ndarray. It is also known by the alias array.

numpy.array is not the same as the Standard Python Library class array.array, which only handles one-dimensional arrays and offers less functionality. ndarray.ndim the number of axes (dimensions) of the array.

The more important attributes of an ndarray object are:

ndarray.ndim the number of axes (dimensions) of the array.

ndarray.shape the dimensions of the array. This is a tuple of integers indicating the size of the array in each dimension. For a matrix with *n* rows and *m* columns, shape will be (n,m). The length of the shape tuple is therefore the number of axes, ndim.

ndarray.size the total number of elements of the array. This is equal to the product of the elements of shape.

ndarray.dtype an object describing the type of the elements in the array. One can create or specify dtype's using standard Python types.

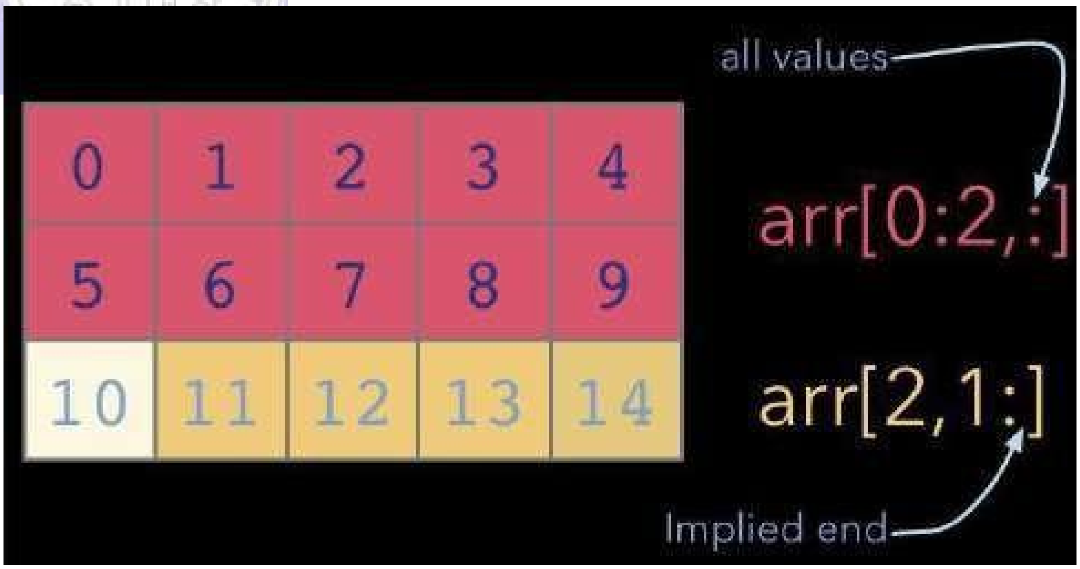
ndarray.itemsize the size in bytes of each element of the array.

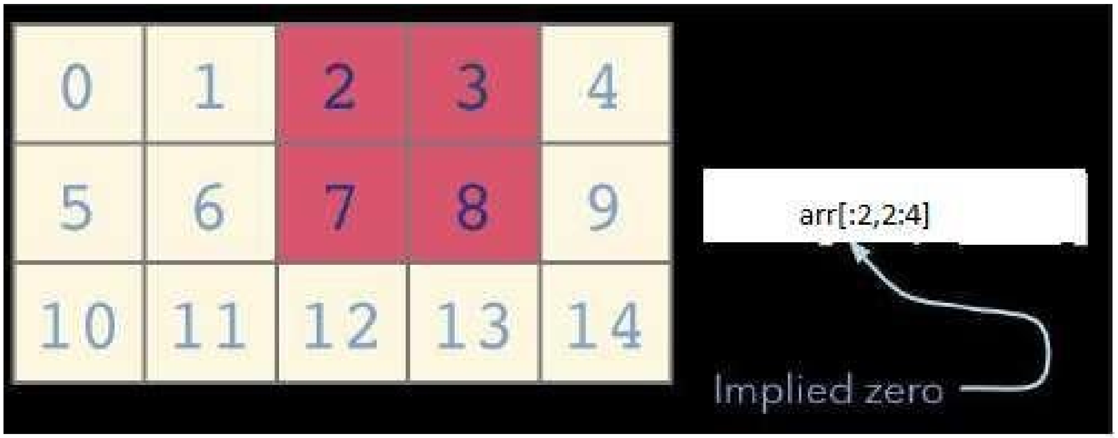
r containing the actual elements of the array. Normally, we won't need to se we will access the elements in an array using indexing facilities.

), numpy.zeros(), numpy.empty() we can create standard arrays of ones, zed numbers respectively.

from list of homogeneous numbers as well.

g in numpy arrays: figure below gives idea about slicing and indexing in





In order to use ndarray and its related attributes and functions, we first have to make sure that numpy is installed. Since numpy is basic library of python it comes along with most of the python IDE. In case it is not installed we can download latest wheel of numpy and install it using pip install.

One it is installed using following statement it can be import and its functionalities can be used.

import numpy as nd

#creating array of zeros

np.zeros(5, float)

similarly we can use following functions to find statistical measures using ndarray.

x.sum(),x.mean(),x.min(0,x.max() etc

one can pass axis=0 or axis=1 to do columnwise and rowwise operations.

reshape() function will resize array as per new dimensions passed as an arguments to it.

vstack() and vstack() for concatenation of two compatible arrays

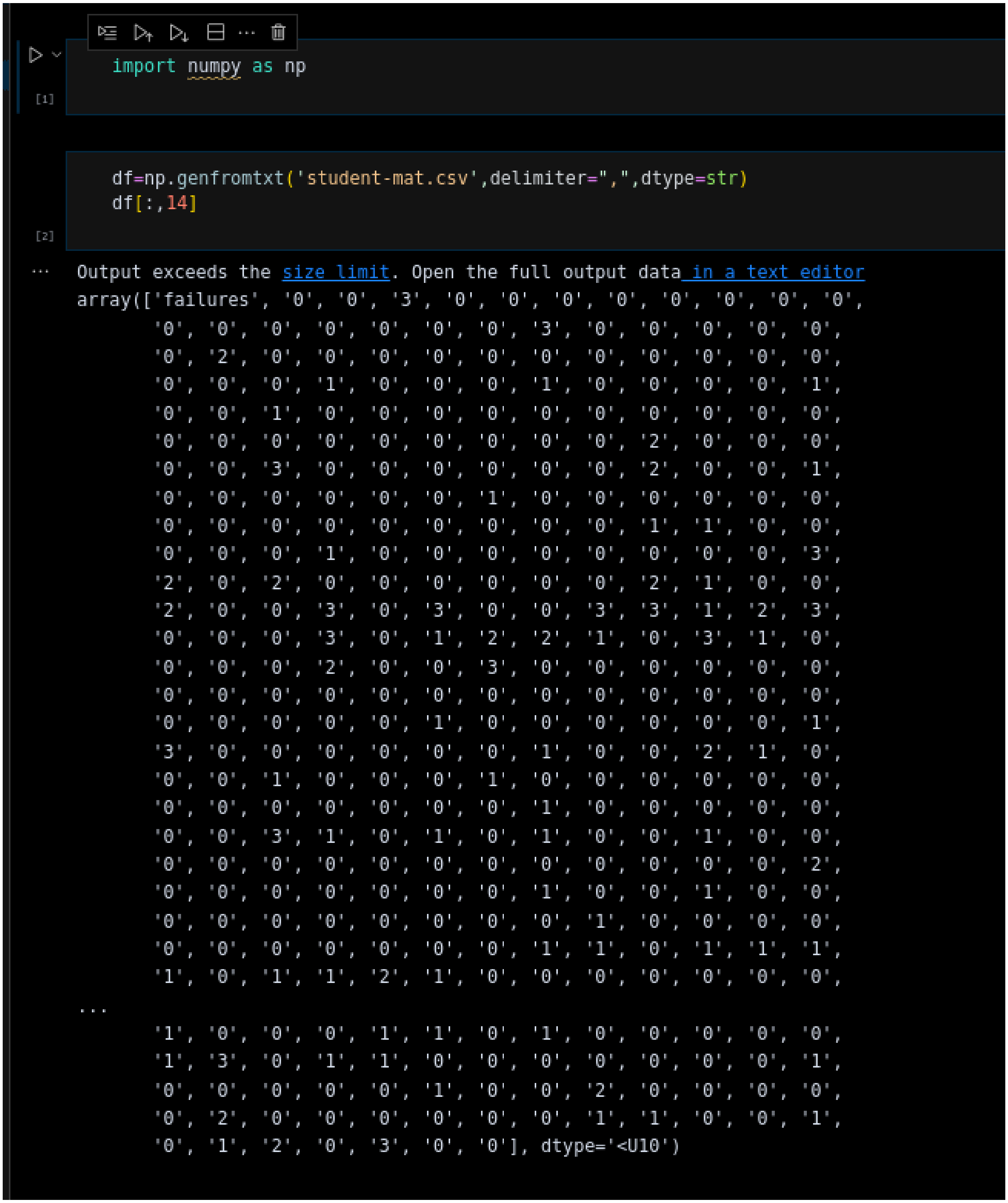
various matrix operations like add(), subtract(),multiply(), divide(), dot() can be performed on 2D arrays in numpy. Numpy allows broadcasting of arrays for uncompatible dimensions which will help while performing these operations.

Activities:

1. Download data set with atleast 1500 rows and 10-20 columns(numeric and non numeric) from valid data sources
2. Perform in detail Exploratory data analysis of this dataset
3. Write down description of your dataset based on analysis done in activity
4. Write aleast 5 different types of conclusions on your dataset

---

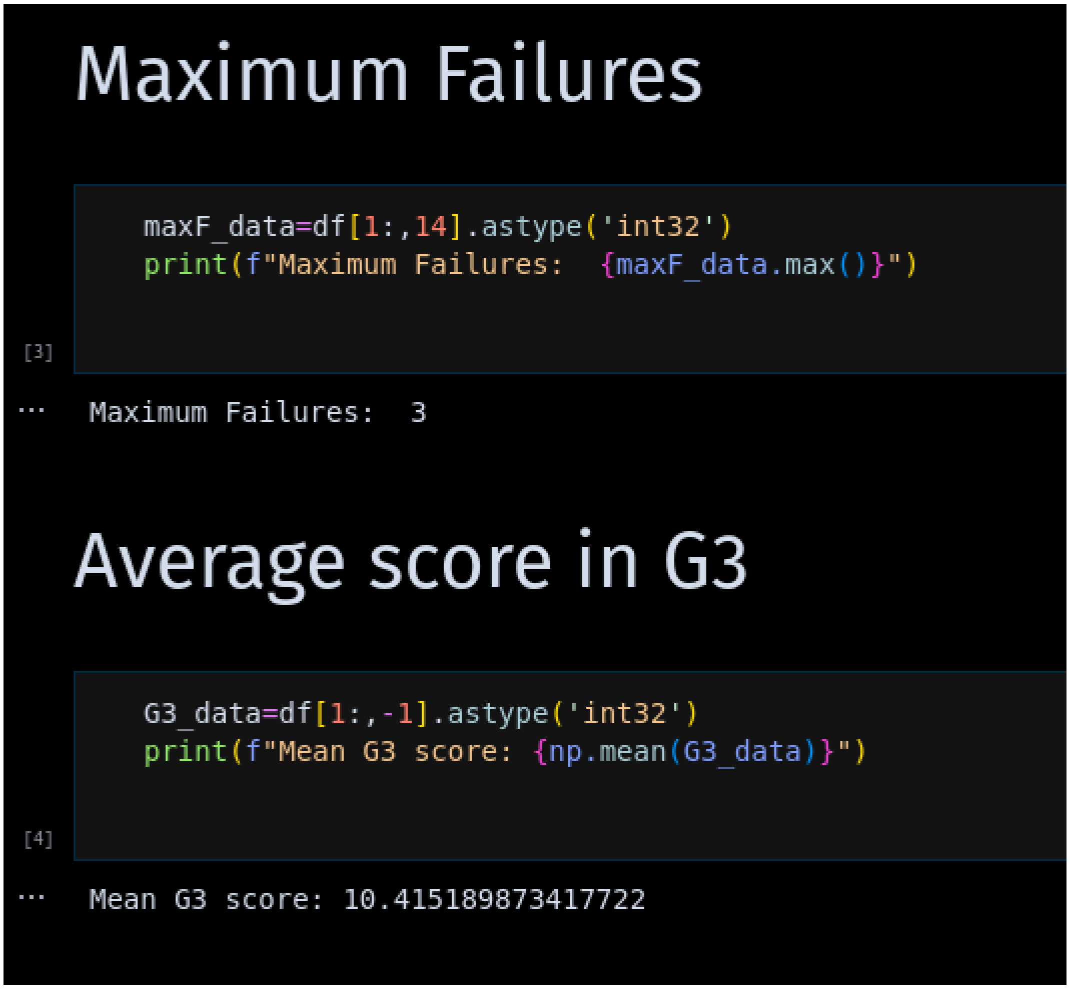
Result: (script and output)



```
import numpy as np
```

```
df=np.genfromtxt('student-mat.csv',delimiter=",",dtype=str)
df[:,14]
```

```
Output exceeds the size limit. Open the full output data in a text editor
array(['failures', '0', '0', '3', '0', '0', '0', '0', '0', '0', '0', '0',
       '0', '0', '0', '0', '0', '0', '0', '3', '0', '0', '0', '0', '0',
       '0', '2', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0',
       '0', '0', '0', '1', '0', '0', '0', '1', '0', '0', '0', '0', '1',
       '0', '0', '1', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0',
       '0', '0', '0', '0', '0', '0', '0', '0', '0', '2', '0', '0', '0',
       '0', '0', '3', '0', '0', '0', '0', '0', '0', '2', '0', '0', '1',
       '0', '0', '0', '0', '0', '0', '1', '0', '0', '0', '0', '0', '0',
       '0', '0', '0', '0', '0', '0', '0', '0', '0', '1', '1', '0', '0',
       '0', '0', '0', '1', '0', '0', '0', '0', '0', '0', '0', '0', '3',
       '2', '0', '2', '0', '0', '0', '0', '0', '0', '2', '1', '0', '0',
       '2', '0', '0', '3', '0', '3', '0', '0', '3', '3', '1', '2', '3',
       '0', '0', '0', '3', '0', '1', '2', '2', '1', '0', '3', '1', '0',
       '0', '0', '0', '2', '0', '0', '3', '0', '0', '0', '0', '0', '0',
       '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0',
       '0', '0', '0', '0', '0', '1', '0', '0', '0', '0', '0', '0', '1',
       '3', '0', '0', '0', '0', '0', '0', '1', '0', '0', '2', '1', '0',
       '0', '0', '1', '0', '0', '0', '1', '0', '0', '0', '0', '0', '0',
       '0', '0', '0', '0', '0', '0', '0', '1', '0', '0', '0', '0', '0',
       '0', '0', '3', '1', '0', '1', '0', '1', '0', '0', '1', '0', '0',
       '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '2',
       '0', '0', '0', '0', '0', '0', '0', '1', '0', '0', '1', '0', '0',
       '0', '0', '0', '0', '0', '0', '0', '0', '1', '0', '0', '0', '0',
       '0', '0', '0', '0', '0', '0', '0', '1', '1', '0', '1', '1', '1',
       '1', '0', '1', '1', '2', '1', '0', '0', '0', '0', '0', '0', '0',
...
       '1', '0', '0', '0', '1', '1', '0', '1', '0', '0', '0', '0', '0',
       '1', '3', '0', '1', '1', '0', '0', '0', '0', '0', '0', '0', '1',
       '0', '0', '0', '0', '0', '1', '0', '0', '2', '0', '0', '0', '0',
       '0', '2', '0', '0', '0', '0', '0', '0', '1', '1', '0', '0', '1',
       '0', '1', '2', '0', '3', '0', '0'], dtype='<U10')
```



Maximum Failures

```
maxF_data=df[1:,14].astype('int32')
print(f"Maximum Failures:  {maxF_data.max()}")
```

[3]

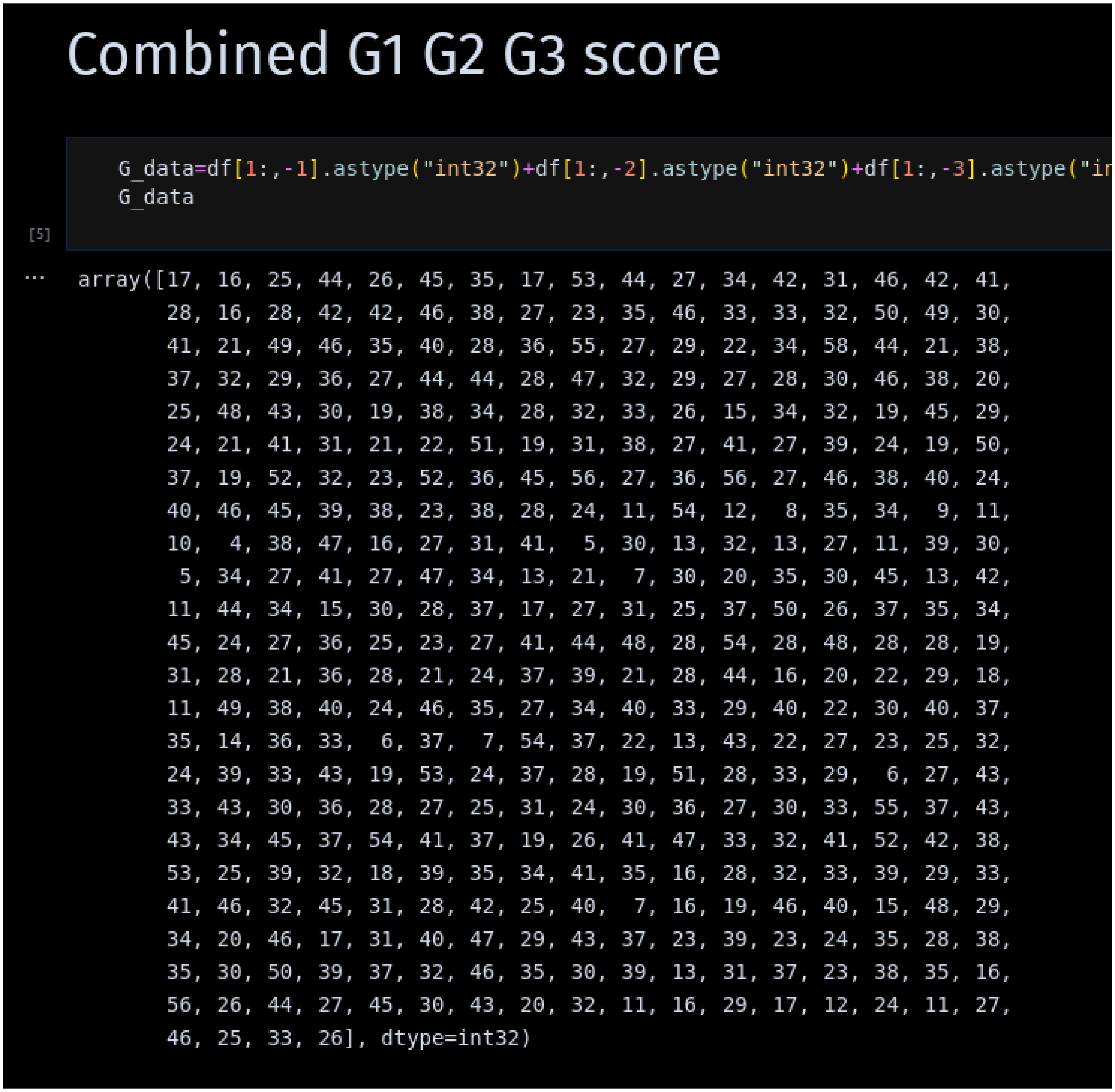
```
Maximum Failures:  3
```

Average score in G3

```
G3_data=df[1:,-1].astype('int32')
print(f"Mean G3 score: {np.mean(G3_data)}")
```

[4]

```
Mean G3 score: 10.415189873417722
```

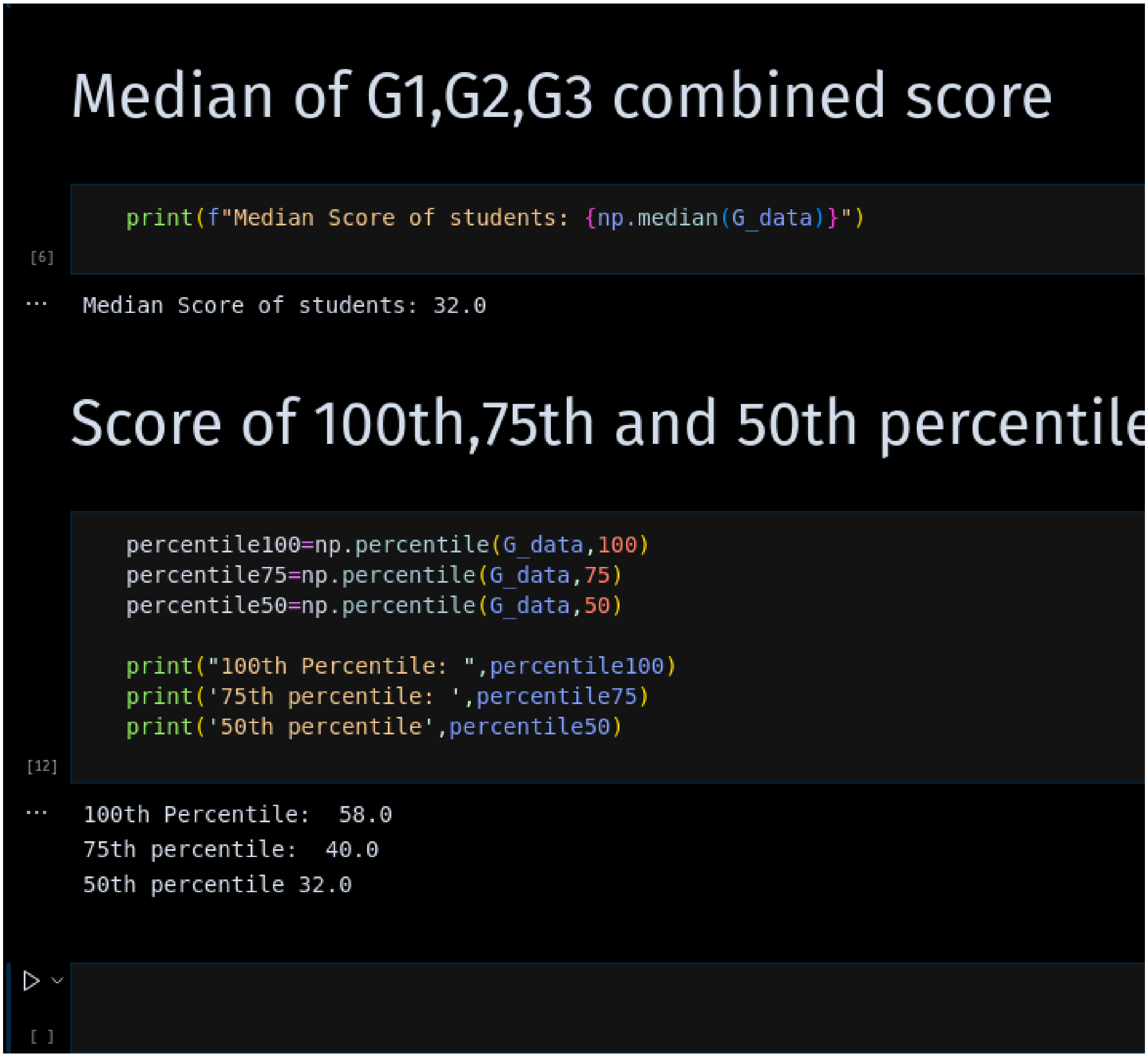


Combined G1 G2 G3 score

```
G_data=df[1:,-1].astype("int32")+df[1:,-2].astype("int32")+df[1:,-3].astype("in
G_data
```

[5]

```
array([17, 16, 25, 44, 26, 45, 35, 17, 53, 44, 27, 34, 42, 31, 46, 42, 41,
       28, 16, 28, 42, 42, 46, 38, 27, 23, 35, 46, 33, 33, 32, 50, 49, 30,
       41, 21, 49, 46, 35, 40, 28, 36, 55, 27, 29, 22, 34, 58, 44, 21, 38,
       37, 32, 29, 36, 27, 44, 44, 28, 47, 32, 29, 27, 28, 30, 46, 38, 20,
       25, 48, 43, 30, 19, 38, 34, 28, 32, 33, 26, 15, 34, 32, 19, 45, 29,
       24, 21, 41, 31, 21, 22, 51, 19, 31, 38, 27, 41, 27, 39, 24, 19, 50,
       37, 19, 52, 32, 23, 52, 36, 45, 56, 27, 36, 56, 27, 46, 38, 40, 24,
       40, 46, 45, 39, 38, 23, 38, 28, 24, 11, 54, 12,  8, 35, 34,  9, 11,
       10,  4, 38, 47, 16, 27, 31, 41,  5, 30, 13, 32, 13, 27, 11, 39, 30,
        5, 34, 27, 41, 27, 47, 34, 13, 21,  7, 30, 20, 35, 30, 45, 13, 42,
       11, 44, 34, 15, 30, 28, 37, 17, 27, 31, 25, 37, 50, 26, 37, 35, 34,
       45, 24, 27, 36, 25, 23, 27, 41, 44, 48, 28, 54, 28, 48, 28, 28, 19,
       31, 28, 21, 36, 28, 21, 24, 37, 39, 21, 28, 44, 16, 20, 22, 29, 18,
       11, 49, 38, 40, 24, 46, 35, 27, 34, 40, 33, 29, 40, 22, 30, 40, 37,
       35, 14, 36, 33,  6, 37,  7, 54, 37, 22, 13, 43, 22, 27, 23, 25, 32,
       24, 39, 33, 43, 19, 53, 24, 37, 28, 19, 51, 28, 33, 29,  6, 27, 43,
       33, 43, 30, 36, 28, 27, 25, 31, 24, 30, 36, 27, 30, 33, 55, 37, 43,
       43, 34, 45, 37, 54, 41, 37, 19, 26, 41, 47, 33, 32, 41, 52, 42, 38,
       53, 25, 39, 32, 18, 39, 35, 34, 41, 35, 16, 28, 32, 33, 39, 29, 33,
       41, 46, 32, 45, 31, 28, 42, 25, 40,  7, 16, 19, 46, 40, 15, 48, 29,
       34, 20, 46, 17, 31, 40, 47, 29, 43, 37, 23, 39, 23, 24, 35, 28, 38,
       35, 30, 50, 39, 37, 32, 46, 35, 30, 39, 13, 31, 37, 23, 38, 35, 16,
       56, 26, 44, 27, 45, 30, 43, 20, 32, 11, 16, 29, 17, 12, 24, 11, 27,
       46, 25, 33, 26], dtype=int32)
```



Median of G1,G2,G3 combined score

```
print(f"Median Score of students: {np.median(G_data)}")
```

```
Median Score of students: 32.0
```

Score of 100th,75th and 50th percentile

```
percentile100=np.percentile(G_data,100)
percentile75=np.percentile(G_data,75)
percentile50=np.percentile(G_data,50)

print("100th Percentile: ",percentile100)
print('75th percentile: ',percentile75)
print('50th percentile',percentile50)
```

```
100th Percentile:  58.0
75th percentile:  40.0
50th percentile 32.0
```

Outcomes:

CO2. Inculcate the knowledge of python libraries like numpy, pandas, matplotlib for scientific- computing and data visualization.

Conclusion: (Conclusion to be based on the objectives and outcomes achieved)

Successfully analysed and got meaningful results using the numpy library.

References:

1. https://www.geeksforgeeks.org/python-numpy/