Department of Computer Science and Engineering

21st Batch

Lab Report 5

Course title : Digital Signal Processing Lab
Course Code : CSE-414

Submitted By	Submitted To
Name : Md. Mahfujur Rahman ID : 192311014 Section : A Semester : 10th Batch : 21st	Name : Salma Akter Lima Designation : Lecturer(Provisional), Varendra University, Rajshahi Name : Sumaiya Tasnim Designation : Lecturer(Provisional), Varendra University, Rajshahi

Submission date: 25-08-2022

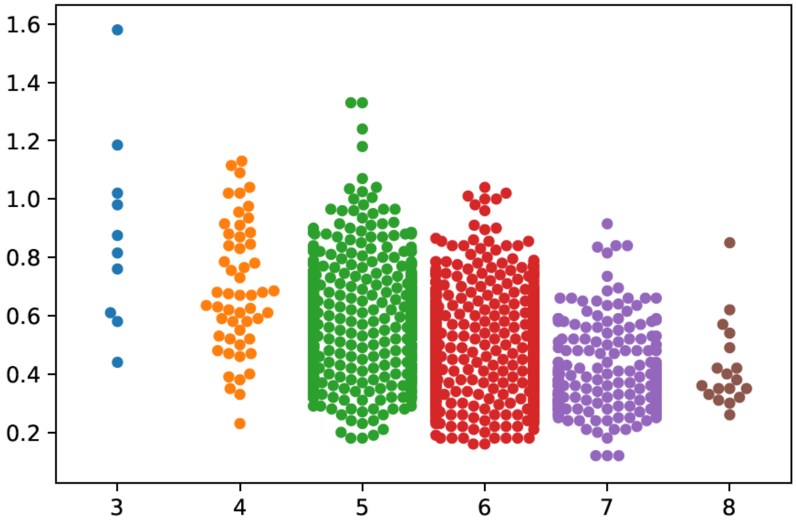
Signature

Problem Statement: Create a basic data set as a CSV file then perform EDA analysis on that data set.

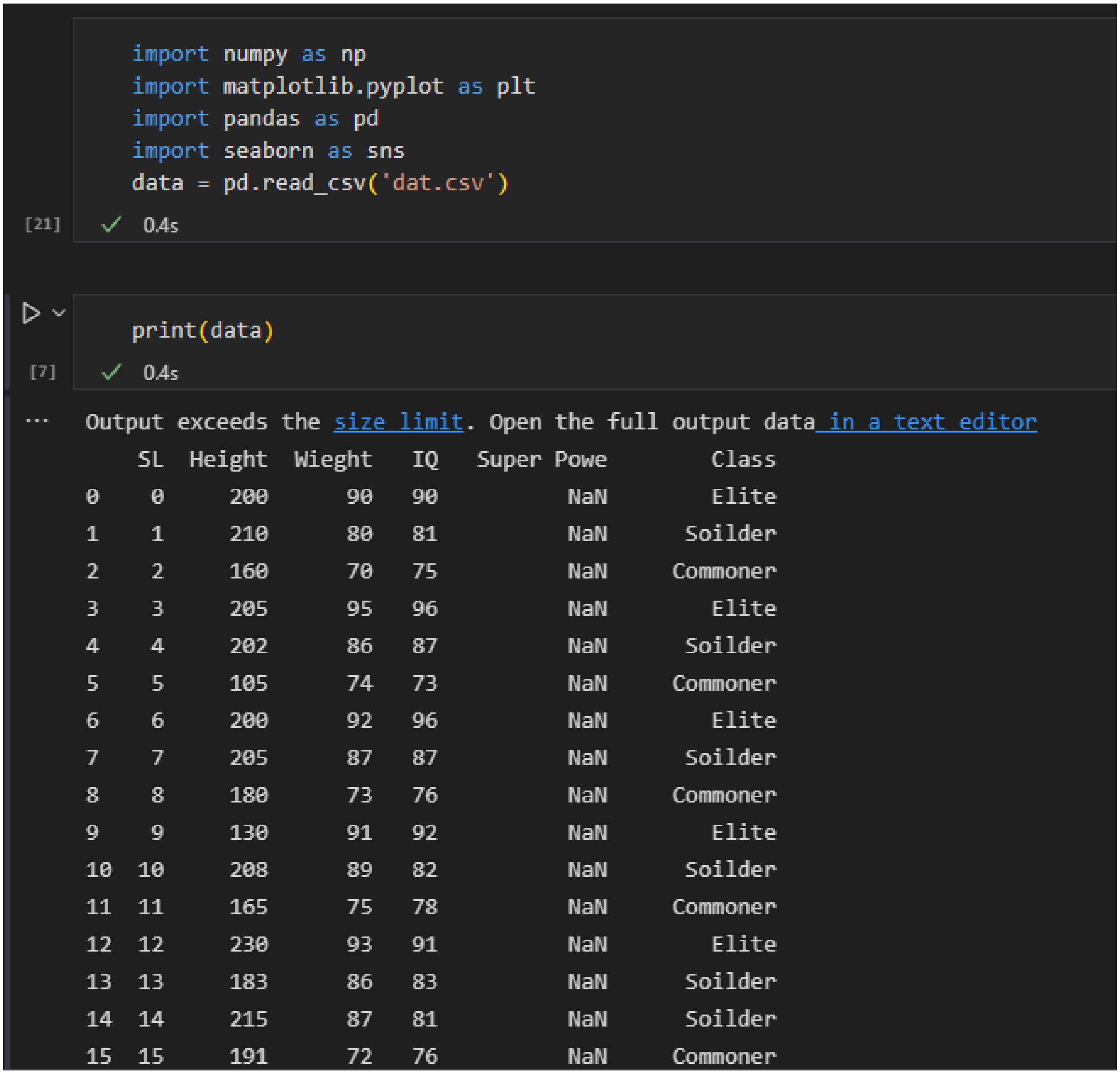
Theory: In statistics, exploratory data analysis is an approach of analyzing data sets to summarize their main characteristics, often using statistical graphics and other data visualization methods. A statistical model can be used or not, but primarily EDA is for seeing what the data can tell us beyond the formal modeling and thereby contrasts traditional hypothesis testing.

Exploratory data analysis has been promoted by John Tukey since 1970 to encourage statisticians to explore the data, and possibly formulate hypotheses that could lead to new data collection and experiments. EDA is different from initial data analysis (IDA), which focuses more narrowly on checking assumptions required for model fitting and hypothesis testing, and handling missing values and making transformations of variables as needed. EDA encompasses IDA.

Exploratory data analysis (EDA) is used by data scientists to analyze and investigate data sets and summarize their main characteristics, often employing data visualization methods. It helps determine how best to manipulate data sources to get the answers you need, making it easier for data scientists to discover patterns, spot anomalies, test a hypothesis, or check assumptions.



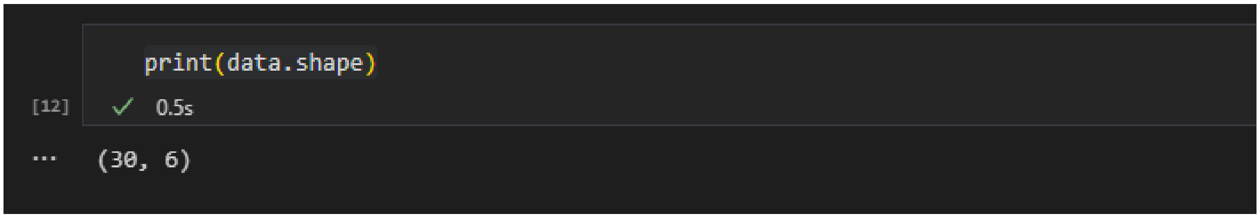
CODE:



```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
data = pd.read_csv('dat.csv')
```

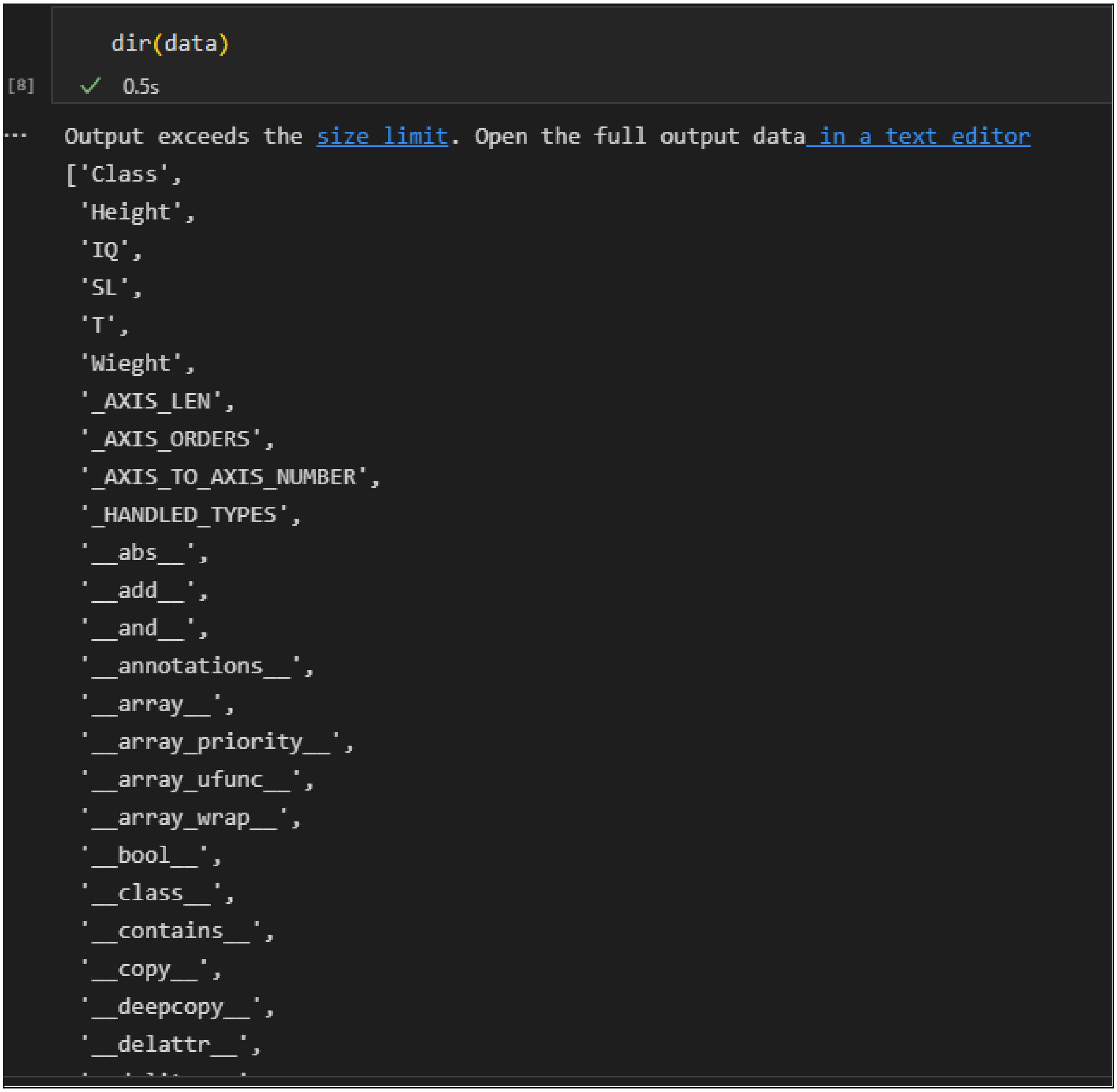
```
print(data)
```

```
Output exceeds the size limit. Open the full output data in a text editor
    SL  Height  Wieght  IQ   Super Powe        Class
0    0     200      90  90          NaN        Elite
1    1     210      80  81          NaN      Soilder
2    2     160      70  75          NaN     Commoner
3    3     205      95  96          NaN        Elite
4    4     202      86  87          NaN      Soilder
5    5     105      74  73          NaN     Commoner
6    6     200      92  96          NaN        Elite
7    7     205      87  87          NaN      Soilder
8    8     180      73  76          NaN     Commoner
9    9     130      91  92          NaN        Elite
10  10     208      89  82          NaN      Soilder
11  11     165      75  78          NaN     Commoner
12  12     230      93  91          NaN        Elite
13  13     183      86  83          NaN      Soilder
14  14     215      87  81          NaN      Soilder
15  15     191      72  76          NaN     Commoner
```



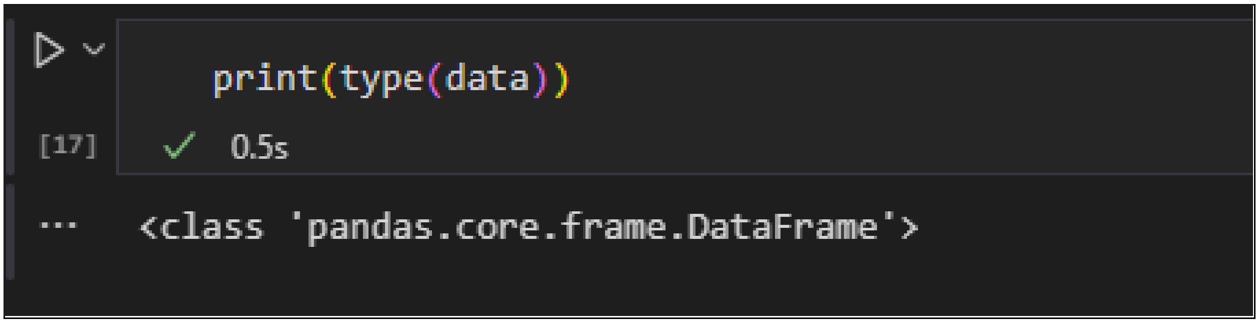
```
print(data.shape)
```

```
(30, 6)
```



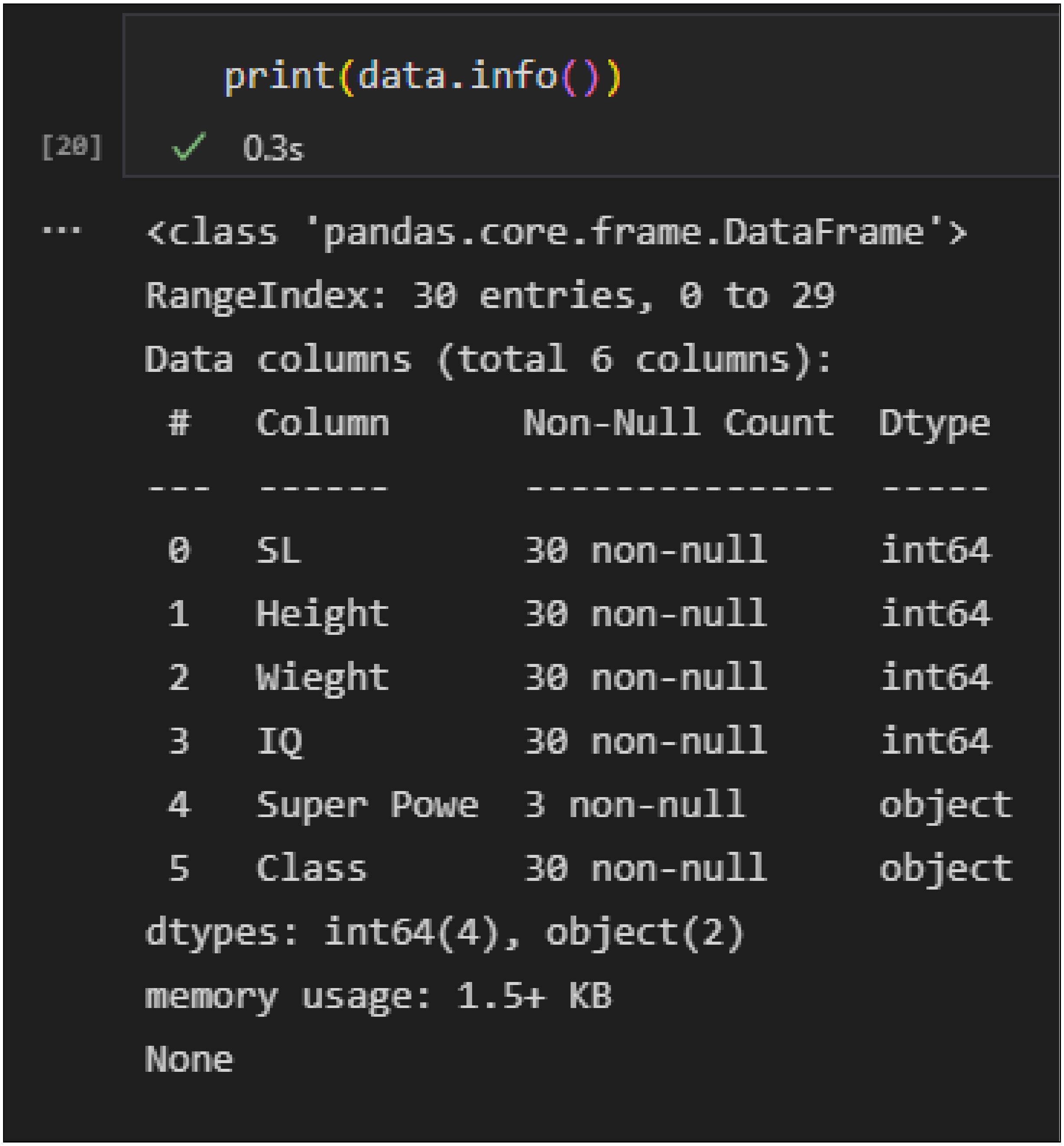
```
dir(data)
```

```
Output exceeds the size limit. Open the full output data in a text editor
['Class',
 'Height',
 'IQ',
 'SL',
 'T',
 'Wieght',
 '_AXIS_LEN',
 '_AXIS_ORDERS',
 '_AXIS_TO_AXIS_NUMBER',
 '_HANDLED_TYPES',
 '__abs__',
 '__add__',
 '__and__',
 '__annotations__',
 '__array__',
 '__array_priority__',
 '__array_ufunc__',
 '__array_wrap__',
 '__bool__',
 '__class__',
 '__contains__',
 '__copy__',
 '__deepcopy__',
 '__delattr__',
```



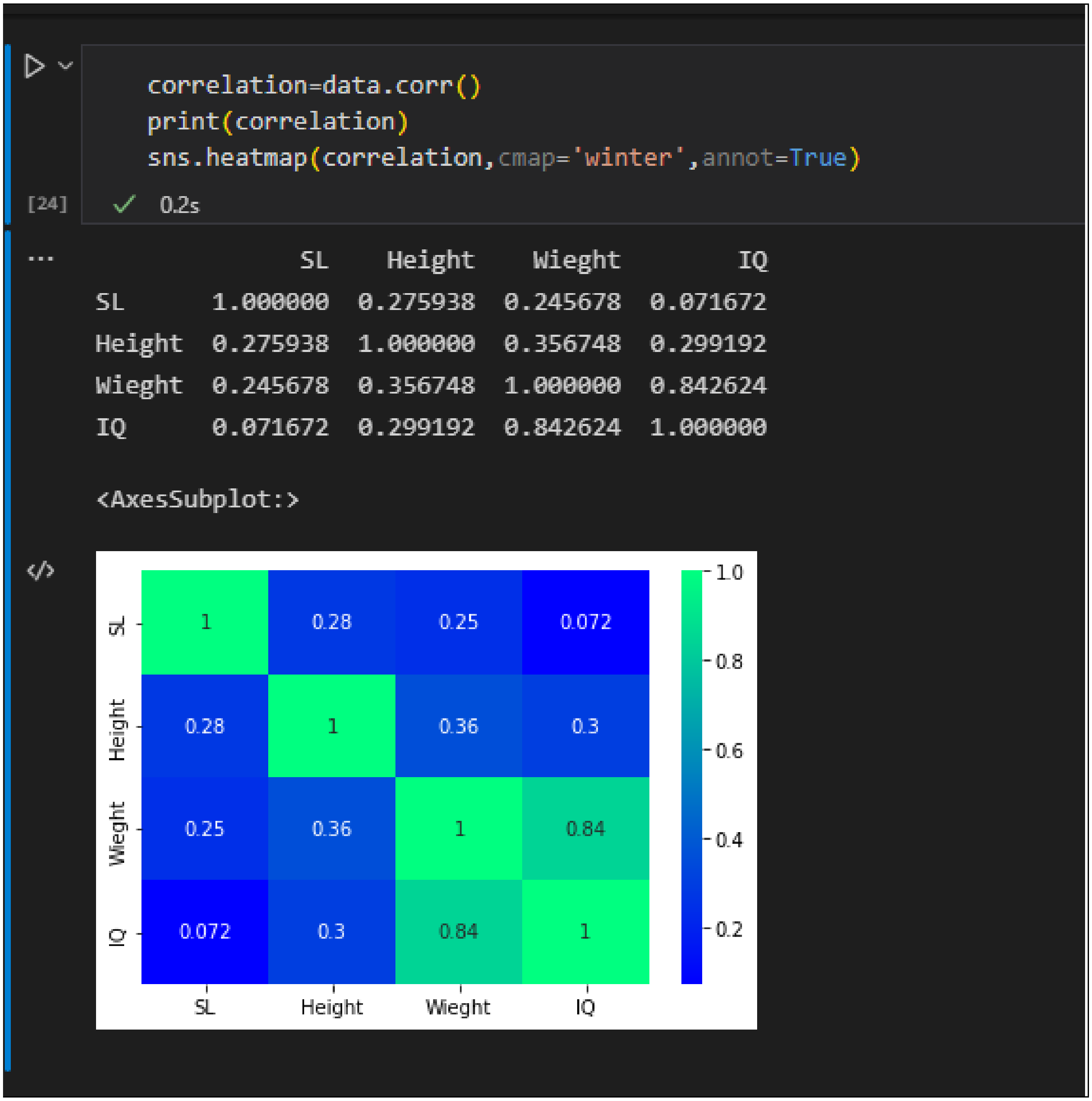
```
print(type(data))
```

```
<class 'pandas.core.frame.DataFrame'>
```



```
print(data.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   SL          30 non-null     int64
 1   Height      30 non-null     int64
 2   Wieght      30 non-null     int64
 3   IQ          30 non-null     int64
 4   Super Powe  3 non-null      object
 5   Class       30 non-null     object
dtypes: int64(4), object(2)
memory usage: 1.5+ KB
None
```



```
correlation=data.corr()
print(correlation)
sns.heatmap(correlation,cmap='winter',annot=True)
```

[24] ✓ 0.2s

```
              SL    Height    Wieght        IQ
SL      1.000000  0.275938  0.245678  0.071672
Height  0.275938  1.000000  0.356748  0.299192
Wieght  0.245678  0.356748  1.000000  0.842624
IQ      0.071672  0.299192  0.842624  1.000000

<AxesSubplot:>
```

Conclusion : By simply using the pandas library we can import our csv file as our data sets then perform different operation on it . by using the seaborn library we can also perform Graphical EDA analysis upon the data set.