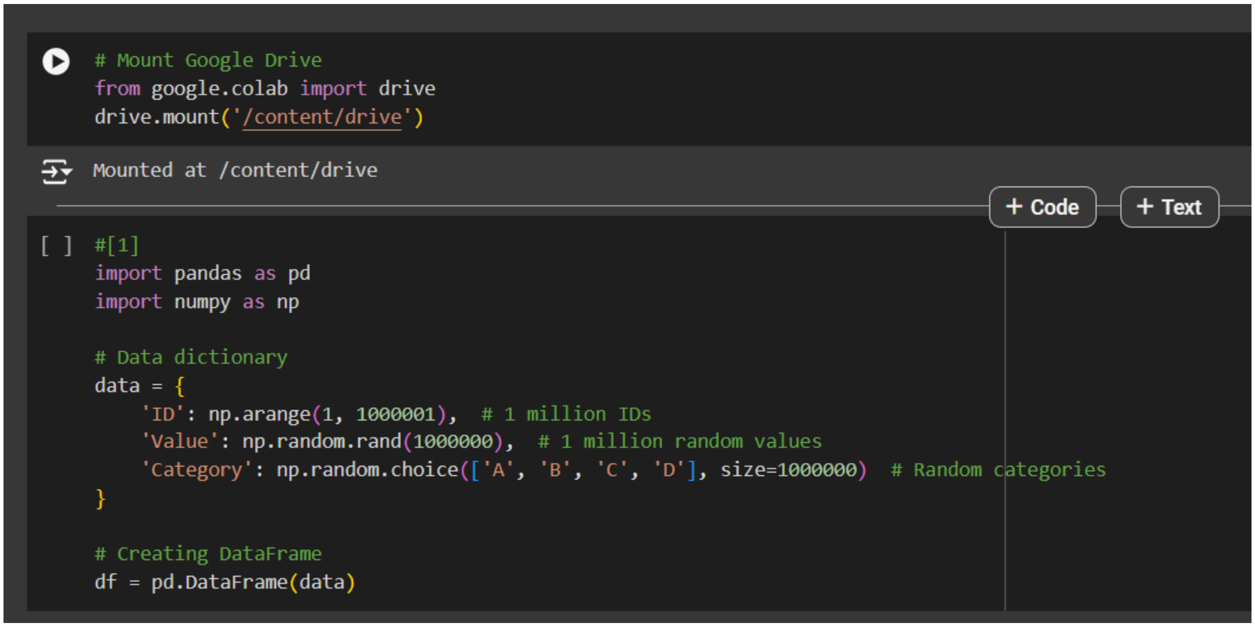
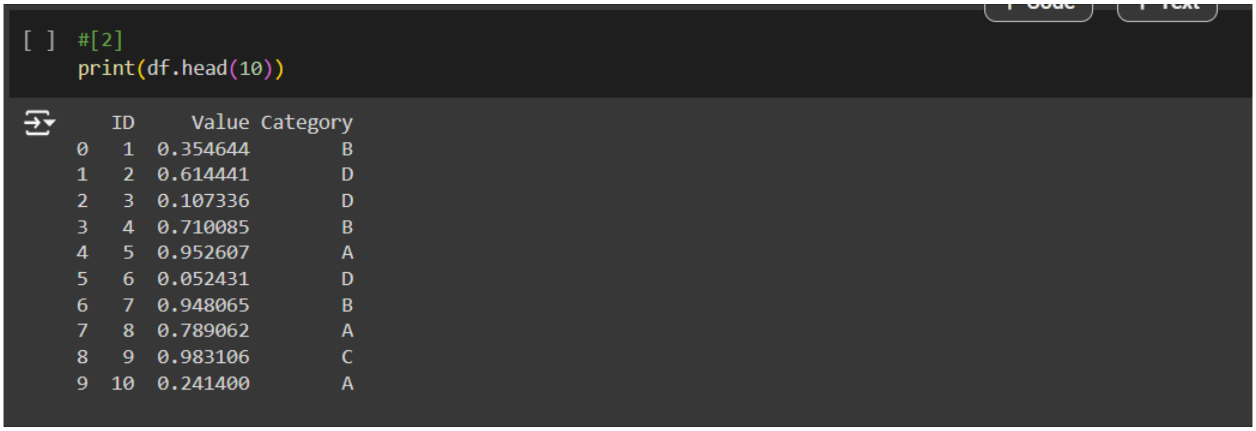
ICP3 REPORT



```
# Mount Google Drive
from google.colab import drive
drive.mount('/content/drive')
```

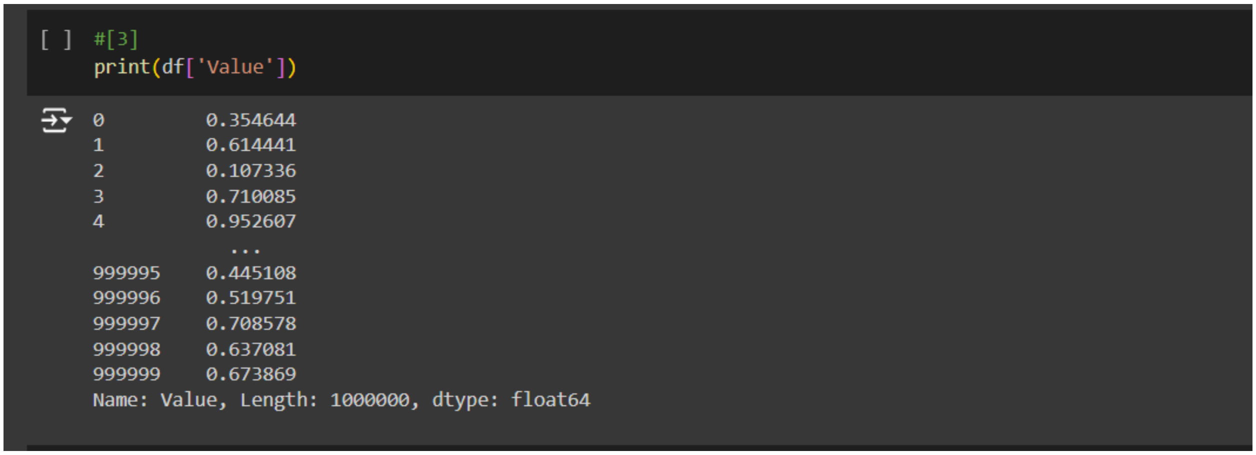
```
Mounted at /content/drive
```

```
[ ] #[1]
    import pandas as pd
    import numpy as np

    # Data dictionary
    data = {
        'ID': np.arange(1, 1000001),  # 1 million IDs
        'Value': np.random.rand(1000000),  # 1 million random values
        'Category': np.random.choice(['A', 'B', 'C', 'D'], size=1000000)  # Random categories
    }

    # Creating DataFrame
    df = pd.DataFrame(data)
```



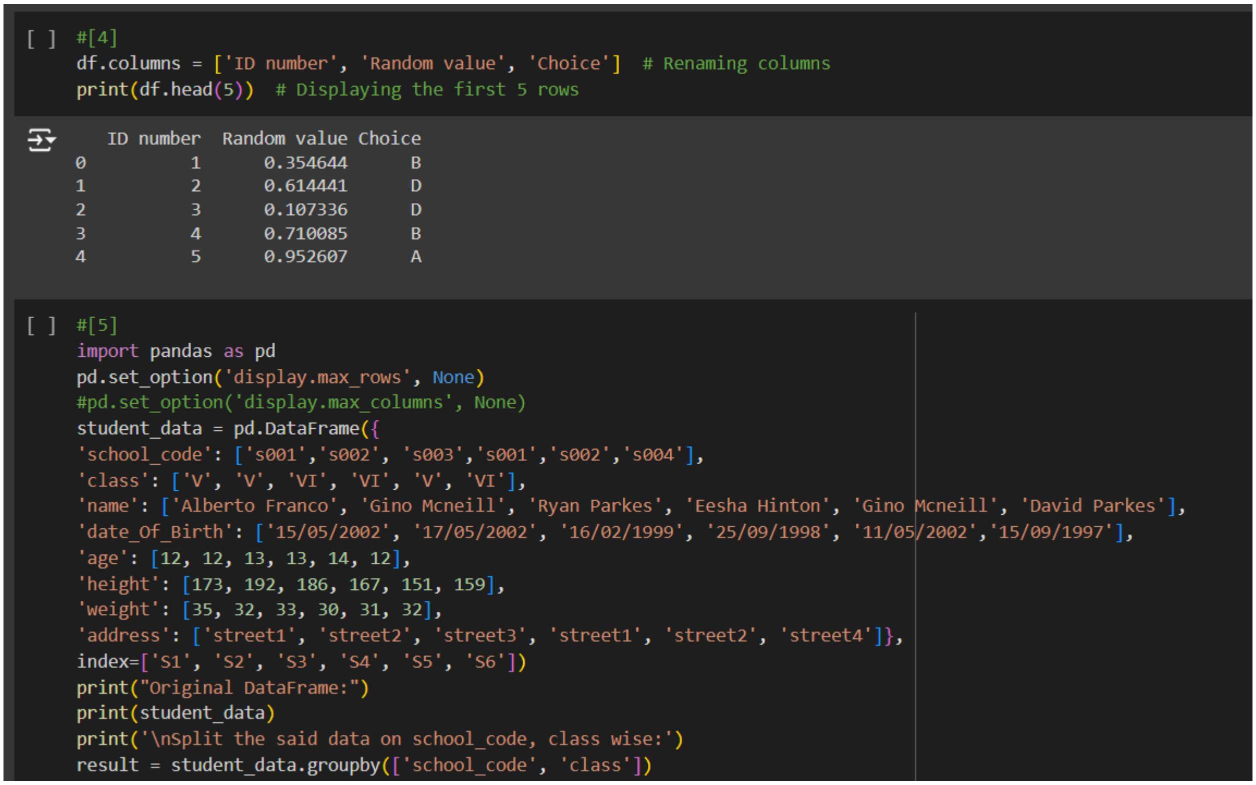
```
[ ] #[2]
    print(df.head(10))
```

```
   ID     Value Category
0   1  0.354644        B
1   2  0.614441        D
2   3  0.107336        D
3   4  0.710085        B
4   5  0.952607        A
5   6  0.052431        D
6   7  0.948065        B
7   8  0.789062        A
8   9  0.983106        C
9  10  0.241400        A
```



```
#[3]
print(df['Value'])
```

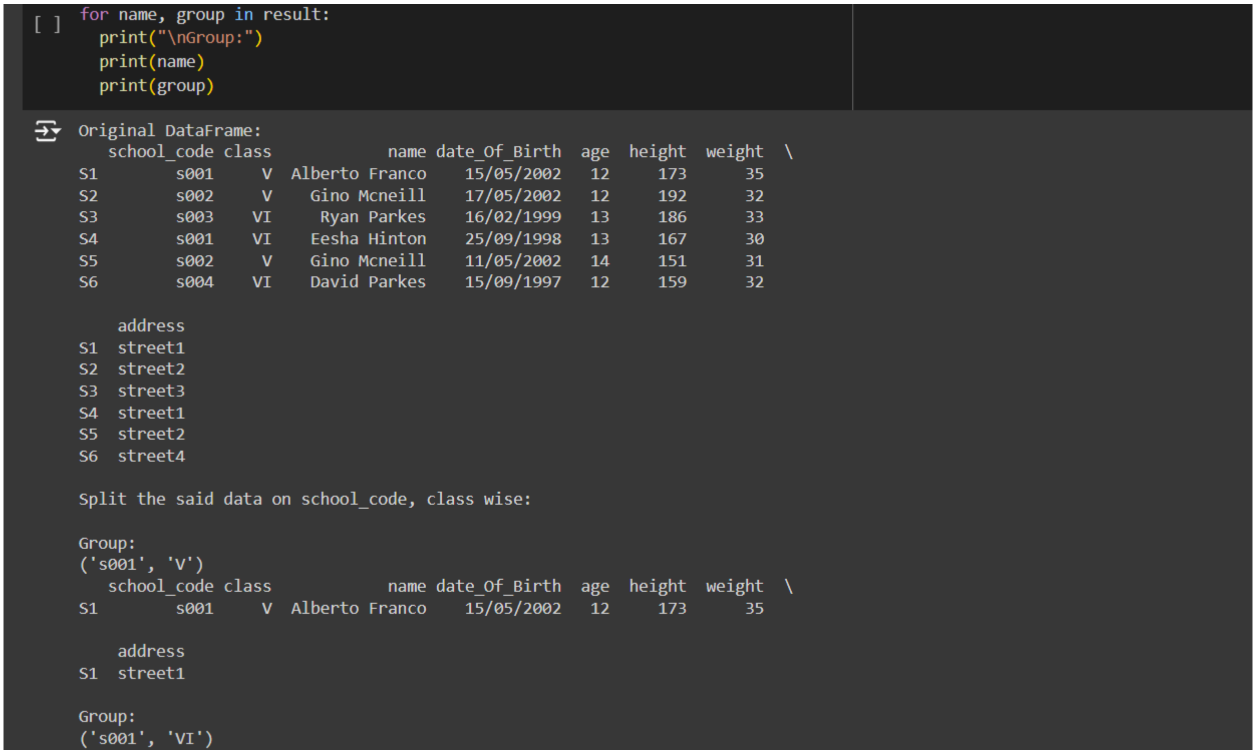
```
0         0.354644
1         0.614441
2         0.107336
3         0.710085
4         0.952607
            ...
999995    0.445108
999996    0.519751
999997    0.708578
999998    0.637081
999999    0.673869
Name: Value, Length: 1000000, dtype: float64
```



```
#[4]
df.columns = ['ID number', 'Random value', 'Choice']  # Renaming columns
print(df.head(5))  # Displaying the first 5 rows
```

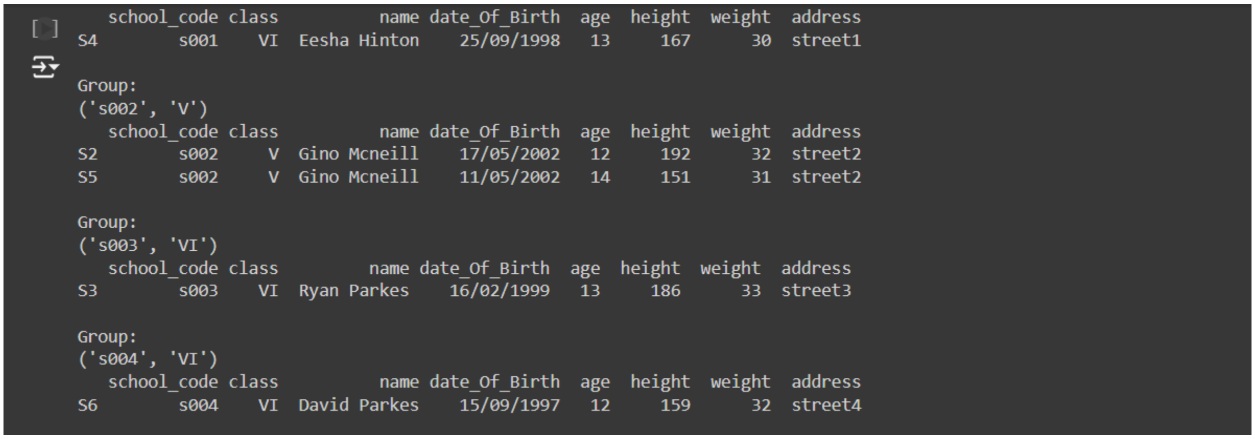
```
   ID number  Random value Choice
0          1      0.354644      B
1          2      0.614441      D
2          3      0.107336      D
3          4      0.710085      B
4          5      0.952607      A
```

```
#[5]
import pandas as pd
pd.set_option('display.max_rows', None)
#pd.set_option('display.max_columns', None)
student_data = pd.DataFrame({
'school_code': ['s001','s002', 's003','s001','s002','s004'],
'class': ['V', 'V', 'VI', 'VI', 'V', 'VI'],
'name': ['Alberto Franco', 'Gino Mcneill', 'Ryan Parkes', 'Eesha Hinton', 'Gino Mcneill', 'David Parkes'],
'date_Of_Birth': ['15/05/2002', '17/05/2002', '16/02/1999', '25/09/1998', '11/05/2002','15/09/1997'],
'age': [12, 12, 13, 13, 14, 12],
'height': [173, 192, 186, 167, 151, 159],
'weight': [35, 32, 33, 30, 31, 32],
'address': ['street1', 'street2', 'street3', 'street1', 'street2', 'street4']},
index=['S1', 'S2', 'S3', 'S4', 'S5', 'S6'])
print("Original DataFrame:")
print(student_data)
print('\nSplit the said data on school_code, class wise:')
result = student_data.groupby(['school_code', 'class'])
```

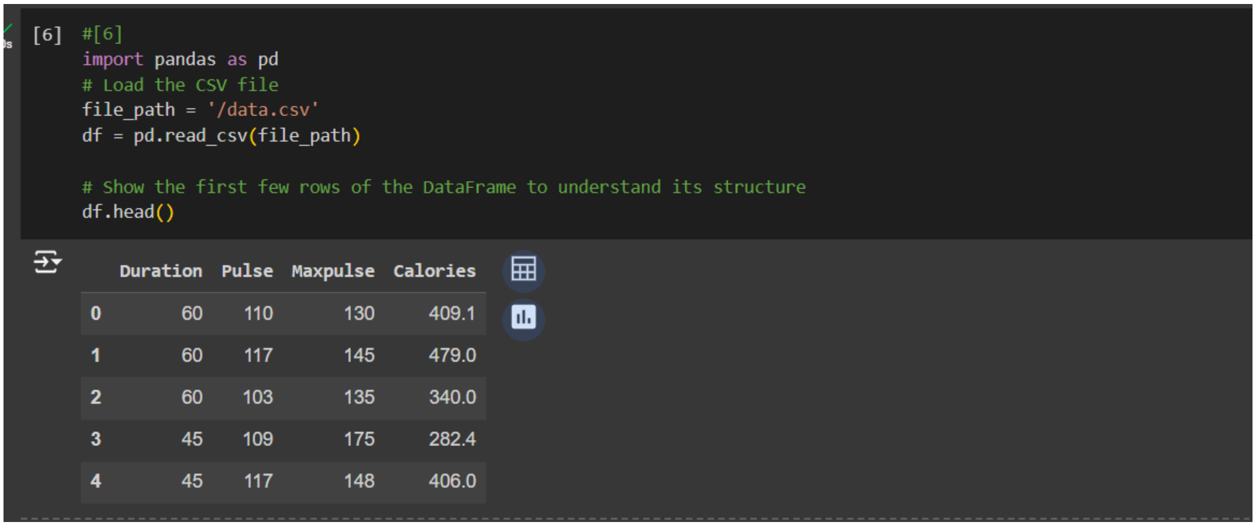


```
for name, group in result:
  print("\nGroup:")
  print(name)
  print(group)
```

```
Original DataFrame:
   school_code class            name date_Of_Birth  age  height  weight  \
S1        s001     V  Alberto Franco    15/05/2002   12     173      35
S2        s002     V    Gino Mcneill    17/05/2002   12     192      32
S3        s003    VI     Ryan Parkes    16/02/1999   13     186      33
S4        s001    VI    Eesha Hinton    25/09/1998   13     167      30
S5        s002     V    Gino Mcneill    11/05/2002   14     151      31
S6        s004    VI    David Parkes    15/09/1997   12     159      32

    address
S1  street1
S2  street2
S3  street3
S4  street1
S5  street2
S6  street4

Split the said data on school_code, class wise:

Group:
('s001', 'V')
   school_code class            name date_Of_Birth  age  height  weight  \
S1        s001     V  Alberto Franco    15/05/2002   12     173      35

    address
S1  street1

Group:
('s001', 'VI')
```

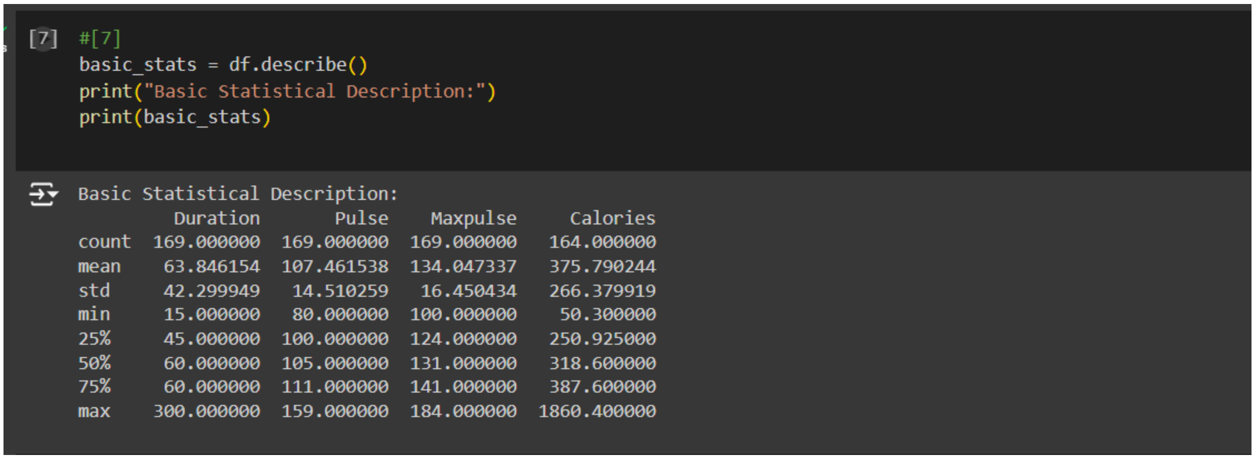


```
   school_code class          name date_Of_Birth  age  height  weight  address
S4        s001    VI  Eesha Hinton    25/09/1998   13     167      30  street1

Group:
('s002', 'V')
   school_code class          name date_Of_Birth  age  height  weight  address
S2        s002     V  Gino Mcneill    17/05/2002   12     192      32  street2
S5        s002     V  Gino Mcneill    11/05/2002   14     151      31  street2

Group:
('s003', 'VI')
   school_code class         name date_Of_Birth  age  height  weight  address
S3        s003    VI  Ryan Parkes    16/02/1999   13     186      33  street3

Group:
('s004', 'VI')
   school_code class          name date_Of_Birth  age  height  weight  address
S6        s004    VI  David Parkes    15/09/1997   12     159      32  street4
```



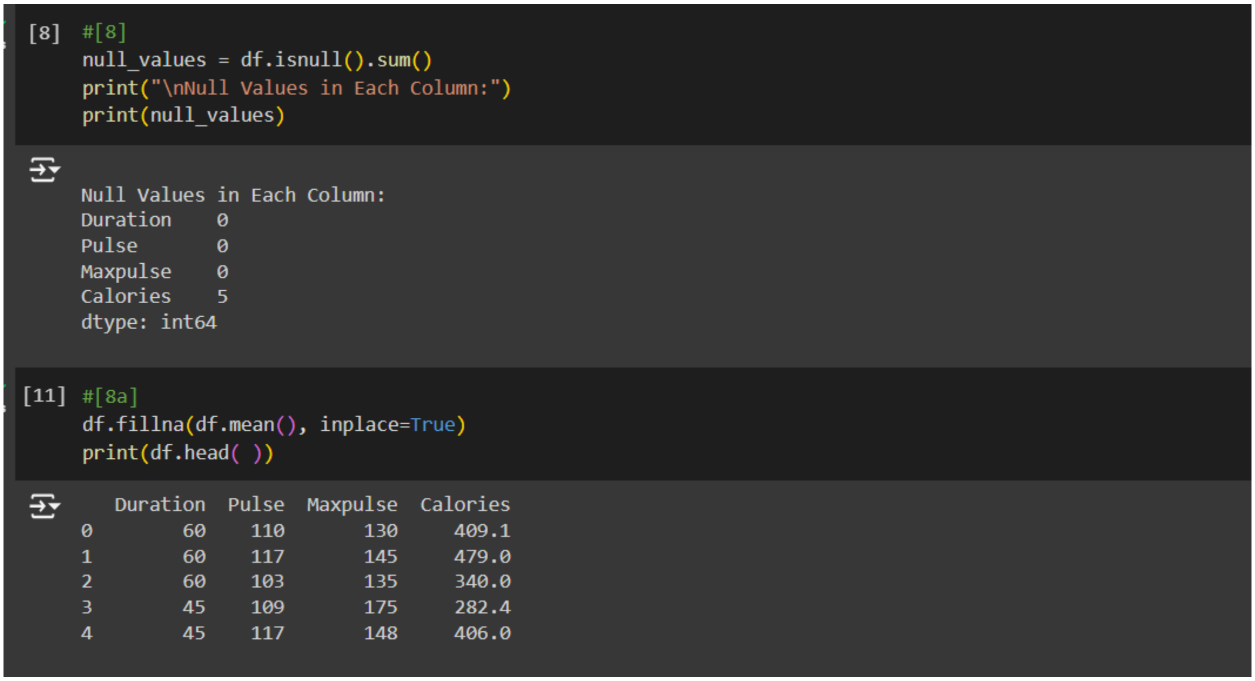
[6]
```python
#[6]
import pandas as pd
# Load the CSV file
file_path = '/data.csv'
df = pd.read_csv(file_path)

# Show the first few rows of the DataFrame to understand its structure
df.head()
```

	Duration	Pulse	Maxpulse	Calories
0	60	110	130	409.1
1	60	117	145	479.0
2	60	103	135	340.0
3	45	109	175	282.4
4	45	117	148	406.0



[7]
```python
#[7]
basic_stats = df.describe()
print("Basic Statistical Description:")
print(basic_stats)
```

```
Basic Statistical Description:
         Duration       Pulse    Maxpulse     Calories
count  169.000000  169.000000  169.000000   164.000000
mean    63.846154  107.461538  134.047337   375.790244
std     42.299949   14.510259   16.450434   266.379919
min     15.000000   80.000000  100.000000    50.300000
25%     45.000000  100.000000  124.000000   250.925000
50%     60.000000  105.000000  131.000000   318.600000
75%     60.000000  111.000000  141.000000   387.600000
max    300.000000  159.000000  184.000000  1860.400000
```

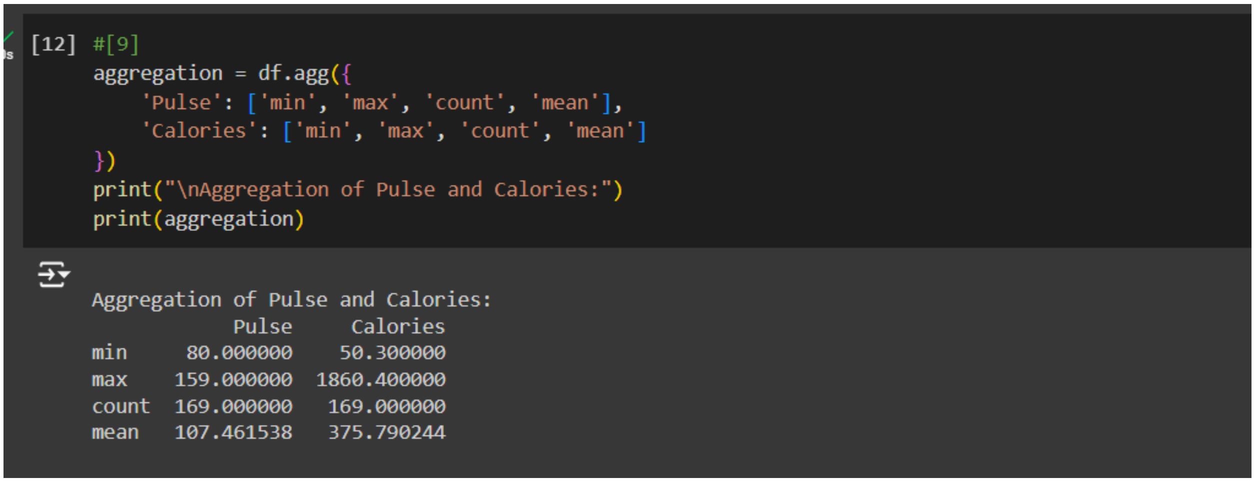


[8]
```python
#[8]
null_values = df.isnull().sum()
print("\nNull Values in Each Column:")
print(null_values)
```

```

Null Values in Each Column:
Duration    0
Pulse       0
Maxpulse    0
Calories    5
dtype: int64
```

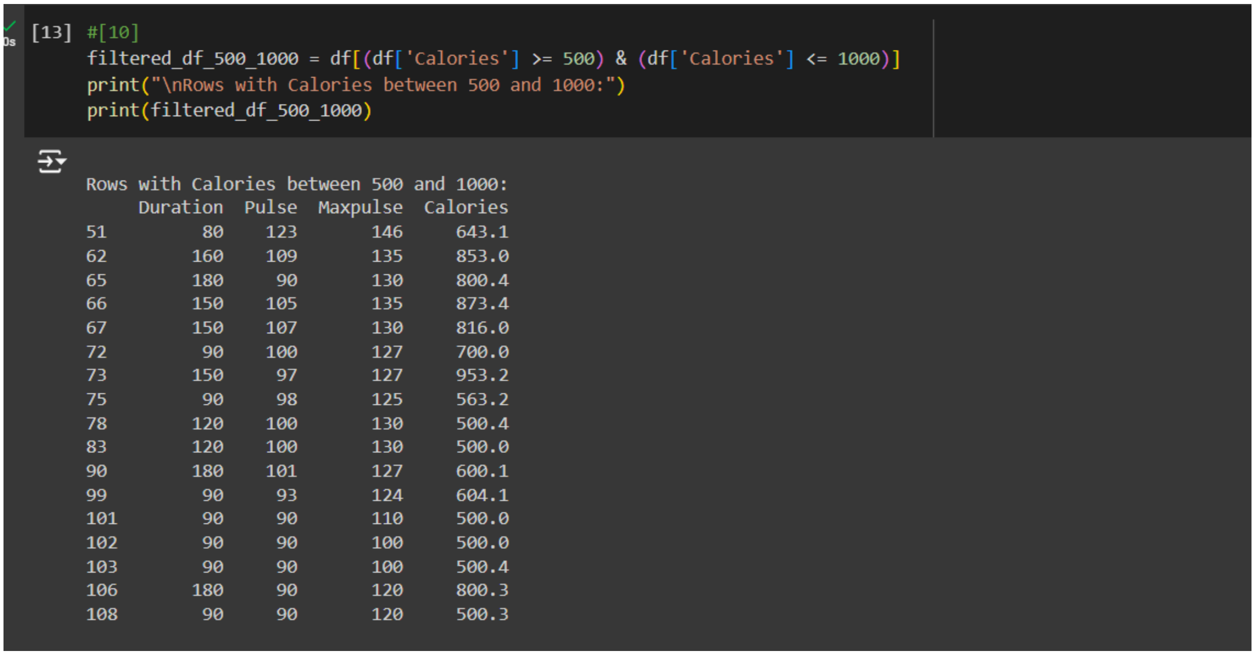
[11]
```python
#[8a]
df.fillna(df.mean(), inplace=True)
print(df.head( ))
```

```
   Duration  Pulse  Maxpulse  Calories
0        60    110       130     409.1
1        60    117       145     479.0
2        60    103       135     340.0
3        45    109       175     282.4
4        45    117       148     406.0
```



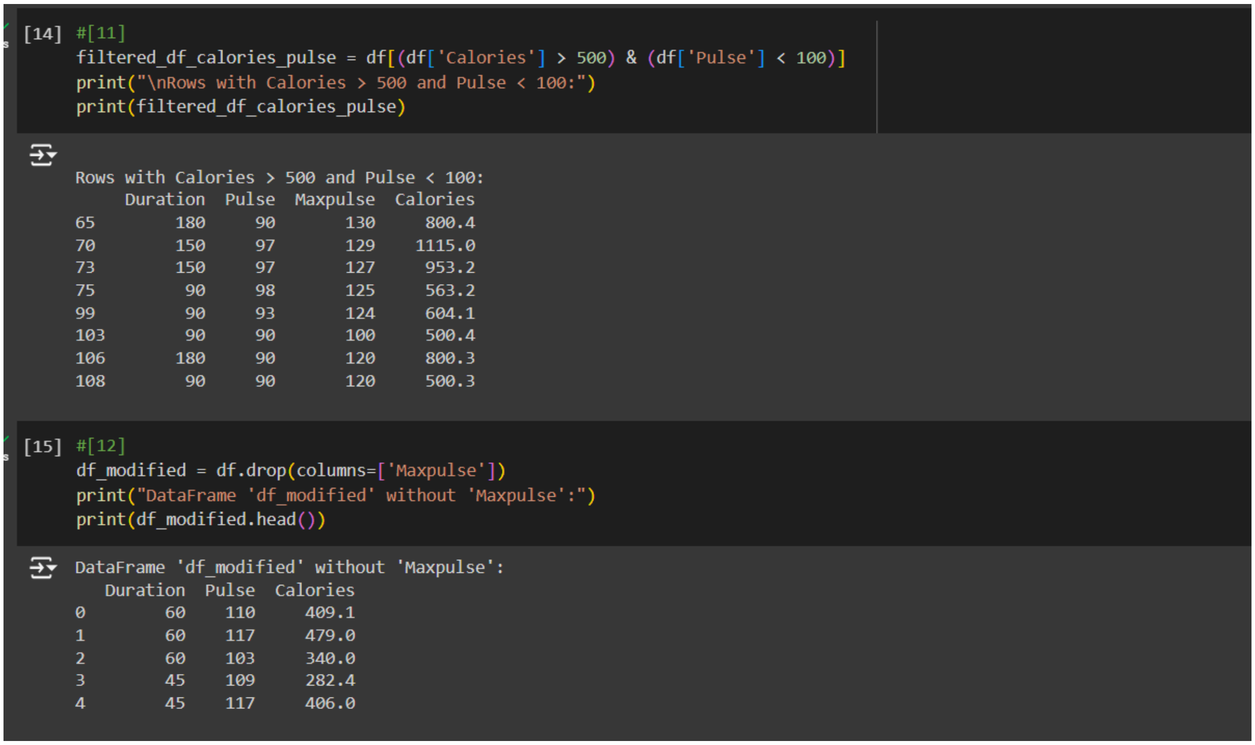
```
[12] #[9]
     aggregation = df.agg({
         'Pulse': ['min', 'max', 'count', 'mean'],
         'Calories': ['min', 'max', 'count', 'mean']
     })
     print("\nAggregation of Pulse and Calories:")
     print(aggregation)
```

```

Aggregation of Pulse and Calories:
            Pulse     Calories
min     80.000000    50.300000
max    159.000000  1860.400000
count  169.000000   169.000000
mean   107.461538   375.790244
```



```
[13] #[10]
     filtered_df_500_1000 = df[(df['Calories'] >= 500) & (df['Calories'] <= 1000)]
     print("\nRows with Calories between 500 and 1000:")
     print(filtered_df_500_1000)
```

```

Rows with Calories between 500 and 1000:
     Duration  Pulse  Maxpulse  Calories
51         80    123       146     643.1
62        160    109       135     853.0
65        180     90       130     800.4
66        150    105       135     873.4
67        150    107       130     816.0
72         90    100       127     700.0
73        150     97       127     953.2
75         90     98       125     563.2
78        120    100       130     500.4
83        120    100       130     500.0
90        180    101       127     600.1
99         90     93       124     604.1
101        90     90       110     500.0
102        90     90       100     500.0
103        90     90       100     500.4
106       180     90       120     800.3
108        90     90       120     500.3
```

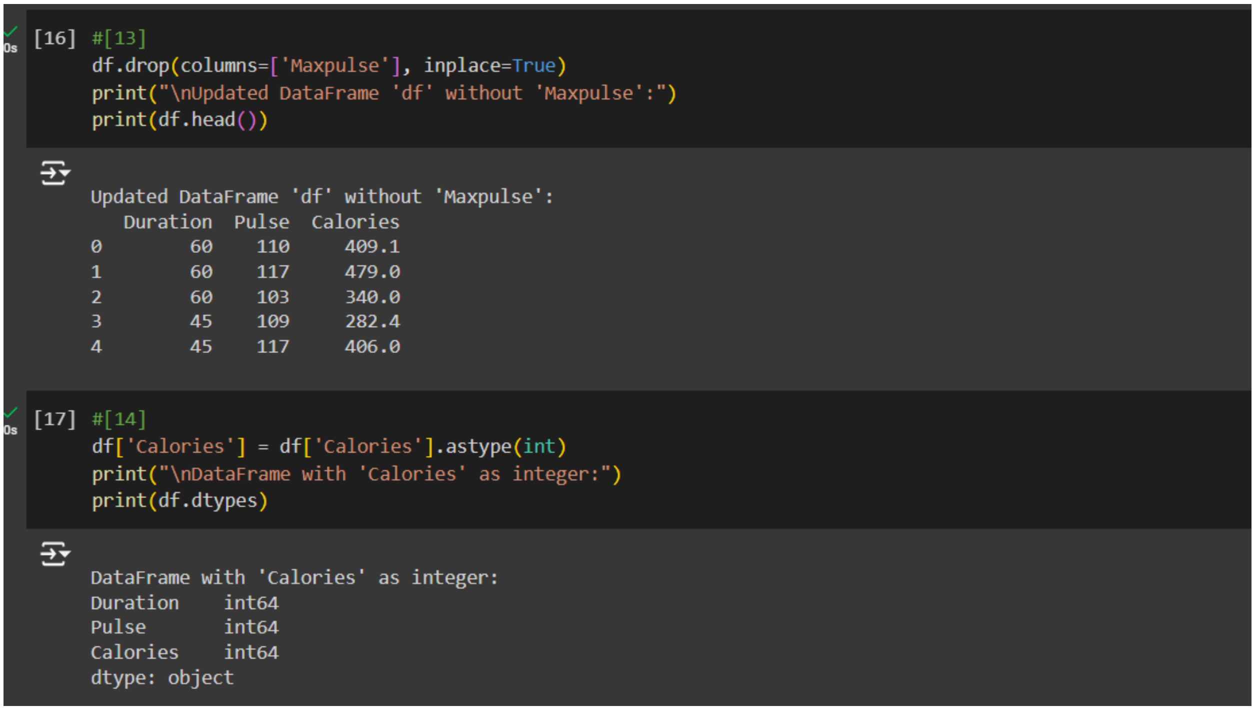


```
[14] #[11]
     filtered_df_calories_pulse = df[(df['Calories'] > 500) & (df['Pulse'] < 100)]
     print("\nRows with Calories > 500 and Pulse < 100:")
     print(filtered_df_calories_pulse)
```

```

Rows with Calories > 500 and Pulse < 100:
     Duration  Pulse  Maxpulse  Calories
65        180     90       130     800.4
70        150     97       129    1115.0
73        150     97       127     953.2
75         90     98       125     563.2
99         90     93       124     604.1
103        90     90       100     500.4
106       180     90       120     800.3
108        90     90       120     500.3
```

```
[15] #[12]
     df_modified = df.drop(columns=['Maxpulse'])
     print("DataFrame 'df_modified' without 'Maxpulse':")
     print(df_modified.head())
```

```
DataFrame 'df_modified' without 'Maxpulse':
   Duration  Pulse  Calories
0        60    110     409.1
1        60    117     479.0
2        60    103     340.0
3        45    109     282.4
4        45    117     406.0
```

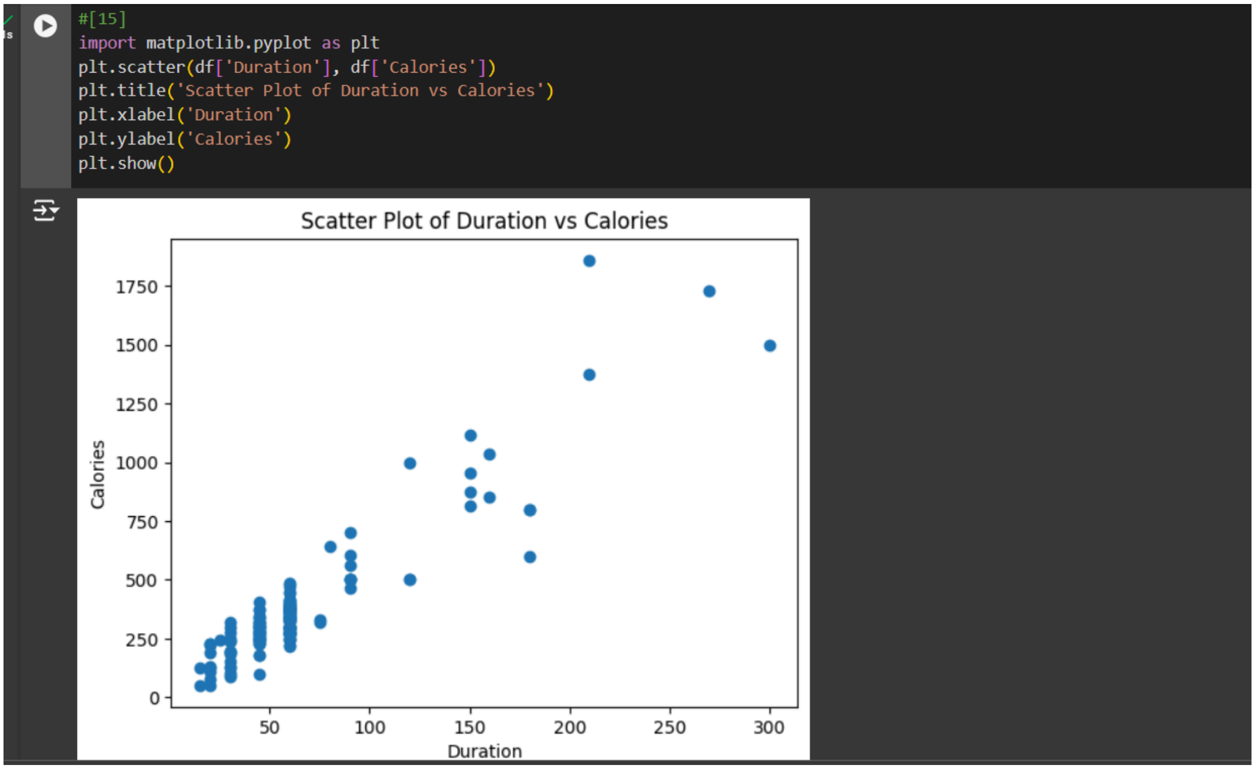


```
[16] #[13]
     df.drop(columns=['Maxpulse'], inplace=True)
     print("\nUpdated DataFrame 'df' without 'Maxpulse':")
     print(df.head())
```

```
Updated DataFrame 'df' without 'Maxpulse':
   Duration  Pulse  Calories
0        60    110     409.1
1        60    117     479.0
2        60    103     340.0
3        45    109     282.4
4        45    117     406.0
```

```
[17] #[14]
     df['Calories'] = df['Calories'].astype(int)
     print("\nDataFrame with 'Calories' as integer:")
     print(df.dtypes)
```

```
DataFrame with 'Calories' as integer:
Duration    int64
Pulse       int64
Calories    int64
dtype: object
```



```
#[15]
import matplotlib.pyplot as plt
plt.scatter(df['Duration'], df['Calories'])
plt.title('Scatter Plot of Duration vs Calories')
plt.xlabel('Duration')
plt.ylabel('Calories')
plt.show()
```

Github Repository link:-https://github.com/niharika0912/BDA.git
Youtube link:-https://youtu.be/mP6ziDaVOjQ