```python
import numpy as np
import pandas as pd

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)  # Ra
}

df = pd.DataFrame(data)
```

```python
[7] print(df.head(10))
```

```
   ID     Value Category
0   1  0.946539        A
1   2  0.316930        B
2   3  0.968075        B
3   4  0.837722        C
4   5  0.560785        A
5   6  0.102848        B
6   7  0.777172        A
7   8  0.683831        B
8   9  0.673348        A
9  10  0.073294        C
```

```
[8] value_column = df['Value']
    print(value_column.head())  # Displaying the first 5 values for brevity
```

```
0    0.946539
1    0.316930
2    0.968075
3    0.837722
4    0.560785
Name: Value, dtype: float64
```

```
df.columns = ['ID number', 'Random value', 'Choice']
print(df.head(5))
```

```
   ID number  Random value Choice
0          1      0.946539      A
1          2      0.316930      B
2          3      0.968075      B
3          4      0.837722      C
4          5      0.560785      A
```

```
[33] filepath = '/content/drive/My Drive/data/data.csv'
     data_frame = pd.read_csv(filepath)
     stats_summary = data_frame.describe()
     print(stats_summary)
```

```
         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
```

```python
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 Hintc
    'date_Of_Birth': ['15/05/2002', '17/05/2002', '16/02/1999', '25/09/19
    'age': [12, 12, 13, 14, 12, 13],
    'height': [173, 192, 186, 167, 151, 159],
    'weight': [35, 32, 33, 30, 31, 32],
    'address': ['street1', 'street2', 'street3', 'street1', 'street2', 'st
}, index=['S1', 'S2', 'S3', 'S4', 'S5', 'S6'])

print("Original DataFrame:")
print(student_data)

print('\nSplit the data based on school_code and class:')
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   14     167      30
S5        s002     V    Gino Mcneill    11/05/2002   12     151      31
S6        s004    VI    David Parkes    15/09/1997   13     159      32
```

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

Split the data based on school_code and class:

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  addr
S4        s001    VI  Eesha Hinton    25/09/1998   14     167      30  stre

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

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

Group:
('s004', 'VI')
   school_code class          name date_Of_Birth  age  height  weight  addr
S6        s004    VI  David Parkes    15/09/1997   13     159      32  stre
```

```
[14] from google.colab import drive
     drive.mount('/content/drive')
```

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

```
filepath = '/content/drive/My Drive/data/data.csv'
with open(filepath, 'r') as file:
    content = file.read()
    print(content)
```

```
Duration,Pulse,Maxpulse,Calories
60,110,130,409.1
60,117,145,479.0
60,103,135,340.0
45,109,175,282.4
45,117,148,406.0
60,102,127,300.0
60,110,136,374.0
45,104,134,253.3
30,109,133,195.1
60,98,124,269.0
60,103,147,329.3
60,100,120,250.7
60,106,128,345.3
60,104,132,379.3
60,98,123,275.0
60,98,120,215.2
60,100,120,300.0
45,90,112,
60,103,123,323.0
45,97,125,243.0
60,108,131,364.2
45,100,119,282.0
60,130,101,300.0
```

```python
# Step 1: Mount Google Drive
from google.colab import drive
drive.mount('/content/drive')

# Step 2: Import pandas
import pandas as pd

# Step 3: Load the CSV file into a DataFrame named 'data_frame'
filepath = '/content/drive/My Drive/data/data.csv'
data_frame = pd.read_csv(filepath)

# Step 4: Check for null values
null_counts = data_frame.isnull().sum()
print("Null values in each column:\n", null_counts)

# Step 5: Replace null values with the mean of each column
data_frame_filled = data_frame.fillna(data_frame.mean())

# Step 6: Verify that there are no more null values
null_counts_after = data_frame_filled.isnull().sum()
print("\nNull values after filling with mean:\n", null_counts_after)
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, c
Null values in each column:
 Duration    0
Pulse        0
Maxpulse     0
Calories     5
dtype: int64

Null values after filling with mean:
 Duration    0
Pulse        0
Maxpulse     0
Calories     0
dtype: int64
```

```python
# Step 1: Mount Google Drive
from google.colab import drive
drive.mount('/content/drive')

# Step 2: Import pandas
import pandas as pd

# Step 3: Load the CSV file into a DataFrame named 'data_frame'
filepath = '/content/drive/My Drive/data/data.csv'
data_frame = pd.read_csv(filepath)

# Step 4: Select at least two columns (e.g., 'Column1' and 'Column2')
selected_columns = data_frame[['Duration', 'Pulse']]

# Step 5: Aggregate the data using min, max, count, and mean
aggregated_data = selected_columns.agg(['min', 'max', 'count', 'mean'])

# Step 6: Display the aggregated results
print(aggregated_data)
```

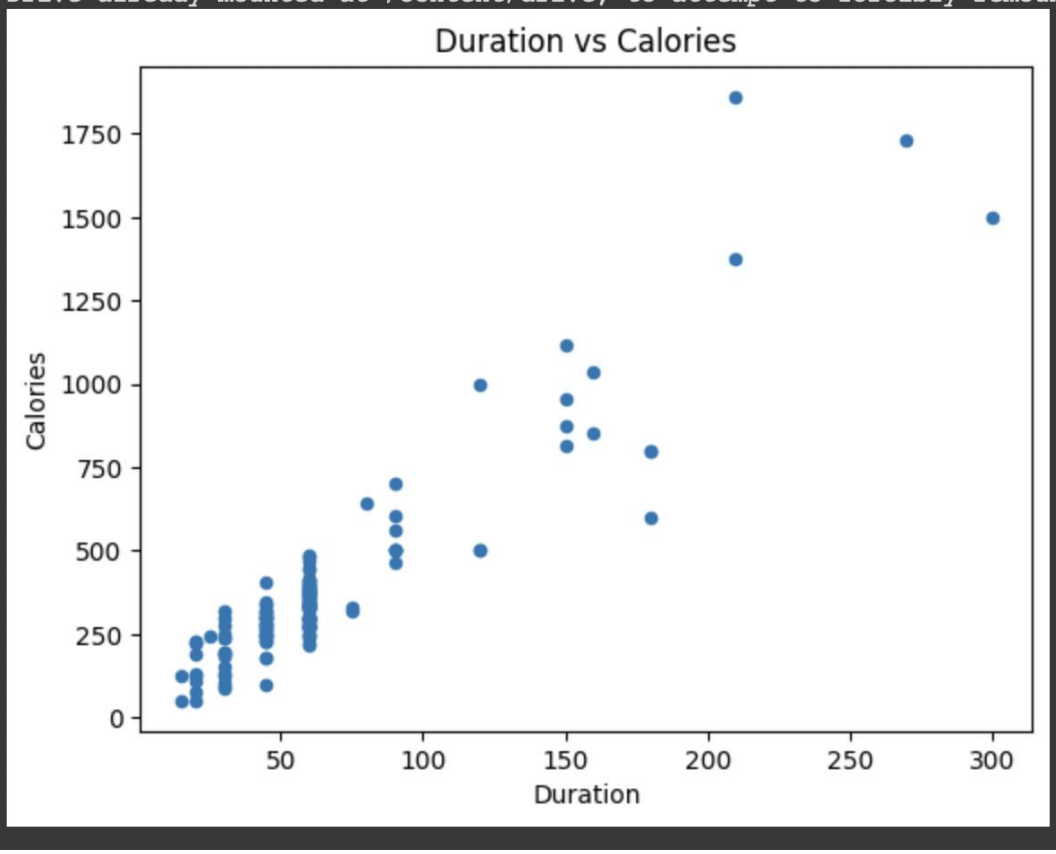
```
Drive already mounted at /content/drive; to attempt to forcibly remount, c
         Duration       Pulse
min     15.000000   80.000000
max    300.000000  159.000000
count  169.000000  169.000000
mean    63.846154  107.461538
```

```
[40] # Step 1: Mount Google Drive (if not done already)
     from google.colab import drive
     drive.mount('/content/drive')

     # Step 2: Import pandas
     import pandas as pd

     # Step 3: Load the CSV file into a DataFrame named 'data_frame'
     filepath = '/content/drive/My Drive/data/data.csv'
     data_frame = pd.read_csv(filepath)

     # Step 4: Create a new DataFrame 'df_modified' excluding the 'Maxpulse' c
     df_modified = data_frame.drop(columns=['Maxpulse'])

     # Step 5: Display the first few rows of the new DataFrame
     print(df_modified.head())
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, 
   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
```

```
# Step 1: Mount Google Drive (if not done already)
from google.colab import drive
drive.mount('/content/drive')

# Step 2: Import the necessary libraries
import pandas as pd
import matplotlib.pyplot as plt

# Step 3: Load the CSV file into a DataFrame named 'data_frame'
filepath = '/content/drive/My Drive/data/data.csv'
data_frame = pd.read_csv(filepath)

# Step 4: Create a scatter plot for "Duration" vs "Calories"
data_frame.plot.scatter(x='Duration', y='Calories', title='Duration vs Ca

# Step 5: Display the plot
plt.show()
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, c
```



```python
# Step 1: Mount Google Drive (if not done already)
from google.colab import drive
drive.mount('/content/drive')

# Step 2: Import pandas
import pandas as pd

# Step 3: Load the CSV file into a DataFrame named 'data_frame'
filepath = '/content/drive/My Drive/data/data.csv'
data_frame = pd.read_csv(filepath)

# Step 4: Filter the DataFrame to select rows with "Calories" between 500
filtered_data = data_frame[(data_frame['Calories'] >= 500) & (data_frame['

# Step 5: Display the filtered data
print(filtered_data)
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, c
     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
```

```python
# Step 1: Mount Google Drive (if not done already)
from google.colab import drive
drive.mount('/content/drive')

# Step 2: Import pandas
import pandas as pd

# Step 3: Load the CSV file into a DataFrame named 'data_frame'
filepath = '/content/drive/My Drive/data/data.csv'
data_frame = pd.read_csv(filepath)

# Step 4: Filter the DataFrame to select rows with "Calories" > 500 and "P
filtered_data = data_frame[(data_frame['Calories'] > 500) & (data_frame['P

# Step 5: Display the filtered data
print(filtered_data)
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, c
     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
```

```python
# Step 1: Mount Google Drive (if not done already)
from google.colab import drive
drive.mount('/content/drive')

# Step 2: Import pandas
import pandas as pd

# Step 3: Load the CSV file into a DataFrame named 'data_frame'
filepath = '/content/drive/My Drive/data/data.csv'
data_frame = pd.read_csv(filepath)

# Step 4: Delete the "Maxpulse" column from the main DataFrame
data_frame.drop(columns=['Maxpulse'], inplace=True)

# Step 5: Display the first few rows of the modified DataFrame to confirm
print(data_frame.head())
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount,
   Duration  Pulse  Calories
0        60    110     409.1
1        60    117     479.0
2        60    103     340.0
3        60    109     282.4
4        60    117     406.0
```

+ Code + Text

```python
# Step 1: Mount Google Drive (if not done already)
from google.colab import drive
drive.mount('/content/drive')

# Step 2: Import pandas
import pandas as pd

# Step 3: Load the CSV file into a DataFrame named 'data_frame'
filepath = '/content/drive/My Drive/data/data.csv'
data_frame = pd.read_csv(filepath)

# Step 4: Convert the datatype of the "Calories" column to int
data_frame['Calories'] = data_frame['Calories'].fillna(0).astype(int)

# Step 5: Verify the change by checking the datatype of the "Calories" co
print(data_frame.dtypes)
print(data_frame['Calories'].head())  # Display the first few rows of the
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount,
Duration    int64
Pulse       int64
Maxpulse    int64
Calories    int64
dtype: object
0    409
1    479
2    340
3    282
4    406
Name: Calories, dtype: int64
```

*Git hub Link:*https://github.com/shashank1615/BDA.git