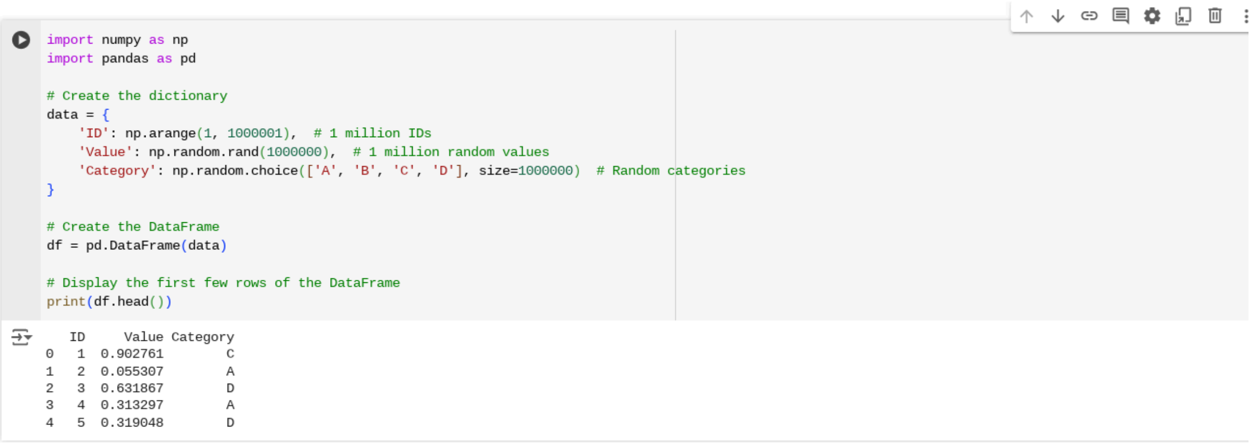
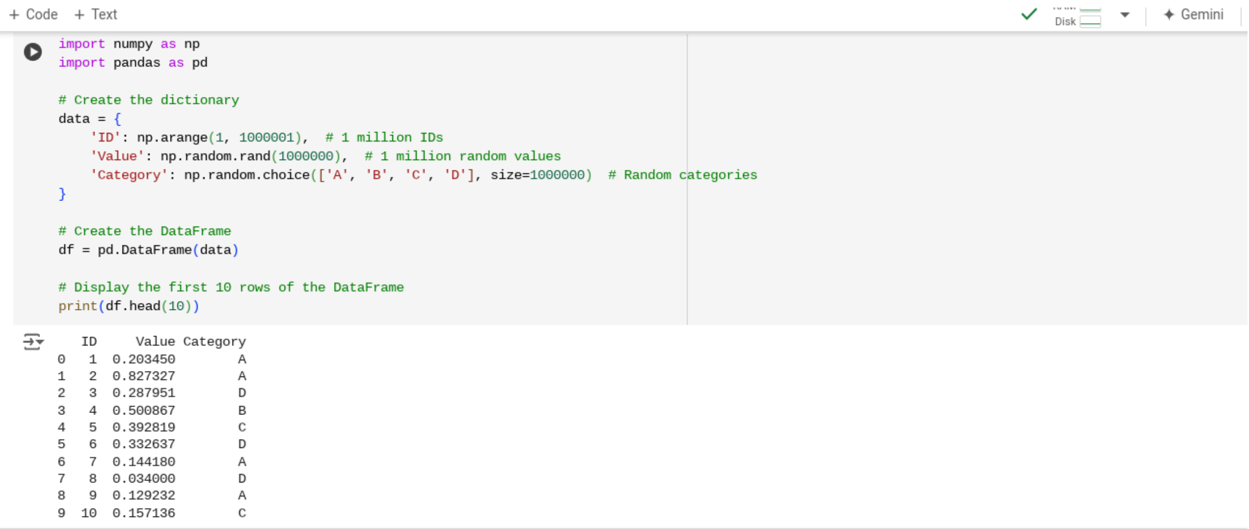
ICP3 REPORT



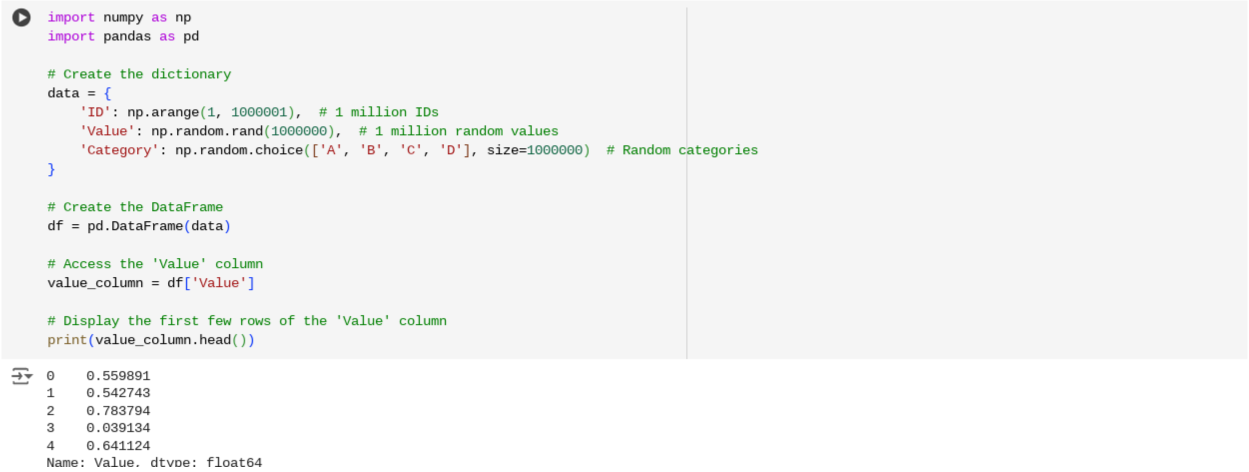
```python
import numpy as np
import pandas as pd

# Create the 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
}

# Create the DataFrame
df = pd.DataFrame(data)

# Display the first few rows of the DataFrame
print(df.head())
```

```
   ID     Value Category
0   1  0.902761        C
1   2  0.055307        A
2   3  0.631867        D
3   4  0.313297        A
4   5  0.319048        D
```



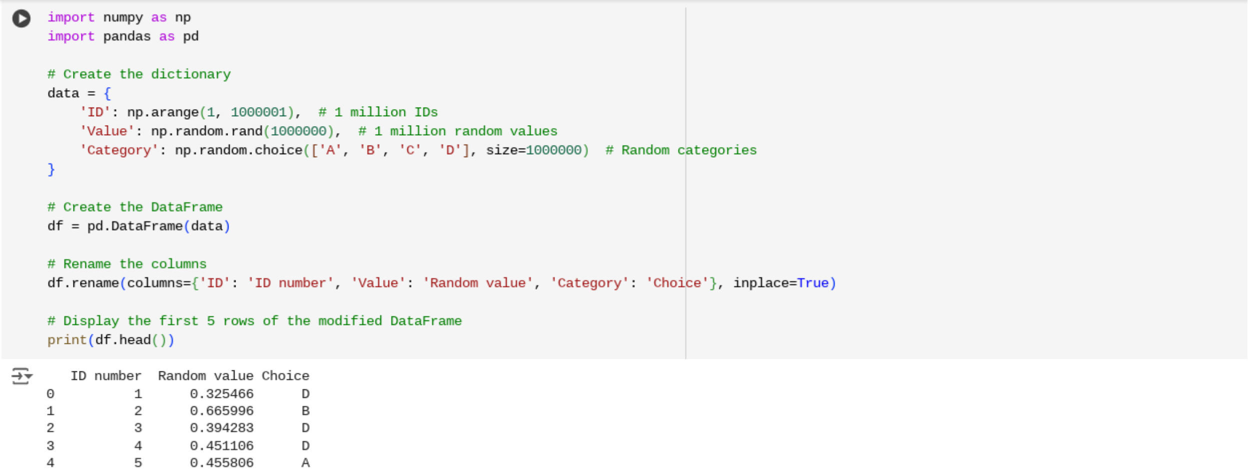
```python
import numpy as np
import pandas as pd

# Create the 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
}

# Create the DataFrame
df = pd.DataFrame(data)

# Display the first 10 rows of the DataFrame
print(df.head(10))
```

```
   ID     Value Category
0   1  0.203450        A
1   2  0.827327        A
2   3  0.287951        D
3   4  0.500867        B
4   5  0.392819        C
5   6  0.332637        D
6   7  0.144180        A
7   8  0.034000        D
8   9  0.129232        A
9  10  0.157136        C
```



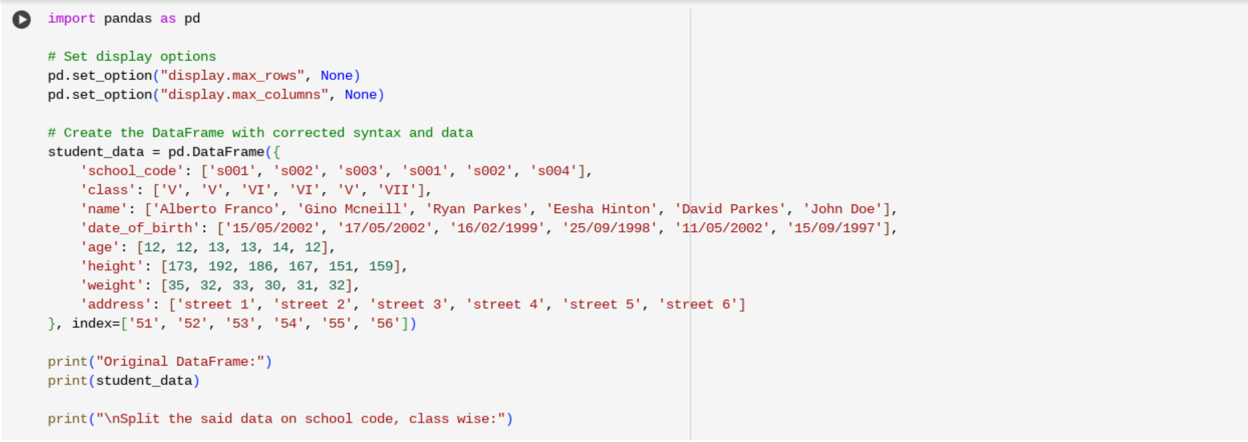
```python
import numpy as np
import pandas as pd

# Create the 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
}

# Create the DataFrame
df = pd.DataFrame(data)

# Access the 'Value' column
value_column = df['Value']

# Display the first few rows of the 'Value' column
print(value_column.head())
```

```
0    0.559891
1    0.542743
2    0.783794
3    0.039134
4    0.641124
Name: Value, dtype: float64
```

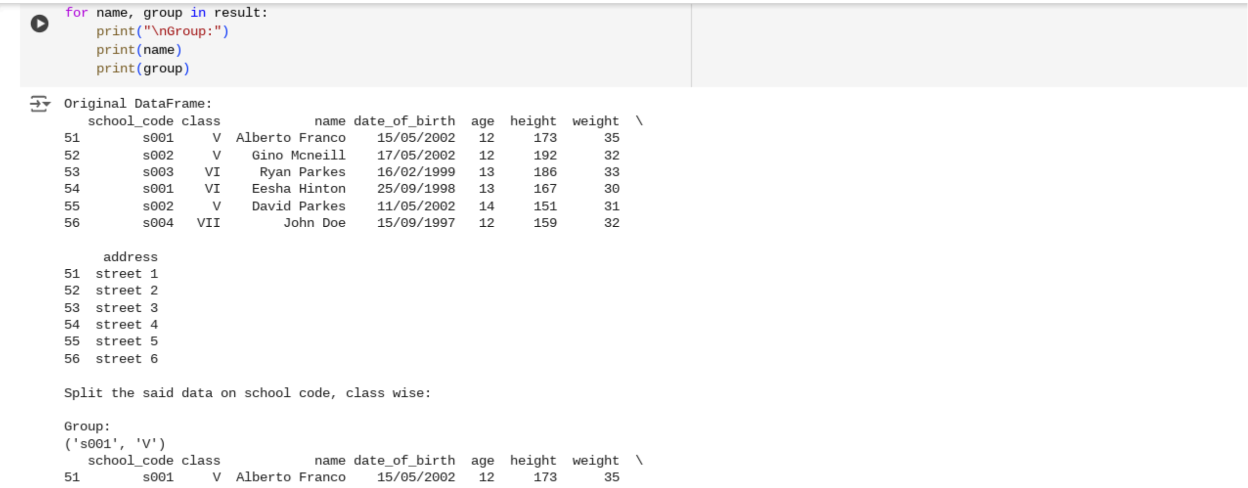


```
import numpy as np
import pandas as pd

# Create the 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
}

# Create the DataFrame
df = pd.DataFrame(data)

# Rename the columns
df.rename(columns={'ID': 'ID number', 'Value': 'Random value', 'Category': 'Choice'}, inplace=True)

# Display the first 5 rows of the modified DataFrame
print(df.head())
```

```
   ID number  Random value Choice
0          1      0.325466      D
1          2      0.665996      B
2          3      0.394283      D
3          4      0.451106      D
4          5      0.455806      A
```

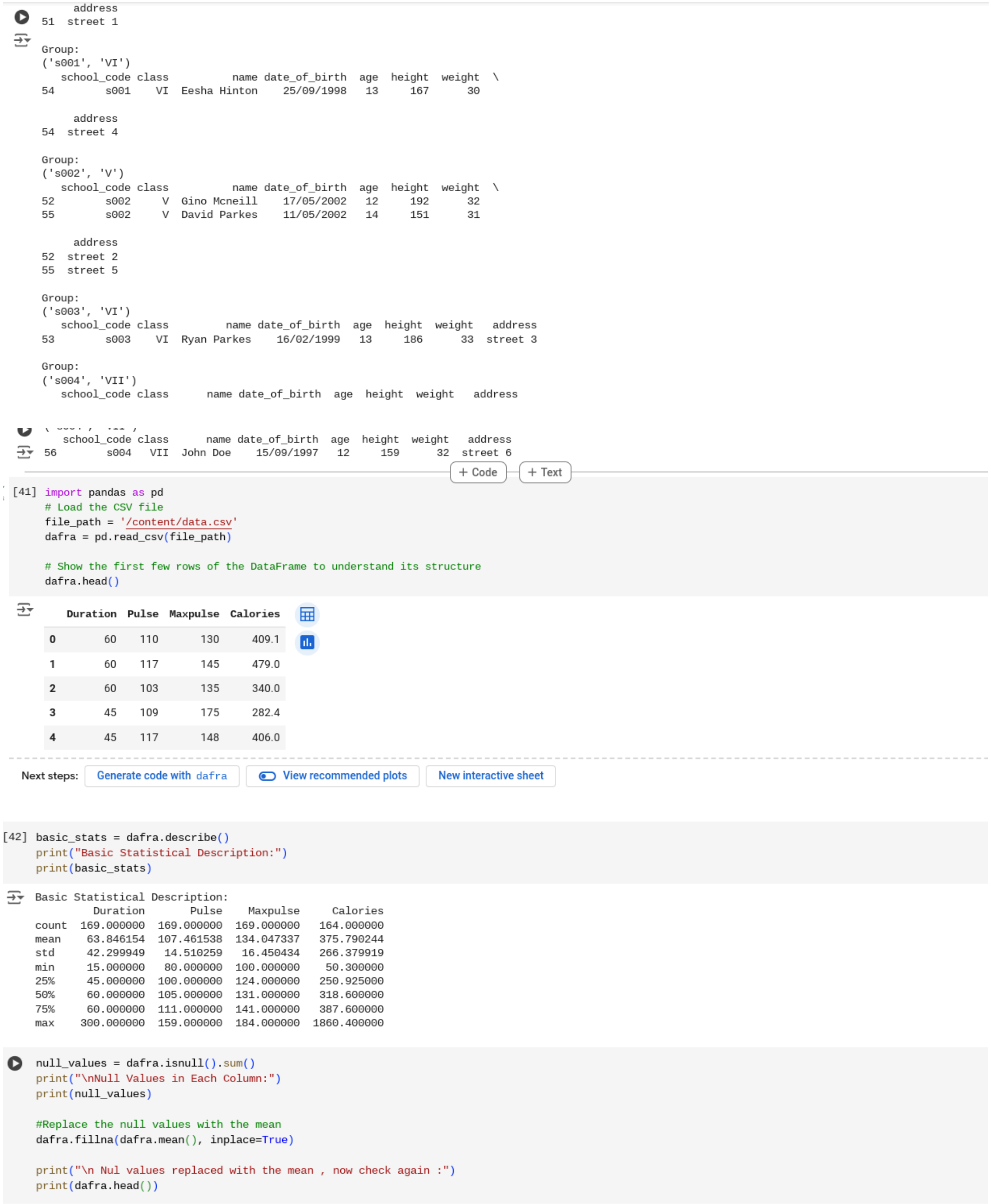


```
import pandas as pd

# Set display options
pd.set_option("display.max_rows", None)
pd.set_option("display.max_columns", None)

# Create the DataFrame with corrected syntax and data
student_data = pd.DataFrame({
    'school_code': ['s001', 's002', 's003', 's001', 's002', 's004'],
    'class': ['V', 'V', 'VI', 'VI', 'V', 'VII'],
    'name': ['Alberto Franco', 'Gino Mcneill', 'Ryan Parkes', 'Eesha Hinton', 'David Parkes', 'John Doe'],
    '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': ['street 1', 'street 2', 'street 3', 'street 4', 'street 5', 'street 6']
}, index=['51', '52', '53', '54', '55', '56'])

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

print("\nSplit the said data on school code, class wise:")
```



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

```
Original DataFrame:
   school_code class            name date_of_birth  age  height  weight  \
51        s001     V  Alberto Franco    15/05/2002   12     173      35   
52        s002     V    Gino Mcneill    17/05/2002   12     192      32   
53        s003    VI     Ryan Parkes    16/02/1999   13     186      33   
54        s001    VI    Eesha Hinton    25/09/1998   13     167      30   
55        s002     V    David Parkes    11/05/2002   14     151      31   
56        s004   VII        John Doe    15/09/1997   12     159      32   

     address  
51  street 1  
52  street 2  
53  street 3  
54  street 4  
55  street 5  
56  street 6  

Split the said data on school code, class wise:

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



```
     address
51  street 1

Group:
('s001', 'VI')
   school_code class          name date_of_birth  age  height  weight  \
54        s001    VI  Eesha Hinton    25/09/1998   13     167      30

     address
54  street 4

Group:
('s002', 'V')
   school_code class          name date_of_birth  age  height  weight  \
52        s002     V  Gino Mcneill    17/05/2002   12     192      32
55        s002     V  David Parkes    11/05/2002   14     151      31

     address
52  street 2
55  street 5

Group:
('s003', 'VI')
   school_code class         name date_of_birth  age  height  weight   address
53        s003    VI  Ryan Parkes    16/02/1999   13     186      33  street 3

Group:
('s004', 'VII')
   school_code class      name date_of_birth  age  height  weight   address
```

```
   school_code class      name date_of_birth  age  height  weight   address
56        s004   VII  John Doe    15/09/1997   12     159      32  street 6
```

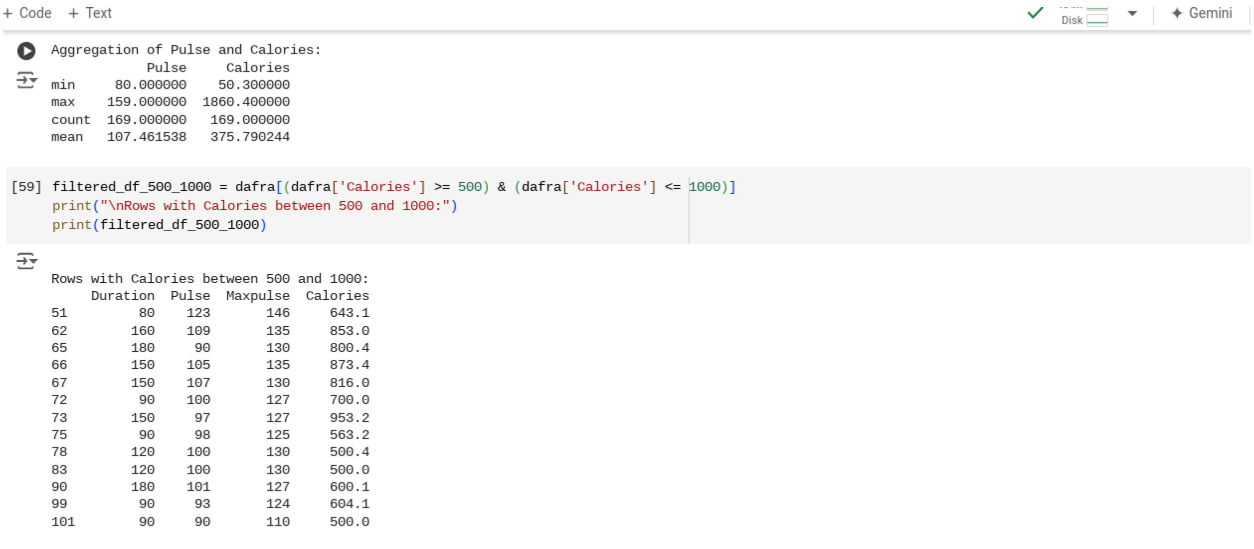
```
[41] import pandas as pd
     # Load the CSV file
     file_path = '/content/data.csv'
     dafra = pd.read_csv(file_path)

     # Show the first few rows of the DataFrame to understand its structure
     dafra.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

```
[42] basic_stats = dafra.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
```

```
null_values = dafra.isnull().sum()
print("\nNull Values in Each Column:")
print(null_values)

#Replace the null values with the mean
dafra.fillna(dafra.mean(), inplace=True)

print("\n Nul values replaced with the mean , now check again :")
print(dafra.head())
```

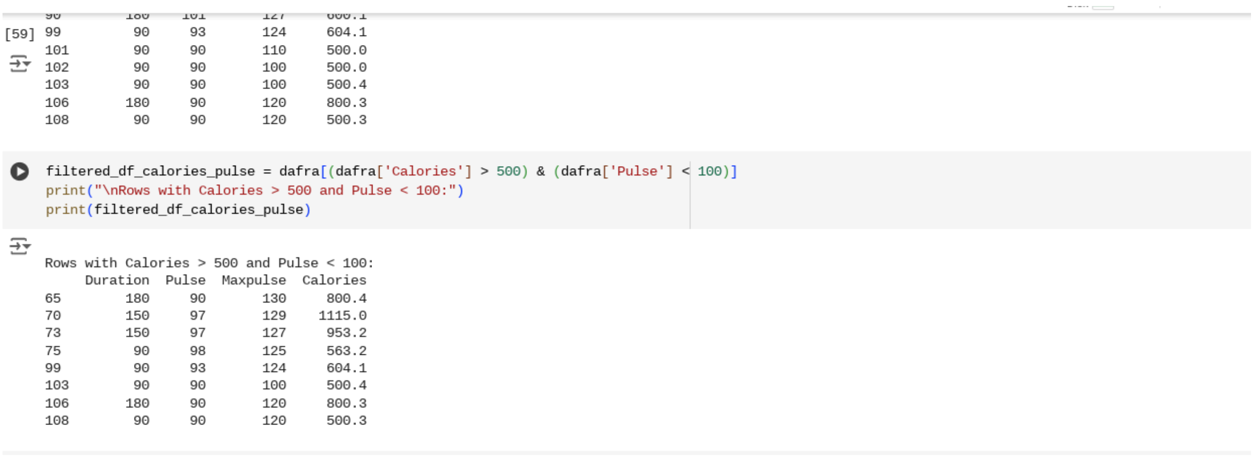


```
Aggregation of Pulse and Calories:
            Pulse     Calories
min     80.000000    50.300000
max    159.000000  1860.400000
count  169.000000   169.000000
mean   107.461538   375.790244
```

[59]

```
filtered_df_500_1000 = dafra[(dafra['Calories'] >= 500) & (dafra['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
```



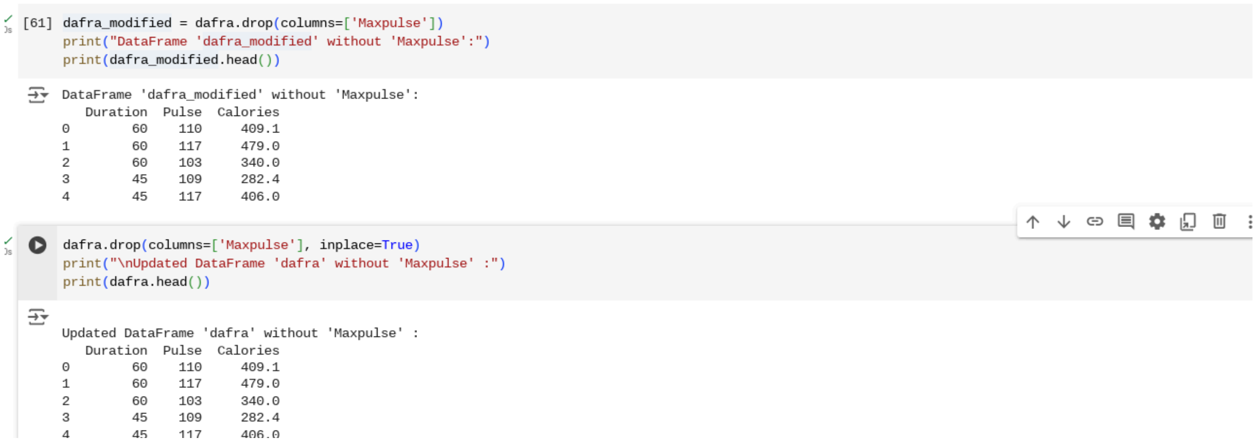
[59]

```
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
```

```
filtered_df_calories_pulse = dafra[(dafra['Calories'] > 500) & (dafra['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
```

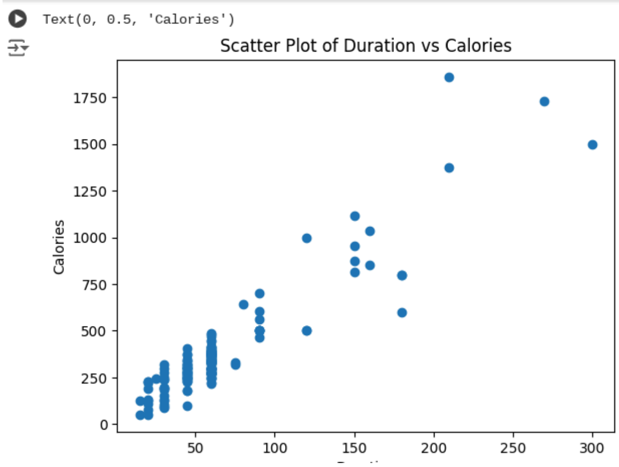


```
[61] dafra_modified = dafra.drop(columns=['Maxpulse'])
     print("DataFrame 'dafra_modified' without 'Maxpulse':")
     print(dafra_modified.head())
```

```
DataFrame 'dafra_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
```

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

```

Updated DataFrame 'dafra' 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
```



```
Text(0, 0.5, 'Calories')
```

My Github link :

https://github.com/Nitish300903/bda.git