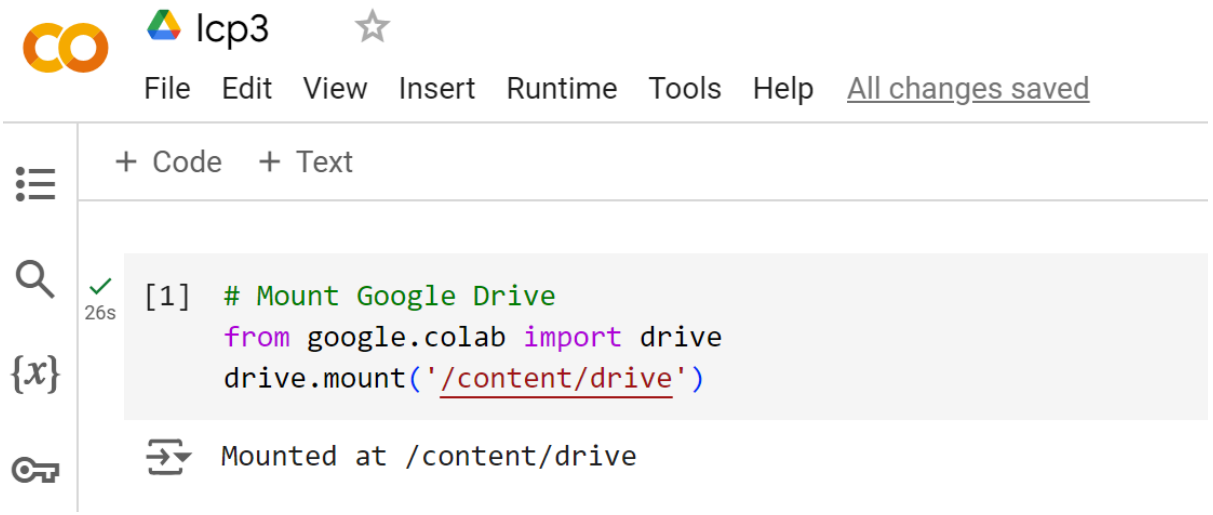
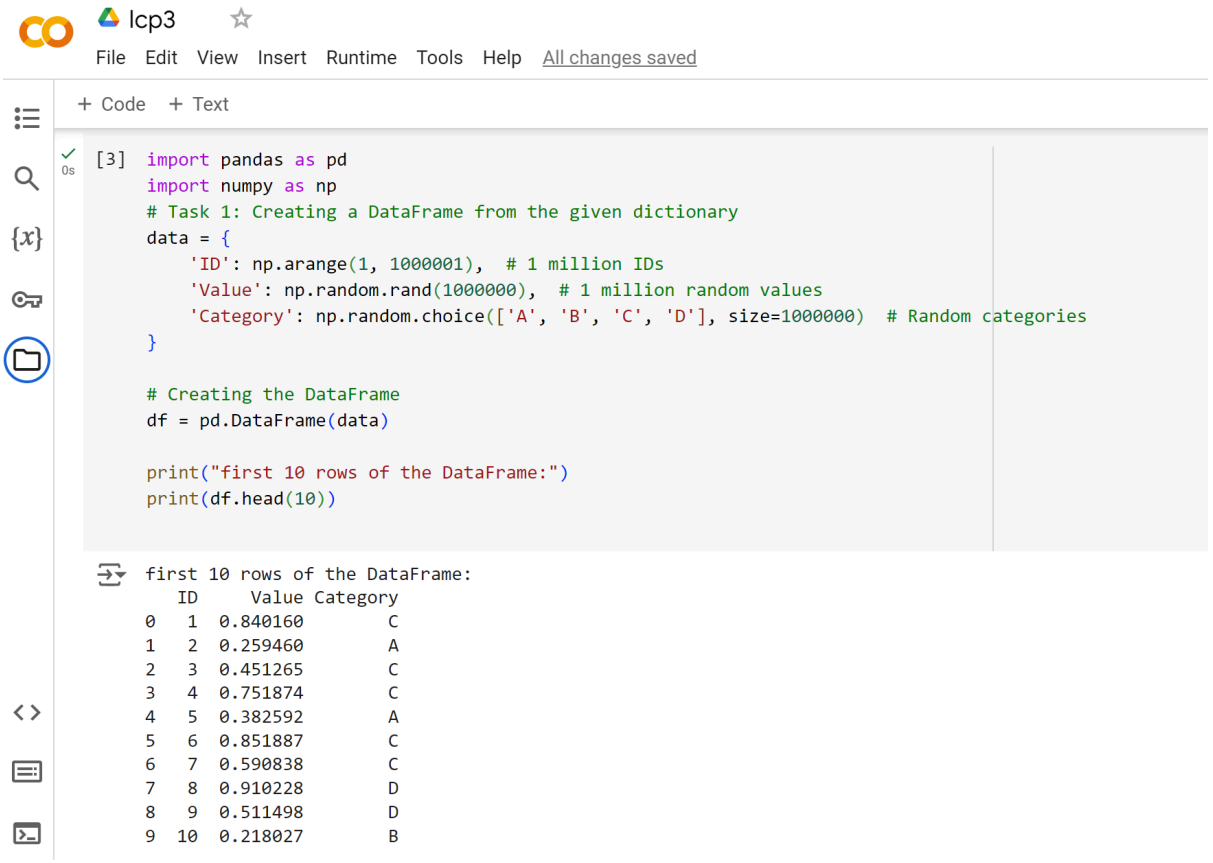
Archana srinivas
#700766576

ICP3





lcp3

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
print("\n 'Value' column:")
print(df['Value'])
```

```
 'Value' column:
0         0.840160
1         0.259460
2         0.451265
3         0.751874
4         0.382592
            ...
999995    0.600133
999996    0.967479
999997    0.003334
999998    0.373484
999999    0.425204
Name: Value, Length: 1000000, dtype: float64
```

lcp3

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
[5] df.columns = ['ID number','random value','Choice']
    print("\n First 5 rows after modifying column names:")
    print(df.head(5))
```

```
 First 5 rows after modifying column names:
   ID number  random value Choice
0          1      0.840160      C
1          2      0.259460      A
2          3      0.451265      C
3          4      0.751874      C
4          5      0.382592      A
```

lcp3

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```python
import pandas as pd

# Set display options for showing full rows and columns
pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)

# Create DataFrame with school and student data
student_data = pd.DataFrame({
    'school_code': ['s001', 's002', 's003', 's001', 's002', 's004'],
    'class': ['V', 'V', 'VI', 'VI', 'VI', '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, 14, 12, 13],
    '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'])

# Display the original DataFrame
print("Original DataFrame:")
print(student_data)

# Group the data by school_code and class
print("\nSplit the said data on school_code, class wise:")
result = student_data.groupby(['school_code', 'class'])

# Display each group
```

lcp3

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```python
# Display each group
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    VI    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 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
```

lcp3

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
   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   14     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

Group:
('s002', 'VI')
   school_code class          name Date_Of_Birth  age  height  weight  address
S5        s002    VI  Gino Mcneill    11/05/2002   12     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   13     159      32  street4
```

lcp3

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
[7] import pandas as pd

    # Load the CSV file
    file_path = '/content/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

lcp3

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
[8]  # Task 1: Show the basic statistical description about the data
     print("Basic Statistical Description of the Data:")
     print(df.describe())
```

```
Basic Statistical Description of the Data:
          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
```

lcp3

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
# Task 2: Check if the data has null values
print("\nChecking for null values:")
print(df.isnull().sum())

# Task 2a: Replace the null values with the mean
df.fillna(df.mean(), inplace=True)
print("\nNull values replaced with mean, check again:")
print(df.head())
```

```
Checking for null values:
Duration    0
Pulse       0
Calories    0
dtype: int64

Null values replaced with mean, check again:
   Duration  Pulse  Calories
0        60    110       409
1        60    117       479
2        60    103       340
3        45    109       282
4        45    117       406
```

lcp3

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
[10] # Task 3: Select at least two columns and aggregate the data using: min, max, count, mean
     print("\nAggregating 'Duration' and 'Calories' columns:")
     aggregated_data = df[['Duration', 'Calories']].agg(['min', 'max', 'count', 'mean'])
     print(aggregated_data)
```

```
Aggregating 'Duration' and 'Calories' columns:
          Duration     Calories
min      15.000000    50.300000
max     300.000000  1860.400000
count   169.000000   169.000000
mean     63.846154   375.790244
```

lcp3

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
# Task 4: Filter the dataframe to select rows with calories between 500 and 1000
print("\nRows with Calories between 500 and 1000:")
filtered_df = df[(df['Calories'] >= 500) & (df['Calories'] <= 1000)]
print(filtered_df)
```

```
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
```

lcp3

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
[12] # Task 5: Filter the dataframe to select rows with calories > 500 and pulse < 100
     print("\nRows with Calories > 500 and Pulse < 100:")
     filtered_df_2 = df[(df['Calories'] > 500) & (df['Pulse'] < 100)]
     print(filtered_df_2)
```

```
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
```

lcp3

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
# Task 6: Create a new 'df_modified' dataframe excluding the 'Maxpulse' column
df_modified = df.drop(columns=['Maxpulse'])
print("\nDataframe 'df_modified' without 'Maxpulse' column:")
print(df_modified.head())
```

```
Dataframe 'df_modified' without 'Maxpulse' column:
   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
```

My youtube link:https://youtu.be/gVx6NCj4Ta8
My GitHub link:https://youtu.be/gVx6NCj4Ta8

lcp3

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
# Task 7: Delete the 'Maxpulse' column from the main df dataframe
df.drop(columns=['Maxpulse'], inplace=True)
print("\nOriginal dataframe after deleting 'Maxpulse':")
print(df.head())
```

```
Original dataframe after deleting '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
```

lcp3

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

0s

```
# Task 8: Convert the datatype of 'Calories' column to int datatype
df['Calories'] = df['Calories'].astype(int)
print("\n'Calories' column after converting to int datatype:")
print(df.dtypes)
```

```
'Calories' column after converting to int datatype:
Duration     int64
Pulse        int64
Calories     int64
dtype: object
```

lcp3

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

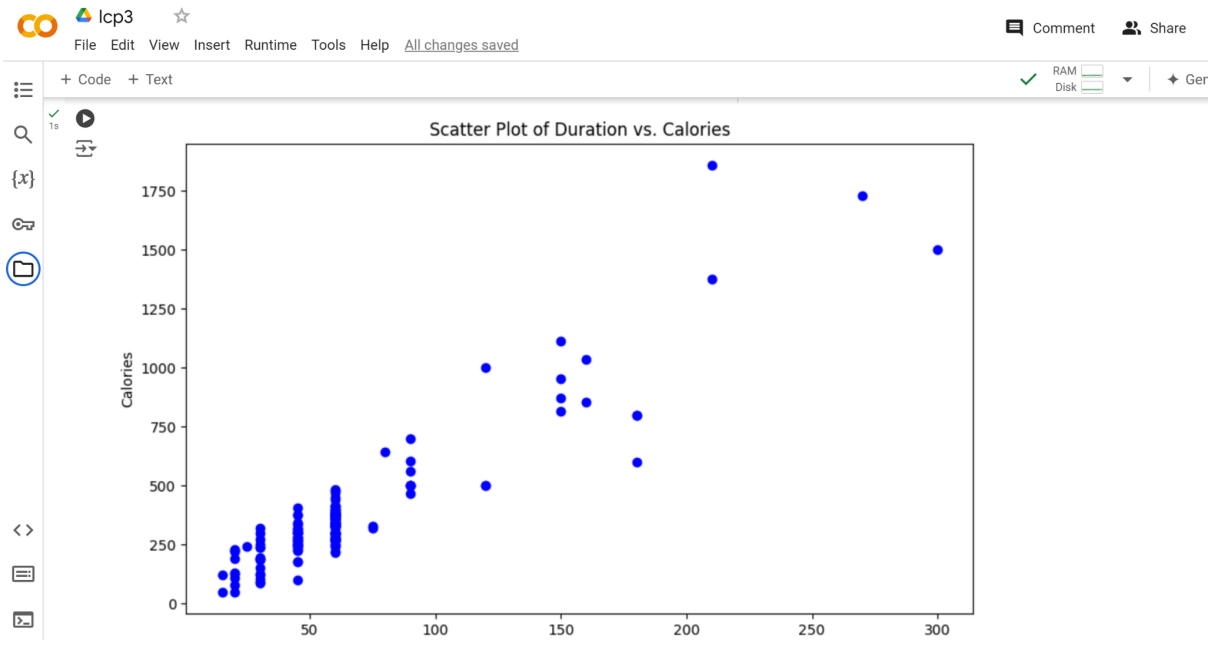
1s

```
# Task 9: Create a scatter plot for the columns 'Duration' and 'Calories'
import matplotlib.pyplot as plt # import the matplotlib library and assign it to the variable plt

plt.figure(figsize=(10, 6))
plt.scatter(df['Duration'], df['Calories'], color='blue')

# Add titles and labels
plt.title('Scatter Plot of Duration vs. Calories')
plt.xlabel('Duration')
plt.ylabel('Calories')

# Display the plot
plt.show()
```


lcp3
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code
+ Text
RAM
Disk
Scatter Plot of Duration vs. Calories
Calories
0
250
500
750
1000
1250
1500
1750
50
100
150
200
250
300