

ICP_4 REPORT

```
import matplotlib.pyplot as plt
import gdown

# Download the data from Google Drive
url = 'https://drive.google.com/uc?id={}'.format('11Q19z_FwFOi1cbItNv1Herc2gH_g01k2')
df = pd.read_csv(url)

# Step 2: Basic Statistical Description
print(df.describe())

# Step 3: Check for Null Values and replace with mean
df.fillna(df.mean(), inplace=True)

# Step 4: Aggregate Data for "Duration" and "Calories"
agg_data = df[['Duration', 'Calories']].agg(['min', 'max', 'count', 'mean'])
print(agg_data)

# Step 5: Filter Data for calories between 500 and 1000
filtered_data_500_1000 = df[(df['Calories'] >= 500) & (df['Calories'] <= 1000)]
print(filtered_data_500_1000)

# Step 6: Filter Data for calories > 500 and pulse < 100
filtered_data_cal_pulse = df[(df['Calories'] > 500) & (df['Pulse'] < 100)]
print(filtered_data_cal_pulse)

# Step 7 & 8: Modify Dataframe
df_modified = df.drop(columns=['Maxpulse'])
df.drop(columns=['Maxpulse'], inplace=True)

# Step 9: Convert Datatype of Calories to int
df['Calories'] = df['Calories'].astype(int)

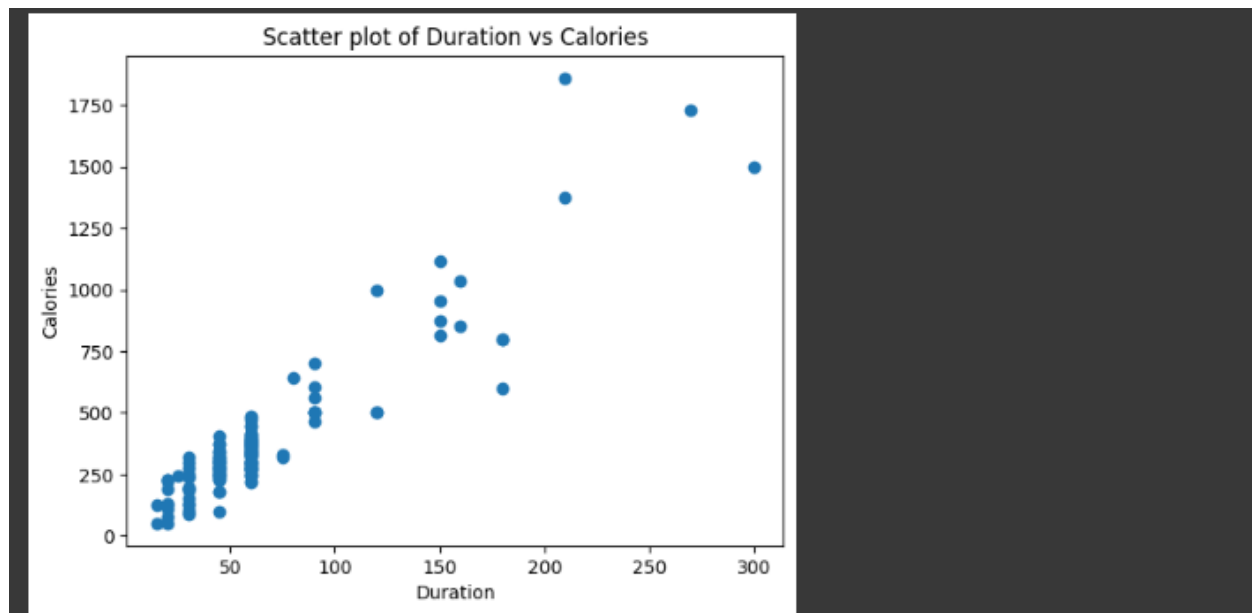
# Step 10: Scatter Plot for Duration and Calories
plt.scatter(df['Duration'], df['Calories'])
plt.xlabel('Duration')
plt.ylabel('Calories')
plt.title("Scatter plot of Duration vs Calories")
plt.show()
```

```
count    169.000000    169.000000    169.000000    164.000000
mean      63.846154    107.461538    134.047337    375.790244
std       42.239949     14.310259     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    164.000000    1860.400000

      Duration  Calories
min    15.000000    50.300000
max   300.000000   1860.400000
count    169.000000    169.000000
mean     63.846154    375.790244

      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

      Duration  Pulse  Maxpulse  Calories
65        180      90      120     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
102         90      90      100     500.4
106        180      90      120     800.3
108         90      90      120     500.3
```

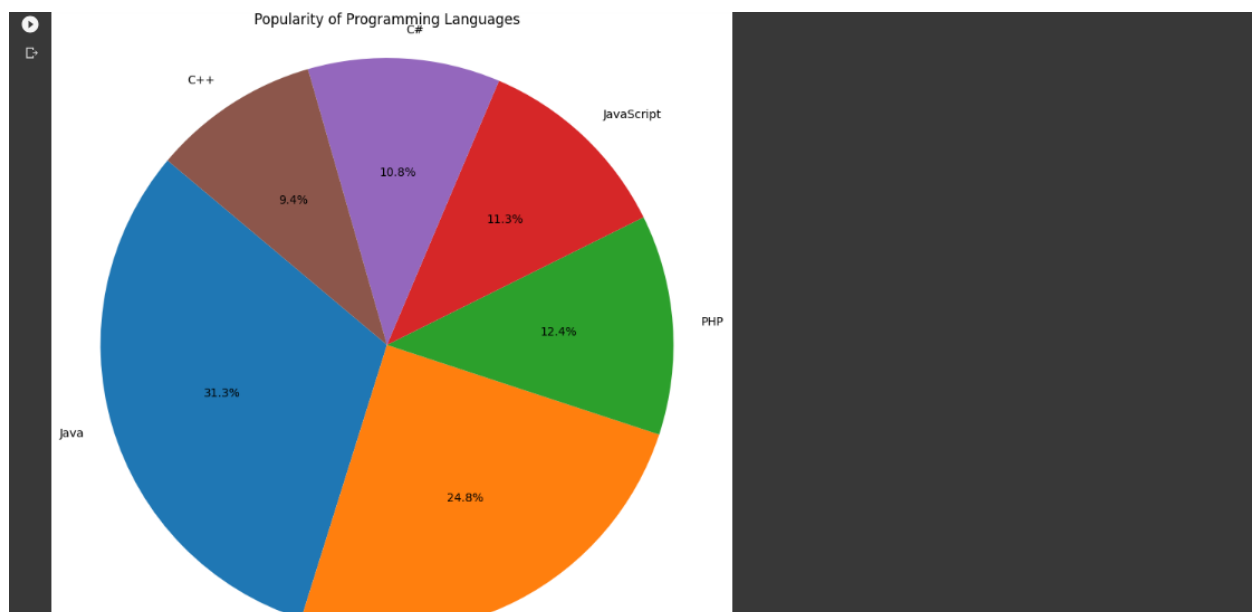


```
[ ] import matplotlib.pyplot as plt

# Sample data
languages = ["Java", "Python", "PHP", "JavaScript", "C#", "C++"]
popularity = [22.2, 17.6, 8.8, 8, 7.7, 6.7]

# Create a pie chart
plt.figure(figsize=(8, 8)) # Set the figure size as square for a better-looking pie chart
plt.pie(popularity, labels=languages, autopct='%1.1f%%', startangle=140)
plt.axis('equal') # Equal aspect ratio ensures that pie is drawn as a circle.
plt.title('Popularity of Programming Languages')

# Show the chart
plt.tight_layout()
plt.show()
```



```

import matplotlib.pyplot as plt

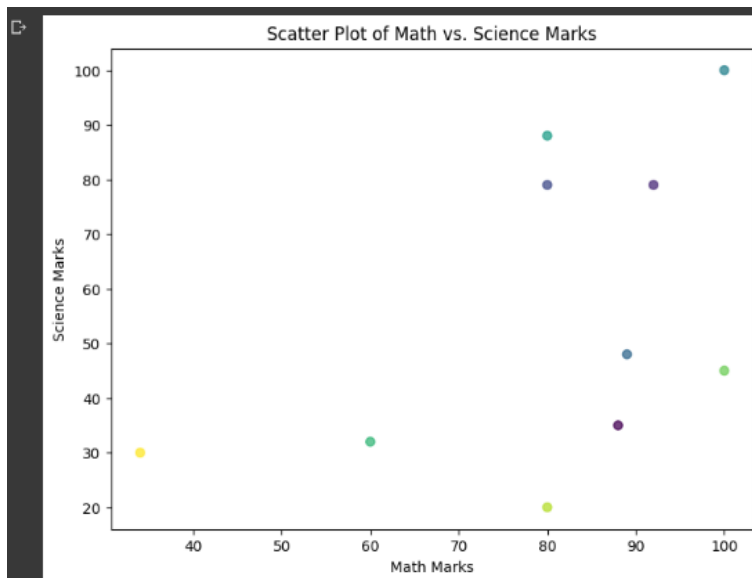
# Sample data
math_marks = [88, 92, 80, 89, 100, 80, 60, 100, 80, 34]
science_marks = [35, 79, 79, 48, 100, 88, 32, 45, 20, 30]
marks_range = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]

# Create a scatter plot
plt.figure(figsize=(8, 6))
plt.scatter(math_marks, science_marks, c=marks_range, cmap='viridis', marker='o', alpha=0.75)

plt.xlabel('Math Marks')
plt.ylabel('Science Marks')
plt.title('Scatter Plot of Math vs. Science Marks')

# Show the plot
plt.show()

```



Repository link: <https://github.com/sxk7912/Bigdata>

Youtube link: <https://youtu.be/xW30SkCSoeY>