LAB Assignment No. 1

Lab Assignment – Dataset Creation s Analysis

Objective

To learn how to create and upload a dataset in Python, perform basic statistical analysis, and visualize data using graphs.

Tasks

🔷 **Q1: Create a Dataset Manually**

- Create a dataset of at least **10 students** with the following columns:
    - Student_ID,
    - Name,
    - Age,
    - Marks_Math,
    - Marks_Science.
- Store the dataset in a **CSV file** named students.csv.

Code:

```
import pandas as pd
data = {
  'Student_ID': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
  'Name': ['Arooba', 'Azariya', 'salar', 'rohan', 'anaya', 'Bashir', 'Sana', 'saima', 'rida',
'saif'],
  'Age': [23, 78, 29, 15, 12, 14, 15, 19, 30, 28],
  'Marks_Math': [85, 25, 50, 58, 85, 90, 95, 75, 85, 75],
  'Marks_Science': [60, 80, 95, 50, 80, 75, 90, 70, 85, 90],
  'CGPA': [3.4, 3.8, 2.8, 3.5, 2.6, 3.5, 3.6, 2.4, 3.7, 3.4]
}
df = pd.DataFrame(data)
print(df)
```

Output:

	Student_ID	Name	Age	Marks_Math	Marks_Science	CGPA
0	1	Arooba	23	85	60	3.4
1	2	Azariya	78	25	80	3.8
2	3	salar	29	50	95	2.8
3	4	rohan	15	58	50	3.5
4	5	anaya	12	85	80	2.6
5	6	Bashir	14	90	75	3.5
6	7	Sana	15	95	90	3.6
7	8	saima	19	75	70	2.4
8	9	rida	30	85	85	3.7
9	10	saif	28	75	90	3.4

Q2: Upload Dataset in Python

- Use **Pandas** to load the dataset.

Code:

```
print(df.info())
print(df.head())
print(df['Marks_Math'].mean())
print(df['Marks_Science'].max())
```

Output:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Student_ID     10 non-null     int64
 1   Name           10 non-null     object
 2   Age            10 non-null     int64
 3   Marks_Math     10 non-null     int64
 4   Marks_Science  10 non-null     int64
 5   CGPA           10 non-null     float64
dtypes: float64(1), int64(4), object(1)
memory usage: 612.0+ bytes
None
   Student_ID     Name  Age  Marks_Math  Marks_Science  CGPA
0           1   Arooba   23          85             60   3.4
1           2  Azariya   78          25             80   3.8
2           3    salar   29          50             95   2.8
3           4    rohan   15          58             50   3.5
4           5    anaya   12          85             80   2.6
72.3
95
```

Q3: Observe Dataset Information

Run the following commands and explain the output:

1. data.info() → Dataset structure
2. data.describe() → Summary statistics (mean, std, min, max, etc.)
3. data['Marks_Math'].mean() → Mean of Math marks
4. data['Marks_Science'].max() → Maximum Science marks

Code:

```
print(df[df['Marks_Math'] > 50])
print(df.loc[df['Marks_Science'].idxmax()])
print(df['Marks_Math'].corr(df['Marks_Science']))
```

output:

```
   Student_ID    Name  Age  Marks_Math  Marks_Science  CGPA
0           1  Arooba   23          85             60   3.4
3           4   rohan   15          58             50   3.5
4           5   anaya   12          85             80   2.6
5           6  Bashir   14          90             75   3.5
6           7    Sana   15          95             90   3.6
7           8   saima   19          75             70   2.4
8           9    rida   30          85             85   3.7
9          10    saif   28          75             90   3.4
Student_ID           3
Name             salar
Age                 29
Marks_Math          50
Marks_Science       95
CGPA               2.8
Name: 2, dtype: object
0.015283907715545006
```

Q4: Perform Some Data Analysis

- Find how many students have Marks_Math > 50.
- Find the student with the **highest Science marks**.
- Calculate the **correlation** between Marks_Math and Marks_Science.

Code:

```
print(df[df['Marks_Math'] > 50])
print(df.loc[df['Marks_Science'].idxmax()])
print(df['Marks_Math'].corr(df['Marks_Science']))
```

Output:

```
   Student_ID    Name  Age  Marks_Math  Marks_Science  CGPA
0           1  Arooba   23          85             60   3.4
3           4   rohan   15          58             50   3.5
4           5   anaya   12          85             80   2.6
5           6  Bashir   14          90             75   3.5
6           7    Sana   15          95             90   3.6
7           8   saima   19          75             70   2.4
8           9    rida   30          85             85   3.7
9          10    saif   28          75             90   3.4
Student_ID           3
Name             salar
Age                 29
Marks_Math          50
Marks_Science       95
CGPA               2.8
Name: 2, dtype: object
0.015283907715545006
```

🔷 Q5: Data Visualization

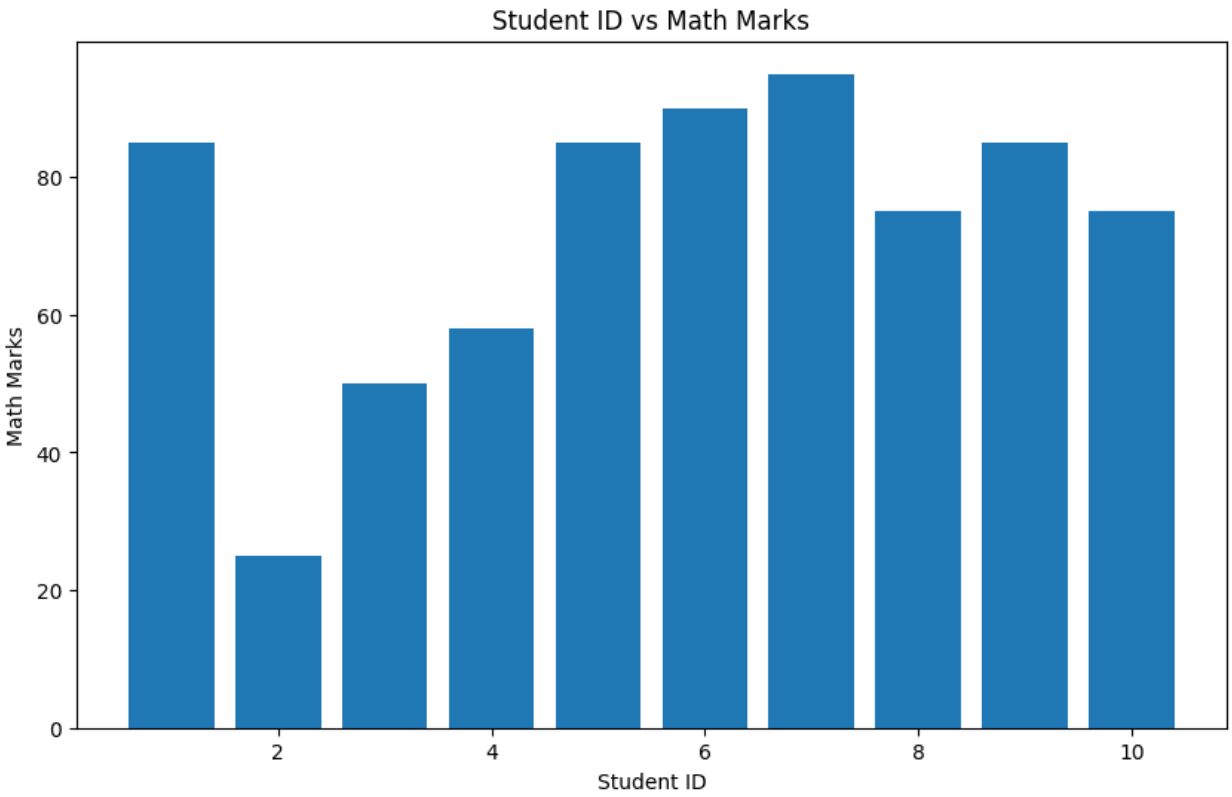
Use **Matplotlib/Seaborn** to create graphs:

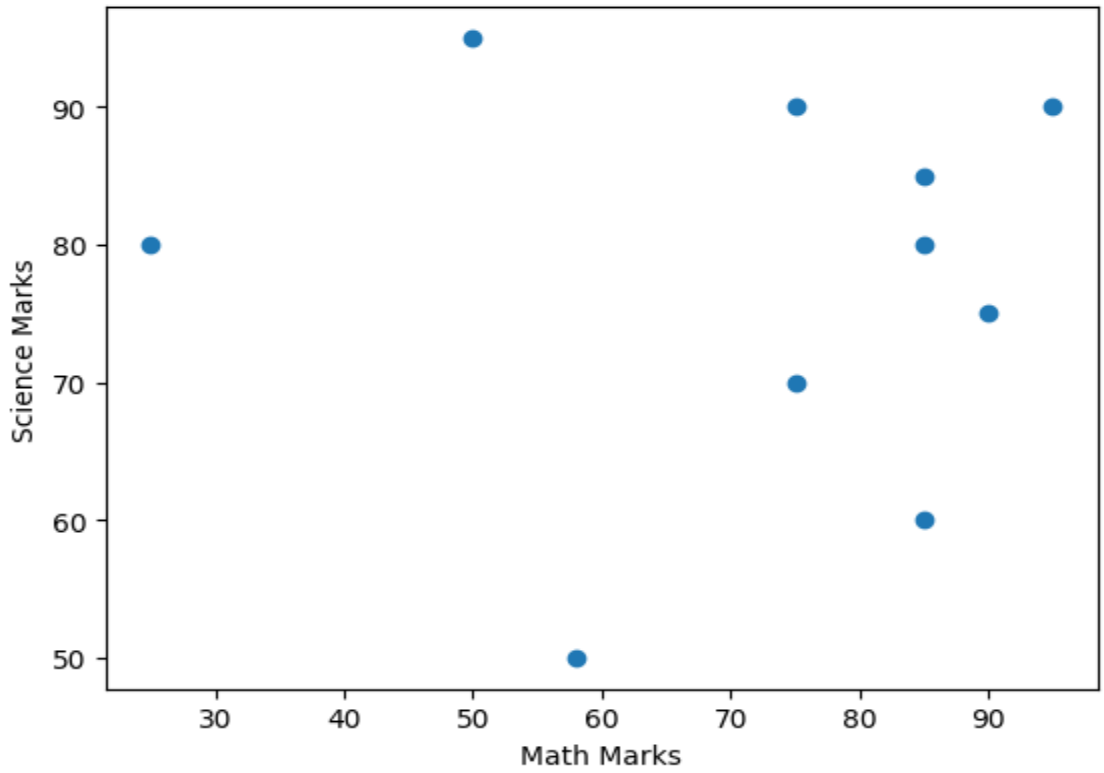
1. A bar chart of Student_ID vs Marks_Math.
2. A histogram of Age.
3. A scatter plot of Marks_Math vs Marks_Science.

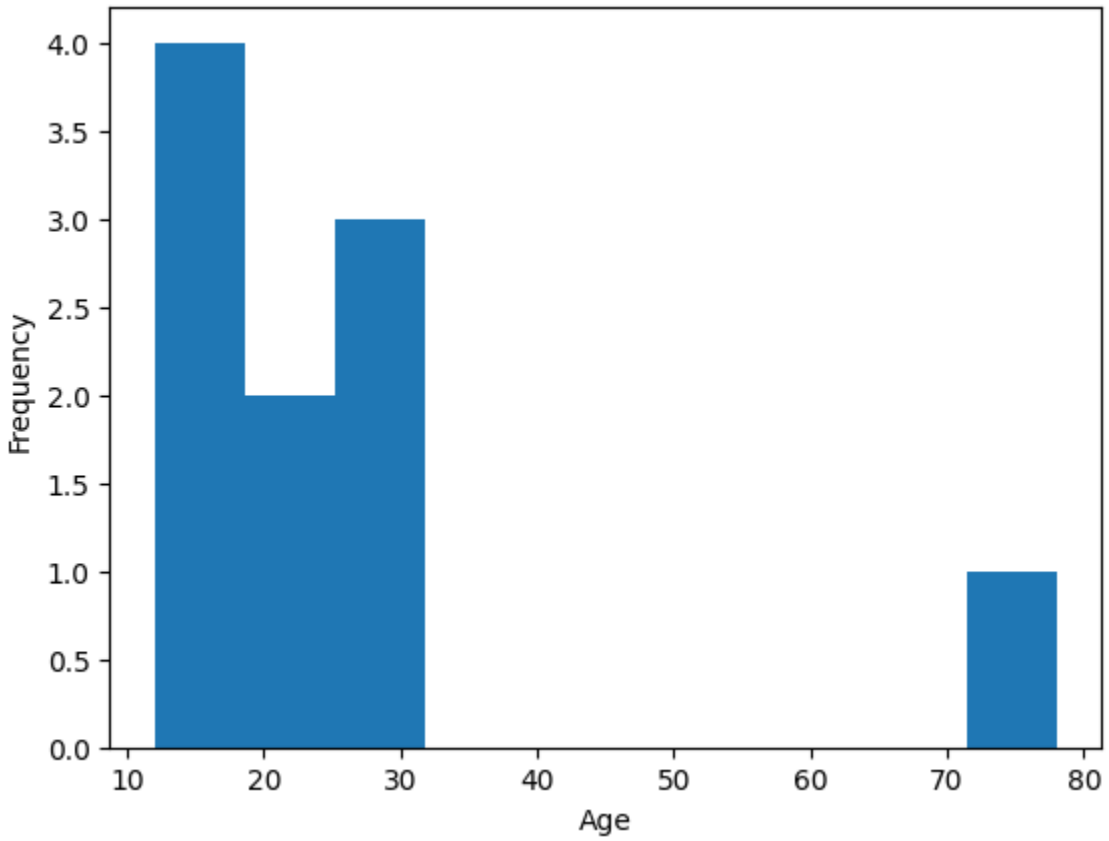
Code:

```
import matplotlib.pyplot as plt
plt.figure(figsize=(10,6))
plt.bar(df['Student_ID'], df['Marks_Math'])
plt.xlabel('Student ID')
plt.ylabel('Math Marks')
plt.title('Student ID vs Math Marks')
plt.show()
import matplotlib.pyplot as plt
plt.scatter(df['Marks_Math'], df['Marks_Science'])
plt.xlabel('Math Marks')
plt.ylabel('Science Marks')
plt.show()
plt.hist(df['Age'])
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
```

Output:


Student ID vs Math Marks
Math Marks
Student ID
0
20
40
60
80
2
4
6
8
10


Science Marks
Math Marks
50
60
70
80
90
30
40
50
60
70
80
90


4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Frequency
10
20
30
40
50
60
70
80
Age