

```
In [1]: import pandas as pd
df=pd.read_csv('student_dataset.csv') # reading the dataset
```

```
In [2]: df.info() # information about the dataset
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype
---  -
 0    ID      1000 non-null    int64
 1   Name     1000 non-null    object
 2   Age      1000 non-null    int64
 3  Marks    1000 non-null    int64
 4  Gender   1000 non-null    object
dtypes: int64(3), object(2)
memory usage: 39.2+ KB
```

```
In [3]: df.head() # displaying the first 5 rows of the dataset
```

```
Out[3]:
```

	ID	Name	Age	Marks	Gender
0	1	Student_1	24	94	Male
1	2	Student_2	21	81	Female
2	3	Student_3	22	90	Male
3	4	Student_4	24	56	Female
4	5	Student_5	20	44	Female

```
In [4]: df.tail() # displaying the last 5 rows of the dataset
```

```
Out[4]:
```

	ID	Name	Age	Marks	Gender
995	996	Student_996	23	82	Female
996	997	Student_997	18	73	Female
997	998	Student_998	24	96	Female
998	999	Student_999	21	65	Male
999	1000	Student_1000	23	80	Male

```
In [5]: df.describe() # statistical summary of the dataset
```

Out[5]:

	ID	Age	Marks
count	1000.000000	1000.0000	1000.000000
mean	500.500000	20.9600	70.232000
std	288.819436	2.0036	17.468638
min	1.000000	18.0000	40.000000
25%	250.750000	19.0000	56.000000
50%	500.500000	21.0000	70.000000
75%	750.250000	23.0000	86.000000
max	1000.000000	24.0000	100.000000

In [6]: `df.describe(include='all')` *# statistical summary of all columns including categoricals*

Out[6]:

	ID	Name	Age	Marks	Gender
count	1000.000000	1000	1000.0000	1000.000000	1000
unique	NaN	1000	NaN	NaN	2
top	NaN	Student_1	NaN	NaN	Female
freq	NaN	1	NaN	NaN	503
mean	500.500000	NaN	20.9600	70.232000	NaN
std	288.819436	NaN	2.0036	17.468638	NaN
min	1.000000	NaN	18.0000	40.000000	NaN
25%	250.750000	NaN	19.0000	56.000000	NaN
50%	500.500000	NaN	21.0000	70.000000	NaN
75%	750.250000	NaN	23.0000	86.000000	NaN
max	1000.000000	NaN	24.0000	100.000000	NaN

In [7]: `df.shape` *# shape of the dataset (rows, columns)*

Out[7]: (1000, 5)

In [8]: `df.dtypes` *# data types of each column*

Out[8]: ID int64
 Name object
 Age int64
 Marks int64
 Gender object
 dtype: object

In [9]: `df.columns` *# list of column names*

Out[9]: Index(['ID', 'Name', 'Age', 'Marks', 'Gender'], dtype='object')

```
In [10]: df['Age'] # accessing a specific column
```

```
Out[10]: 0      24
          1      21
          2      22
          3      24
          4      20
          ..
        995     23
        996     18
        997     24
        998     21
        999     23
        Name: Age, Length: 1000, dtype: int64
```

```
In [11]: df['Marks'].mean() # mean of the 'Marks' column
```

```
Out[11]: np.float64(70.232)
```

```
In [12]: df['ID'].min() # minimum value in the 'ID' column
```

```
Out[12]: np.int64(1)
```

```
In [13]: df['Marks'].max() # maximum value in the 'Marks' column
```

```
Out[13]: np.int64(100)
```

```
In [14]: df['Age'].median() # median of the 'Age' column
```

```
Out[14]: np.float64(21.0)
```

```
In [15]: df['Marks'].value_counts() # count of unique values in the 'ID' column
```

```
Out[15]: Marks
        65     28
        90     27
        86     26
        76     23
        79     23
          ..
        80     10
        66     10
        97     10
        55      9
        87      8
        Name: count, Length: 61, dtype: int64
```

```
In [16]: df['Age'].sum() # sum of all values in the 'Age' column
```

```
Out[16]: np.int64(20960)
```