

In [1]:

```
import pandas as pd
import numpy as np
df=pd.read_csv("Desktop/Mall_Customers.csv")
df
```

Out[1]:

	CustomerID	Genre	Age	Annual Income (k\$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40
...
195	196	Female	35	120	79
196	197	Female	45	126	28
197	198	Male	32	126	74
198	199	Male	32	137	18
199	200	Male	30	137	83

200 rows × 5 columns

In [2]:

```
df.mean()
```

C:\Users\OMPRAKASH KHANDALE\AppData\Local\Temp\ipykernel_16808\972437606.py:1: FutureWarning: The default value of numeric_only in DataFrame.mean is deprecated. In a future version, it will default to False. In addition, specifying 'numeric_only=None' is deprecated. Select only valid columns or specify the value of numeric_only to silence this warning.

```
df.mean()
```

Out[2]:

```
CustomerID          100.50
Age                 38.85
Annual Income (k$)   60.56
Spending Score (1-100)  50.20
dtype: float64
```

In [4]:

```
df.loc[:, 'Age'].mean()
```

Out[4]:

38.85

In [5]:

```
df.mean(axis=1)[0:4]
```

C:\Users\OMPRAKASH KHANDALE\AppData\Local\Temp\ipykernel_16808\1148177455.py:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError. Select only valid columns before calling the reduction.
df.mean(axis=1)[0:4]

Out[5]:

```
0    18.50  
1    29.75  
2    11.25  
3    30.00  
dtype: float64
```

In [6]:

```
df.median()
```

C:\Users\OMPRAKASH KHANDALE\AppData\Local\Temp\ipykernel_16808\530051474.py:1: FutureWarning: The default value of numeric_only in DataFrame.median is deprecated. In a future version, it will default to False. In addition, specifying 'numeric_only=None' is deprecated. Select only valid columns or specify the value of numeric_only to silence this warning.
df.median()

Out[6]:

```
CustomerID      100.5  
Age             36.0  
Annual Income (k$)  61.5  
Spending Score (1-100)  50.0  
dtype: float64
```

In [7]:

```
df.loc[:, 'Age'].median()
```

Out[7]:

```
36.0
```

In [8]:

```
df.loc[:, 'Age'].mode()
```

Out[8]:

```
0    32  
Name: Age, dtype: int64
```

In [9]:

```
df.max()
```

Out[9]:

```
CustomerID      200
Genre           Male
Age             70
Annual Income (k$)  137
Spending Score (1-100)  99
dtype: object
```

In [10]:

```
df.min()
```

Out[10]:

```
CustomerID      1
Genre           Female
Age             18
Annual Income (k$)  15
Spending Score (1-100)  1
dtype: object
```

In [11]:

```
df.loc[:, 'Age'].min()
```

Out[11]:

```
18
```

In [12]:

```
df.std()
```

```
C:\Users\OMPRAKASH KHANDALE\AppData\Local\Temp\ipykernel_16808\3390915376.py:1: FutureWarning: The default value of numeric_only in DataFrame.std is deprecated. In a future version, it will default to False. In addition, specifying 'numeric_only=None' is deprecated. Select only valid columns or specify the value of numeric_only to silence this warning.
```

```
df.std()
```

Out[12]:

```
CustomerID      57.879185
Age             13.969007
Annual Income (k$)  26.264721
Spending Score (1-100)  25.823522
dtype: float64
```

In [13]:

```
df.loc[:, 'Age'].std()
```

Out[13]:

```
13.969007331558883
```

In [14]:

```
df.var()
```

C:\Users\OMPRAKASH KHANDALE\AppData\Local\Temp\ipykernel_16808\1568254755.py:1: FutureWarning: The default value of numeric_only in DataFrame.var is deprecated. In a future version, it will default to False. In addition, specifying 'numeric_only=None' is deprecated. Select only valid columns or specify the value of numeric_only to silence this warning.

```
df.var()
```

Out[14]:

```
CustomerID      3350.000000
Age              195.133166
Annual Income (k$)  689.835578
Spending Score (1-100)  666.854271
dtype: float64
```

In [15]:

```
df.groupby(['Genre'])['Age'].mean()
```

Out[15]:

```
Genre
Female    38.098214
Male      39.806818
Name: Age, dtype: float64
```

In [29]:

```
df.groupby(['Genre'])['Annual Income (k$)'].mean()
```

Out[29]:

```
Genre
Female    59.250000
Male      62.227273
Name: Annual Income (k$), dtype: float64
```

In [35]:

```
from sklearn import preprocessing
pf=df['Genre'].unique()
pf
```

Out[35]:

```
array([1, 0])
```

In [37]:

```
rf = preprocessing.LabelEncoder()
df['Genre']=rf.fit_transform(df['Genre'])
df['Genre'].unique()
```

Out[37]:

```
array([1, 0], dtype=int64)
```

In []:

In [44]:

```
import pandas as pd
import numpy as np
df_iris=pd.read_csv("Desktop/Iris.csv")
df_iris
```

Out[44]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 6 columns

In [45]:

```
print('Iris-setosa')
setosa = df_iris['Species'] == 'Iris-setosa'
print(df_iris[setosa].describe())
```

Iris-setosa					
	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	50.00000	50.00000	50.000000	50.000000	50.00000
mean	25.50000	5.00600	3.418000	1.464000	0.24400
std	14.57738	0.35249	0.381024	0.173511	0.10721
min	1.00000	4.30000	2.300000	1.000000	0.10000
25%	13.25000	4.80000	3.125000	1.400000	0.20000
50%	25.50000	5.00000	3.400000	1.500000	0.20000
75%	37.75000	5.20000	3.675000	1.575000	0.30000
max	50.00000	5.80000	4.400000	1.900000	0.60000

In [48]:

```
print('Iris Versicolor')
versicolor = df_iris['Species'] == 'Iris-versicolor'
print(df_iris[versicolor].describe())
```

Iris Versicolor

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	50.00000	50.000000	50.000000	50.000000	50.000000
mean	75.50000	5.936000	2.770000	4.260000	1.326000
std	14.57738	0.516171	0.313798	0.469911	0.197753
min	51.00000	4.900000	2.000000	3.000000	1.000000
25%	63.25000	5.600000	2.525000	4.000000	1.200000
50%	75.50000	5.900000	2.800000	4.350000	1.300000
75%	87.75000	6.300000	3.000000	4.600000	1.500000
max	100.00000	7.000000	3.400000	5.100000	1.800000

In [49]:

```
print('Iris Verginica')
versicolor = df_iris['Species'] == 'Iris-verginica'
print(df_iris[versicolor].describe())
```

Iris Verginica

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	0.0	0.0	0.0	0.0	0.0
mean	NaN	NaN	NaN	NaN	NaN
std	NaN	NaN	NaN	NaN	NaN
min	NaN	NaN	NaN	NaN	NaN
25%	NaN	NaN	NaN	NaN	NaN
50%	NaN	NaN	NaN	NaN	NaN
75%	NaN	NaN	NaN	NaN	NaN
max	NaN	NaN	NaN	NaN	NaN

In [50]:

```
df_iris.loc[:, 'SepalLengthCm'].mean()
```

Out[50]:

5.843333333333334

In [52]:

```
df_iris.dtypes
```

Out[52]:

```
Id          int64
SepalLengthCm  float64
SepalWidthCm  float64
PetalLengthCm  float64
PetalWidthCm  float64
Species      object
dtype: object
```

In []:

In []: