

```
In [ ]: #Name: Atharv Santosh Danave
        #Roll no: 11
        #Practical no: 03
        #Academic year: 2024-25
```

```
In [1]: import pandas as pd
        import statistics as st
```

```
In [4]: df=pd.read_csv("Mall_Customers.csv")
```

```
In [5]: df
```

```
Out[5]:
```

	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 [59]: df.mean(numeric_only=True)
```

```
Out[59]: CustomerID          100.50
         Age                38.85
         Annual Income (k$)    60.56
         Spending Score (1-100) 50.20
         dtype: float64
```

```
In [11]: df.loc[:, 'Age'].mean()
```

```
Out[11]: 38.85
```

```
In [58]: df.mean(axis=1, numeric_only=True)[0:4]
```

```
Out[58]: 0    18.50
         1    29.75
         2    11.25
         3    30.00
         dtype: float64
```

```
In [57]: df.mean(axis=1, numeric_only=True)[0:4]
```

```
Out[57]: 0    18.50
         1    29.75
         2    11.25
         3    30.00
         dtype: float64
```

```
In [56]: df.median(numeric_only=True)
```

```
Out[56]: CustomerID      100.5
         Age             36.0
         Annual Income (k$)  61.5
         Spending Score (1-100)  50.0
         dtype: float64
```

```
In [21]: df.loc[:, 'Age'].median()
```

```
Out[21]: 36.0
```

```
In [55]: df.median(axis=1, numeric_only=True)[0:4]
```

```
Out[55]: 0    17.0
         1    18.0
         2    11.0
         3    19.5
         dtype: float64
```

```
In [20]: df.mode()
```

```
Out[20]:
```

	CustomerID	Genre	Age	Annual Income (k\$)	Spending Score (1-100)
0	1	Female	32.0	54.0	42.0
1	2	NaN	NaN	78.0	NaN
2	3	NaN	NaN	NaN	NaN
3	4	NaN	NaN	NaN	NaN
4	5	NaN	NaN	NaN	NaN
...
195	196	NaN	NaN	NaN	NaN
196	197	NaN	NaN	NaN	NaN
197	198	NaN	NaN	NaN	NaN
198	199	NaN	NaN	NaN	NaN
199	200	NaN	NaN	NaN	NaN

200 rows × 5 columns

```
In [23]: df.loc[:, 'Age'].mode()
```

```
Out[23]: 0    32
         dtype: int64
```

```
In [24]: df.min()
```

```
Out[24]: CustomerID      1
        Genre          Female
        Age            18
        Annual Income (k$)  15
        Spending Score (1-100)  1
        dtype: object
```

```
In [25]: df.loc[:, 'Age'].min(skipna = False)
```

```
Out[25]: 18
```

```
In [26]: df.max()
```

```
Out[26]: CustomerID      200
        Genre          Male
        Age            70
        Annual Income (k$)  137
        Spending Score (1-100)  99
        dtype: object
```

```
In [27]: df.loc[:, 'Age'].max(skipna = False)
```

```
Out[27]: 70
```

```
In [54]: df.std(numeric_only=True)
```

```
Out[54]: CustomerID      57.879185
        Age            13.969007
        Annual Income (k$)  26.264721
        Spending Score (1-100)  25.823522
        dtype: float64
```

```
In [29]: df.loc[:, 'Age'].std()
```

```
Out[29]: 13.96900733155888
```

```
In [60]: df.std(axis=1, numeric_only=True)[0:4]
```

```
Out[60]: 0      15.695010
        1      35.074920
        2       8.057088
        3     32.300671
        dtype: float64
```

```
In [31]: df.groupby(['Genre'])['Age'].mean()
```

```
Out[31]: Genre
        Female      38.098214
        Male       39.806818
        Name: Age, dtype: float64
```

```
In [42]: df_u = df.rename(columns={'Annual Income k$': 'Income'}, inplace=False)
```

```
In [53]: df_u.groupby('Genre')['Annual Income (k$)'].mean()
```

```
Out[53]: Genre
        Female      59.250000
        Male       62.227273
        Name: Annual Income (k$), dtype: float64
```

```
In [38]: print(df.columns)
```

```
Index(['CustomerID', 'Genre', 'Age', 'Annual Income (k$)',  
      'Spending Score (1-100)'],  
      dtype='object')
```

```
In [40]: from sklearn import preprocessing  
enc = preprocessing.OneHotEncoder()  
enc_df = pd.DataFrame(enc.fit_transform(df[['Genre']]).toarray())  
enc_df
```

```
Out[40]:
```

	0	1
0	0.0	1.0
1	0.0	1.0
2	1.0	0.0
3	1.0	0.0
4	1.0	0.0
...
195	1.0	0.0
196	1.0	0.0
197	0.0	1.0
198	0.0	1.0
199	0.0	1.0

200 rows × 2 columns

```
In [41]: df_encode = df_u.join(enc_df)  
df_encode
```

Out[41]:

	CustomerID	Genre	Age	Annual Income (k\$)	Spending Score (1-100)	0	1
0	1	Male	19	15	39	0.0	1.0
1	2	Male	21	15	81	0.0	1.0
2	3	Female	20	16	6	1.0	0.0
3	4	Female	23	16	77	1.0	0.0
4	5	Female	31	17	40	1.0	0.0
...
195	196	Female	35	120	79	1.0	0.0
196	197	Female	45	126	28	1.0	0.0
197	198	Male	32	126	74	0.0	1.0
198	199	Male	32	137	18	0.0	1.0
199	200	Male	30	137	83	0.0	1.0

200 rows × 7 columns