

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

```
In [4]: df=pd.read_csv("Q7.csv")
df.head()
```

```
Out[4]:
```

	Unnamed: 0	Points	Score	Weigh
0	Mazda RX4	3.90	2.620	16.46
1	Mazda RX4 Wag	3.90	2.875	17.02
2	Datsun 710	3.85	2.320	18.61
3	Hornet 4 Drive	3.08	3.215	19.44
4	Hornet Sportabout	3.15	3.440	17.02

```
In [5]: df.describe()
#gives most of the required values
```

```
Out[5]:
```

	Points	Score	Weigh
count	32.000000	32.000000	32.000000
mean	3.596563	3.217250	17.848750
std	0.534679	0.978457	1.786943
min	2.760000	1.513000	14.500000
25%	3.080000	2.581250	16.892500
50%	3.695000	3.325000	17.710000
75%	3.920000	3.610000	18.900000
max	4.930000	5.424000	22.900000

```
In [23]: round(df.mean(),1)
#avg points are 3.6, avg score is 3.2 and avg weight is 17.8
```

```
Out[23]: Points      3.6
Score        3.2
Weigh       17.8
dtype: float64
```

```
In [7]: df.mode()  
#there is nothing in common to give a mode
```

```
Out[7]:
```

	Unnamed: 0	Points	Score	Weigh
0	AMC Javelin	3.07	3.44	17.02
1	Cadillac Fleetwood	3.92	NaN	18.90
2	Camaro Z28	NaN	NaN	NaN
3	Chrysler Imperial	NaN	NaN	NaN
4	Datsun 710	NaN	NaN	NaN
5	Dodge Challenger	NaN	NaN	NaN
6	Duster 360	NaN	NaN	NaN
7	Ferrari Dino	NaN	NaN	NaN
8	Fiat 128	NaN	NaN	NaN
9	Fiat X1-9	NaN	NaN	NaN
10	Ford Pantera L	NaN	NaN	NaN
11	Honda Civic	NaN	NaN	NaN
12	Hornet 4 Drive	NaN	NaN	NaN
13	Hornet Sportabout	NaN	NaN	NaN
14	Lincoln Continental	NaN	NaN	NaN
15	Lotus Europa	NaN	NaN	NaN
16	Maserati Bora	NaN	NaN	NaN
17	Mazda RX4	NaN	NaN	NaN
18	Mazda RX4 Wag	NaN	NaN	NaN
19	Merc 230	NaN	NaN	NaN
20	Merc 240D	NaN	NaN	NaN
21	Merc 280	NaN	NaN	NaN
22	Merc 280C	NaN	NaN	NaN
23	Merc 450SE	NaN	NaN	NaN
24	Merc 450SL	NaN	NaN	NaN
25	Merc 450SLC	NaN	NaN	NaN
26	Pontiac Firebird	NaN	NaN	NaN
27	Porsche 914-2	NaN	NaN	NaN
28	Toyota Corolla	NaN	NaN	NaN
29	Toyota Corona	NaN	NaN	NaN
30	Valiant	NaN	NaN	NaN
31	Volvo 142E	NaN	NaN	NaN

```
In [25]: round(df.median(),1)
#almost similar values to mean it means that there is no outlier
```

```
Out[25]: Points      3.7
Score      3.3
Weigh     17.7
dtype: float64
```

```
In [28]: round(df.var(),2)
#we use variance to see how individual numbers relate to each other within a data
```

```
Out[28]: Points      0.29
Score      0.96
Weigh      3.19
dtype: float64
```

```
In [29]: round(df.std(),2)
#we use standard variance to see how close is it to the mean, close it is more reliable
#std of weight is the farthest that means that weight data has huge gaps between
```

```
Out[29]: Points      0.53
Score      0.98
Weigh      1.79
dtype: float64
```

```
In [20]: range1=df.Points.max()-df.Points.min()
range2=df.Score.max()-df.Score.min()
range3=df.Weigh.max()-df.Weigh.min()
print(range1,"\t",range2,"\t",range3)
#range isn't that reliable its better to use IQR instead for accurate results
#Like here weight is farthest from the real value
```

```
2.17      3.9110000000000005      8.399999999999999
```