

In [1]:

```
1 # Q7) Calculate Mean, Median, Mode, Variance, Standard Deviation, Range &  
2 # comment about the values / draw inferences, for the given dataset  
3 # - For Points,Score,Weigh>  
4 #     Find Mean, Median, Mode, Variance, Standard Deviation, and Range  
5 #     and also Comment about the values/ Draw some inferences.  
6  
7
```

In [2]:

```
1 import pandas as pd
```

In [36]:

```
1 car_data = pd.read_csv("Q7.csv")
2 car_data
```

Out[36]:

	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
5	Valiant	2.76	3.460	20.22
6	Duster 360	3.21	3.570	15.84
7	Merc 240D	3.69	3.190	20.00
8	Merc 230	3.92	3.150	22.90
9	Merc 280	3.92	3.440	18.30
10	Merc 280C	3.92	3.440	18.90
11	Merc 450SE	3.07	4.070	17.40
12	Merc 450SL	3.07	3.730	17.60
13	Merc 450SLC	3.07	3.780	18.00
14	Cadillac Fleetwood	2.93	5.250	17.98
15	Lincoln Continental	3.00	5.424	17.82
16	Chrysler Imperial	3.23	5.345	17.42
17	Fiat 128	4.08	2.200	19.47
18	Honda Civic	4.93	1.615	18.52
19	Toyota Corolla	4.22	1.835	19.90
20	Toyota Corona	3.70	2.465	20.01
21	Dodge Challenger	2.76	3.520	16.87
22	AMC Javelin	3.15	3.435	17.30
23	Camaro Z28	3.73	3.840	15.41
24	Pontiac Firebird	3.08	3.845	17.05
25	Fiat X1-9	4.08	1.935	18.90
26	Porsche 914-2	4.43	2.140	16.70
27	Lotus Europa	3.77	1.513	16.90
28	Ford Pantera L	4.22	3.170	14.50
29	Ferrari Dino	3.62	2.770	15.50
30	Maserati Bora	3.54	3.570	14.60
31	Volvo 142E	4.11	2.780	18.60

In [37]:

```
1 car_data.dtypes
```

Out[37]:

```
Unnamed: 0      object
Points          float64
Score           float64
Weigh           float64
dtype: object
```

In [38]:

```
1 car_data.shape
```

Out[38]:

```
(32, 4)
```

In [39]:

```
1 car_data.isna().sum()
```

Out[39]:

```
Unnamed: 0      0
Points          0
Score           0
Weigh           0
dtype: int64
```

In [40]:

```
1 car_data.mean(numeric_only=True)    # Mean of the table
```

Out[40]:

```
Points      3.596563
Score       3.217250
Weigh      17.848750
dtype: float64
```

In [41]:

```
1 car_data.describe(include='all') # Overall review of the table
```

Out[41]:

	Unnamed: 0	Points	Score	Weigh
count	32	32.000000	32.000000	32.000000
unique	32	NaN	NaN	NaN
top	Mazda RX4	NaN	NaN	NaN
freq	1	NaN	NaN	NaN
mean	NaN	3.596563	3.217250	17.848750
std	NaN	0.534679	0.978457	1.786943
min	NaN	2.760000	1.513000	14.500000
25%	NaN	3.080000	2.581250	16.892500
50%	NaN	3.695000	3.325000	17.710000
75%	NaN	3.920000	3.610000	18.900000
max	NaN	4.930000	5.424000	22.900000

In [42]:

```
1 car_data.var(numeric_only=True) # variance of the table
```

Out[42]:

```
Points    0.285881
Score     0.957379
Weigh     3.193166
dtype: float64
```

In [43]:

```
1 car_data.mode( axis = 0,numeric_only= True,) # Mode of the table
```

Out[43]:

	Points	Score	Weigh
0	3.07	3.44	17.02
1	3.92	NaN	18.90

In [44]:

```
1 car_data.median(numeric_only=True)           # Median of the table
```

Out[44]:

```
Points      3.695
Score       3.325
Weigh      17.710
dtype: float64
```

In [45]:

```
1 car_data.std(numeric_only=True)             # standar Daviation of the table
```

Out[45]:

```
Points      0.534679
Score       0.978457
Weigh       1.786943
dtype: float64
```

In [33]:

```
1
```

In []:

```
1
```