Import some libraries. Don't forget to import Scipy, and Stats for using Descriptive Statistics

In [2]:
```
import scipy
from scipy import stats

import numpy as np

import pandas as pd
from pandas import Series, DataFrame
```

Read CSV document from local computer

In [3]:
```
address = 'C:/Users/muham/mtcars.csv'
cars = pd.read_csv(address)
cars.columns = ['car_names', 'mpg', 'cyl', 'disp', 'hp', 'drat', 'wp', 'qsec', '
```

Descriptive Statistics that describe the values of observation

Looking for sum, median, mean, max and describe

In [4]:
```
cars.sum()
```

Out[4]:
```
car_names    Mazda RX4Mazda RX4 WagDatsun 710Hornet 4 Drive...
mpg                                                      642.9
cyl                                                        198
disp                                                    7383.1
hp                                                        4694
drat                                                    115.09
wp                                                     102.952
qsec                                                    571.16
vs                                                          14
am                                                          13
gear                                                       118
carb                                                        90
dtype: object
```

```
In [5]: cars.sum(axis = 1)
```

```
Out[5]: 0     328.980
        1     329.795
        2     259.580
        3     426.135
        4     590.310
        5     385.540
        6     656.920
        7     270.980
        8     299.570
        9     350.460
        10    349.660
        11    510.740
        12    511.500
        13    509.850
        14    728.560
        15    726.644
        16    725.695
        17    213.850
        18    195.165
        19    206.955
        20    273.775
        21    519.650
        22    506.085
        23    646.280
        24    631.175
        25    208.215
        26    272.570
        27    273.683
        28    670.690
        29    379.590
        30    694.710
        31    288.890
        dtype: float64
```

```
In [6]: cars.median()
```

```
Out[6]: mpg      19.200
        cyl       6.000
        disp    196.300
        hp      123.000
        drat      3.695
        wp        3.325
        qsec     17.710
        vs        0.000
        am        0.000
        gear      4.000
        carb      2.000
        dtype: float64
```

With describe() funtion, we call all in one, include Quartiles. Look at the data table below. 25%, 50% and 75% means Q1, Q2 and Q3

In [7]:
```
cars.describe()
```

Out[7]:

	mpg	cyl	disp	hp	drat	wp	qsec	vs
count	32.000000	32.000000	32.000000	32.000000	32.000000	32.000000	32.000000	32.000000
mean	20.090625	6.187500	230.721875	146.687500	3.596563	3.217250	17.848750	0.437500
std	6.026948	1.785922	123.938694	68.562868	0.534679	0.978457	1.786943	0.504016
min	10.400000	4.000000	71.100000	52.000000	2.760000	1.513000	14.500000	0.000000
25%	15.425000	4.000000	120.825000	96.500000	3.080000	2.581250	16.892500	0.000000
50%	19.200000	6.000000	196.300000	123.000000	3.695000	3.325000	17.710000	0.000000
75%	22.800000	8.000000	326.000000	180.000000	3.920000	3.610000	18.900000	1.000000
max	33.900000	8.000000	472.000000	335.000000	4.930000	5.424000	22.900000	1.000000

In [8]:
```
cyl = cars['cyl']
cyl.mean()
```

Out[8]:
```
6.1875
```

Descriptive Statistics that describe a variable spread

Looking for Standard Deviation, Variance and Variable_counts

In [9]:
```
cars.std()
```

Out[9]:
```
mpg       6.026948
cyl       1.785922
disp    123.938694
hp       68.562868
drat      0.534679
wp        0.978457
qsec      1.786943
vs        0.504016
am        0.498991
gear      0.737804
carb      1.615200
dtype: float64
```

In [10]:
```
cars.var()
```

Out[10]:
```
mpg         36.324103
cyl          3.189516
disp     15360.799829
hp        4700.866935
drat         0.285881
wp           0.957379
qsec         3.193166
vs           0.254032
am           0.248992
gear         0.544355
carb         2.608871
dtype: float64
```

In [14]:
```
cyl.value_counts()
```

Out[14]:
```
8    14
4    11
6     7
Name: cyl, dtype: int64
```

In [19]:
```
cyl.describe()
```

Out[19]:
```
count    32.000000
mean      6.187500
std       1.785922
min       4.000000
25%       4.000000
50%       6.000000
75%       8.000000
max       8.000000
Name: cyl, dtype: float64
```