```
Chapter 5 - Basic Math and Statistics

Segement 3 - Generating summary statistics using pandas and scipy
```

In [1]:
```
import numpy as np
import pandas as pd
from pandas import Series, DataFrame

import scipy
from scipy import stats
```

In [2]:
```
address = 'C:/Users/danal/Desktop/ExerciseFiles/Data/mtcars.csv'

cars = pd.read_csv(address)
cars.columns = ['car_names', 'mpg', 'cyl', 'disp', 'hp', 'drat', 'wt', 'qsec', 'v

cars.head()
```

Out[2]:

	car_names	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
0	Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
1	Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
2	Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
3	Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
4	Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2

```
1 Looking at summary statistics that describe a variable's numeric values
```

In [3]:
```
cars.sum()
```

Out[3]:
```
car_names    Mazda RX4Mazda RX4 WagDatsun 710Hornet 4 Drive...
mpg                                                      642.9
cyl                                                        198
disp                                                    7383.1
hp                                                        4694
drat                                                    115.09
wt                                                     102.952
qsec                                                    571.16
vs                                                          14
am                                                          13
gear                                                       118
carb                                                        90
dtype: object
```

In [4]:
```
cars.sum(axis=1)
```

Out[4]:
```
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 [5]:
```
cars.median()
```

Out[5]:
```
mpg      19.200
cyl       6.000
disp    196.300
hp      123.000
drat      3.695
wt        3.325
qsec     17.710
vs        0.000
am        0.000
gear      4.000
carb      2.000
dtype: float64
```

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

Out[6]:
```
mpg      20.090625
cyl       6.187500
disp    230.721875
hp      146.687500
drat      3.596563
wt        3.217250
qsec     17.848750
vs        0.437500
am        0.406250
gear      3.687500
carb      2.812500
dtype: float64
```

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

Out[7]:
```
car_names    Volvo 142E
mpg                33.9
cyl                   8
disp                472
hp                  335
drat               4.93
wt                5.424
qsec               22.9
vs                    1
am                    1
gear                  5
carb                  8
dtype: object
```

In [8]:
```
mpg = cars.mpg
mpg.idxmax()
```

Out[8]:
```
19
```

```
Looking at summary statistics that describe variable distribution
```

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

Out[9]:
```
mpg       6.026948
cyl       1.785922
disp    123.938694
hp       68.562868
drat      0.534679
wt        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
wt          0.957379
qsec        3.193166
vs          0.254032
am          0.248992
gear        0.544355
carb        2.608871
dtype: float64
```

In [11]:
```
gear = cars.gear
gear.value_counts()
```

Out[11]:
```
3    15
4    12
5     5
Name: gear, dtype: int64
```

In [14]:
```
cars.describe()
```

Out[14]:

	mpg	cyl	disp	hp	drat	wt	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