Assignment 2

In [62]:

```
from prettytable import PrettyTable as pt
import math
import pandas as pd
```

In [63]:

```
values = [15,23,23,64,23,65,22,34,24,62,45,63,45,73,46,73,56,73,56,73,45,72]
car = Pens = {'Brand': ['Audi','Mercedes','Tata','Jaguar','McLaren'],
        'Price': [50000,40000,20000,35000,60000],
        'Rating': [60,65,45,55,65]
        }
df = pd.DataFrame(car)
```

Mean

Mean can also be understood as the average of certain numbers.

The arithmetic mean, also known as average or arithmetic average, is a central value of a finite set of numbers

Formula:

$$m = \frac{\text{sum of the terms}}{\text{number of terms}}$$

m = mean

In [64]:

```
def mean(values):
    sum = 0
    avg = 0
    for i in values:
        sum += i
    avg = sum/len(values)
    return avg
print("Mean: {}".format(mean(values)))
```

```
Mean: 48.86363636363637
```

Median

The median is the value separating the higher half from the lower half of a data sample, a population, or a probability distribution.

For a data set, it may be thought of as "the middle" value.

Formula:

$$\text{Med}(X) = \begin{cases} X[\frac{n}{2}] & \text{if n is even} \\ \frac{(X[\frac{n-1}{2}]+X[\frac{n+1}{2}])}{2} & \text{if n is odd} \end{cases}$$

X = ordered list of values in data set

n = number of values in data set

In [65]:

```
def median(values):
    median = 0
    median_values = values
    values.sort()
    length = len(median_values)
    if length%2:
        median = values[int((length+1)/2)]+ values[int((length-1)/2)]/2
        return median
    median = values[int(length/2)]
    return median

print("Median: {}".format(median(values)))
```

```
Median: 56
```

In [66]:

```
mode_values = [1,1,1,1,1,1,1,1,1,2,2,2,3,4,5,3,3,3,4,5,6,7,8,8,6,7,5,3,4,5,6,8,9,0,8
```

Mode

The mode is the value that appears most often in a set of data values

In [67]:

```
def mode(mode_values):
    greatest = 0
    mode_val = list()
    freq = {}
    for item in mode_values:
        if (item in freq):
            freq[item] += 1
        else:
            freq[item] = 1
    for i,j in freq.items():
        if j > greatest:
            greatest = j
            mode_val.clear()
            mode_val.insert(0,i)
        elif j==greatest:
            mode_val.append(i)

    print("Mode : {}".format(mode_val))
    print("No of Occurances : {}".format(greatest))
mode(mode_values)
```

```
Mode : [1, 5]
No of Occurances : 9
```

Variance

Variance tells you the degree of spread in your data set. The more spread the data, the larger the variance is in relation to the mean

Formula

$$\sigma^2 = \frac{\Sigma (X - \mu)^2}{N}$$

σ2 = population variance

Σ = sum of…

X = each value

μ = population mean

N = number of values in the population

In [68]:

```
def variance(values):
    mean_val = mean(values)
    print("Mean: {}".format(mean_val))
    myTable = pt(["i", "Variance"])
    for i in values:
        row = [i, mean_val-i]
        myTable.add_row(row)
    print(myTable)

variance(values)
```

```
Mean: 48.86363636363637
+----+---------------------+
| i  |       Variance      |
+----+---------------------+
| 15 |  33.86363636363637  |
| 22 |  26.863636363636367 |
| 23 |  25.863636363636367 |
| 23 |  25.863636363636367 |
| 23 |  25.863636363636367 |
| 24 |  24.863636363636367 |
| 34 |  14.863636363636367 |
| 45 |  3.863636363636367  |
| 45 |  3.863636363636367  |
| 45 |  3.863636363636367  |
| 46 |  2.863636363636367  |
| 56 |  -7.136363636363633 |
| 56 |  -7.136363636363633 |
| 62 | -13.136363636363633 |
| 63 | -14.136363636363633 |
| 64 | -15.136363636363633 |
| 65 | -16.136363636363633 |
| 72 | -23.136363636363633 |
| 73 | -24.136363636363633 |
| 73 | -24.136363636363633 |
| 73 | -24.136363636363633 |
| 73 | -24.136363636363633 |
+----+---------------------+
```

Standard Deviation

Square root of variation

In statistics, the standard deviation is a measure of the amount of variation or dispersion of a set of values.

Formula:

$$\sigma = \sqrt{\frac{\sum (x_i - \mu)^2}{N}}$$

In [69]:

```
def sd(values):
    mean_val = mean(values)
    print("Mean: {}".format(mean_val))
    myTable = pt(["i", "Variance"])
    for i in values:
        row = [i, math.sqrt(abs((mean_val-i)))]
        myTable.add_row(row)
    print(myTable)

sd(values)
```

```
Mean: 48.86363636363637
+----+--------------------+
| i  |      Variance      |
+----+--------------------+
| 15 | 5.819247061573892  |
| 22 | 5.183014216036492  |
| 23 | 5.085630380162952  |
| 23 | 5.085630380162952  |
| 23 | 5.085630380162952  |
| 24 | 4.986344990435015  |
| 34 | 3.8553386833890957 |
| 45 | 1.965613482767242  |
| 45 | 1.965613482767242  |
| 45 | 1.965613482767242  |
| 46 | 1.6922282244532996 |
| 56 | 2.6713973190754747 |
| 56 | 2.6713973190754747 |
| 62 | 3.624412178045377  |
| 63 | 3.759835586347312  |
| 64 | 3.8905479866419372 |
| 65 | 4.017009290051945  |
| 72 | 4.810027404949335  |
| 73 | 4.912877327632315  |
| 73 | 4.912877327632315  |
| 73 | 4.912877327632315  |
| 73 | 4.912877327632315  |
+----+--------------------+
```

In [70]:

```
df
```

Out[70]:

	Brand	Price	Rating
0	Audi	50000	60
1	Mercedes	40000	65
2	Tata	20000	45
3	Jaguar	35000	55
4	McLaren	60000	65

Correlation

- It does not mean that the changes in one variable actually cause the changes in the other variable. Sometimes it is clear that there is a causal relationship.

In [71]:

```
df.corr()
```

Out[71]:

	Price	Rating
Price	1.00000	0.85707
Rating	0.85707	1.00000

Covariance

- It provides insight into how two variables are related to one another
- A positive covariance means that the two variables at hand are positively related,and they move in the same direction

In [72]:

```
df.cov()
```

Out[72]:

	Price	Rating
Price	230000000.0	108750.0
Rating	108750.0	70.0