PRACTICAL NO.1

To perform and Analysis of Simple Linear Regression

Importing the Libraries

In [2]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
```

In [3]:
```
import os
```

In [4]:
```
os.getcwd()
```

Out[4]:
```
'C:\\Users\\SAICOM\\Downloads'
```

In [6]:
```
os.chdir('C:\\Users\\SAICOM\\Downloads')
```

In [7]:
```
df=pd.read_csv("Salary_Data.csv")
```

In [8]:
```
df.head()
```

Out[8]:

	YearsExperience	Salary
0	1.1	39343.0
1	1.3	46205.0
2	1.5	37731.0
3	2.0	43525.0
4	2.2	39891.0

In [9]:
```
df.tail()
```

Out[9]:

	YearsExperience	Salary
25	9.0	105582.0
26	9.5	116969.0
27	9.6	112635.0
28	10.3	122391.0
29	10.5	121872.0

In [10]:
```
df.head(30)
```

Out[10]:

	YearsExperience	Salary
0	1.1	39343.0
1	1.3	46205.0
2	1.5	37731.0
3	2.0	43525.0
4	2.2	39891.0
5	2.9	56642.0
6	3.0	60150.0
7	3.2	54445.0
8	3.2	64445.0
9	3.7	57189.0
10	3.9	63218.0
11	4.0	55794.0
12	4.0	56957.0
13	4.1	57081.0
14	4.5	61111.0
15	4.9	67938.0
16	5.1	66029.0
17	5.3	83088.0
18	5.9	81363.0
19	6.0	93940.0
20	6.8	91738.0
21	7.1	98273.0
22	7.9	101302.0
23	8.2	113812.0
24	8.7	109431.0
25	9.0	105582.0
26	9.5	116969.0
27	9.6	112635.0
28	10.3	122391.0
29	10.5	121872.0

In [11]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 2 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   YearsExperience  30 non-null     float64
 1   Salary           30 non-null     float64
dtypes: float64(2)
memory usage: 612.0 bytes
```

In [12]:
```
df.describe()
```

Out[12]:

	YearsExperience	Salary
count	30.000000	30.000000
mean	5.313333	76003.000000
std	2.837888	27414.429785
min	1.100000	37731.000000
25%	3.200000	56720.750000
50%	4.700000	65237.000000
75%	7.700000	100544.750000
max	10.500000	122391.000000

In [13]:
```
df.shape
```

Out[13]:
```
(30, 2)
```

In [14]:
```
df.size
```

Out[14]:
```
60
```

In [15]:
```
df.ndim
```

Out[15]:
```
2
```

In [16]:
```
df.isnull().sum()
```

Out[16]:
```
YearsExperience    0
Salary             0
dtype: int64
```

In [17]:
```
#Assiging values in X & Y

X=df.drop("YearsExperience", axis=1)
y=df["Salary"]
```

In [18]:
```
print(X)
```

```
      Salary
0    39343.0
1    46205.0
2    37731.0
3    43525.0
4    39891.0
5    56642.0
6    60150.0
7    54445.0
8    64445.0
9    57189.0
10   63218.0
11   55794.0
12   56957.0
13   57081.0
14   61111.0
15   67938.0
16   66029.0
17   83088.0
18   81363.0
19   93940.0
20   91738.0
21   98273.0
22  101302.0
23  113812.0
24  109431.0
25  105582.0
26  116969.0
27  112635.0
28  122391.0
29  121872.0
```

In [19]:
```
print(y)
```

```
0     39343.0
1     46205.0
2     37731.0
3     43525.0
4     39891.0
5     56642.0
6     60150.0
7     54445.0
8     64445.0
9     57189.0
10    63218.0
11    55794.0
12    56957.0
13    57081.0
14    61111.0
15    67938.0
16    66029.0
17    83088.0
18    81363.0
19    93940.0
20    91738.0
21    98273.0
22   101302.0
23   113812.0
24   109431.0
25   105582.0
26   116969.0
27   112635.0
28   122391.0
29   121872.0
Name: Salary, dtype: float64
```

In [20]:
```
#Splitting testdata into X_train,X_test,y_train,y_test
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=.3,random_state=42)
```

In [21]:
```
print(X_train)
```

```
      Salary
0    39343.0
4    39891.0
16   66029.0
5    56642.0
13   57081.0
11   55794.0
22  101302.0
1    46205.0
2    37731.0
25  105582.0
3    43525.0
21   98273.0
26  116969.0
18   81363.0
29  121872.0
20   91738.0
7    54445.0
10   63218.0
14   61111.0
19   93940.0
6    60150.0
```

In [22]:
```
print(X_test)
```

```
      Salary
27  112635.0
15   67938.0
23  113812.0
17   83088.0
8    64445.0
9    57189.0
28  122391.0
24  109431.0
12   56957.0
```

In [23]:
```
print(y_train)
```

```
0     39343.0
4     39891.0
16    66029.0
5     56642.0
13    57081.0
11    55794.0
22   101302.0
1     46205.0
2     37731.0
25   105582.0
3     43525.0
21    98273.0
26   116969.0
18    81363.0
29   121872.0
20    91738.0
7     54445.0
10    63218.0
14    61111.0
19    93940.0
6     60150.0
Name: Salary, dtype: float64
```

In [24]:
```
print (y_test)
```

```
27   112635.0
15    67938.0
23   113812.0
17    83088.0
8     64445.0
9     57189.0
28   122391.0
24   109431.0
12    56957.0
Name: Salary, dtype: float64
```

In [25]:
```
from sklearn.linear_model import LinearRegression
lr = linearRegression()
lr.fit(X_train, y_train)
```

Out[25]:
```
* LinearRegression
LinearRegression()
```

In [26]:
```
#Assigning Coefficient (slope) to m
m = lr.coef_
```

In [27]:
```
print("Coefficient  :" , m)
```

```
Coefficient  : [1.]
```

In [28]:
```
#Assigning Y-intercept to a
c = lr.intercept_
```

In [29]:
```
print("Intercept  : ", c)
```

```
Intercept  :  1.4551915228366852e-11
```