

```
In [1]: #Aim:To perform Simple Linear Regression and find out the coefficients of it.
```

```
In [ ]: #Name: Achal Subhash Kharwade  
#Roll No: 36  
#Sec: B  
#Date :09-09-2023
```

```
In [3]: import pandas as ps  
import matplotlib.pyplot as plt  
import seaborn as sns  
import numpy as np  
import os
```

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

```
Out[4]: 'C:\\Users\\Lenovo\\DSS 5th Sem'
```

```
In [35]: os.chdir("D:\\DSS\\DSS PRAC PG")
```

```
In [6]: df=ps.read_csv("SalaryDataset.csv")
```

```
In [7]: df.columns
```

```
Out[7]: Index(['Unnamed: 0', 'YearsExperience', 'Salary'], dtype='object')
```

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

```
Out[8]:
```

	Unnamed: 0	YearsExperience	Salary
0	0	1.2	39344.0
1	1	1.4	46206.0
2	2	1.6	37732.0
3	3	2.1	43526.0
4	4	2.3	39892.0

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

```
Out[9]:
```

	Unnamed: 0	YearsExperience	Salary
25	25	9.1	105583.0
26	26	9.6	116970.0
27	27	9.7	112636.0
28	28	10.4	122392.0
29	29	10.6	121873.0

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

```
Out[10]:
```

	Unnamed: 0	YearsExperience	Salary
0	0	1.2	39344.0
1	1	1.4	46206.0
2	2	1.6	37732.0
3	3	2.1	43526.0
4	4	2.3	39892.0
5	5	3.0	56643.0
6	6	3.1	60151.0
7	7	3.3	54446.0

8	8	3.3	64446.0
9	9	3.8	57190.0
10	10	4.0	63219.0
11	11	4.1	55795.0
12	12	4.1	56958.0
13	13	4.2	57082.0
14	14	4.6	61112.0
15	15	5.0	67939.0
16	16	5.2	66030.0
17	17	5.4	83089.0
18	18	6.0	81364.0
19	19	6.1	93941.0
20	20	6.9	91739.0
21	21	7.2	98274.0
22	22	8.0	101303.0
23	23	8.3	113813.0
24	24	8.8	109432.0
25	25	9.1	105583.0
26	26	9.6	116970.0
27	27	9.7	112636.0
28	28	10.4	122392.0
29	29	10.6	121873.0

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 3 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   Unnamed: 0      30 non-null    int64
1   YearsExperience  30 non-null    float64
2   Salary          30 non-null    float64
dtypes: float64(2), int64(1)
memory usage: 852.0 bytes
```

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

```
Out[12]:
```

	Unnamed: 0	YearsExperience	Salary
count	30.000000	30.000000	30.000000
mean	14.500000	5.413333	76004.000000
std	8.803408	2.837888	27414.429785
min	0.000000	1.200000	37732.000000
25%	7.250000	3.300000	56721.750000
50%	14.500000	4.800000	65238.000000
75%	21.750000	7.800000	100545.750000
max	29.000000	10.600000	122392.000000

In [13]: `df.shape`

Out[13]: (30, 3)

In [14]: `df.size`

Out[14]: 90

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

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

```
In [17]: df.isnull()
```

```
Out[17]:
```

	Unnamed: 0	YearsExperience	Salary
0	False	False	False
1	False	False	False
2	False	False	False
3	False	False	False
4	False	False	False
5	False	False	False
6	False	False	False
7	False	False	False
8	False	False	False
9	False	False	False
10	False	False	False
11	False	False	False
12	False	False	False
13	False	False	False
14	False	False	False
15	False	False	False
16	False	False	False
17	False	False	False
18	False	False	False
19	False	False	False
20	False	False	False
21	False	False	False
22	False	False	False
23	False	False	False
24	False	False	False
25	False	False	False
26	False	False	False
27	False	False	False
28	False	False	False
29	False	False	False

```
In [18]: df.isnull().sum()
```

```
Out[18]: Unnamed: 0      0
YearsExperience  0
Salary          0
dtype: int64
```

```
In [19]: #Assigning values in X & Y
x=df.iloc[:, :-1]
y=df.iloc[:, -1]
#X = df['YearsExperience']
#Y = df['Salary']
```

```
In [20]: print(x)
```

```
      Unnamed: 0  YearsExperience
0              0                1.2
```

1	1	1.4
2	2	1.6
3	3	2.1
4	4	2.3
5	5	3.0
6	6	3.1
7	7	3.3
8	8	3.3
9	9	3.8
10	10	4.0
11	11	4.1
12	12	4.1
13	13	4.2
14	14	4.6
15	15	5.0
16	16	5.2
17	17	5.4
18	18	6.0
19	19	6.1
20	20	6.9
21	21	7.2
22	22	8.0
23	23	8.3
24	24	8.8
25	25	9.1
26	26	9.6
27	27	9.7
28	28	10.4
29	29	10.6

In [21]: `print(y)`

```
0    39344.0
1    46206.0
2    37732.0
3    43526.0
4    39892.0
5    56643.0
6    60151.0
7    54446.0
8    64446.0
9    57190.0
10   63219.0
11   55795.0
12   56958.0
13   57082.0
14   61112.0
15   67939.0
16   66030.0
17   83089.0
18   81364.0
19   93941.0
20   91739.0
21   98274.0
22  101303.0
23  113813.0
24  109432.0
25  105583.0
26  116970.0
27  112636.0
28  122392.0
29  121873.0
Name: Salary, dtype: float64
```

In [22]: `#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 [23]: `print(x_train)`

```
Unnamed: 0  YearsExperience
0          0             1.2
4          4             2.3
16         16             5.2
5          5             3.0
13         13             4.2
11         11             4.1
22         22             8.0
1          1             1.4
2          2             1.6
25         25             9.1
```

3	3	2.1
21	21	7.2
26	26	9.6
18	18	6.0
29	29	10.6
20	20	6.9
7	7	3.3
10	10	4.0
14	14	4.6
19	19	6.1
6	6	3.1

In [24]: `print(x_test)`

	Unnamed: 0	YearsExperience
27	27	9.7
15	15	5.0
23	23	8.3
17	17	5.4
8	8	3.3
9	9	3.8
28	28	10.4
24	24	8.8
12	12	4.1

In [25]: `print(y_train)`

```

0      39344.0
4      39892.0
16     66030.0
5      56643.0
13     57082.0
11     55795.0
22    101303.0
1      46206.0
2      37732.0
25    105583.0
3      43526.0
21     98274.0
26    116970.0
18     81364.0
29    121873.0
20     91739.0
7      54446.0
10     63219.0
14     61112.0
19     93941.0
6      60151.0
Name: Salary, dtype: float64

```

In [26]: `print(y_test)`

```

27    112636.0
15     67939.0
23    113813.0
17     83089.0
8      64446.0
9      57190.0
28    122392.0
24    109432.0
12     56958.0
Name: Salary, dtype: float64

```

In [27]: `from sklearn.linear_model import LinearRegression`

In [28]: `lr = LinearRegression()`

In [29]: `lr.fit(x_train, y_train)`

Out[29]: `LinearRegression`
`LinearRegression()`

```
In [30]: #Assigning Coefficient (slope) to m  
m = lr.coef_
```

```
In [31]: print("Coefficient :",m)  
  
Coefficient : [ -704.23954045 11567.35602913]
```

```
In [32]: #Assigning Y-intercept to a  
c = lr.intercept_
```

```
In [33]: print("Intercept :",c)  
  
Intercept : 23114.289065884543
```

```
In [ ]:
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js