

```
In [1]: #EXP_7
```

```
In [ ]: #Aim:Simple linear Regression
```

```
In [1]: #Name:Dev Sanjay Vaidya  
#Roll no.:69  
#Subject:ET1  
#Sec:B  
#Date:04/09/25
```

```
In [7]: import pandas as pd
```

```
In [8]: import os
```

```
In [9]: os.getcwd()
```

```
Out[9]: 'C:\\\\Users\\\\LENOVO'
```

```
In [10]: os.chdir ("C:\\\\Users\\\\LENOVO\\\\Desktop")
```

```
In [11]: df=pd.read_csv("Salary_Data.csv")
```

```
In [12]: df.head(10)
```

```
Out[12]:
```

	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

```
In [13]: df.tail(10)
```

```
Out[13]:
```

	YearsExperience	Salary
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 [14]: df.shape
```

```
Out[14]: (30, 2)
```

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

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

```
In [16]: df.columns
```

```
Out[16]: Index(['YearsExperience', 'Salary'], dtype='object')
```

```
In [17]: df.size
```

```
Out[17]: 60
```

```
In [18]: df.info
```

```
Out[18]: <bound method DataFrame.info of      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 [19]: df.describe()
```

```
Out[19]:
```

	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 [20]: df
```

Out[20]:

	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 [21]: df.isna

```
Out[21]: <bound method DataFrame.isna of      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 [22]: df.isna().any()
```

```
Out[22]: YearsExperience      False
Salary          False
dtype: bool
```

```
In [23]: df.isna().sum()
```

```
Out[23]: YearsExperience      0
Salary          0
dtype: int64
```

Independant and dependant Variables

```
In [24]: x=df.drop('Salary',axis=1)
```

```
In [25]: x.head()
```

```
Out[25]: YearsExperience
```

	YearsExperience
0	1.1
1	1.3
2	1.5
3	2.0
4	2.2

```
In [26]: y=df.Salary
```

```
In [27]: y.head()
```

```
Out[27]: 0    39343.0
1    46205.0
2    37731.0
3    43525.0
4    39891.0
Name: Salary, dtype: float64
```

```
In [28]: from sklearn.model_selection import train_test_split
```

```
In [29]: x_train,x_test,y_train,y_test=train_test_split(x,y,random_state=0,test_size=
```

```
In [30]: print(x_train.shape)

(21, 1)
```

```
In [31]: x_test.shape
```

```
Out[31]: (9, 1)
```

Model fitting

```
In [32]: from sklearn.linear_model import LinearRegression
LR=LinearRegression()
LR.fit(x_train,y_train)
```

```
Out[32]: ▾ LinearRegression ⓘ ⓘ
LinearRegression()
```

```
In [33]: #Assingning coefficient(slope)to m
m=LR.coef_
```

```
In [34]: print("coefficient:",m)
```

```
coefficient: [9360.26128619]
```

```
In [35]: #Assingning y-intercept to a  
c=LR.intercept_
```

```
In [36]: print("Intercept:",c)
```

```
Intercept: 26777.39134119764
```

Evaluation Metrics

```
In [37]: from sklearn import metrics
```

```
In [38]: Accuracy = LR.score(x_test, y_test)  
Accuracy
```

```
Out[38]: 0.9740993407213511
```

```
In [ ]:
```