

# Simple Linear Regression

In [1]: `#Exp no.:9`

In [2]: `#Aim: To perform Simple Linear Regression`

In [3]: `#Name: Rohit Pradip Malokar  
#Roll No.:18  
#Sec: 3B  
#Subject:ET - 1  
#Date: 05/10/2024`

In [4]: `import pandas as pd`

In [5]: `import os`

In [6]: `os.getcwd()`

Out[6]: `'C:\\Users\\riyak'`

In [7]: `os.chdir("C:\\Users\\riyak\\OneDrive\\Desktop")`

In [8]: `data=pd.read_csv("Salary.csv")`

In [9]: `data.head()`

Out[9]:

	<b>YearsExperience</b>	<b>Salary</b>
<b>0</b>	1.1	39343
<b>1</b>	1.3	46205
<b>2</b>	1.5	37731
<b>3</b>	2.0	43525
<b>4</b>	2.2	39891

In [10]: `data.tail()`

Out[10]:

	<b>YearsExperience</b>	<b>Salary</b>
<b>30</b>	11.2	127345
<b>31</b>	11.5	126756
<b>32</b>	12.3	128765
<b>33</b>	12.9	135675
<b>34</b>	13.5	139465

In [11]: `data.describe()`

Out[11]:

	<b>YearsExperience</b>	<b>Salary</b>
<b>count</b>	35.000000	35.000000
<b>mean</b>	6.308571	83945.600000
<b>std</b>	3.618610	32162.673003
<b>min</b>	1.100000	37731.000000
<b>25%</b>	3.450000	57019.000000
<b>50%</b>	5.300000	81363.000000
<b>75%</b>	9.250000	113223.500000
<b>max</b>	13.500000	139465.000000

In [12]: `data.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 35 entries, 0 to 34
Data columns (total 2 columns):
#   Column          Non-Null Count  Dtype
---  -
0   YearsExperience  35 non-null     float64
1   Salary          35 non-null     int64
dtypes: float64(1), int64(1)
memory usage: 692.0 bytes
```

In [13]: `data.shape`

Out[13]: (35, 2)

In [14]: `data.size`

Out[14]: 70

In [ ]:

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

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

```
In [16]: data.isnull()
```

Out[16]:

	<b>YearsExperience</b>	<b>Salary</b>
0	False	False
1	False	False
2	False	False
3	False	False
4	False	False
5	False	False
6	False	False
7	False	False
8	False	False
9	False	False
10	False	False
11	False	False
12	False	False
13	False	False
14	False	False
15	False	False
16	False	False
17	False	False
18	False	False
19	False	False
20	False	False
21	False	False
22	False	False
23	False	False
24	False	False
25	False	False
26	False	False
27	False	False
28	False	False
29	False	False
30	False	False
31	False	False
32	False	False
33	False	False
34	False	False

```
In [17]: data.isnull().any()
```

```
Out[17]: YearsExperience    False  
Salary                    False  
dtype: bool
```

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

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

```
In [19]: a='ashish'
```

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

```
ashish
```

```
In [21]: a[-1]
```

```
Out[21]: 'h'
```

```
In [22]: a[1]
```

```
Out[22]: 's'
```

```
In [23]: a[1:3]
```

```
Out[23]: 'sh'
```

```
In [24]: a[1:4]
```

```
Out[24]: 'shi'
```

```
In [26]: #Assiging values in X & Y  
X = data.iloc[:, :-1].values  
y = data.iloc[:, -1].values
```

```
#X = data['YearsExperience']  
#y = data['Salary']
```

```
In [27]: print(X)
```

```
[[ 1.1]
 [ 1.3]
 [ 1.5]
 [ 2. ]
 [ 2.2]
 [ 2.9]
 [ 3. ]
 [ 3.2]
 [ 3.2]
 [ 3.7]
 [ 3.9]
 [ 4. ]
 [ 4. ]
 [ 4.1]
 [ 4.5]
 [ 4.9]
 [ 5.1]
 [ 5.3]
 [ 5.9]
 [ 6. ]
 [ 6.8]
 [ 7.1]
 [ 7.9]
 [ 8.2]
 [ 8.7]
 [ 9. ]
 [ 9.5]
 [ 9.6]
 [10.3]
 [10.5]
 [11.2]
 [11.5]
 [12.3]
 [12.9]
 [13.5]]
```

```
In [28]: print(y)
```

```
[ 39343  46205  37731  43525  39891  56642  60150  54445  64445  57189
  63218  55794  56957  57081  61111  67938  66029  83088  81363  93940
  91738  98273 101302 113812 109431 105582 116969 112635 122391 121872
 127345 126756 128765 135675 139465]
```

```
In [31]: #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
```

```
In [32]: print(X_train)
```

```
[[12.9]
 [ 1.1]
 [ 2.2]
 [ 5.3]
 [ 9.6]
 [ 2.9]
 [ 4. ]
 [ 1.3]
 [ 1.5]
 [12.3]
 [ 2. ]
 [11.2]
 [ 8.2]
 [11.5]
 [ 3.9]
 [ 7.9]
 [ 5.9]
 [ 9. ]
 [ 3. ]
 [ 6.8]
 [13.5]
 [ 3.2]
 [ 4.5]
 [10.3]]
```

```
In [33]: print(X_test)
```

```
[[ 9.5]
 [ 4.1]
 [ 8.7]
 [ 7.1]
 [ 4.9]
 [10.5]
 [ 6. ]
 [ 4. ]
 [ 3.2]
 [ 5.1]
 [ 3.7]]
```

```
In [34]: print(y_train)
```

```
[135675  39343  39891  83088 112635  56642  55794  46205  37731 128765
  43525 127345 113812 126756  63218 101302  81363 105582  60150  91738
 139465  54445  61111 122391]
```

```
In [35]: print (y_test)
```

```
[116969  57081 109431  98273  67938 121872  93940  56957  64445  66029
  57189]
```



```
In [36]: from sklearn.linear_model import LinearRegression  
lr = LinearRegression()  
lr.fit(X_train, y_train)
```

Out[36]: LinearRegression()

**In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.**

**On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.**

```
In [37]: #Assigning Coefficient (slope) to m  
m = lr.coef_
```

```
In [38]: print("Coefficient :", m)
```

Coefficient : [8555.33918938]

```
In [39]: #Assigning Y-intercept to a  
c = lr.intercept_
```

```
In [40]: print("Intercept :", c)
```

Intercept : 29602.07353482097

```
In [41]: lr.score(X_test,y_test) * 100
```

Out[41]: 91.71426108885095

```
In [ ]:
```