Simple Linear Regression

In [1]:

```
#Aim: To perform simple Linear Regression
#Exp no:9
#Name:Khushi Chandrashekhar Satpute
#Sec:B
#Roll no:43
#Sub:ET-1
#Date:05/10/2024
```

In [3]:

```
import os
```

In [4]:

```
import pandas as pd
```

In [5]:

```
os.getcwd()
```

Out[5]:

```
'C:\\Users\\asus'
```

In [9]:

```
os.chdir("C:\\Users\\asus\\Desktop")
```

In [11]:

```
data=pd.read_csv("Salary.csv")
```

In [13]:

```
data.head()
```

Out[13]:

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

In [15]:

```
data.head(50)
```

Out[15]:

	YearsExperience	Salary
0	1.1	39343
1	1.3	46205
2	1.5	37731

	YearsExperience	Salary
3	2.0	43525
4	2.2	39891
5	2.9	56642
6	3.0	60150
7	3.2	54445
8	3.2	64445
9	3.7	57189
10	3.9	63218
11	4.0	55794
12	4.0	56957
13	4.1	57081
14	4.5	61111
15	4.9	67938
16	5.1	66029
17	5.3	83088
18	5.9	81363
19	6.0	93940
20	6.8	91738
21	7.1	98273
22	7.9	101302
23	8.2	113812
24	8.7	109431
25	9.0	105582
26	9.5	116969
27	9.6	112635
28	10.3	122391
29	10.5	121872
30	11.2	127345
31	11.5	126756
32	12.3	128765
33	12.9	135675
34	13.5	139465

In [17]:

```
data.tail()
```

Out[17]:

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

In [19]:

```
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 [21]:

```
data.describe()
```

Out[21]:

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

In [23]:

```
data.shape
```

Out[23]:

```
(35, 2)
```

In [25]:

```
data.size
```

Out[25]:

```
70
```

In [27]:

```
data.ndim
```

Out[27]:

2

In [29]:

```
data.isnull().sum()
```

Out[29]:

```
YearsExperience    0
Salary             0
dtype: int64
```

In [31]:

```
X = data.iloc[:, :-1].values
y = data.iloc[:, -1].values
```

In [33]:

```
data.isnull()
```

Out[33]:

	YearsExperience	Salary
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

	YearsExperience	Salary
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 [35]:
```
data.isnull().any()
```

Out[35]:
```
YearsExperience    False
Salary             False
dtype: bool
```

In [37]:
```
a="Ashish"
```

In [39]:
```
print(a)
```

```
Ashish
```

In [41]:
```
a[0]
```

Out[41]:
```
'A'
```

In [43]:
```
a[-1]
```

Out[43]:
```
'h'
```

In [45]:
```
a[1:4]
```

Out[45]:
```
'shi'
```

In [47]:
```
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 [49]:

```
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 [51]:

```
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
```

In [52]:

```
#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 [53]:

```
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 [54]:

```
print(X_test)
```

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

In [59]:

```
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 [61]:

```
print(y_test)
```

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

In [63]:

```
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
lr.fit(X_train, y_train)
```

Out[63]:

▾ LinearRegression

```
LinearRegression()
```

In [65]:

```
m = lr.coef_
```

In [67]:

```
print("Coefficient  :" , m)
```

```
Coefficient  : [8555.33918938]
```

In [69]:

```
#Assigning Y-intercept to a
c = lr.intercept_
```

In [71]:

```
print("Intercept : ", c)
```

```
Intercept :  29602.07353482097
```

In [73]:

```
lr.score(X_test,y_test) * 100
```

Out[73]:

```
91.71426108885095
```

In [ ]: