```
In [1]: #Name :Ankita Gulde
        #Roll no :44
        #section : 3A
```

```
In [ ]: #Aim : To perform simple linear regression
```

```
In [2]: import pandas as pd
```

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

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

Out[4]: 'C:\\Users\\HP'

In [5]: ```python
os.chdir("C:\\Users\\HP\\Desktop")
```

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

In [8]: ```python
df
```

Out[8]:

| | YearsExperience | Salary |
|---|---|---|
| 0 | 1.1 | 39343 |
| 1 | 1.3 | 46205 |
| 2 | 1.5 | 37731 |
| 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 |

|     | YearsExperience | Salary |
| --- | --- | --- |
| 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 [9]:
```python
df.head()
```

Out[9]:

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

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

Out[10]:

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

In [11]:
```
df.shape
```

Out[11]:
```
(35, 2)
```

In [12]:
```
df.size
```

Out[12]:
```
70
```

In [13]:
```
df.info()
```

In [11]:
```
df.shape
```

Out[11]:
```
(35, 2)
```

```
<class 'pandas.core.frame.DataFrame'> RangeIndex: 35 entries,
0 to 34
Data columns (total 2 columns):
 #   Column            Non-Null Count    Dtype
---  ------            --------------    ------

 1.  YearsExperience   35 non-null       float64
 2.  Salary            35 non-null       int64
dtypes: float64(1), int64(1)
memory usage: 692.0 bytes
```

In [14]:

```
df.describe()
```

Out[14]:

	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 [15]:

```
df.ndim
```

Out[15]:
2

In [16]:

```
df.isnull()
```

Out[16]:

	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

	YearsExperience	Salary
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]:
```python
df.isnull().any()
```

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

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

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

In [19]:
```python
a="ashish"
```

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

ashish

In [21]: a[0]

Out[21]: 'a'

In [22]: a[-1]

Out[22]: 'h'

In [23]: a[1:3]

Out[23]: 'sh'

In [25]:
```
a[1:4]
```

Out[25]: 'shi'

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

#X = df['YearsExperience']
#y = df['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 [29]:
```
import matplotlib.pyplot as plt
import seaborn as sns

import numpy as np
```

In [37]:
```python
#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 [39]:
```python
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 [40]:
```python
print(X_test)
```

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

In [41]:
```python
print(y_train)
```

[13567	39343	3989	83088 112635	56642	5579	46205	3773	12876
5		1			4		1	5
43525	12734	113812	126756	10130	8136	10558	6015	9173
	5		63218	2	3	2	0	8
139465	54445	61111 122391]						

```
In [42]: print(y_test)
```

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

```
In [43]: from sklearn.linear_model import LinearRegression lr =
         LinearRegression()
         lr.fit(X_train, y_train)
```

Out[43]:
```
▾ LinearRegression
LinearRegression()
```

In [44]:
```python
#Assigning Coefficient (slope) to m
m = lr.coef_
```

In [45]:
```python
print("Coefficient            :" , m)
```
```
Coefficient       : [8555.33918938]
```

In [46]:
```python
#Assigning Y-intercept to a
c = lr.intercept_
```

In [47]:
```python
print("Intercept : ", c)
```
```
Intercept :        29602.07353482097
```

In [48]:
```python
lr.score(X_test,y_test) * 100
```

Out[48]: 91.71426108885095

In [ ]: