# simple linear regression

In [2]:
```
#Name : Ashwini V Kayande
#Roll No : 60
#Section : 3A
#Date :05/10/2024
```

In [4]:
```
#Aim : To perform simple linear regression
```

In [6]:
```
import pandas as pd
```

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

In [10]:
```
os.getcwd()
```

Out[10]:
```
'C:\\Users\\user'
```

In [12]:
```
os.chdir("C:\\Users\\user\\desktop")
```

In [14]:
```
df=pd.read_csv("salary.csv")
```

In [16]:
```
df
```

Out[16]:

| | 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 |
| **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 |

| | YearsExperience | Salary |
|---|---|---|
| **33** | 12.9 | 135675 |
| **34** | 13.5 | 139465 |

In [18]:
```
df.head()
```

Out[18]:

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

In [20]:
```
df.tail()
```

Out[20]:

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

In [22]:
```
df.shape
```

Out[22]: (35, 2)

In [24]:
```
df.size
```

Out[24]: 70

In [26]:
```
df.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 [28]:
```
df.describe()
```

Out[28]:

| | 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 [30]:
```
df.ndim
```

Out[30]: 2

In [32]:
```
df.isnull()
```

Out[32]:

| | 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 |
| **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 |

| | YearsExperience | Salary |
|---|---|---|
| **33** | False | False |
| **34** | False | False |

In [34]:
```
df.isnull().any()
```

Out[34]:
```
YearsExperience    False
Salary             False
dtype: bool
```

In [36]:
```
df.isnull().sum()
```

Out[36]:
```
YearsExperience    0
Salary             0
dtype: int64
```

In [38]:
```
a="ashish"
```

In [40]:
```
print(a)
```

```
ashish
```

In [42]:
```
a[0]
```

Out[42]:
```
'a'
```

In [46]:
```
a[-1]
```

Out[46]:
```
'h'
```

In [48]:
```
a[1:3]
```

Out[48]:
```
'sh'
```

In [50]:
```
a[1:4]
```

Out[50]:
```
'shi'
```

In [52]:
```
#Assiging values in X & Y
x = df.iloc[:, :-1].values
y = df.iloc[:, -1].values
#X = df['YearsExperience']
#y = df['Salary']
```

In [54]:
```
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 [56]:
```
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 [58]:
```
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
```

In [67]:
```
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_sta
```

In [69]:
```
print(X_train)
```

```
[[ 3.7]
 [ 6. ]
 [ 9.5]
 [ 3.2]
 [ 7.1]
 [ 1.1]
 [ 3.9]
 [13.5]
 [11.5]
 [ 4.1]
 [ 2.9]
 [ 6.8]
 [11.2]
 [ 7.9]
 [ 5.1]
 [ 4. ]
 [10.3]
 [ 2. ]
 [ 1.5]
 [ 5.3]
 [10.5]
 [ 3. ]
 [ 9.6]
 [ 4. ]]
```

In [71]:
```
print(X_test)
```

```
[[ 3.2]
 [ 4.9]
 [ 4.5]
 [ 5.9]
 [ 8.7]
 [ 9. ]
 [12.3]
 [ 1.3]
 [ 8.2]
 [ 2.2]
 [12.9]]
```

In [73]:
```
print(y_train)
```

```
[ 57189  93940 116969  64445  98273  39343  63218 139465 126756  57081
  56642  91738 127345 101302  66029  56957 122391  43525  37731  83088
 121872  60150 112635  55794]
```

In [75]:
```
print(y_test)
```

```
[ 54445  67938  61111  81363 109431 105582 128765  46205 113812  39891
 135675]
```

In [77]:
```
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
lr.fit(X_train, y_train)
```

Out[77]:
```
▼ LinearRegression
LinearRegression()
```

In [79]:
```
#Assigning Coefficient (slope) to m
m = lr.coef_
```

In [81]:
```
print("Coefficient :" , m)
```

```
Coefficient : [8819.02108191]
```

In [83]:
```
#Assigning Y-intercept to a
c = lr.intercept_
```

In [85]:
```
print("Intercept : ", c)
```

```
Intercept :  28804.524425100586
```

In [87]:
```
lr.score(X_test,y_test) * 100
```

Out[87]:
```
95.84189499872238
```

In [ ]:
```

```