In [30]:

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
import seaborn as sns
```

In [34]:

```
db=pd.read_csv("Salary.csv")
db.shape
```

Out[34]:

```
(35, 2)
```

In [37]:

```
db.head()
```

Out[37]:

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

In [38]:

```
X=db["YearsExperience"]
y=db["Salary"]
db.head()
```

Out[38]:

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

In [39]:

```
##import random
##list1 =[1,2,3,4,5];
##random.shuffle (list1)
##print("the reshuffled list: ",list1)
```

In [40]:

```
from sklearn.model_selection import train_test_split
#using train test split function
X_train,X_test,y_train,y_test = train_test_split(X,y,random_state=42,test_size=0.2)
```

In [41]:

```
#printing train and test sets
print('X_train :\n',X_train.head())
print('X_test :\n',X_test.head())
print('y_train :\n',y_train.head())
print('y_test :\n',y_test.head())
```

```
X_train :
 12     4.0
8       3.2
16      5.1
9       3.7
33     12.9
Name: YearsExperience, dtype: float64
X_test :
 26     9.5
13     4.1
24     8.7
21     7.1
15     4.9
Name: YearsExperience, dtype: float64
y_train :
 12     56957
8       64445
16      66029
9       57189
33     135675
Name: Salary, dtype: int64
y_test :
 26     116969
13      57081
24     109431
21      98273
15      67938
Name: Salary, dtype: int64
```

In [42]:

```
##X[0].shape
```

In [43]:

```
X_train.shape[0]+X_test.shape[0]+y_train.shape[0]+y_test.shape[0]
```

Out[43]:

```
70
```

In [44]:

```
X_train.head()
```

Out[44]:

```
12     4.0
8      3.2
16     5.1
9      3.7
33    12.9
Name: YearsExperience, dtype: float64
```

In [45]:

```
X_train.shape
```

Out[45]:

```
(28,)
```

In [46]:

```
X_test.shape
```

Out[46]:

```
(7,)
```

In [47]:

```
y_train.shape
```

Out[47]:

```
(28,)
```

In [48]:

```
y_test.shape
```

Out[48]:

```
(7,)
```

In [49]:

```
X_train.head()
```

Out[49]:

```
12     4.0
8      3.2
16     5.1
9      3.7
33    12.9
Name: YearsExperience, dtype: float64
```

In [50]:

```
type(X)
```

Out[50]:

```
pandas.core.series.Series
```

In [51]:

```
type(y)
```

Out[51]:

```
pandas.core.series.Series
```

In [52]:

```
X_train=np.array(X_train)
X_test=np.array(X_test)
y_train=np.array(y_train)
y_test=np.array(y_test)
```

In [53]:

```
X_train.shape
```

Out[53]:

```
(28,)
```

In [54]:

```
type(X_train)
```

Out[54]:

```
numpy.ndarray
```

In [55]:

```
X_train=X_train.reshape(-1,1)
X_test=X_test.reshape(-1,1)
y_train=y_train.reshape(-1,1)
y_test=y_test.reshape(-1,1)
```

SIMPLE LINEAR REGRESSION

In [56]:

```
from sklearn.linear_model import LinearRegression
```

In [57]:

```
model=LinearRegression()
model.fit(X_train,y_train)
print(model.intercept_)
```

```
[29078.62603441]
```

In [58]:

```
model.score(X_train,y_train)
```

Out[58]:

```
0.9716360897963994
```

In [59]:

```
model.coef_
```

Out[59]:

```
array([[8578.76747669]])
```

In [60]:

```
##make prediction using the testing set
y_pred=model.predict(X_test)
```

In [61]:

```
model.coef_
```

Out[61]:

```
array([[8578.76747669]])
```

In [62]:

```
from sklearn.metrics import mean_absolute_error , r2_score , mean_absolute_error as mae
mae= mean_absolute_error(y_test,y_pred)
print(mae)
```

```
6692.364094497284
```

In [63]:

```
##squared true returns MSE value,False returns RMSE value.
mse = mean_squared_error(y_true=y_test,y_pred=y_pred,squared=False)
print(mse)
```

```
7467.381824057221
```