Simple Linear Regression

Concept of linear regresion

we can perform linear regresion on salary vs no of year of experiance
y = b0+bx
b0- line intersect to y
b1-slop of the line
x-observation
or
y = mx + c
where
y = dependant variable
x = independant variable

we simply plot a linear relation between x vs y and try to cover up maximum points in linear regression

Simple linear intuition (best fitting line)

connect maximum points with the linear line vertically and **sum(yi-y^)^2** and minimum sum of squre that line is best fitting line
yi-experiance
y^-salary

In [1]:
```
#importing the libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [30]:
```
#importing data set
```

```
data_set = pd.read_csv('salary_Data.csv')
print(data_set.head(4))
print(data_set.shape)
print(data_set.tail(6))
```

```
   YearsExperience   Salary
0              1.1  39343.0
1              1.3  46205.0
2              1.5  37731.0
3              2.0  43525.0
(30, 2)
    YearsExperience    Salary
24              8.7  109431.0
25              9.0  105582.0
26              9.5  116969.0
27              9.6  112635.0
28             10.3  122391.0
29             10.5  121872.0
```

In [4]:
```
X = data_set.iloc[:,:-1].values #independant variable
y = data_set.iloc[:,1].values
print(X.shape)
print(y.shape)
```

```
(30, 1)
(30,)
```

In [19]:
```
#Spliting the dataset into training and testing data set
from sklearn.model_selection import train_test_split
X_train, X_test ,y_train, y_test= train_test_split(X,y ,test_size = 1/3
,random_state = 0)
print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(20, 1)
(10, 1)
```

```
(20,)
(10,)
```

#feature Scaling from sklearn.preprocessing import StandardScaler sc_x = StandardScaler() x_train = sc_x.fit_transform(X) x_test = sc_x.fit(x_test) sc_y = StandardScaler() y_train = sc_y.fit_transform(y_train) y_train = sc_y.fit(y_train) print(x_tarin.shape) print(y_train.shape)#Fit means our simple linear regression model learn the or fit the trainning data in regression to predict our test data set here it predict no of salary on the basis of no year experiance

In [22]:
```
# we can import libraries from sci-kit learn called linear model for li
near regression and we create a class called as regressor
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
#class contains some method we can fit using this method in linear regr
ession
regressor.fit(X_train,y_train)
```

Out[22]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=F
alse)
```

Predicting the test set result

In [45]:
```
y_pred =regressor.predict(X_test)
print(y_pred)
```

```
[ 40835.10590871 123079.39940819  65134.55626083  63265.36777221
 115602.64545369 108125.8914992  116537.23969801  64199.96201652
  76349.68719258 100649.1375447 ]
```

Visualising the training set result

In [52]:
```
#we can use scatter plot to visualising data
plt.scatter(X_train,y_train,color = 'red') #we can take train data set
 to predict experiance vs salary
plt.plot(X_train,regressor.predict(X_train),color = 'blue') #we want to
 prdict regrssion value after training data set
```

```
plt.xlabel('Year of Experiance')
plt.ylabel('Salary')
plt.title('Experiance VS Salary')
plt.show()
```

Summery

1.Red dot indicates real values of year of experiance and salary.
2.pridected values indicate the blue linear line
3.for predicting the values red dots are projected to blue linear line and we can check salary vs year of experiance

Visualising the test set result

In [60]:
```
plt.scatter(X_test,y_test,color = 'red') #we can check the result of te
st data set which can we train on training data set
plt.plot(X_train,regressor.predict(X_train),color = 'blue')#here we can
 not change the train data set because test data set allready train on
```

```
 this data so no need to change
#we can use scatter plot to visualising data

plt.xlabel('Year of Experiance')
plt.ylabel('Salary')
plt.title('Year of Experiance VS Salary')
plt.show()
```

Observation

1.predicited salary is approximatly equal to salary of employee