In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import sklearn
from sklearn.metrics import mean_squared_error,mean_absolute_error
```

In [2]:
```
dataset = pd.read_csv("http://bit.ly/w-data")#importing the data
dataset.shape #gives no of rows and columns
```

Out[2]:
```
(25, 2)
```

In [3]:
```
dataset.head(25)#displaying the data
```

Out[3]:

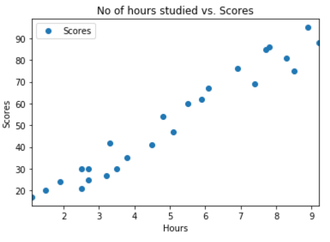
	Hours	Scores
0	2.5	21
1	5.1	47
2	3.2	27
3	8.5	75
4	3.5	30
5	1.5	20
6	9.2	88
7	5.5	60
8	8.3	81
9	2.7	25
10	7.7	85
11	5.9	62
12	4.5	41
13	3.3	42
14	1.1	17
15	8.9	95
16	2.5	30
17	1.9	24
18	6.1	67
19	7.4	69
20	2.7	30
21	4.8	54
22	3.8	35
23	6.9	76
24	7.8	86

In [4]:
```
dataset.describe()#statistical details of data
```

Out[4]:

	Hours	Scores
count	25.000000	25.000000
mean	5.012000	51.480000
std	2.525094	25.286887
min	1.100000	17.000000
25%	2.700000	30.000000
50%	4.800000	47.000000
75%	7.400000	75.000000
max	9.200000	95.000000

In [5]:
```
dataset.plot(x='Hours', y='Scores', style='o')#plotting a graphx
plt.title('No of hours studied vs. Scores')
plt.xlabel('Hours')
plt.ylabel('Scores')
plt.show()
```



In [6]:
```
x=dataset.iloc[:,:-1].values
y=dataset.iloc[:,1].values
```

In [7]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)
```

In [8]:
```
print(x_train.shape)
print(y_train.shape)
print(x_test.shape)
print(y_test.shape)
```

```
(20, 1)
(20,)
(5, 1)
(5,)
```

In [9]:
```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(x_train,y_train)
```

Out[9]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,
         normalize=False)
```

In [10]:
```
print("Intercept ",regressor.intercept_)
print("Coefficient ",regressor.coef_)
```

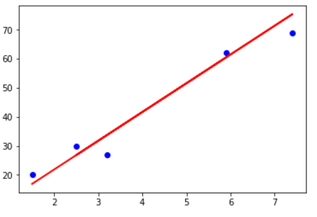
```
Intercept  2.018160041434683
Coefficient  [9.91065648]
```

In [15]:
```
y_pred = regressor.predict(x_test)
df = pd.DataFrame({'Actual':y_test.flatten(),'Predicted':y_pred.flatten()})
df
```

Out[15]:

	Actual	Predicted
0	20	16.884145
1	27	33.732261
2	69	75.357018
3	30	26.794801
4	62	60.491033

In [16]:
```
plt.scatter(x_test,y_test,color='blue')
plt.plot(x_test,y_pred,color='red',linewidth=2)
plt.show()
```



In [17]:
```
from sklearn import metrics
print('Mean Absolute Error:',metrics.mean_absolute_error(y_test,y_pred))
print('Mean Squared Error:',metrics.mean_squared_error(y_test,y_pred))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
```

```
Mean Absolute Error: 4.183859899002975
Mean Squared Error: 21.5987693072174
Root Mean Squared Error: 4.6474476121003665
```