Name: Pratimesh Shivaji Gaykawad

In this task we will predict the percentage of marks that student is expected to score based upon number of hours they studied.This is a simple linear regression task as it involves just 2 variables.

Data can be found at http://bit.ly/w-data

In [22]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [23]:
```
#set up url to dataset and read data
url="http://bit.ly/w-data"
data = pd.read_csv(url)
```

In [24]:
```
#show data
data
```

Out[24]:

	Hours	Scores
0	2.5	21
1	5.1	47
2	3.2	27

	Hours	Scores
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 [25]:

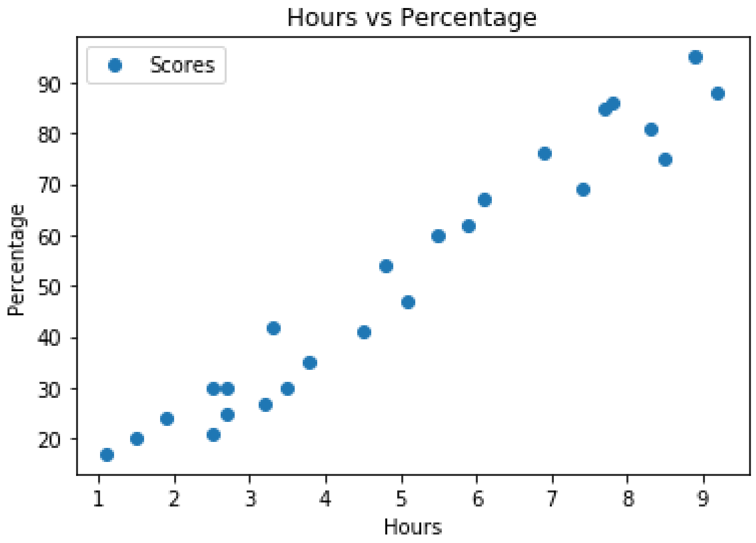
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 25 entries, 0 to 24
Data columns (total 2 columns):
 #   Column   Non-Null Count  Dtype  
---  ------   --------------  -----  
 0   Hours    25 non-null     float64
 1   Scores   25 non-null     int64  
dtypes: float64(1), int64(1)
memory usage: 528.0 bytes
```

In [26]:

```
#plotting scatter plot
data.plot(x='Hours',y='Scores',style='o')
plt.title('Hours vs Percentage')
plt.xlabel('Hours')
plt.ylabel('Percentage')
plt.show()
```

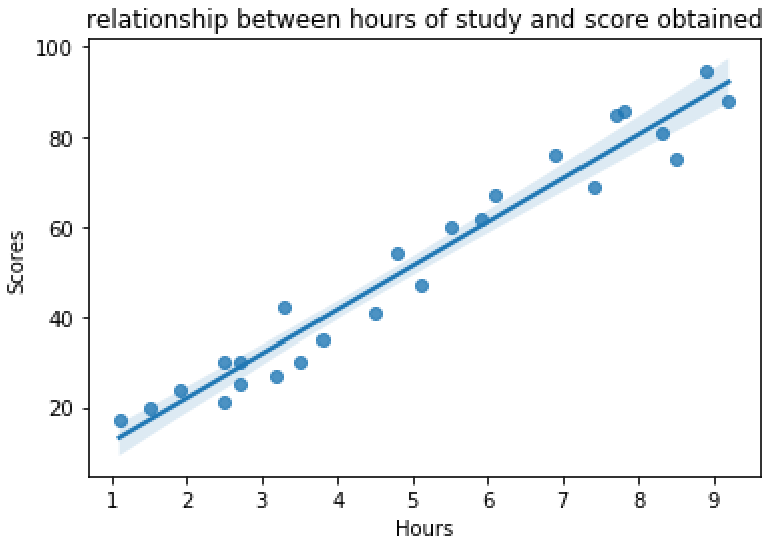


In [27]:

```
#plotting regression plot
sns.regplot(x=data['Hours'],y=data['Scores'])
plt.title('relationship between hours of study and score obtained')
```

Out[27]:

```
Text(0.5, 1.0, 'relationship between hours of study and score obtaine
d')
```



In [28]:
```
X = data.iloc[:, :-1].values
y = data.iloc[:, 1].values
```

In [29]:
```
#splitting data into train and test data set
from sklearn.model_selection import train_test_split
```

In [30]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2
, random_state=0)
```

In [31]:
```
from sklearn.linear_model import LinearRegression
```

In [32]:
```
reg = LinearRegression()
```

In [33]:
```
#model training
reg.fit(X_train, y_train)
```

Out[33]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normaliz
e=False)
```

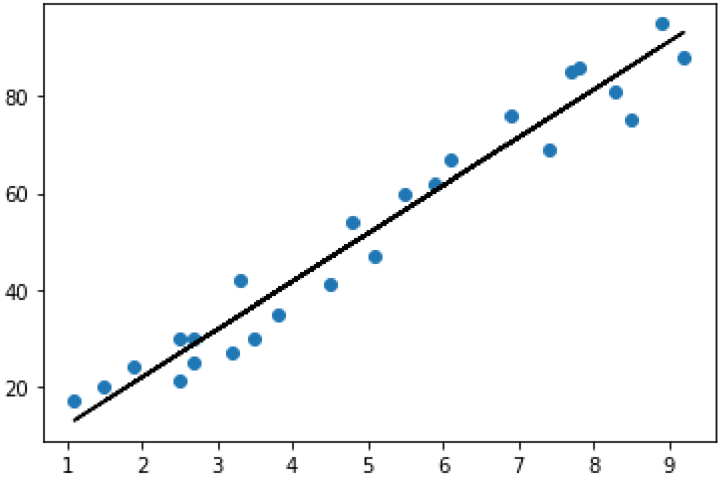
In [34]:

```
y_pred = reg.predict(X_test)
```

In [35]:

```
line = reg.coef_*X + reg.intercept_

#plotting for test data
plt.scatter(X , y)
plt.plot(X,line,color='black')
plt.show()
```



In [36]:

```
print('Training Score')
print(reg.score(X_train, y_train))
print('Test Score')
print(reg.score(X_test, y_test))
```

```
Training Score
0.9515510725211553
Test Score
0.9454906892105356
```

In [37]:

```
print("No of Hours =",9.25)
print("Predicted score =",reg.predict([[9.25]]))
```

```
No of Hours = 9.25
```

```
Predicted score = [93.69173249]
```

In [39]:

```
from sklearn import metrics
print('MAE: ',metrics.mean_absolute_error(y_test, y_pred))
print('MSE: ',metrics.mean_squared_error(y_test, y_pred))
print('RMSE: ',np.sqrt(metrics.mean_absolute_error(y_test, y_pred)))
```

```
MAE:  4.183859899002975
MSE:  21.5987693072174
RMSE:  2.0454485813637495
```