DataScience and Business Analytics Intership

Import all the libraries

In [25]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

Read the data from csv file and visualization:

In [26]:

```
data = pd.read_csv(r"http://bit.ly/w-data")
```

In [27]:

```
data.head()
```

Out[27]:

	Hours	Scores
0	2.5	21
1	5.1	47
2	3.2	27
3	8.5	75
4	3.5	30

Discover and visualize the data to gain insights.

In [28]:

```
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 [29]:

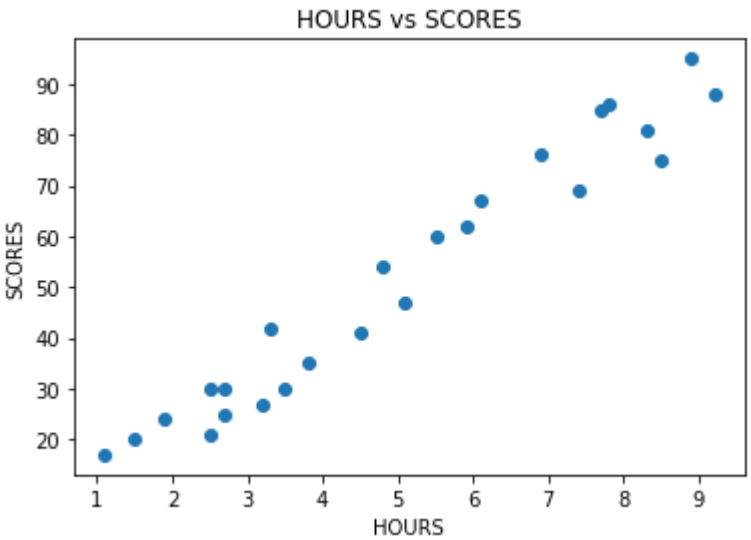
```
data.describe()
```

Out[29]:

	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 [30]:

```
plt.scatter(x=data.Hours,y=data.Scores)
plt.xlabel("HOURS")
plt.ylabel("SCORES")
plt.title("HOURS vs SCORES")
plt.show()
```



PREPARE THE DATA FOR MACHINE LEARNING ALGORITHM

In [31]:

```
data.columns
```

Out[31]:

```
Index(['Hours', 'Scores'], dtype='object')
```

In [32]:

```
data.shape
```

Out[32]:

```
(25, 2)
```

In [33]:

```
# Split the data for train and test.

train,test = train_test_split(data,test_size=0.25,random_state=123)
```

In [34]:

```
train.shape
```

Out[34]:

```
(18, 2)
```

In [35]:

```
test.shape
```

Out[35]:

```
(7, 2)
```

In [36]:

```
train_x= test.drop("Scores",axis=1)
train_y = train["Scores"]
```

In [37]:

```
test_x= test.drop("Scores",axis=1)
test_y = test["Scores"]
```

Step 4:Training the Algorithm

We have to split our data into training and testing sets,and now is finally the time to train our algorithm.

In [62]:

```
lr = LinearRegression()
```

In [64]:

```
lr.fit(train_x,test_y)
```

Out[64]:

```
LinearRegression()
```

In [65]:

```
lr.coef_
```

Out[65]:

```
array([9.94850632])
```

In [66]:

```
lr.intercept_
```

Out[66]:

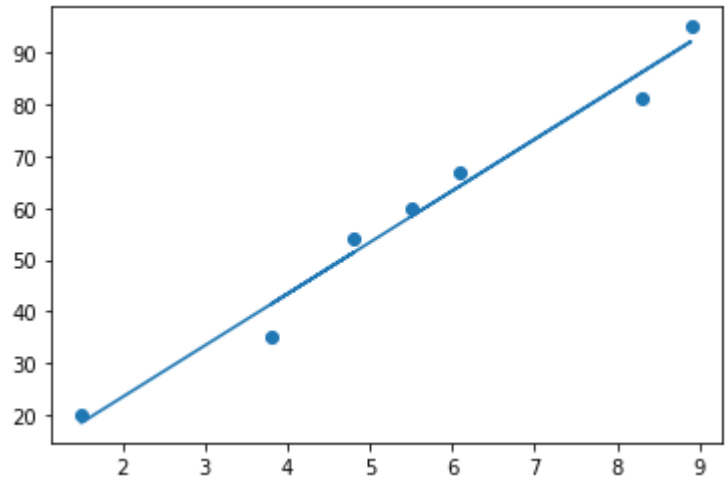
```
3.5718720327222186
```

In [70]:

```
#Plotting the regression line #formula for line is y=m*x=c

line = lr.coef_*train_x + lr.intercept_

#Plotting for the test data
plt.scatter(train_x,test_y)
plt.plot(train_x,line);
plt.show()
```



Making Predictions

In [71]:

```
pr = lr.predict(test_x)
```

In [72]:

```
list(zip(test_y,pr))
```

Out[72]:

```
[(20, 18.494631509750924),
 (54, 51.32470235921407),
 (35, 41.37619604119494),
 (67, 64.25776057263894),
 (95, 92.11357826309253),
 (81, 86.14447447228105),
 (60, 58.28865678182747)]
```

Evaluating the Model:

In [73]:

```
from sklearn.metrics import mean_squared_error
```

In [74]:

```
mean_squared_error(test_y,pr,squared=False)
```

Out[74]:

```
3.6902351539585694
```

Result

In [76]:

```
hour = [9.25]
own_pr= lr.predict([hour])
print("Number of Hours = {}".format([hour]))
print("Predicted Score = {}".format(own_pr[0]))
```

```
Number of Hours = [[9.25]]
Predicted Score = 95.59555547439922
```