In [4]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

import seaborn as sns
import scipy.stats as stats
import statsmodels.formula.api as smf
```

DATA IMPORT

In [6]:
```
df=pd.read_csv('https://raw.githubusercontent.com/AdiPersonalWorks/Random/master/studen
```

In [13]:
```
df
df.head()
```

Out[13]:

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

In [11]:
```
df.isnull().sum()
```

Out[11]:
```
Hours     0
Scores    0
dtype: int64
```

In [10]:
```
df.dtypes
```

Out[10]:
```
Hours     float64
Scores      int64
dtype: object
```

In [12]:
```
df.corr()
```

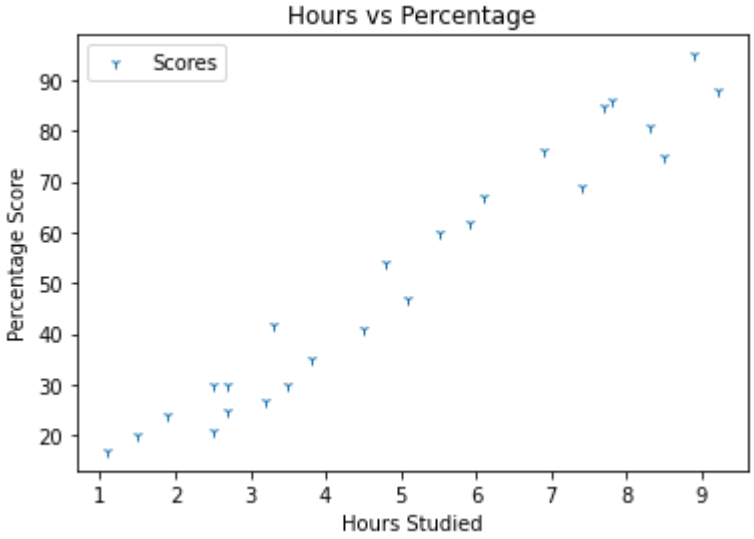
Out[12]:

	Hours	Scores
Hours	1.000000	0.976191
Scores	0.976191	1.000000

plot data

In [29]:
```
#Plotting the distribution of scores
df.plot(x='Hours', y='Scores', style='1')
plt.title('Hours vs Percentage')
plt.xlabel('Hours Studied')
```

```
plt.ylabel('Percentage Score')
plt.show()
```



from graph we can see that there is positive linear relationship between X & Y

preparing Data

defining the dependent and independent variables

In [26]:
```
# Using sklearn
from sklearn import linear_model
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score
```

In [27]:
```
# to divide input and output varibales.
X = df.iloc[:, :-1].values
Y = df.iloc[:, 1].values
```

Training the model

In [30]:
```
# Splitting the data into test and training set using train_test_split
X_train,X_test,Y_train,Y_test= train_test_split(X,Y, test_size= 0.2)
```

In [31]:
```
X_train.shape,Y_train.shape
```

Out[31]: ((20, 1), (20,))

In [32]:
```
X_test.shape,Y_test.shape
```

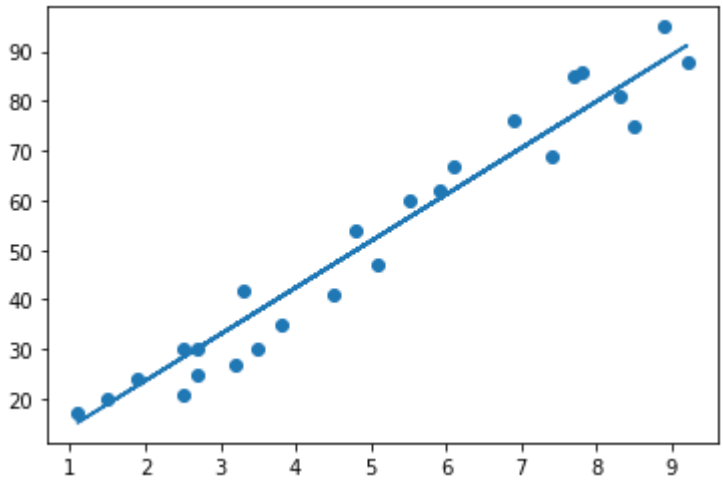
Out[32]: ((5, 1), (5,))

To fit the model

In [34]:
```
model= linear_model.LinearRegression()
model.fit(X_train,Y_train)
```

Out[34]:
```
LinearRegression()
```

In [37]:
```
#Plotting the regression line
line = model.coef_*X+model.intercept_
#Plotting for test data
plt.scatter(X,Y)
plt.plot(X,line);
plt.show()
```



In [38]:
```
# Predicting the scores for test set
y_pred= model.predict(X_test)
y_pred
```

Out[38]:
```
array([28.33401786, 30.21098214, 34.90339286, 78.07357143, 15.19526786])
```

In [39]:
```
# Comparing actual versus predicted
df = pd.DataFrame({'Actual': Y_test, 'Predicted': y_pred})
df
```

Out[39]:

	Actual	Predicted
0	21	28.334018
1	25	30.210982
2	27	34.903393
3	86	78.073571
4	17	15.195268

In [40]:
```
print('Coefficient:', model.coef_)
print('Intercept:', model.intercept_)
print('Mean Squared Error (MSE):%.2f'% mean_squared_error(Y_test,y_pred))
print('Coefficient of Determination (R^2): ', r2_score(Y_test,y_pred))
```

```
Coefficient: [9.38482143]
Intercept: 4.871964285714284
Mean Squared Error (MSE):41.90
Coefficient of Determination (R^2):  0.9362240929487871
```

In [46]:
```
#Let's predict the score for 9.25 hpurs
print('Score of student who studied for 9.25 hours a dat', Regressor.predict([[9.25]]))
```

```
Score of student who studied for 9.25 hours a dat [92.90985477]
```

In [ ]: