TASK 1
Prediction Using Supervised Machine learning

BY: MARY JOHN

In [71]:

```
# Importing the libraries required
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sn
%matplotlib inline
```

In [5]:

```
dt = pd.read_excel("StudentStudyHours.xlsx")
dt.head(6)
```

Out[5]:

	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

In [7]:

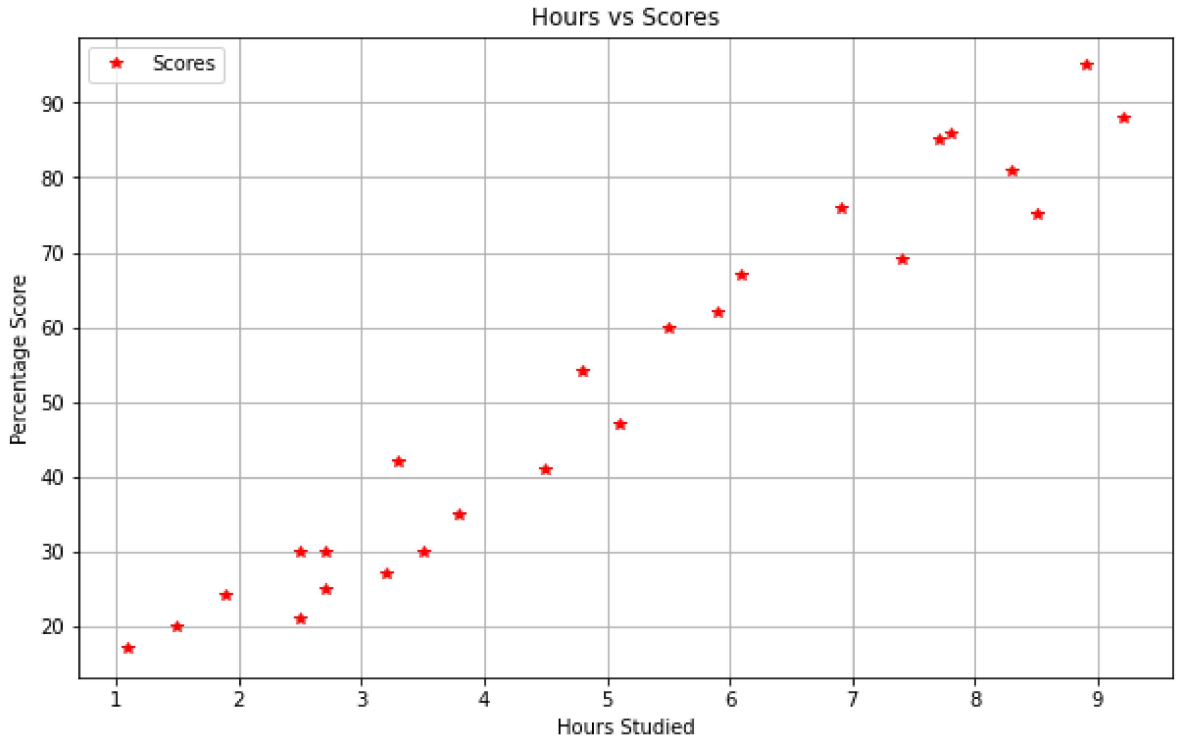
```
1 dt.describe()
```

Out[7]:

	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 [15]:

```
1 plt.rcParams["figure.figsize"] = (10,6)
2 dt.plot(x='Hours', y='Scores', style='*',color='red')
3 plt.title('Hours vs Scores')
4 plt.xlabel('Hours Studied')
5 plt.ylabel('Percentage Score')
6 plt.grid()
7 plt.show()
```



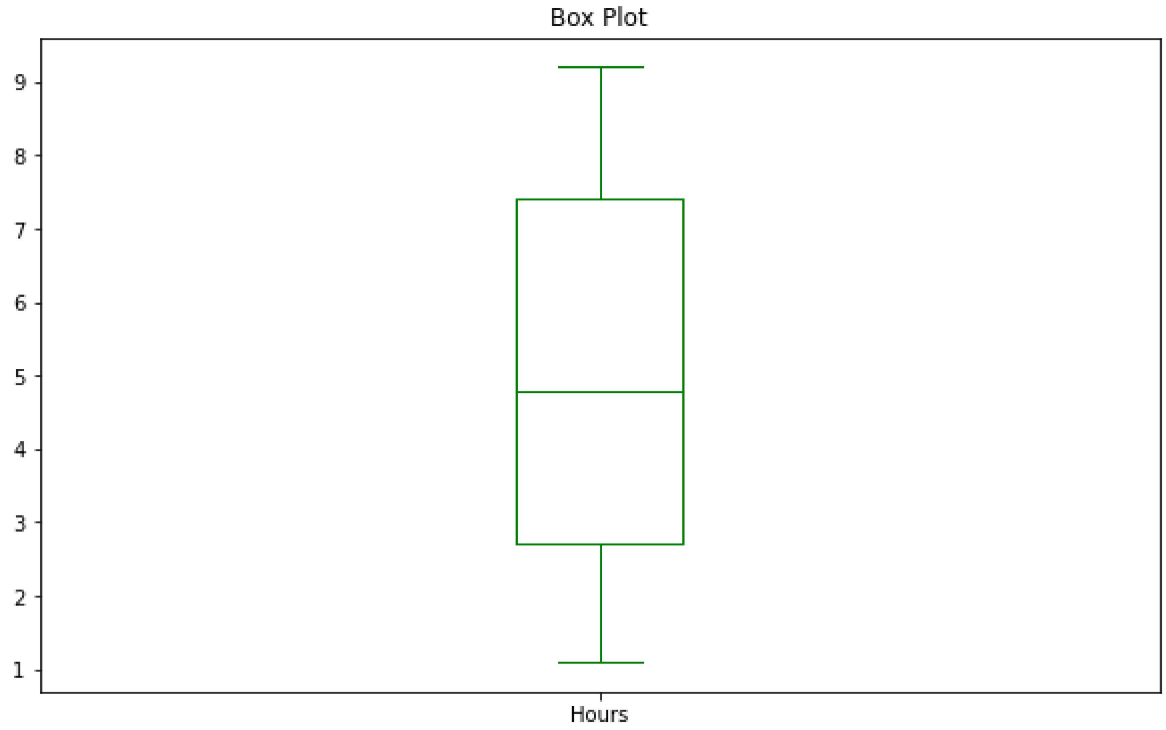
This shows that there is a positive relation between the hours studied by a student and the percentage that student scores

In [21]:

```
# BOXPLOT FOR NO. OF HOURS
dt.Hours.plot.box(color="green")
plt.title("Box Plot")
```

Out[21]:

```
Text(0.5, 1.0, 'Box Plot')
```



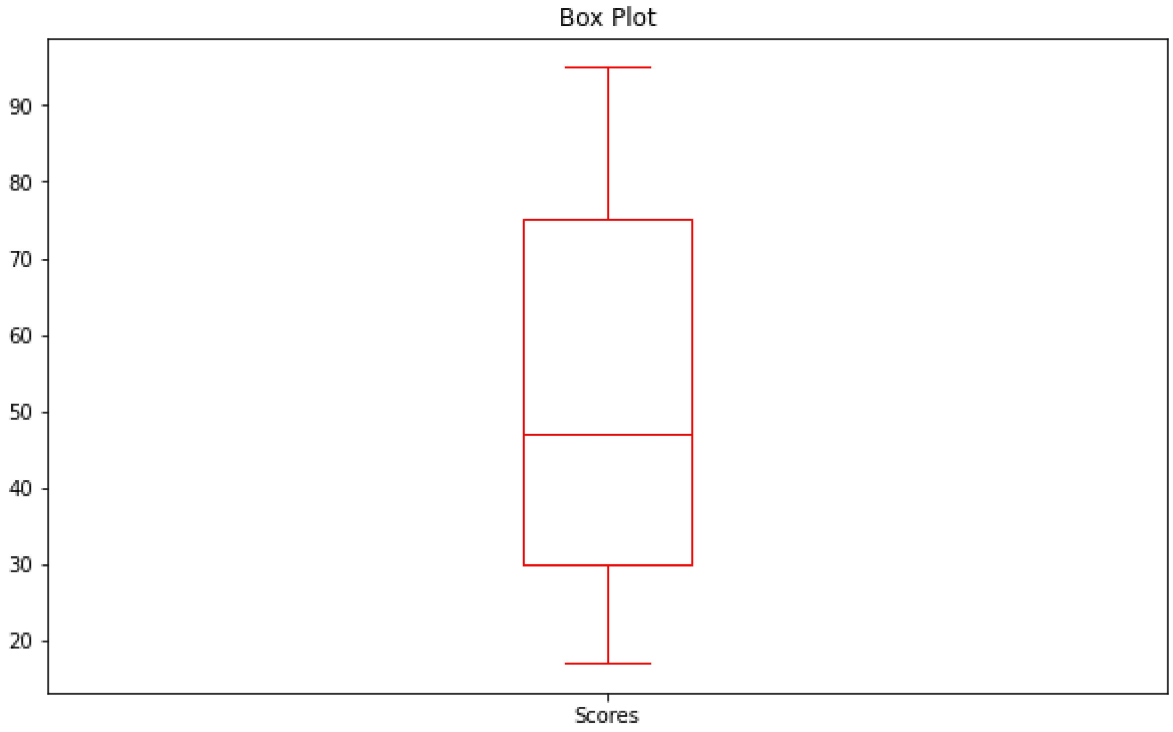
This shows that the median hours of study per day by a student is almost 5 hours. It also shows that there is no presence of outliers and that it is not normally distributed since median is not equal to mean.

In [19]:

```
# BOXPLOT FOR SCORES
dt.Scores.plot.box(color='red')
plt.title("Box Plot")
```

Out[19]:

```
Text(0.5, 1.0, 'Box Plot')
```

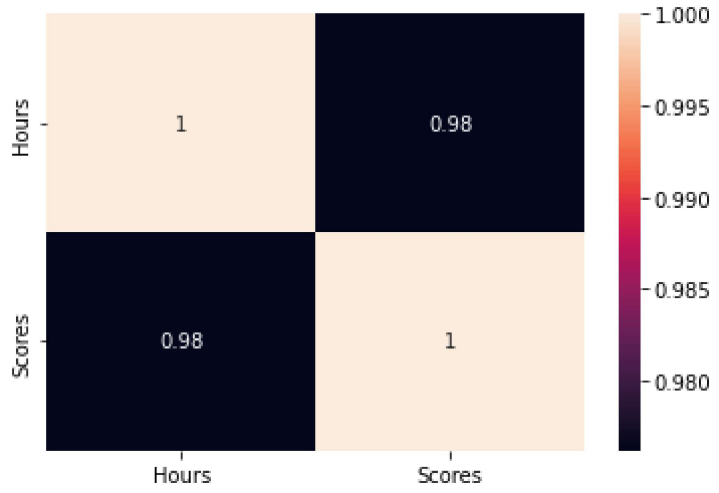


This shows that the percentage of score by a student is around 48%. It also shows that there is no presence of outliers and that it is not normally distributed since median is not equal to mean.

In [23]:

```
print("The correlation Heat Map is:")
corrMatrix = dt.corr()
sn.heatmap(corrMatrix, annot=True)
plt.show()
```

```
The correlation Heat Map is:
```



The correlation coeffiecient is 0.98. This implies that the hours of study and the percentage scored by a student is highly positively correlated.

Preparing the data

The next step is to divide the data into "attributes" (inputs) and "labels" (outputs).

In [54]:

```
X = dt.iloc[:, :-1].values
y = dt.iloc[:, 1].values
```

Now that we have our attributes and labels, the next step is to split this data into training and test sets. We'll do this by using Scikit-Learn's built-in train_test_split() method:

Splitting data into training and testing set

In [55]:

```
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 [57]:

```
print("Dimension of training set of Scores = ",X_train.ndim)
print("Dimension of training set of Hours = ",y_train.ndim)
```

```
Dimension of training set of Scores =  2
Dimension of training set of Hours =  1
```

Training the Algorithm

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

In [58]:

```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(x_train, y_train)

print("The training is complete.")
```

```
The training is complete.
```

Making prediction on training set and checking the RMSE

In [61]:

```
from sklearn.metrics import r2_score
y_pred=model.predict(X_test)
r2_score(y_test,y_pred)
```

Out[61]:

```
0.9454906892105354
```

In [63]:

```
from sklearn.metrics import mean_squared_error
mean_squared_error(y_test,y_pred,squared=False)
```

Out[63]:

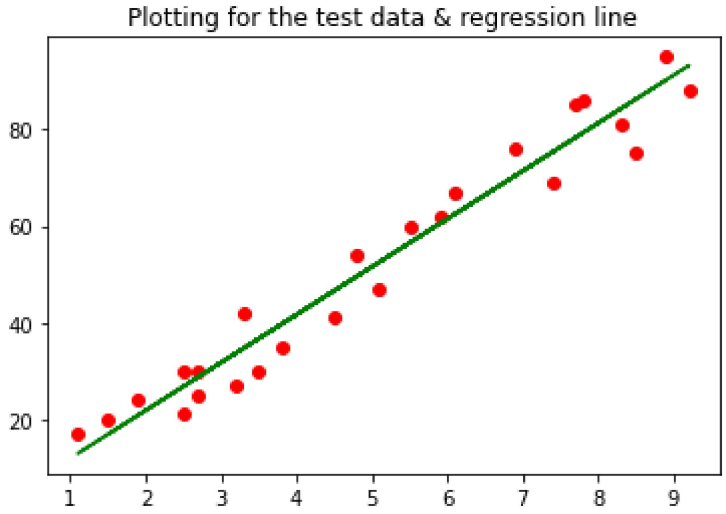
```
4.647447612100373
```

The model is has an accuracy score of 0.9454 that is it 94.54% good fit with it's RMSE of 4.64%

In [73]:

```
# Plotting the regression line
line = regressor.coef_*X + regressor.intercept_

# Plotting for the test data
plt.scatter(X, y,color="red")
plt.plot(X, line, color="green")
plt.title("Plotting for the test data & regression line")
plt.show()
```



PREDICTION

In [66]:

```
# Prediction of score if student studies 9.25 hours per day
hours = 9.25
model.predict([[hours]])
```

Out[66]:

```
array([93.69173249])
```

A student scores 93.69% if they study for 9.25 hours per day

In [ ]:

```

```

In [ ]:

```
1
```