Importing Required Libraries

```
In [3]: 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
        from sklearn import metrics
```
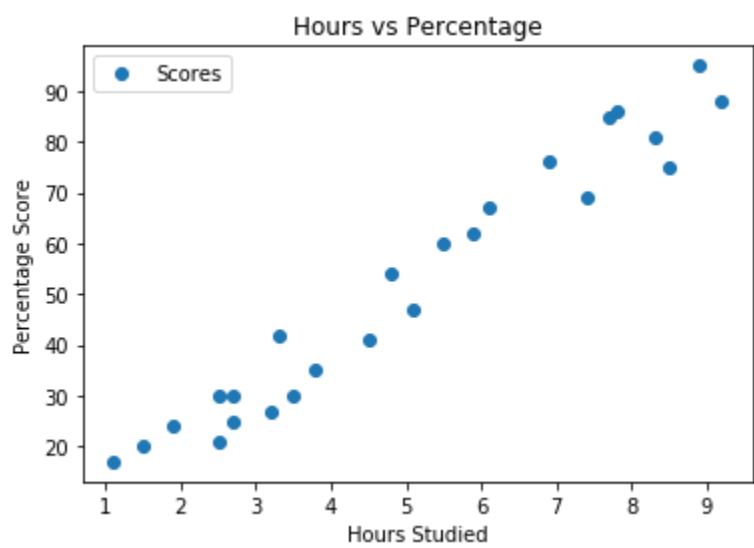
Importing the Dataset

```
In [4]: data = pd.read_csv("http://bit.ly/w-data")
        data.head(10)
```

Out[4]:

	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

Data Interpretation

```
In [5]: # Plotting the distribution of scores
        data.plot(x='Hours', y='Scores', style='o')
        plt.title('Hours vs Percentage')
        plt.xlabel('Hours Studied')
        plt.ylabel('Percentage Score')
        plt.show()
```



```
In [6]: #Preparing the data
        x = data.iloc[:, :-1].values
        y = data.iloc[:, 1].values
```

### Splitting the dataset into Training set and Testing set

```
In [11]: x_train, x_test, y_train, y_test = train_test_split(x, y,
                                    test_size=0.2, random_state=0)
```

Training the Dataset

```
In [15]: regressor = LinearRegression()
         regressor.fit(x_train, y_train)
```

Out[15]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)

```
In [16]: plt.scatter(x_train, y_train, color='red')
         plt.plot(x_train, regressor.predict(x_train), color='blue')
         plt.title('Hours vs Score (Training set)')
         plt.xlabel('Hours Studied')
         plt.ylabel('Percentage score')
         plt.show()
```
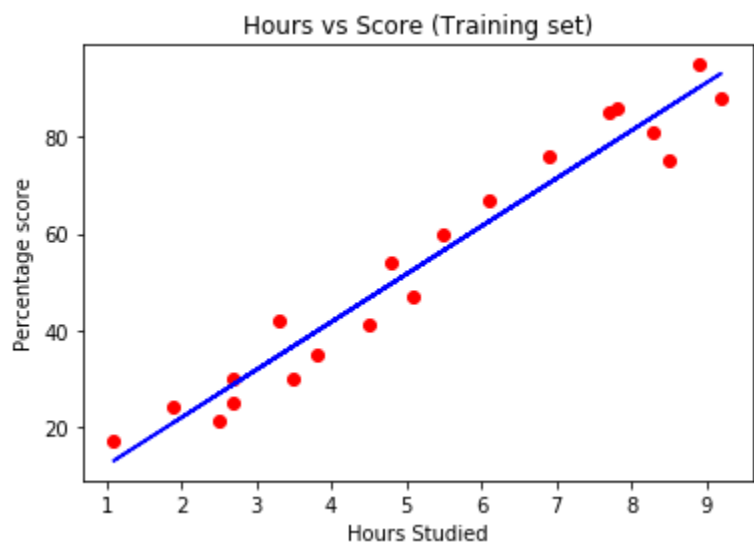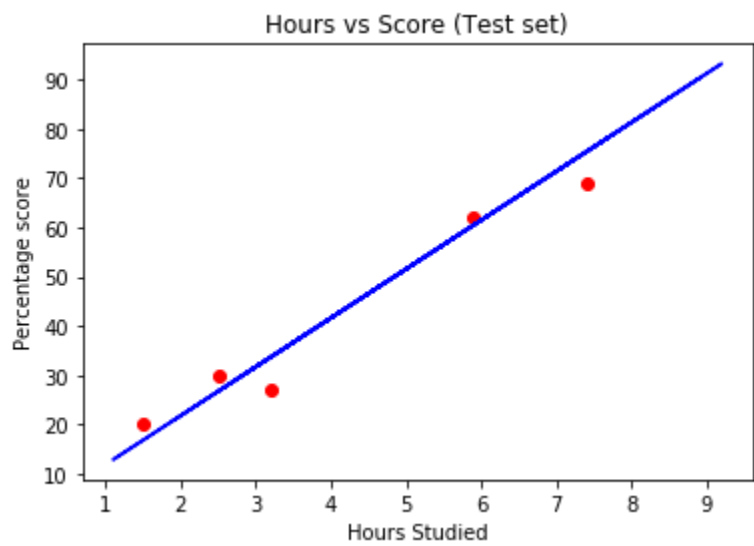

Testing the Dataset

```
In [18]: pred= regressor.predict(x_test)
         print(y_test[0:5], '\n', pred[0:5])
```

```
[20 27 69 30 62]
 [16.88414476 33.73226078 75.357018   26.79480124 60.49103328]
```

```
In [19]: plt.scatter(x_test, y_test, color='red')
         plt.plot(x_train, regressor.predict(x_train), color='blue')
         plt.title('Hours vs Score (Test set)')
         plt.xlabel('Hours Studied')
         plt.ylabel('Percentage score')
         plt.show()
```


Model Evaluation

```
In [20]: print('mean absolute error:',metrics.mean_absolute_error(y_test,pred))
         print('mean squared error:',metrics.mean_squared_error(y_test,pred))
         print('Root mean squared:',np.sqrt(metrics.mean_squared_error(y_test,pred)))
         print('R-2 Score:',metrics.r2_score(y_test,pred))
```

```
mean absolute error: 4.183859899002975
mean squared error: 21.5987693072174
Root mean squared: 4.6474476121003665
R-2 Score: 0.9454906892105356
```

Q)Predicted score if a student study for 9.25 hrs in a day

Ans:

```
In [17]: m = regressor.predict([[9.25]])
         print('Predicted score if a student study for 9.25 hrs in a day:',m)
```

```
Predicted score if a student study for 9.25 hrs in a day: [93.69173249]
```