Author: Rajdeep Saha

Task 1: Prediction using Supervised Machine Learning

GRIP @ The Sparks Foundation

In this regression task I have predicted the percentage of marks that a student is expected to score based upon the number of hours they studied. This is a simple linear regression problem as it has just two variables.

Step 1- Reading the Data

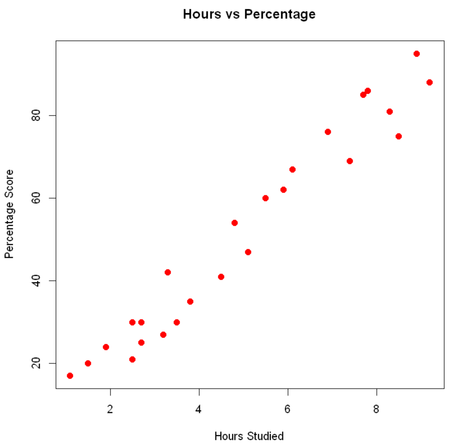
In [1]:
```
setwd("C:/Users/user/OneDrive/Desktop/Sparks")
data=c=read.csv("student_scores - student_scores.csv")
attach(data)
print("Data imported successfully")
head(data,10)
```

```
[1] "Data imported successfully"
```

Hours	Scores
2.5	21
5.1	47
3.2	27
8.5	75
3.5	30
1.5	20
9.2	88
5.5	60
8.3	81
2.7	25

Step 2- Input Data Visualization

In [2]:
```
x=data$Hours
y=data$Scores
plot(x,y,xlab="Hours Studied",ylab="Percentage Score", main="Hours vs Percentage",
     col="red",pch=20,cex=1.75)
```



From the graph, we can conclude that there exist a positive linear relation between the number of hours studied and percentage of score.

Step 3- Train- Test Split

In [3]:
```
set.seed(2021)
index=sample(1:nrow(data),size=floor(0.8*nrow(data)))
train.set=data[index,]
test.set=data[-index,]
print("Train-Test split successful.")
```

```
[1] "Train-Test split successful."
```

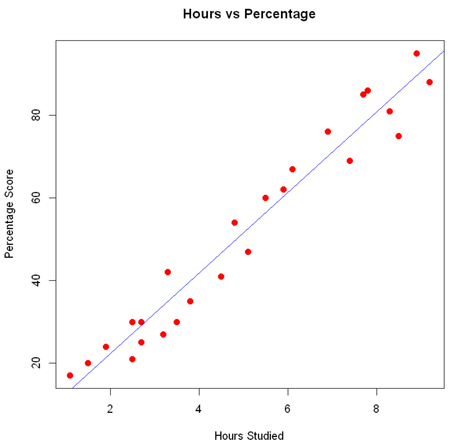
Step 4- Model Training

In [4]:
```
linear.model=lm(Scores~Hours,data=train.set)
summary=summary(linear.model)
print("Training complete.")
```

```
[1] "Training complete."
```

Step 5- Plotting the Line of Regression

In [5]:
```
plot(x,y,xlab="Hours Studied",ylab="Percentage Score",main="Hours vs Percentage",
     col="red",pch=20,cex=1.75)
abline(linear.model,col="blue")
```



Step 6- Making Predictions

After training our model, we can test the model by makng predictions using test set.

In [6]:
```
predicted=predict(linear.model, newdata=test.set,type="response")
predicted
actual=test.set[,2]
data.frame(actual,predicted)
```

1	27.1439825844081
8	56.387390125688
13	46.63958711928
15	13.4970590651442
21	29.0935430871601

	actual	predicted
1	21	27.14398
8	60	56.38739
13	41	46.63959
15	17	13.49706
21	30	29.09354

In [7]:
```
#Testing the model with given observation.
hours=9.25
predicted_score=predict(linear.model,data.frame(Hours=c(hours)))
paste("Numer of hours studied:",hours)
paste("Predicted score is:",predicted_score)
```

'Numer of hours studied: 9.25'
'Predicted score is: 92.9416495522879'

Step 7- Evaluating the Model

The final step is to evaluate the performance of algorithm. This step is particularly important to compare how well different algorithms perform on a particular dataset. I have calculated different errors to compare the model performance.

In [8]:
```
test_mse=mean((actual-predicted)^2)
test_rmse=sqrt(test_mse)
test_mae=mean(abs(actual-predicted))
paste("Mean Absolute Error:",test_mae)
paste("Mean Squared Error:",test_mse)
paste("Root Mean Squared Error:",test_rmse)
```

'Mean Absolute Error: 3.96111558366877'
'Mean Squared Error: 19.1393359724046'
'Root Mean Squared Error: 4.37485268008016'

Conclusion

I have carried out prediction using supervised Machine Learning task and evaluated the model's performance. From my above analysis, I can say that if a student studied for 9.25 hours, he/she will secure 92.94 marks.

Thank You

In [ ]:
```
```