Simple Linear Regression

##Importing the dataset

library(readxl)  
data= read\_excel("C:/Users/user/Desktop/Book1.xlsx")  
head(data)

## # A tibble: 6 x 2  
## Hours Scores  
## <dbl> <dbl>  
## 1 2.5 21  
## 2 5.1 47  
## 3 3.2 27  
## 4 8.5 75  
## 5 3.5 30  
## 6 1.5 20

summary(data)

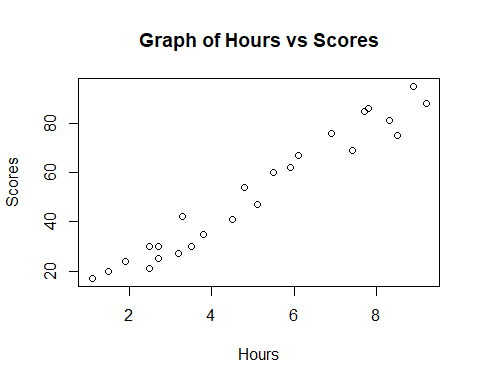
## Hours Scores   
## Min. :1.100 Min. :17.00   
## 1st Qu.:2.700 1st Qu.:30.00   
## Median :4.800 Median :47.00   
## Mean :5.012 Mean :51.48   
## 3rd Qu.:7.400 3rd Qu.:75.00   
## Max. :9.200 Max. :95.00

sum(is.na(data)) #for finding is there any null value or not

## [1] 0

##Plotting the data

plot(data$Hours,data$Scores,xlab="Hours",ylab="Scores",main="Graph of Hours vs Scores")

 ##Determining the correlation between dependent and inependent variable

cor.test(data$Hours,data$Scores)

##   
## Pearson's product-moment correlation  
##   
## data: data$Hours and data$Scores  
## t = 21.583, df = 23, p-value < 2.2e-16  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.9459248 0.9896072  
## sample estimates:  
## cor   
## 0.9761907

#Since the p-value<0.05,the null hypothesis is rejected, which means a significant correlation exists between Sores and Hours. And also the correlation coefficient is 0.97, so the assumption of linearity is fullfilled.

## Splitting the data into training and test dataset

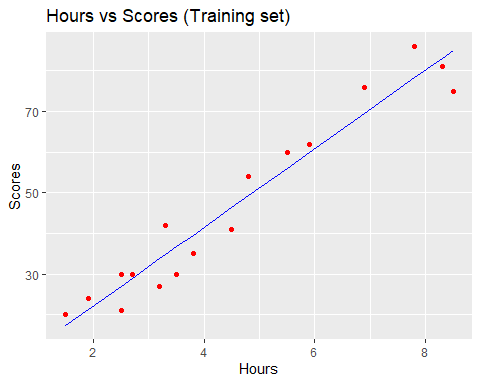
set.seed(555)  
ind=sample(1:2,nrow(data),replace=T,prob=c(0.7,0.3))  
training=data[ind==1,]  
test=data[ind==2,]

# Fitting Simple Linear Regression to the Training set  
regressor = lm(formula = Scores ~ Hours,data = training)  
# Predicting the Test set results  
y\_pred = predict(regressor,test)  
data.frame(test$Scores,y\_pred) #comparing Actual vs Predicted scores

## test.Scores y\_pred  
## 1 47 52.16771  
## 2 88 91.82284  
## 3 25 28.95495  
## 4 85 77.31486  
## 5 17 13.47978  
## 6 95 88.92124  
## 7 67 61.83969  
## 8 69 74.41327

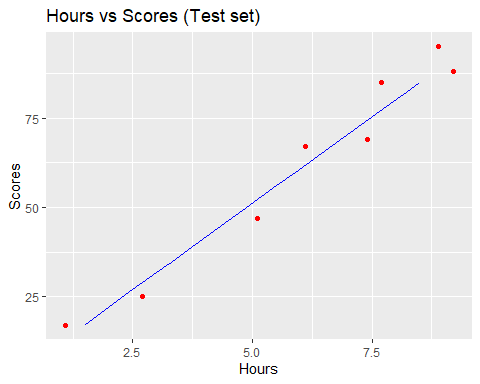
# Visualising the Training set results

library(ggplot2)  
ggplot() +  
geom\_point(aes(x = training$Hours ,  
 y = training$Scores),  
 colour = 'red') +  
geom\_line(aes(x = training$Hours,  
 y = predict(regressor, newdata = training)),  
 colour = 'blue') +  
ggtitle('Hours vs Scores (Training set)') +  
xlab('Hours') +  
ylab('Scores')



## Visualising the Test set results

library(ggplot2)  
ggplot() +  
geom\_point(aes(x = test$Hours, y = test$Scores),  
colour = 'red') +  
geom\_line(aes(x = training$Hours,  
 y = predict(regressor, newdata = training)),  
colour = 'blue') +  
ggtitle('Hours vs Scores (Test set)') +  
xlab('Hours') +  
ylab('Scores')



## Predicting the results

library(margins)

## Warning: package 'margins' was built under R version 4.1.3

newhour=data.frame(Hours=9.25)  
newscore=predict(regressor,newhour)  
newscore

## 1   
## 92.30644