StudentScore\_LinearRegression\_TSF\_Task1.R

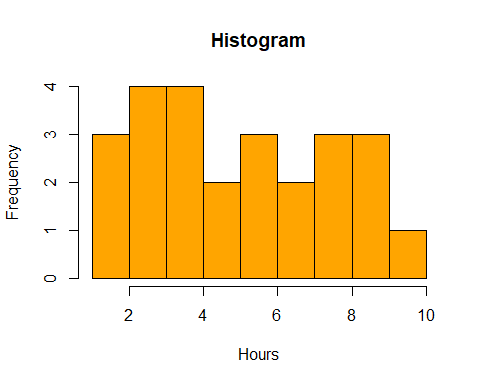
#Task 1 - Problem Statement -Predict the percentage of a student based on the no. of study hours  
  
library(gridExtra)   
  
# Load the .csv file  
stud\_score<-read.csv("https://raw.githubusercontent.com/AdiPersonalWorks/Random/master/student\_scores%20-%20student\_scores.csv")  
View(stud\_score)  
  
#Summary of the data set  
summary(stud\_score)

## 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

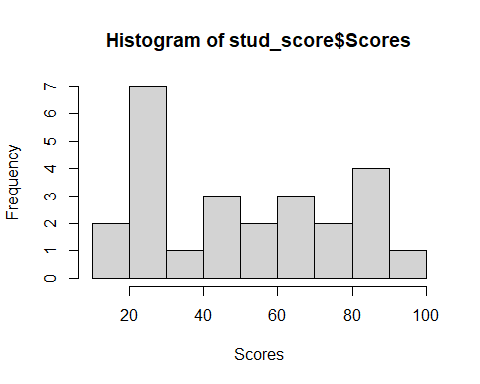
#Structure of the dataset  
str(stud\_score)

## 'data.frame': 25 obs. of 2 variables:  
## $ Hours : num 2.5 5.1 3.2 8.5 3.5 1.5 9.2 5.5 8.3 2.7 ...  
## $ Scores: int 21 47 27 75 30 20 88 60 81 25 ...

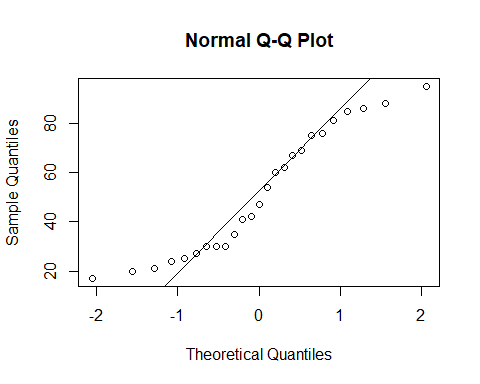
#One variable of type numeric and another of type int  
  
#Histogram to see the trend between the variables  
hist(stud\_score$Hours,xlab="Hours",ylab = "Frequency",col = "orange",main = "Histogram")



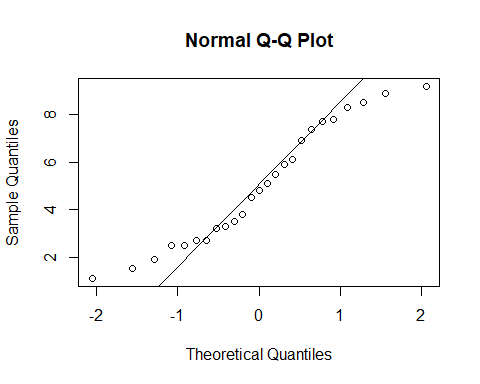
hist(stud\_score$Scores,xlab="Scores")



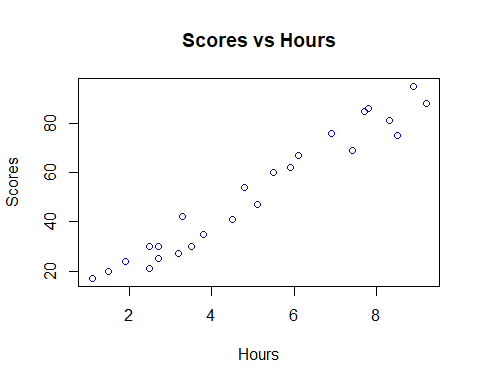
#To see the normality of the data points  
qqnorm(stud\_score$Scores)  
qqline(stud\_score$Scores) #Almost all the points fall along line, so we can assume normality



qqnorm(stud\_score$Hours)  
qqline(stud\_score$Hours) #Almost all the points fall along line, so we can assume normality



#To see the trend   
plot(stud\_score,main='Scores vs Hours',col='blue') # shows a linear relation



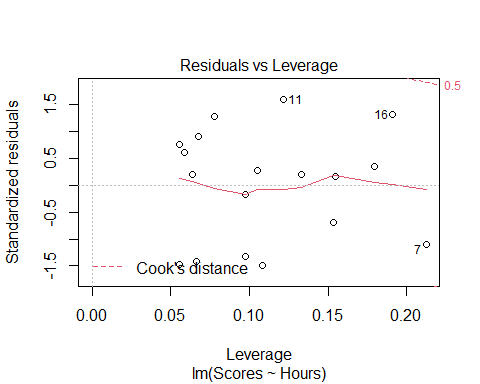
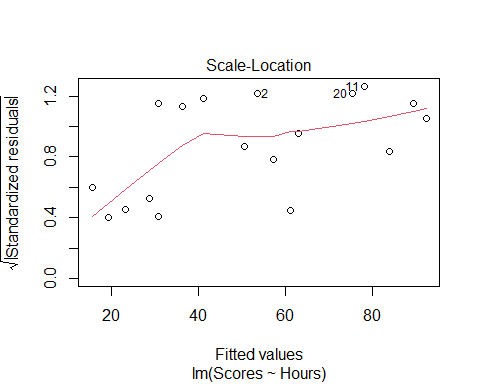
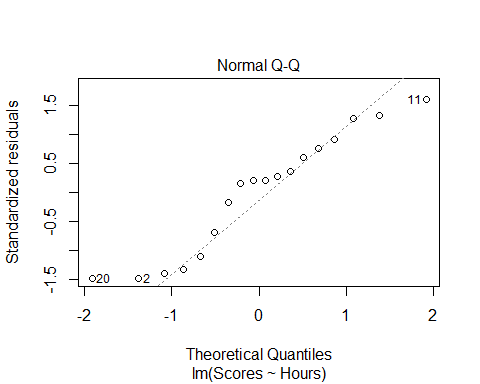
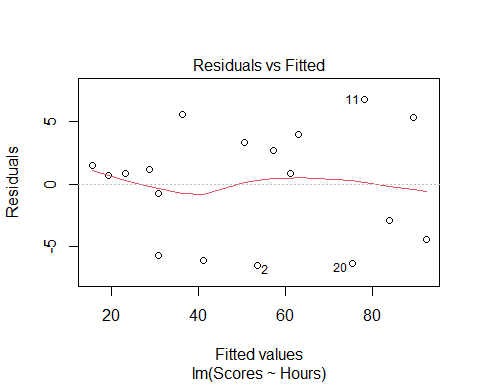
#Correlation between 2 variables  
cor(stud\_score) # shows extreme positive correlation

## Hours Scores  
## Hours 1.0000000 0.9761907  
## Scores 0.9761907 1.0000000

#Perform Linear Regression model  
#Score being the dependent or target variable and Hours is the predictor or independent variable.  
  
  
set.seed(10)  
train\_index<-sample(1:nrow(stud\_score), 0.75\*nrow(stud\_score))  
test\_index<-setdiff(1:nrow(stud\_score),train\_index)  
train\_data<-stud\_score[train\_index,]  
test\_data<-stud\_score[test\_index,]  
model<-lm(Scores~Hours,data=train\_data)  
summary(model)

##   
## Call:  
## lm(formula = Scores ~ Hours, data = train\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -6.5161 -4.0680 0.8805 3.1710 6.7910   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 5.0802 2.3170 2.193 0.0435 \*   
## Hours 9.4972 0.4186 22.685 1.36e-13 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 4.532 on 16 degrees of freedom  
## Multiple R-squared: 0.9698, Adjusted R-squared: 0.968   
## F-statistic: 514.6 on 1 and 16 DF, p-value: 1.36e-13

plot(model)



#From the results , its evident that the intercept is 5.0802 and the P value of hours is less than 0.05 at 95 % confidence interval.  
# Adjusted R Square value is 0.968 with 16 degrees of freedom. Since the Adjusted square value is close to 1 it shows better goodness of fit.  
# Linear Regression can be expressed as y=ax+b  
# Scores = 5.0802 + 9.4972 \* Hours  
#If Hours = 9.25 hrs/day , then Scores = 5.0802 + 9.4972 \* 9.25  
  
Scores1<-5.0802 + 9.4972 \* 9.25  
Scores1

## [1] 92.9293

#Scores = 92.92