Supervised learning : - (Linear Regression) –

Task 2 of GRIP

Pallabi Mandal

10/08/2020

Table of Contents : -

1. Import Data
2. EDA (Univariate and Bivariate analysis)
3. Correlation check
4. Data splicing
5. Build Linear regression model
6. Check Prediction

```
library(ggplot2)
library(coefplot)
library(corrplot)
library(psych)
library(tidyverse)
library(MASS)
library(lattice)
library(DataExplorer)
library(ModelMetrics)
library(lmtest)
library(caret)
```

```r
library(recipes)
library(caTools)
library(Hmisc)
library(GGally)

#Import data

setwd("P:/R file")
getwd()
```

```
## [1] "P:/R file"
```

```r
library(readr)
Score_data <- read.csv("Student_score.csv")
view(Score_data)
str(Score_data)
```

```
## '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 ...
```

```r
summary(Score_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
```

```r
dim(Score_data)
```

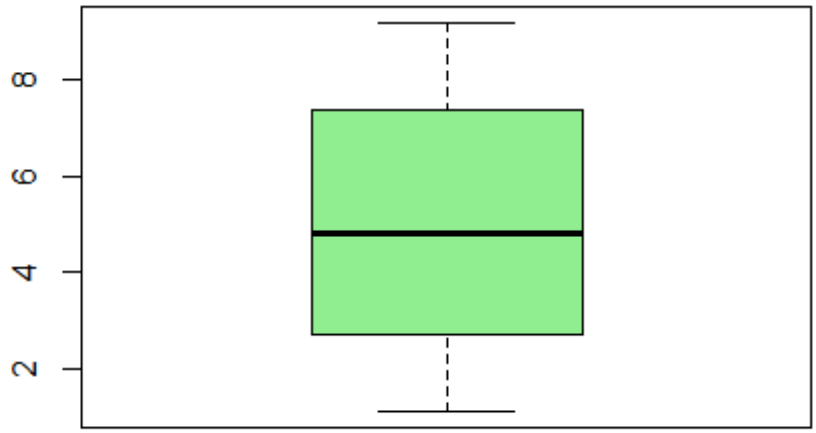
```
## [1] 25  2
```

```r
names(Score_data)
```

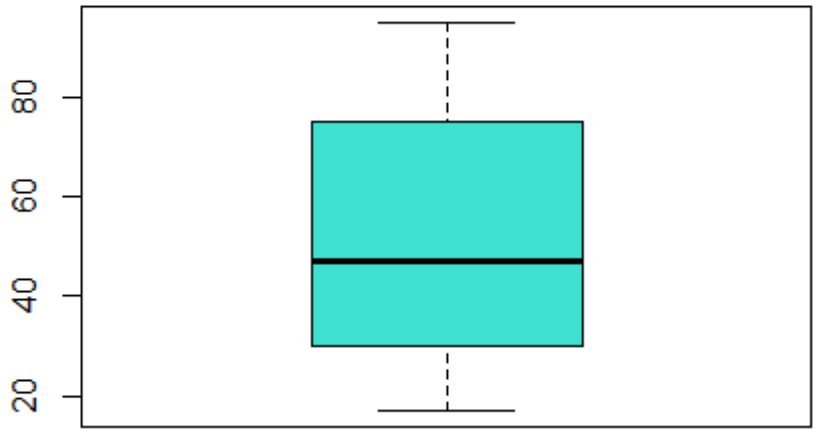
```
## [1] "Hours"  "Scores"
```

```r
attach(Score_data)
```

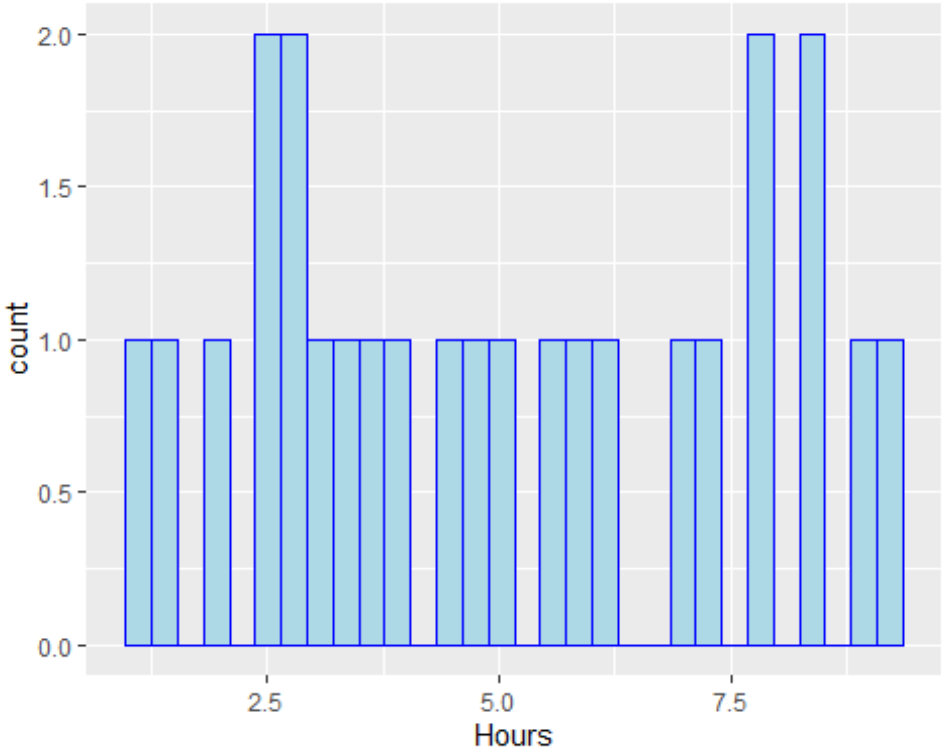
```r
#EDA
#Univariate and Bivariate analysis
boxplot(Hours, col = "lightgreen")
```



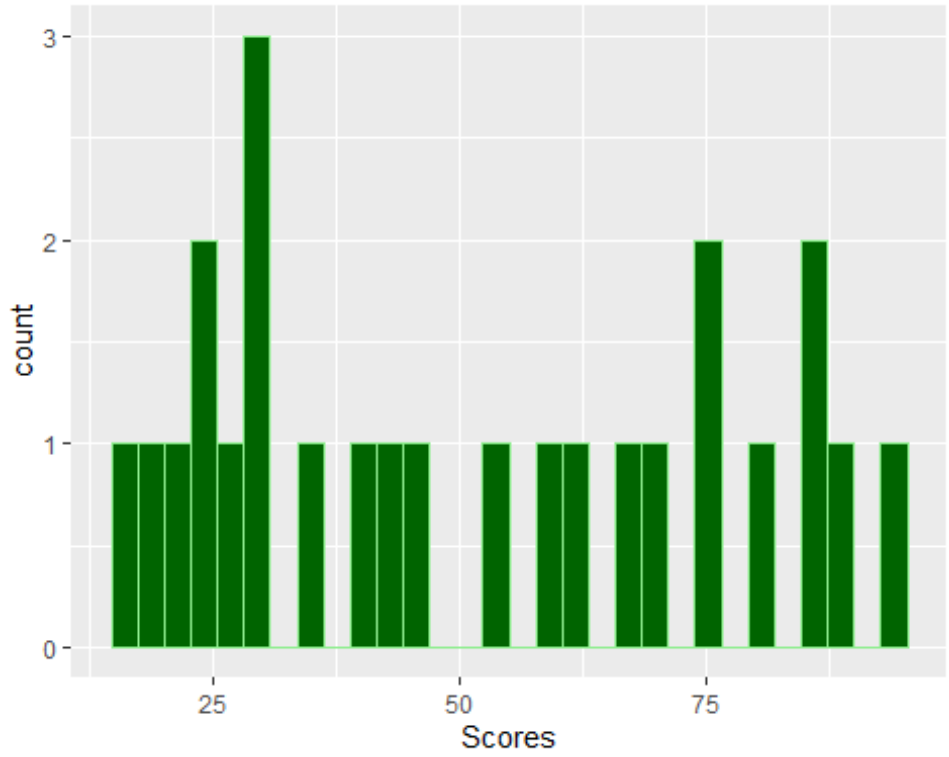

```r
boxplot(Scores, col = "turquoise")
```



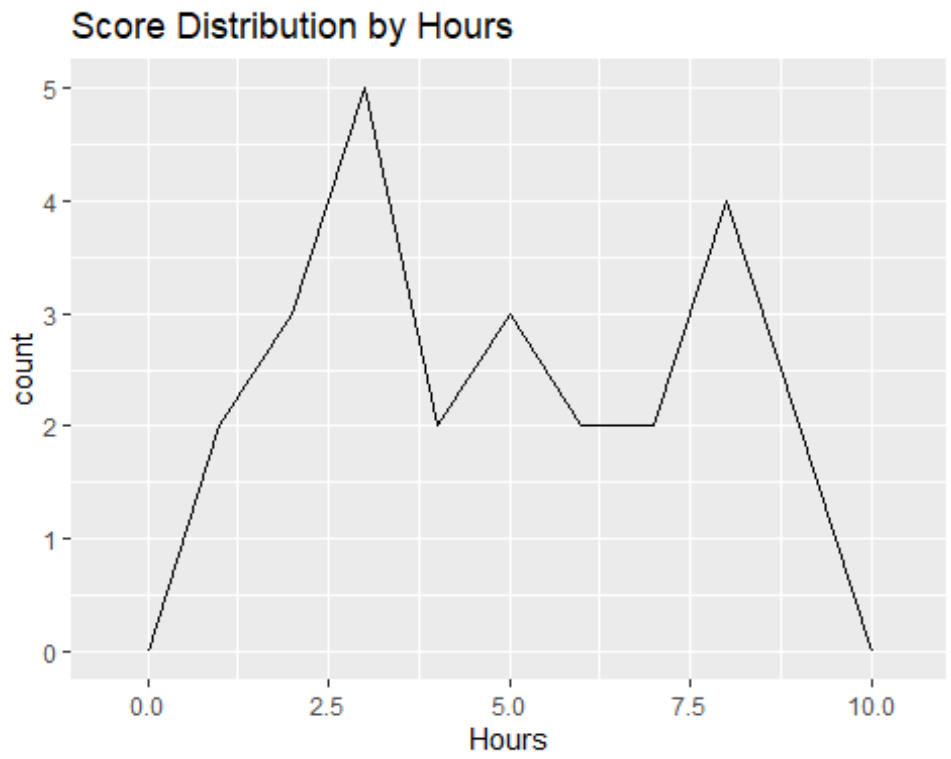

```
ggplot(Score_data, aes(x= Hours)) + geom_histogram(bins = 30, fill = "lightbl
ue",col = "blue")
```



```
ggplot(Score_data, aes(x = Scores)) + geom_histogram(bins = 30, fill =
"darkgreen", col = "lightgreen")
```




```
ggplot(Score_data, aes(Hours, colour = Scores)) +
  geom_freqpoly(binwidth = 1) + labs(title="Score Distribution by Hours")
```

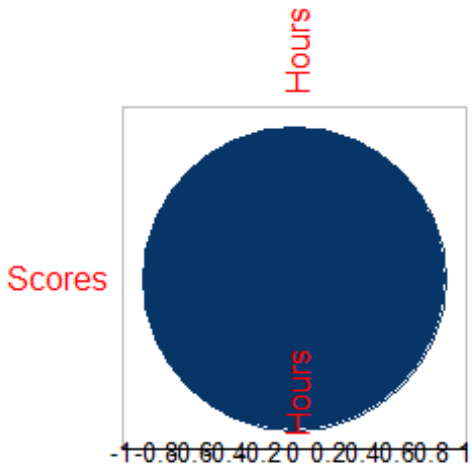



```
scatter.smooth(x=Score_data$Hours, y=Score_data$Scores, main="Scores ~ Hours"
)
```

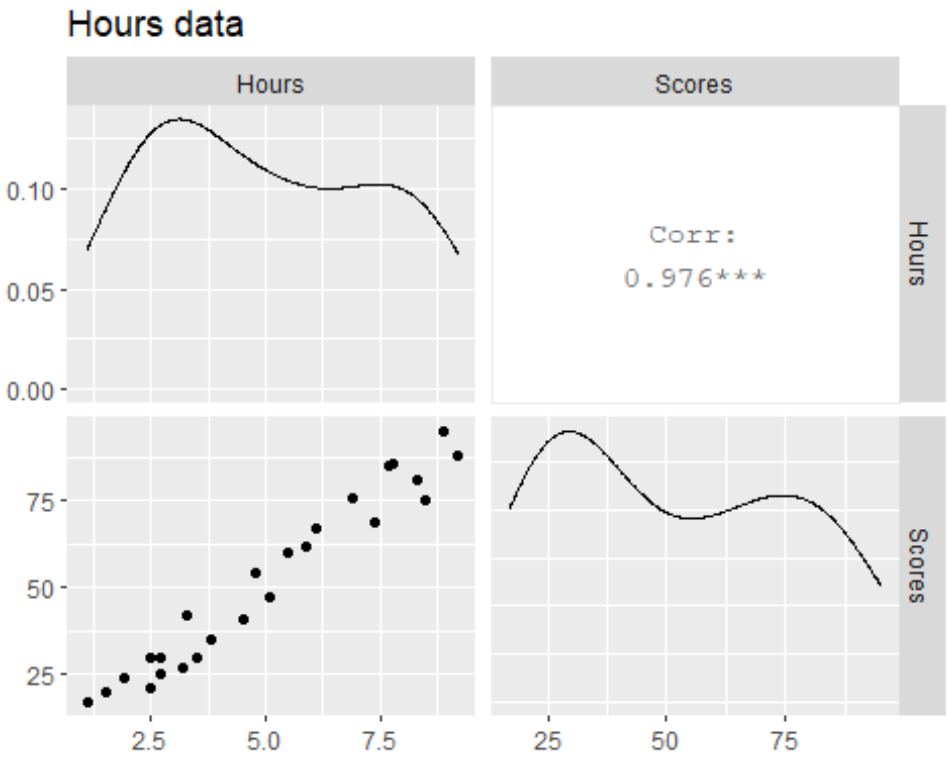
```r
#Correlation
Score.cor = cor(Score_data)
Score.cor = cor(Score_data, method = c("pearson"))
library(Hmisc)
Score.rcorr = rcorr(as.matrix(Score_data))
Score.rcorr
```

```
##        Hours Scores
## Hours   1.00   0.98
## Scores  0.98   1.00
##
## n= 25
##
##
## P
##        Hours Scores
## Hours         0
## Scores  0
```

```r
corrplot(Score.cor, type="lower", diag = FALSE)
```




```r
ggpairs(data=Score_data, columns=1:2, title="Hours data")
```




```r
cor(Score_data$Hours, Score_data$Scores)
```

```
## [1] 0.9761907
```

```r
#Data splicing
split <- sample.split(Score_data$Scores, SplitRatio = 0.70)
length(split)
```

```
## [1] 25
```

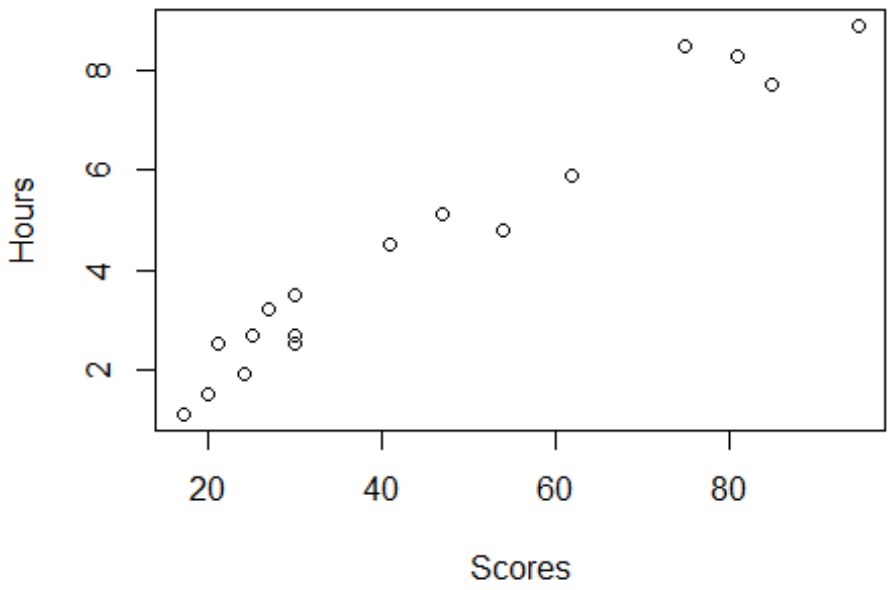
```r
train<-subset(Score_data, split == TRUE)
test<-subset(Score_data, split == FALSE)

#Model building
set.seed(123)
Mod1 <- lm(Scores ~ ., data = train)
summary(Mod1)
```

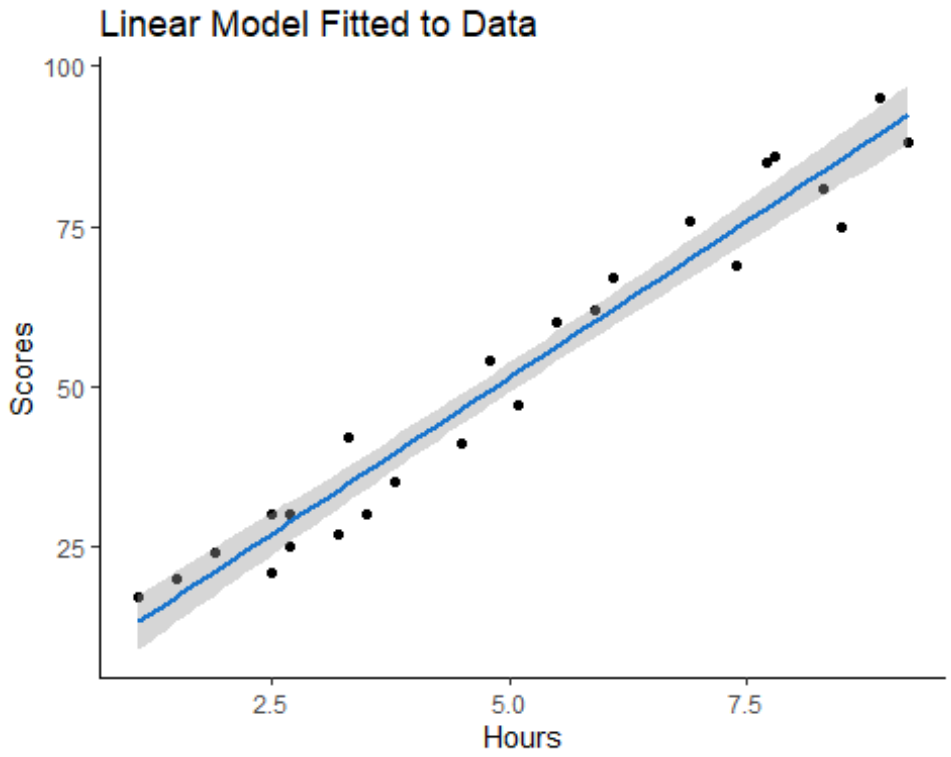
```
##
## Call:
## lm(formula = Scores ~ ., data = train)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.431 -4.626  1.836  3.776  8.330
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)   1.9704     2.7098   0.727    0.478
## Hours         9.7012     0.5324  18.221 1.21e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.504 on 15 degrees of freedom
## Multiple R-squared:  0.9568, Adjusted R-squared:  0.9539
## F-statistic:   332 on 1 and 15 DF,  p-value: 1.211e-11
```

```
#Model plots
with(train,plot(Scores, Hours))
    abline(0, 1)
```




```
#Fitted line
ggplot(data = Score_data, aes(x = Hours, y = Scores)) +
geom_point() +
stat_smooth(method = "lm", col = "dodgerblue3") +
theme(panel.background = element_rect(fill = "white"),
axis.line.x=element_line(),
axis.line.y=element_line()) +
ggtitle("Linear Model Fitted to Data")
```



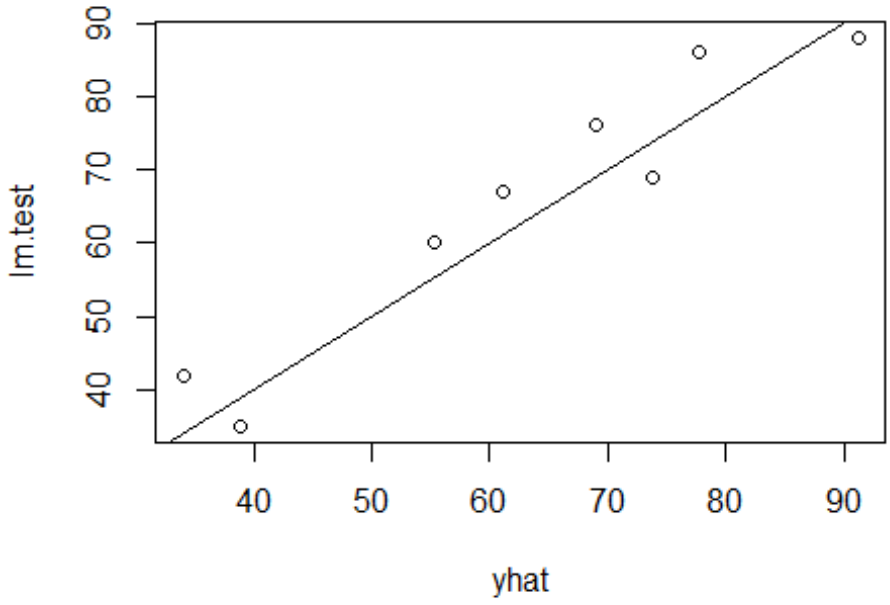

```
#K fold validation
set.seed(123)
train.control <- trainControl(method = "cv", number = 10)
#Train the model
model <- train(Scores~., data = train, method = "lm", trControl = train.contr
ol)
print(model)
```

```
## Linear Regression
##
## 17 samples
##  1 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 15, 15, 15, 16, 15, 16, ...
## Resampling results:
##
##   RMSE      Rsquared  MAE
##   5.973125  1         5.712376
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```
#Predict MSE
yhat = predict(Mod1,newdata= test)
lm.test=test$Scores # These are the actual values
lm.test # Take a quick look
```

```
## [1] 88 60 42 67 69 35 76 86
```

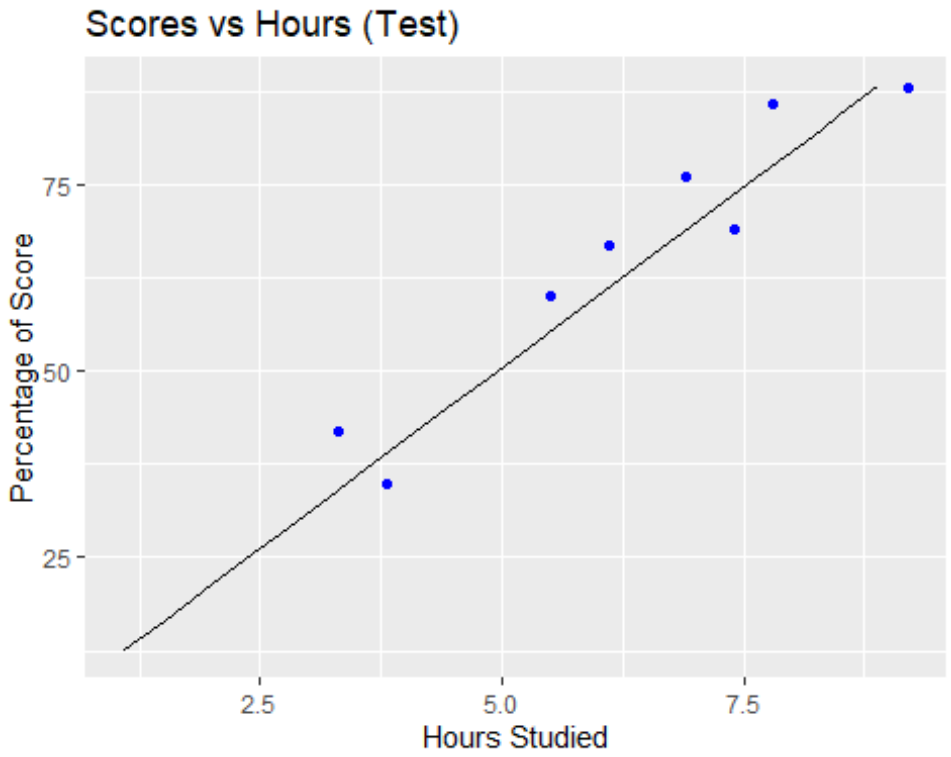
```
plot(yhat,lm.test) # Let's plot predicted vs. actual
abline(0,1) # And draw a line
```




```
mean((yhat-lm.test)^2) # And calculate the MSE
```

```
## [1] 36.03034
```

```
#plot visualization for Test data predictions of Hours with respect to percen
tage of score
ggplot() +
  geom_point(aes(x = test$Hours, y = test$Scores),
             colour = 'blue') +
  geom_line(aes(x = train$Hours, y = predict(Mod1, newdata = train)),
            colour = 'black') +
  ggtitle('Scores vs Hours (Test)') +
  xlab('Hours Studied') +
  ylab('Percentage of Score')
```




```r
#Make prediction for 9.25 hours
new_data <- data.frame("Hours"=9.25,"Scores"=0)
score_predict <- predict(Mod1,newdata = new_data)
new_data <- data.frame("Hours"=9.25,"Scores"= score_predict)
print(new_data)
```

```
##   Hours   Scores
## 1  9.25 92.76682
```

```r
#Calculate RMSE, MAE score on Test data
predictions <- predict(Mod1, test)
postResample(test$Scores, predictions)
```

```
##      RMSE  Rsquared       MAE
## 5.537494 0.953319 5.297265
```