```
# Linear regression
library(MASS)
library(tidyverse)
library(caTools)
library(lattice)
data("Boston")
view(Boston)
?Boston
#correlation line using corrplot to visualize
cr<-cor(Boston)
cr
library(corrplot)
corrplot(cr,type = "lower")
corrplot(cr,method = "number")
#creating scatter plotmatrix
attach(Boston)
splom(~Boston[c(1:6,14)],groups=NULL,data=Boston,axis.line.tck=0,axis.text.alpha=0)
splom(~Boston[c(7:14)],groups = NULL,data = Boston,axis.line.tck=0,axis.text.alpha=0)
#to view the correlation of variables
plot(Boston$lstat,Boston$medv,cex=0.5,xlab = "crime rate", ylab = "Price")
#studying lstat and medv
plot(lstat,medv)
abline(lm(medv~lstat),col=("red"))
#simple regression model
lrmodel<-lm(medv~lstat,data = Boston)
summary(lrmodel)
#prediction
result_reg<-predict(lrmodel,Boston)
final_data<-cbind(actual=Boston$medv,predicted=result_reg)
final_data<-as.data.frame(final_data)
```

```
view(final_data)

str(final_data)

#to compare prediction values and actual values, we use plots

plot(final_data$actual,type = "l",lty=1.8,col="green")

lines(final_data$predict,type="l",col="blue")

#finding errors

errors<-(final_data$actual - final_data$predicted)

final_data<-cbind(final_data,errors)

view(final_data)

plot(final_data$errors)

hist(final_data$errors)

# test of normality

#jarque.bera.test

library(tseries)

jarque.bera.test(final_data$errors)

qqnorm(final_data$errors)

qqline(final_data$errors)

shapiro.test(final_data$errors)

ks.test(final_data$errors,"pnorm", mean = mean(final_data$errors), sd =
sd(final_data$errors))

#Test of autocorrelation

#ACF (Autocorrelation Function):

acf(final_data$errors)

abline(h = 0, col = "red")  # Add a red line at y=0 for reference

#PACF (Partial Autocorrelation Function):

pacf(final_data$errors)

abline(h = 0, col = "red")  # Add a red line at y=0 for reference

library(urca)

# Replace "your_data"

Box.test(acf(final_data$errors)$acf)  # Test for significance of ACF values

Box.test(pacf(final_data$errors)$pacf)  # Test for significance of PACF values
```

```r
library(lmtest)
bg_result <- bgtest(lrmodel, order = 4, type = "Chisq")
# View the test results
print(bg_result)
#Test for homoscedasticity
plot(lrmodel$residuals ~ fitted(lrmodel))
bptest(lrmodel)
#root mean square error
rmse<-sqrt(mean(final_data$errors^2))
rmse
# Linear Regression in Machine Learning
#We will split the data into training and testing sets
set.seed(123)
split<-sample.split(Boston$medv,SplitRatio = 0.7)
split
#to divide the data with the ratio 0,7
training_data<-subset(Boston,split=="TRUE")
testing_data<-subset(Boston,split=="FALSE")
#finding multicollinarity
library(caret)
Boston_a=subset(Boston,select = -c(medv))
numericdata<-Boston_a[sapply(Boston_a,is.numeric)]
descrcor<-cor(numericdata)
descrcor
#VIF variance inflation factors =1 no correlation among variables
library('car')
model<-lm(medv~.,data = training_data)
vif(model)
summary(model)
predict1<-predict(model,testing_data)
```

```
predict1
#to compare prediction values and actual values, we use plots
plot(testing_data$medv,type = "l",lty=1.8,col="green")
lines(predict1,type="l",col="blue")
#model2 after removing unsignificant variables( age, indus)
model2<-lm(medv~crim+zn+rm+dis+ptratio+black+lstat,data = training_data)
summary(model2)
predict2<-predict(model,testing_data)
predict2
#to compare prediction values and actual values, we use plots
plot(testing_data$medv,type = "l",lty=1.8,col="green")
lines(predict2,type="l",col="blue")
# plotting results
plot(model2)
par(mfrow=c(2,2))
plot(model2)
par(mfrow=c(1,1))
bg_result <- bgtest(model2, order = 2, type = "Chisq")
# View the test results
print(bg_result)
#Test for homoscedasticity
plot(model2$residuals ~ fitted(model2))
bptest(model2)
#root mean squre error
rmse<-sqrt(mean(final_data$errors^2))
rmse
```