Regression Model Assignment 2

Anthony Ndegwa

2024-11-15

library(forecast)

## Registered S3 method overwritten by 'quantmod':  
## method from  
## as.zoo.data.frame zoo

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(ggplot2)  
library(tidyr)  
library(lmtest)

## Loading required package: zoo

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

# Reading the dataset  
df <- read.csv("/home/anthony/data\_science/Data\_Science\_Projects/0x01-r\_programming\_data\_science\_projects/Regression/BMI-Data.csv")  
head(df, 5)

## Age Height Weight Bmi  
## 1 61 1.85 109.3 <NA>  
## 2 60 1.71 79.02 27.0236996  
## 3 60 1.55 74.7 31.09261186  
## 4 60 1.46 35.9 16.84180897  
## 5 60 1.58 97.1 38.89601025

# increasing the view of the plot in an interactive way  
X11(width = 8, height = 10)  
  
# layout of the plots  
par(mfrow= c(2,2))

# sum of all the na in the dataframe  
sum(is.na(df))

## [1] 44

# Checking the data type og weight and bmi  
typeof(df$Weight)

## [1] "character"

typeof(df$Bmi)

## [1] "character"

# Converting the character or string data type to numetic  
df$Weight <- as.numeric(df$Weight)

## Warning: NAs introduced by coercion

df$Bmi <- as.numeric(df$Bmi)

## Warning: NAs introduced by coercion

# Filing the missing value with the mean of the column  
df <- df %>% replace\_na(list(Weight = mean(df$Weight, na.rm=TRUE)))  
df <- df %>% replace\_na(list(Bmi = mean(df$Bmi, na.rm=TRUE)))

# removing the not a number na or nan  
df <- na.omit(df)

sum(is.na(df))

## [1] 0

y = df$Bmi  
  
# model <- lm(y ~ Age + Weight + Height, data = df)  
model <- lm(y ~ Weight, data = df)  
  
# getting the statistics of the model  
print(summary(model))

##   
## Call:  
## lm(formula = y ~ Weight, data = df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -17.986 -1.230 -0.386 0.559 33.652   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 5.312682 0.348051 15.26 <2e-16 \*\*\*  
## Weight 0.269109 0.004137 65.04 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3.502 on 739 degrees of freedom  
## Multiple R-squared: 0.8513, Adjusted R-squared: 0.8511   
## F-statistic: 4231 on 1 and 739 DF, p-value: < 2.2e-16

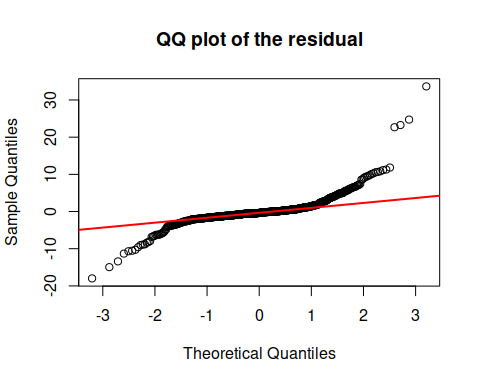
# converting the weight data into a data frame in order to predict the future  
predict\_value <- data.frame(Weight=200)  
  
# Making the prediction of weight = 200  
prediction <- predict(model, newdata = predict\_value)  
  
# display the prediction of bmi when weight == 200  
print(prediction)

## 1   
## 59.13452

#Residual are the difference btw the predicted value with the actual value  
# getting the residual from the model  
model\_residual <- residuals(model)  
  
model\_residual\_frame <- data.frame(model\_residual = model\_residual)  
# Display the residual of the model  
print(head(model\_residual\_frame, 5))

## model\_residual  
## 1 -8.3783845  
## 2 0.4460096  
## 3 5.6774736  
## 4 1.8681073  
## 5 7.4528261

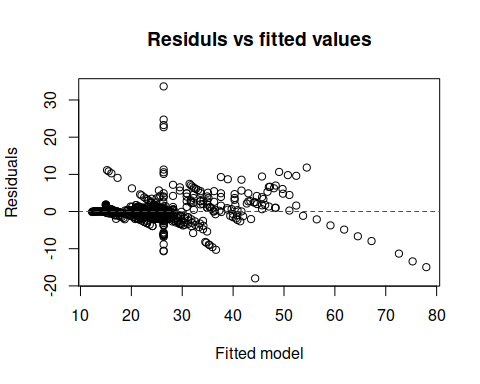
#QQ-plot helps to decipher whether the residual are normaly distributed  
#Ploting a QQ-plot  
qqnorm(model\_residual, main='QQ plot of the residual')  
  
# adding an abline to the plot to act as a reference line  
qqline(model\_residual, col='red', lwd=2)



# checking for normality an going to use shaporo test  
shapiro.test(model\_residual)

##   
## Shapiro-Wilk normality test  
##   
## data: model\_residual  
## W = 0.73419, p-value < 2.2e-16

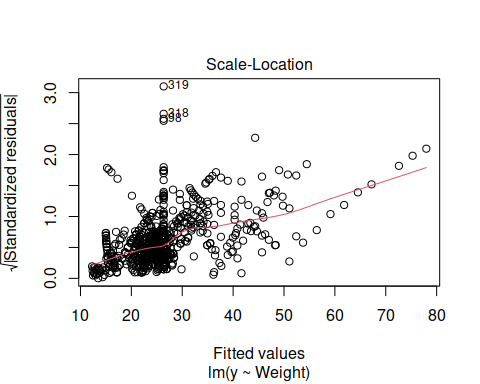
# model assumptions  
# a) Linearity  
plot(fitted(model), model\_residual, xlab = "Fitted model", ylab = "Residuals",  
 main="Residuls vs fitted values")  
  
abline(h=0, col='red', lty=2)



# b) Independence by using durbin watson  
  
dwtest(model)

##   
## Durbin-Watson test  
##   
## data: model  
## DW = 1.507, p-value = 8.076e-12  
## alternative hypothesis: true autocorrelation is greater than 0

# c) Homoscedasticity by using breusch pagan test  
  
plot(model, which = 3)



bptest(model)

##   
## studentized Breusch-Pagan test  
##   
## data: model  
## BP = 25.182, df = 1, p-value = 5.216e-07

# Quiz9  
# Describe how R can be used to optimize parameter estimation using maximum likelihood estimation method   
# (Refer to maximum likelihood estimation notes upload in lecture 2)

# Step 1  
# Simulating some data from a normal distribution  
set.seed(42)  
x <- rnorm(100, mean = 5, sd = 2) # Mean = 5, SD = 2

# Step 2  
# Defining the negative likelihood function  
negative\_log\_likelihood <- function(params) {  
 # mean  
 mu <- params[1]  
 # Standard deviation  
 sigma <- params[2]  
   
 # Making sure that the sigma is positive  
 if (sigma <= 0){  
 return(Inf)  
 }  
   
 # computation of the negative log likelihood  
 nil <- sum(dnorm(x, mean = mu, sd = sigma, log = TRUE))  
   
 return(nil)  
}

# Step 3  
# Optimize the parameter using Optim()  
  
initial\_params <- c(mu = 0, sigma = 1)  
result <- optim(par = initial\_params, fn = negative\_log\_likelihood,  
 method = "L-BFGS-B",  
 lower = c(-Inf, 1e-6))  
  
#Extracting the estimated parameter  
estimated\_mu <- result$par[1]  
estimated\_sigma <- result$par[2]  
  
  
# Display of the the esimated mean  
cat("The estimated mean (mu) is: ", estimated\_mu, "\n")

## The estimated mean (mu) is: -5.11568e+16

cat("The estimated standard deviation(sigma) is: ", estimated\_sigma)

## The estimated standard deviation(sigma) is: 1e-06