Laboratorio 2

Ricardo Kaleb Flores Alfonso

2024-10-14

```
library("moments")
library("nortest")
library("tidyr")
library("moments")
library("nortest")
library("MASS")
library("mlbench")
library("VGAM")
```

```
## Cargando paquete requerido: stats4
```

```
## Cargando paquete requerido: splines
```

```
library("lmtest")
```

```
## Cargando paquete requerido: zoo
```

```
##
## Adjuntando el paquete: 'zoo'
```

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

```
##
## Adjuntando el paquete: 'lmtest'
```

```
## The following object is masked from 'package:VGAM':
##
##     lrtest
```

```
library(e1071)
```

```
##
## Adjuntando el paquete: 'e1071'
```

```
## The following objects are masked from 'package:moments':
##
##     kurtosis, moment, skewness
```

```
library(mnormt)
library(MVN)
library(GPArotation)
library(performance)
library(polycor)
library(ggcorrplot)
```

```
## Cargando paquete requerido: ggplot2
```

```r
library(psych)
```

```
##
## Adjuntando el paquete: 'psych'
```

```
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
## The following object is masked from 'package:polycor':
##
##     polyserial
```

```
## The following objects are masked from 'package:GPArotation':
##
##     equamax, varimin
```

```
## The following objects are masked from 'package:VGAM':
##
##     fisherz, logistic, logit
```

```r
library(ggplot2)
library(polycor)
library(ggcorrplot)
library(MVN)
library(ggplot2)
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```r
library(rattle)
```

```
## Cargando paquete requerido: tibble
```

```
## Cargando paquete requerido: bitops
```

```
## Rattle: A free graphical interface for data science with R.
## Versión 5.5.1 Copyright (c) 2006-2021 Togaware Pty Ltd.
## Escriba 'rattle()' para agitar, sacudir y  rotar sus datos.
```

```
##
## Adjuntando el paquete: 'rattle'
```

```
## The following object is masked from 'package:VGAM':
##
##     wine
```

```r
library(psych)
library(MVN)
library(stats)
library(FactoMineR)
library(ggplot2)
library(factoextra)
```

Problema 1

A) Analice la correlación entre peso y potencia

```
library(MASS)
df <- Cars93

weight <- df$Weight
hp <- df$Horsepower
x <- df[c("Weight","Horsepower")]
print("La correlación entre peso y caballos de fuerza")
```

```
## [1] "La correlación entre peso y caballos de fuerza"
```

```
cor(weight,hp)
```

```
## [1] 0.7387975
```

B) Proponga un modelo de regresión simple

```
library(lmtest)

modelo1 <- lm(hp ~ weight)
summary(modelo1)
```

```
##
## Call:
## lm(formula = hp ~ weight)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -93.017 -20.921  -1.515   8.356 136.028
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -57.738203  19.622752  -2.942  0.00413 **
## weight        0.065595   0.006272  10.458  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 35.49 on 91 degrees of freedom
## Multiple R-squared:  0.5458, Adjusted R-squared:  0.5408
## F-statistic: 109.4 on 1 and 91 DF,  p-value: < 2.2e-16
```

C) Realice la validación de los supuestos del modelo

Individual, conjunta y correlación

```
summary(modelo1)
```

```
##
## Call:
## lm(formula = hp ~ weight)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -93.017 -20.921  -1.515   8.356 136.028
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -57.738203  19.622752  -2.942  0.00413 **
## weight        0.065595   0.006272  10.458  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 35.49 on 91 degrees of freedom
## Multiple R-squared:  0.5458, Adjusted R-squared:  0.5408
## F-statistic: 109.4 on 1 and 91 DF,  p-value: < 2.2e-16
```

T Test

- H_0 := El coeficiente es igual a 0.
- H_A := El coeficiente no es igual a 0.

Linealidad

- H_0 := La relación entre la variable independiente y dependiente es lineal
- H_A := La relación es no lineal

```
resettest(modelo1)
```

```
##
##  RESET test
##
## data:  modelo1
## RESET = 1.5744, df1 = 2, df2 = 89, p-value = 0.2128
```

Media de cero de los residuos

T Test

- H_0 := La media de los errores es igual a 0.
- H_A := La media de los errores no es igual a 0.

```
t.test(modelo1$residuals)
```

```
##
##  One Sample t-test
##
## data:  modelo1$residuals
## t = 9.5597e-17, df = 92, p-value = 1
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -7.269239  7.269239
## sample estimates:
##    mean of x
## 3.498919e-16
```

Normalidad de residuos

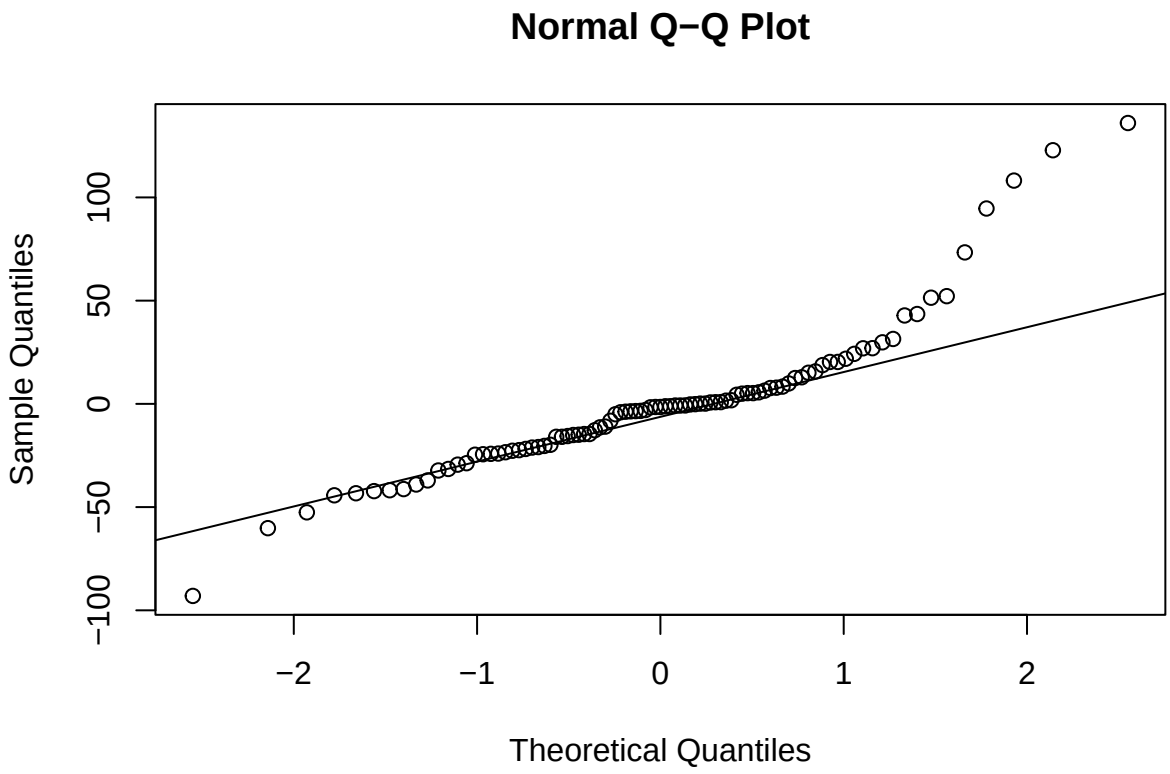
T Test

- H_0 := Los residuos tienen una distribución normal
- H_A := Los residuos no tienen una distribución normal

```r
shapiro.test(modelo1$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  modelo1$residuals
## W = 0.88138, p-value = 4.597e-07
```

```r
qqnorm(modelo1$residuals)
qqline(modelo1$residuals)
```




```r
hist(modelo1$residuals)
```

Breusch-Pagan

- H_0 := Los datos tienen homocedasticidad.
- H_A := Los datos no tienen homocedasticidad.

```
bptest(modelo1)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  modelo1
## BP = 7.7789, df = 1, p-value = 0.005286
```

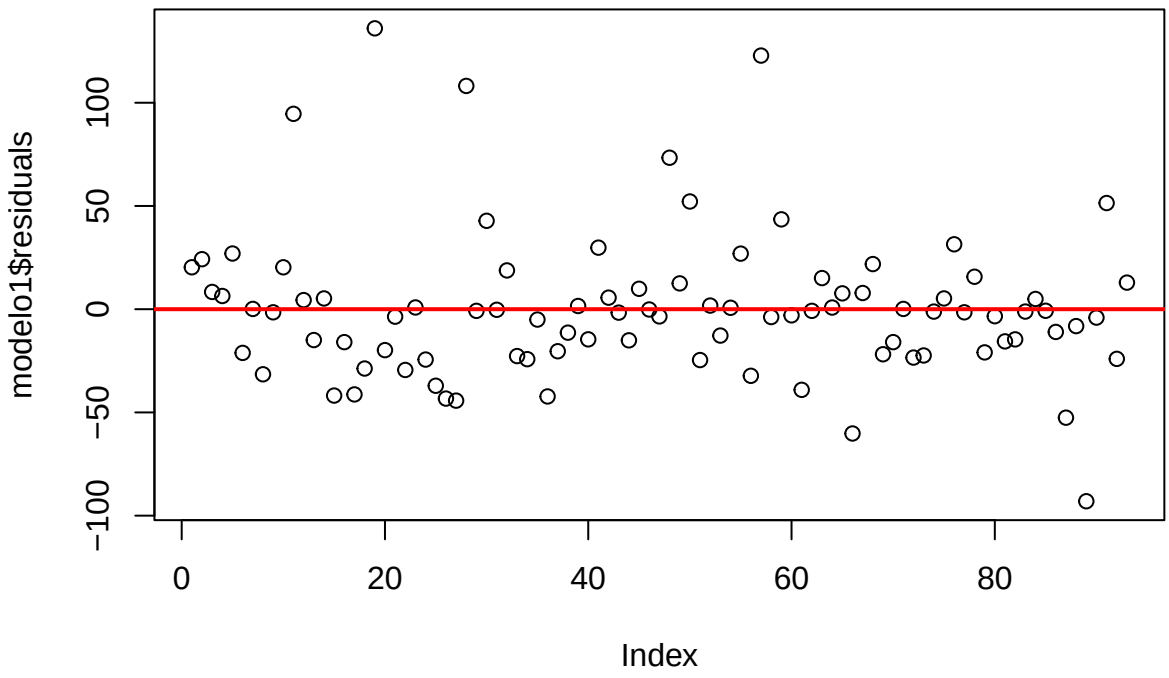
Independencia Test de Durbin Watson

- H_0 := No existe autocorrelación en los datos
- H_A := Existe autocorrelacion en los datos.

```
dwtest(modelo1)
```

```
##
##  Durbin-Watson test
##
## data:  modelo1
## DW = 2.1052, p-value = 0.6894
## alternative hypothesis: true autocorrelation is greater than 0
```

```
plot(modelo1$residuals)
abline(h=0, col = "red", lwd = 2)
```



#Prueba Breusch-Godfrey

```
bgtest(modelo1)
```

```
##
##  Breusch-Godfrey test for serial correlation of order up to 1
##
## data:  modelo1
## LM test = 0.28452, df = 1, p-value = 0.5938
```

D) Identifique los datos atípicos e influyentes

Datos atipicos

Metodo de desviación estandar

```
residuals_values<- rstandard(modelo1)
```

Metodo de estandarización

```
rstudents_values<-rstudent(modelo1)
```

Datos influyentes

Por grado de leverage

```
hat_values<-hatvalues(modelo1)
```

Por distance de Cook

```r
cooks_values<-cooks.distance(modelo1)
```

Resumen de resultados

```r
tabla= data.frame(residuals_values,rstudents_values, hat_values, cooks_values)
```

Variable dependiente contra las variables predictoras

```r
library(car)
```

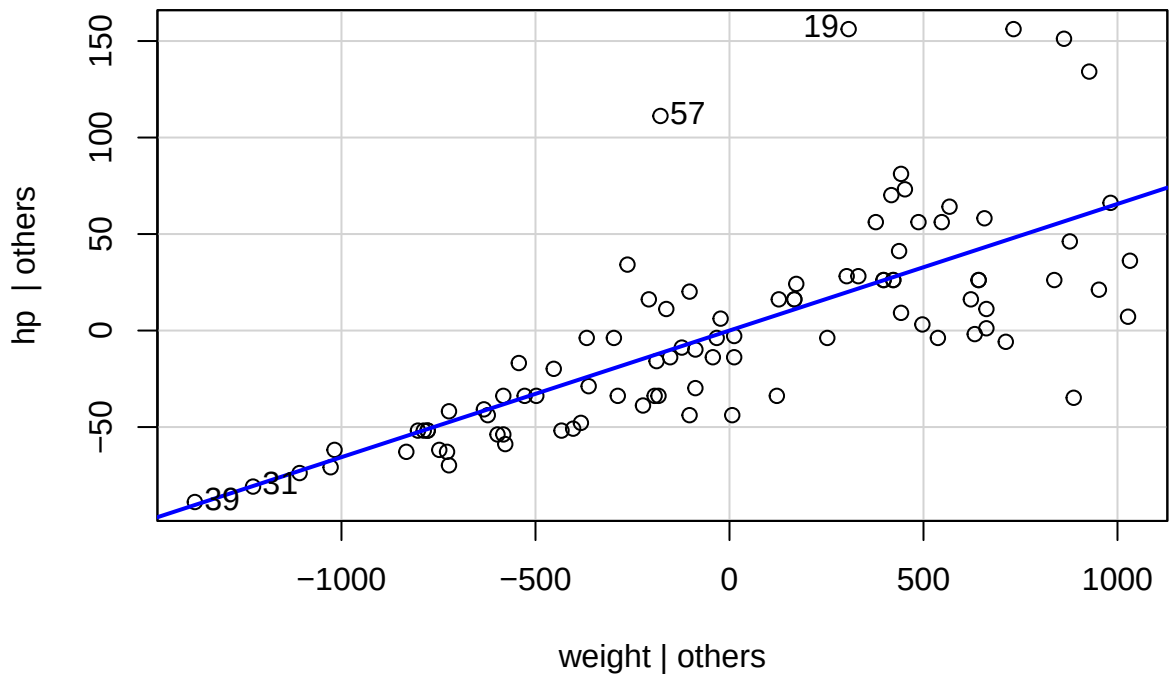
```
## Cargando paquete requerido: carData
```

```
##
## Adjuntando el paquete: 'car'
```

```
## The following object is masked from 'package:psych':
##
##     logit
```

```
## The following object is masked from 'package:VGAM':
##
##     logit
```

```r
avPlots(modelo1)
```



Residuos estandarizados absolutos e identifica aquellos cuyo valor absoluto es mayor a 3.

```r
plot(abs(residuals_values), ylab = "Residuos estandarizados absolutos",
     xlab = "Índice de observación", pch = 19, col = "blue", main = "Residuos estandarizados")

# Identificar aquellos cuyo valor absoluto es mayor a 3
```

```r
abline(h = 3, col = "red", lty = 2)
abline(h = -3, col = "red", lty = 2)
```

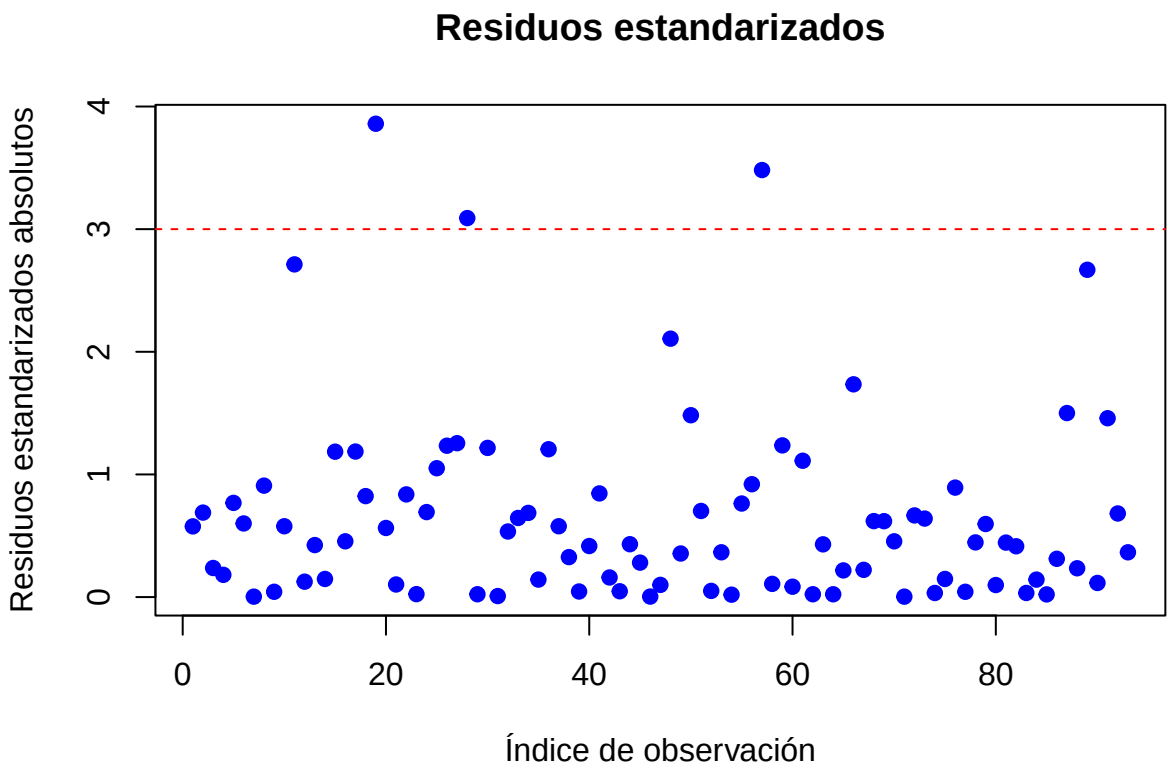
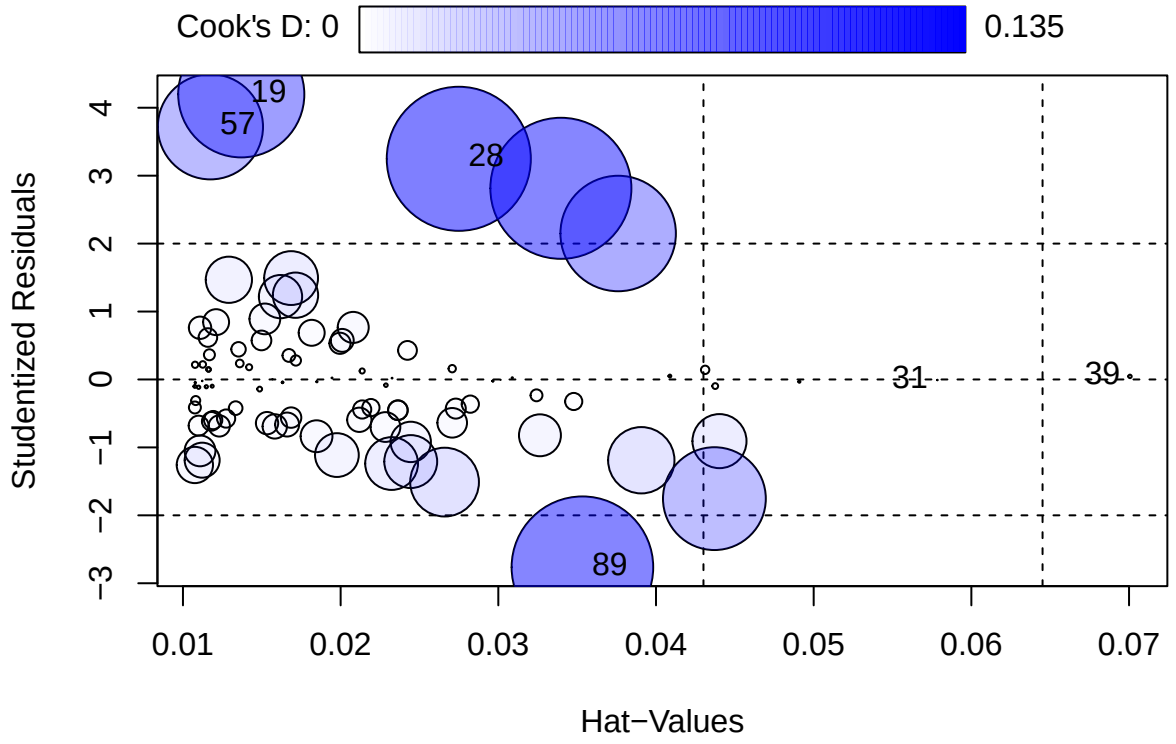


Gráfico de influencia

```r
influencePlot(modelo1, id=TRUE)
```




```
##         StudRes        Hat        CookD
## 19  4.196938831 0.01369854 1.034359e-01
## 28  3.248248107 0.02749431 1.349813e-01
## 31 -0.008199298 0.05784921 2.086887e-06
## 39  0.045191368 0.07005860 7.778142e-05
## 57  3.719207655 0.01174131 7.201528e-02
## 89 -2.764150849 0.03533386 1.304124e-01
```

Se usó el criterio de residuos estandarizados, asi como ser observó si los valores tenian influencia hacia el modelo, para decidir si eliminarlos o no

E) Calcule:

1) Intervalos de confianza para los coeficeintes de la regresión

```
level <- 1-0.05
confint(modelo1, level= level)
```

```
##                    2.5 %       97.5 %
## (Intercept) -96.71638922 -18.76001719
## weight        0.05313529   0.07805411
```

2) Intevalos de confianza para la respuesta media de la regresión

```
colMeans(predict(modelo1 ,interval = "confidence", level=level))
```

```
##      fit      lwr      upr
```

```
## 143.8280 133.7913 153.8646
```

3) Intervalos de predicción

```
colMeans(predict(modelo1, interval = "prediction", level=level))
```

```
## Warning in predict.lm(modelo1, interval = "prediction", level = level): predictions on current data
```

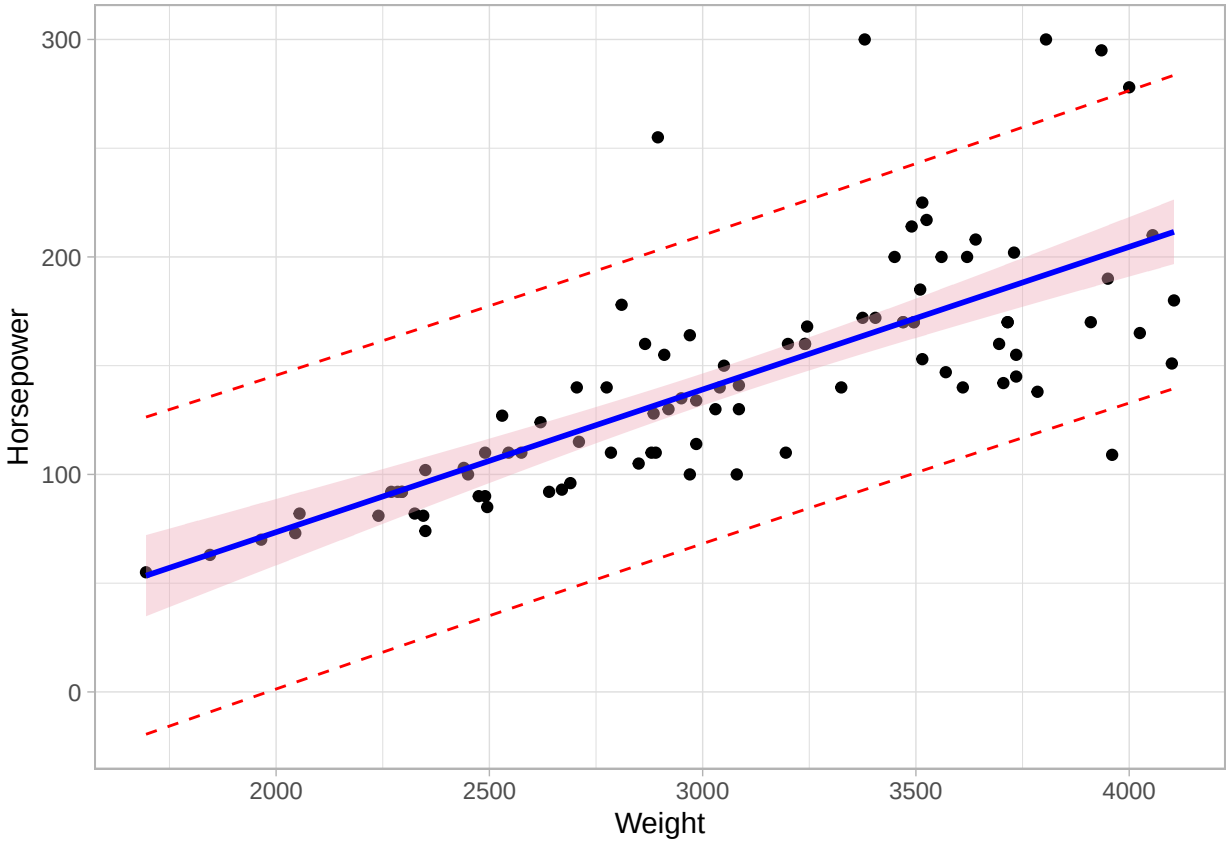
```
##       fit       lwr       upr
## 143.82796  72.57862 215.07730
```

F) Realice un gráfico para mostrar los intervalos de confianza

```
Yp <- predict(modelo1,interval="prediction",level=level)
```

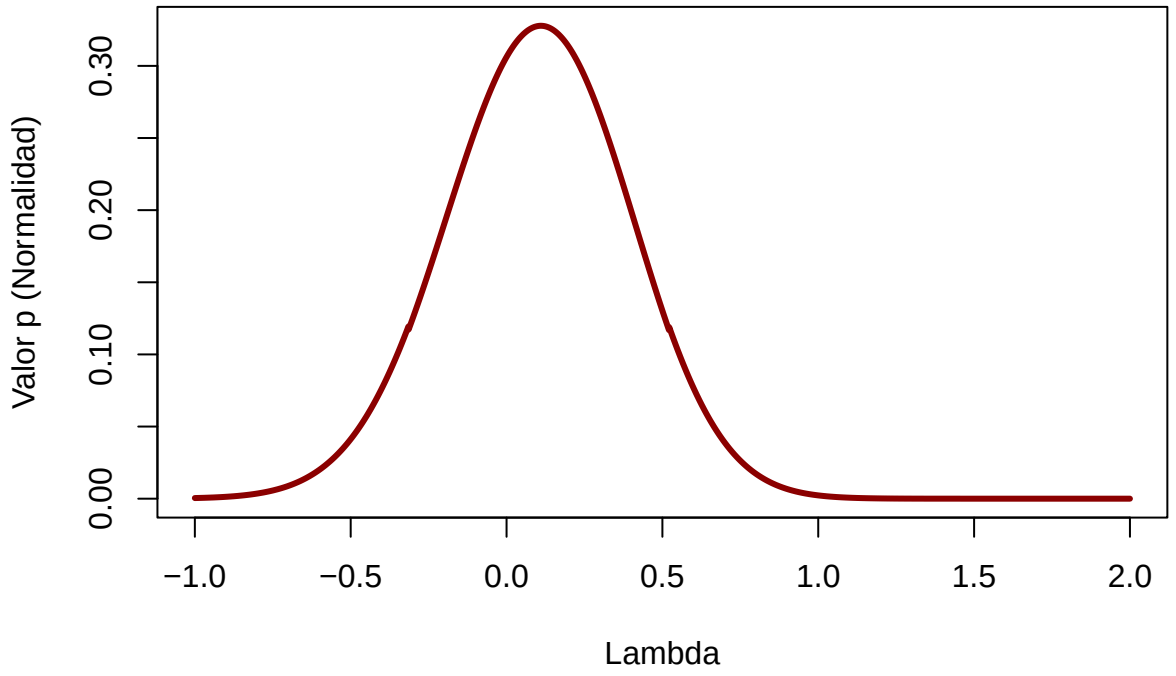
```
## Warning in predict.lm(modelo1, interval = "prediction", level = level): predictions on current data
```

```
datos1=cbind(x,Yp)
library(ggplot2)
ggplot(datos1,aes(x=Weight,y=Horsepower))+
geom_point()+
geom_line(aes(y=lwr), color="red", linetype="dashed")+
geom_line(aes(y=upr), color="red", linetype="dashed")+
geom_smooth(method=lm, formula=y~x, se=TRUE, level=0.95,
col='blue', fill='pink2') + theme_light()
```



G) Proponga un segundo modelo implementando una transformación a la variable de potencia, para cumplir normalidad

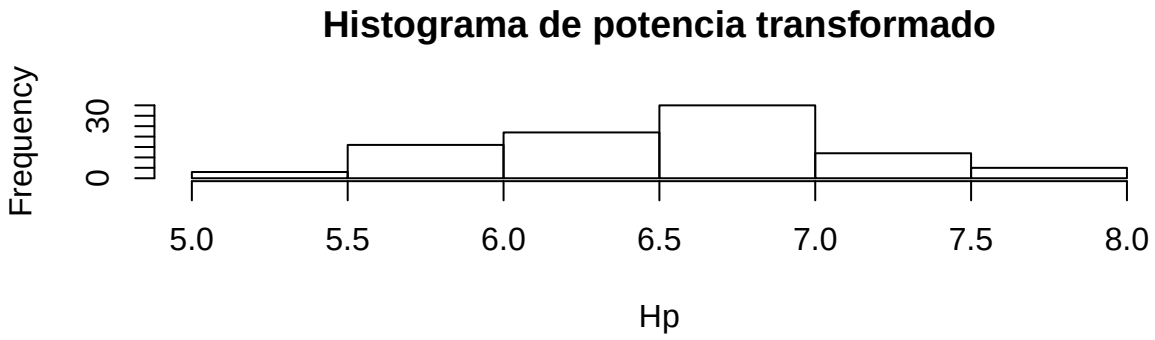
```
lp <- seq(-1,2,0.001)
nlp <- length(lp)
n=length(x$Horsepower)
D <- matrix(as.numeric(NA),ncol=2,nrow=nlp)
d <- NA
for (i in 1:nlp){
  d=yeo.johnson(x$Horsepower,lambda=lp[i])
  p=ad.test(d)
  D[i,]=c(lp[i],p$p.value)}

N = as.data.frame(D)
plot(N$V1,N$V2,type="l",col="darkred",lwd=3,xlab="Lambda",ylab="Valor p (Normalidad)")
```

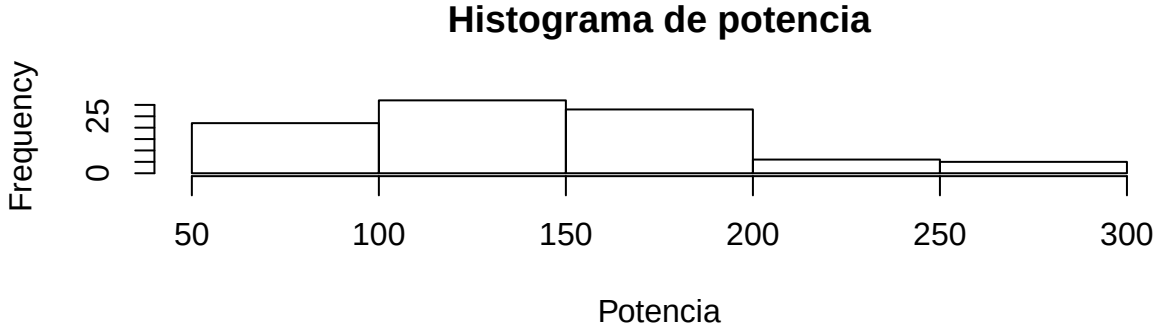



```
G = data.frame(subset(N,N$V2==max(N$V2)))
G
```

```
##        V1        V2
## 1111 0.11 0.3277887
```

```
Hp <- yeo.johnson(x$Horsepower,G$V1)
par(mfrow=c(2,1))
hist(Hp,col=0,main="Histograma de potencia transformado")
text(x=3, y=6, expression(speed2= frac((x+1)^0.438-1,0.438)))
hist(x$Horsepower,col=0,main="Histograma de potencia", xlab="Potencia")
```





H) Contraste los resultados de la validación del segundo modelo con el obtenido inicialmente

```
modelo2 <- lm(Hp ~ weight)
summary(modelo2)
```

```
##
## Call:
## lm(formula = Hp ~ weight)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.11022 -0.23509  0.00715  0.17045  1.25898
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.976e+00  2.016e-01   19.73   <2e-16 ***
## weight      8.307e-04  6.443e-05   12.89   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3646 on 91 degrees of freedom
## Multiple R-squared:  0.6462, Adjusted R-squared:  0.6423
## F-statistic: 166.2 on 1 and 91 DF,  p-value: < 2.2e-16
```

Validación

Individual, conjunta y correlación

```
summary(modelo2)
```

```
##
## Call:
## lm(formula = Hp ~ weight)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.11022 -0.23509  0.00715  0.17045  1.25898
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.976e+00  2.016e-01   19.73   <2e-16 ***
## weight      8.307e-04  6.443e-05   12.89   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3646 on 91 degrees of freedom
## Multiple R-squared:  0.6462, Adjusted R-squared:  0.6423
## F-statistic: 166.2 on 1 and 91 DF,  p-value: < 2.2e-16
```

T Test

- H_0 := El coeficiente es igual a 0.
- H_A := El coeficiente no es igual a 0.

Linealidad

- H_0 := La relación entre la variable independiente y dependiente es lineal
- H_A := La relación es no lineal

```
resettest(modelo2)
```

```
##
##  RESET test
##
## data:  modelo2
## RESET = 5.3149, df1 = 2, df2 = 89, p-value = 0.0066
```

Media de cero de los residuos

T Test

- H_0 := La media de los errores es igual a 0.
- H_A := La media de los errores no es igual a 0.

```
t.test(modelo2$residuals)
```

```
##
##  One Sample t-test
##
## data:  modelo2$residuals
## t = -1.0757e-16, df = 92, p-value = 1
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
```

```
##  -0.07466982  0.07466982
## sample estimates:
##     mean of x
## -4.044191e-18
```

Normalidad de residuos

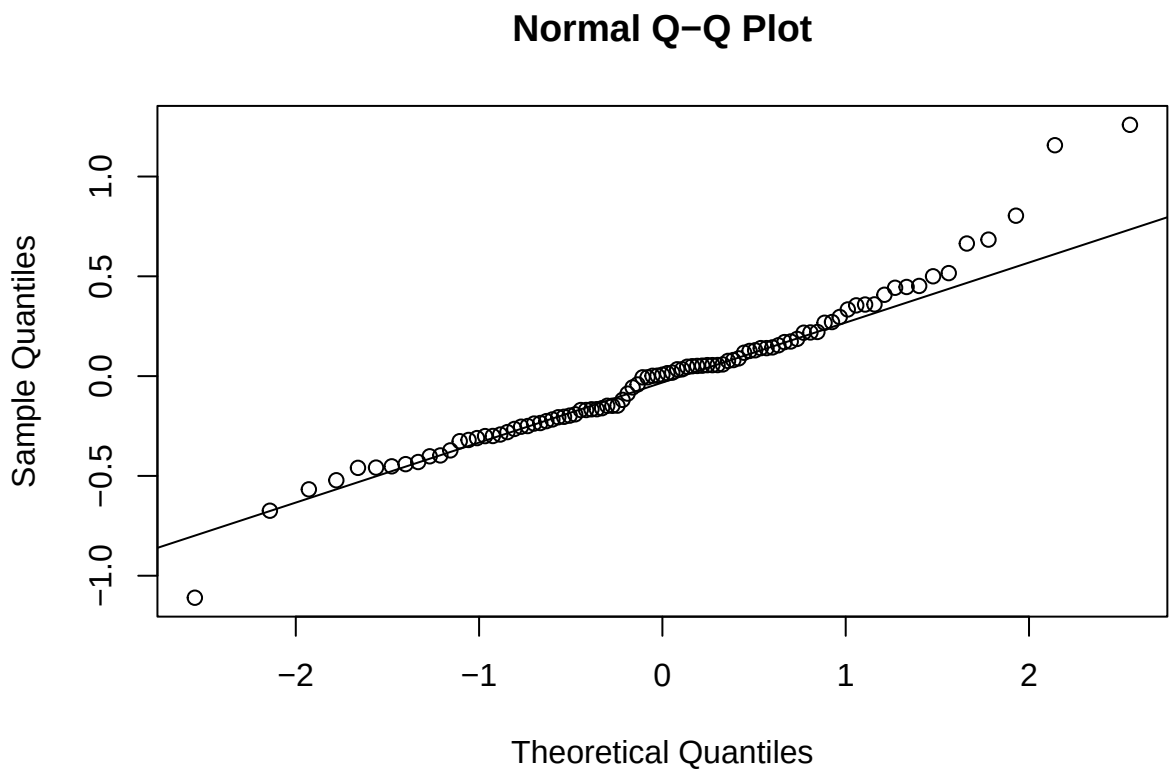
T Test

- H_0 := Los residuos tienen una distribución normal
- H_A := Los residuos no tienen una distribución normal

```
shapiro.test(modelo2$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  modelo2$residuals
## W = 0.95993, p-value = 0.006036
```

```
qqnorm(modelo2$residuals)
qqline(modelo2$residuals)
```



```r
hist(modelo2$residuals)
```

Breusch-Pagan

- H_0 := Los datos tienen homocedasticidad.
- H_A := Los datos no tienen homocedasticidad.

```r
bptest(modelo2)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  modelo2
## BP = 5.3402, df = 1, p-value = 0.02084
```

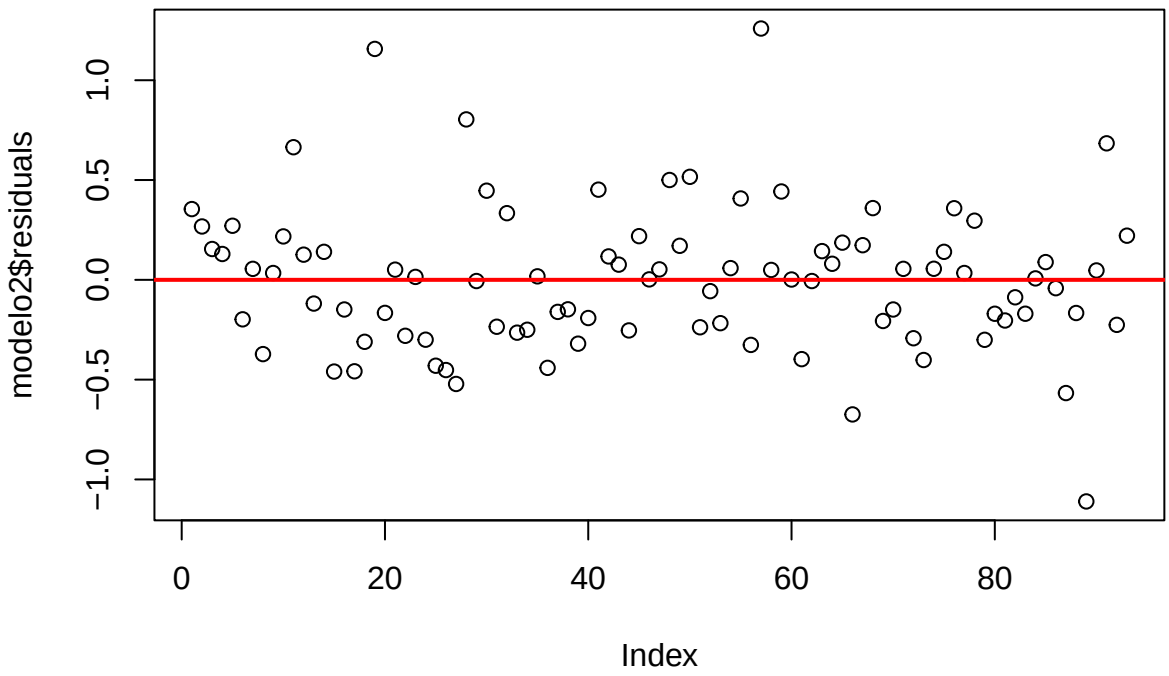
Independencia Test de Durbin Watson

- H_0 := No existe autocorrelación en los datos
- H_A := Existe autocorrelacion en los datos.

```r
dwtest(modelo2)
```

```
##
##  Durbin-Watson test
##
## data:  modelo2
## DW = 2.0414, p-value = 0.5733
## alternative hypothesis: true autocorrelation is greater than 0
```

```r
plot(modelo2$residuals)
abline(h=0, col = "red", lwd = 2)
```



#Prueba Breusch-Godfrey

```r
bgtest(modelo2)
```

```
##
##  Breusch-Godfrey test for serial correlation of order up to 1
##
## data:  modelo2
## LM test = 0.072691, df = 1, p-value = 0.7875
```

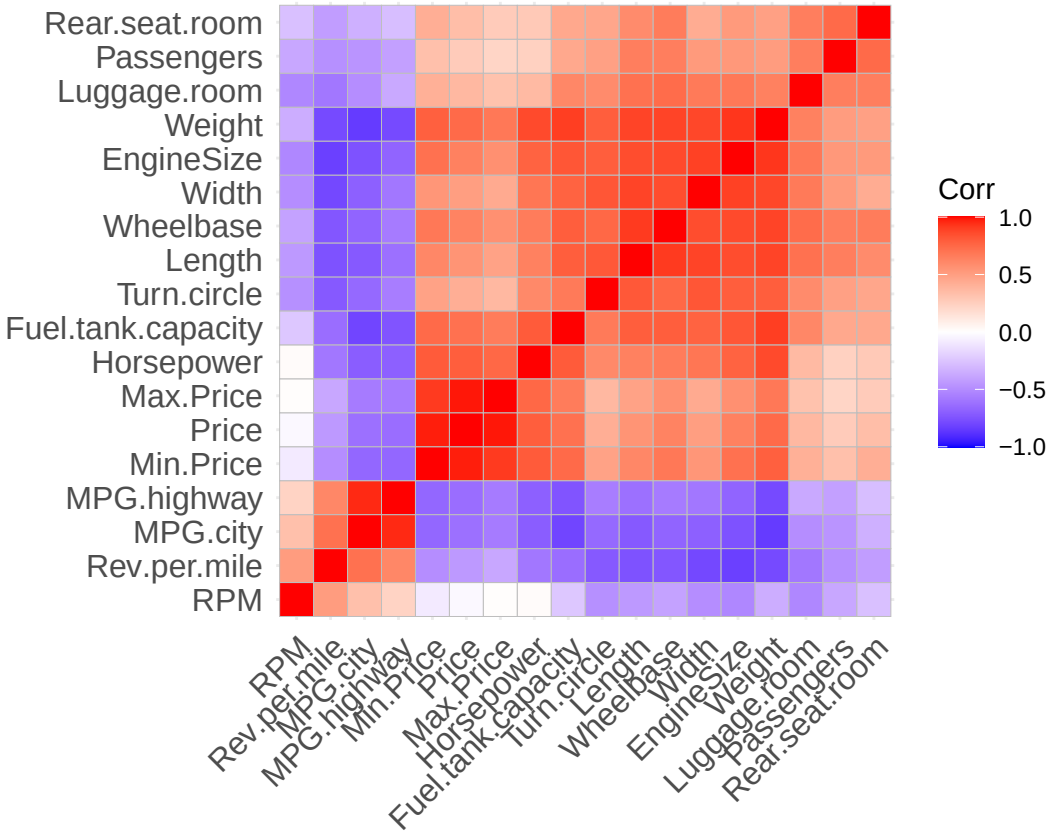
Problema 2)

A) Realice el análisis de correlación entre las variables numéricas y seleccione un conjunto de variables numéricas puedan explicar la variabilidad del precio del vehículo

```r
df <- Cars93
x <-  dplyr::select_if(df, is.numeric)
```

```r
#Cargamos librerias
library(polycor)
library(ggcorrplot)
```

```r
mat_cor <- hetcor(x)$correlations
ggcorrplot(mat_cor,hc.order = T)
```



Dadas estas variables se tomaran las variables de MPG CITY, MPG highway, Horsepower, Fuel tank y weight

B) A partir de las variables, ajuste un modelo de regresión lineal múltiple

```r
variables <- x[c("MPG.city","MPG.highway","Horsepower","Fuel.tank.capacity","Weight")]

modelo1 <- lm(df$Price ~ variables$MPG.city+variables$MPG.highway+variables$Horsepower+variables$Fuel.ta
```

C) Realice la validación de los supuestos del modelo

Individual, conjunta y correlación T Test

- H_0 := El coeficiente es igual a 0.
- H_A := El coeficiente no es igual a 0.

```r
summary(modelo1)
```

```
##
## Call:
## lm(formula = df$Price ~ variables$MPG.city + variables$MPG.highway +
##     variables$Horsepower + variables$Fuel.tank.capacity + variables$Weight)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -15.038  -2.739  -0.711   1.612  32.031
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  -0.191265  11.614764  -0.016    0.987
## variables$MPG.city            0.038738   0.376379   0.103    0.918
## variables$MPG.highway        -0.145186   0.361341  -0.402    0.689
## variables$Horsepower          0.125466   0.018286   6.861 9.51e-10 ***
## variables$Fuel.tank.capacity  0.035551   0.446798   0.080    0.937
## variables$Weight              0.001438   0.002751   0.523    0.602
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.025 on 87 degrees of freedom
## Multiple R-squared:  0.6321, Adjusted R-squared:  0.6109
## F-statistic: 29.89 on 5 and 87 DF,  p-value: < 2.2e-16
```

Linealidad

- H_0 := La relación entre la variable independiente y dependiente es lineal
- H_A := La relación es no lineal

```
resettest(modelo1)
```

```
##
##  RESET test
##
## data:  modelo1
## RESET = 1.7998, df1 = 2, df2 = 85, p-value = 0.1716
```

Media de cero de los residuos

T Test

- H_0 := La media de los errores es igual a 0.
- H_A := La media de los errores no es igual a 0.

```
t.test(modelo1$residuals)
```

```
##
##  One Sample t-test
##
## data:  modelo1$residuals
## t = -8.9202e-17, df = 92, p-value = 1
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -1.206683  1.206683
## sample estimates:
##     mean of x
## -5.419612e-17
```

Normalidad de residuos

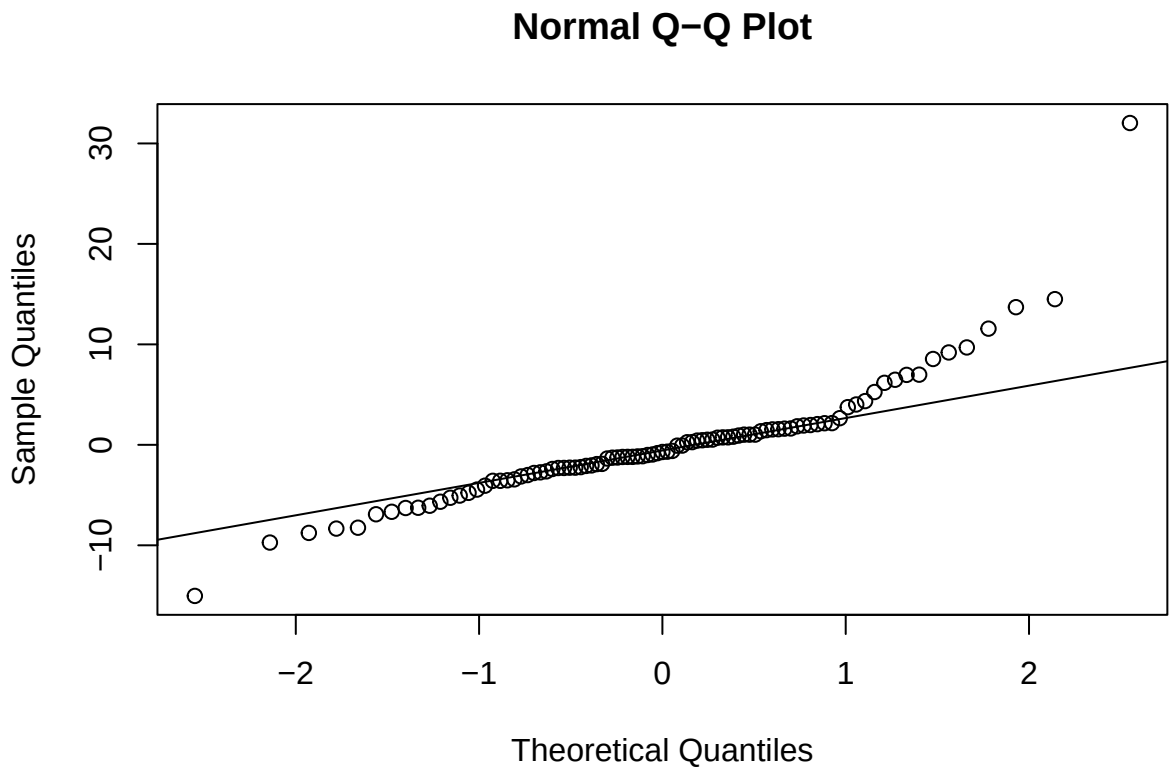
T Test

- H_0 := Los residuos tienen una distribución normal
- H_A := Los residuos no tienen una distribución normal

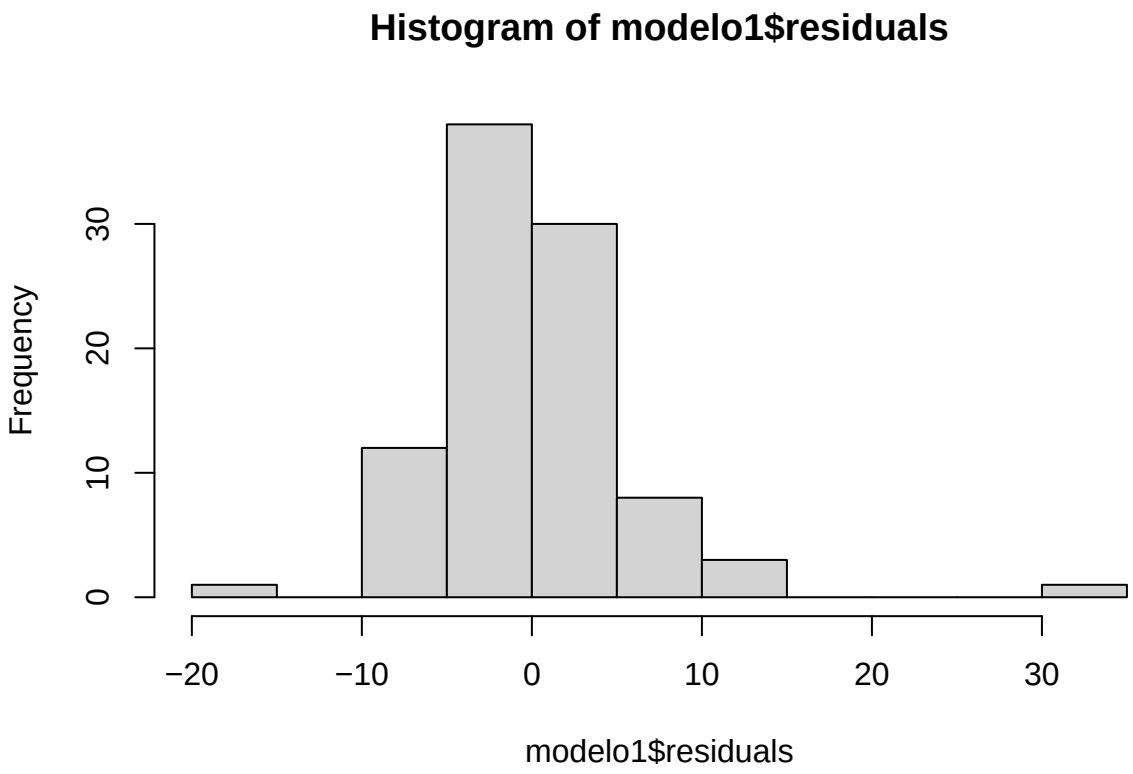
```r
shapiro.test(modelo1$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  modelo1$residuals
## W = 0.85556, p-value = 4.565e-08
```

```r
qqnorm(modelo1$residuals)
qqline(modelo1$residuals)
```




```r
hist(modelo1$residuals)
```



Breusch-Pagan

- H_0 := Los datos tienen homocedasticidad.
- H_A := Los datos no tienen homocedasticidad.

```
bptest(modelo1)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  modelo1
## BP = 8.2217, df = 5, p-value = 0.1444
```

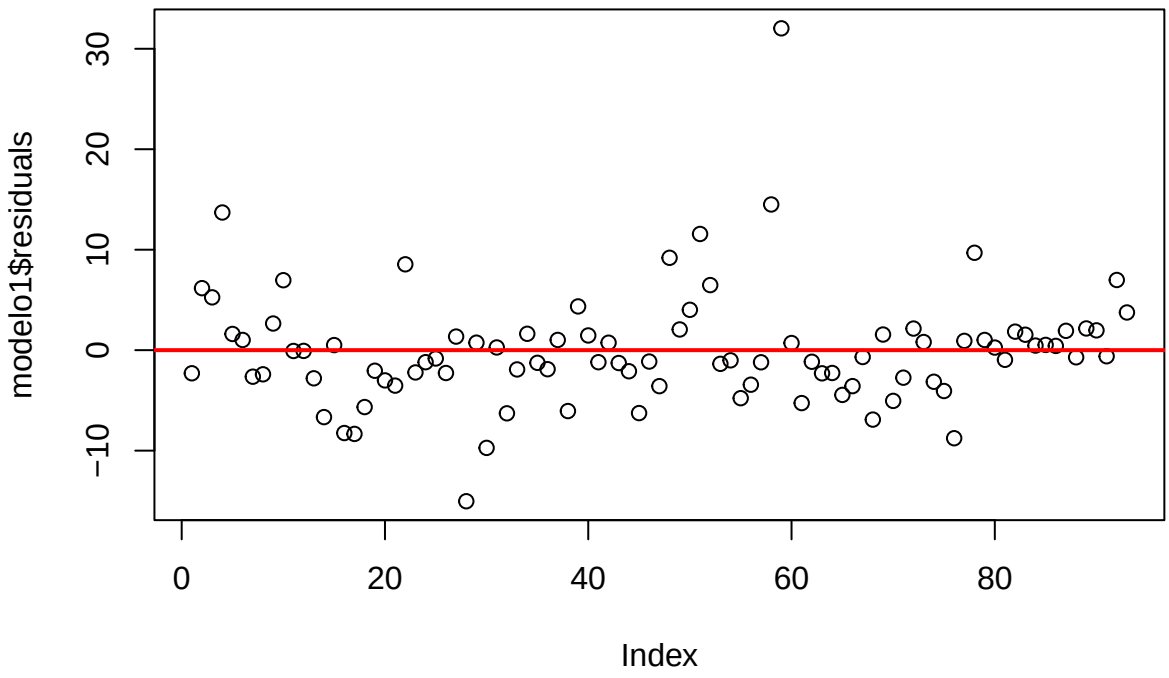
Independencia Test de Durbin Watson

- H_0 := No existe autocorrelación en los datos
- H_A := Existe autocorrelacion en los datos.

```
dwtest(modelo1)
```

```
##
##  Durbin-Watson test
##
## data:  modelo1
## DW = 1.4117, p-value = 0.00137
## alternative hypothesis: true autocorrelation is greater than 0
```

```
plot(modelo1$residuals)
abline(h=0, col = "red", lwd = 2)
```



D) Identifique los datos atípicos y datos influyentes y describa los criterios implementados para su determinación

Metodo de desviación estandar

```
residuals_values<- rstandard(modelo1)
```

Metodo de estandarización

```
rstudents_values<-rstudent(modelo1)
```

Datos influyentes Por grado de leverage

```
hat_values<-hatvalues(modelo1)
```

Por distance de Cook

```
cooks_values<-cooks.distance(modelo1)
```

Resumen de resultados

```
tabla= data.frame(residuals_values,rstudents_values, hat_values, cooks_values)
tabla
```

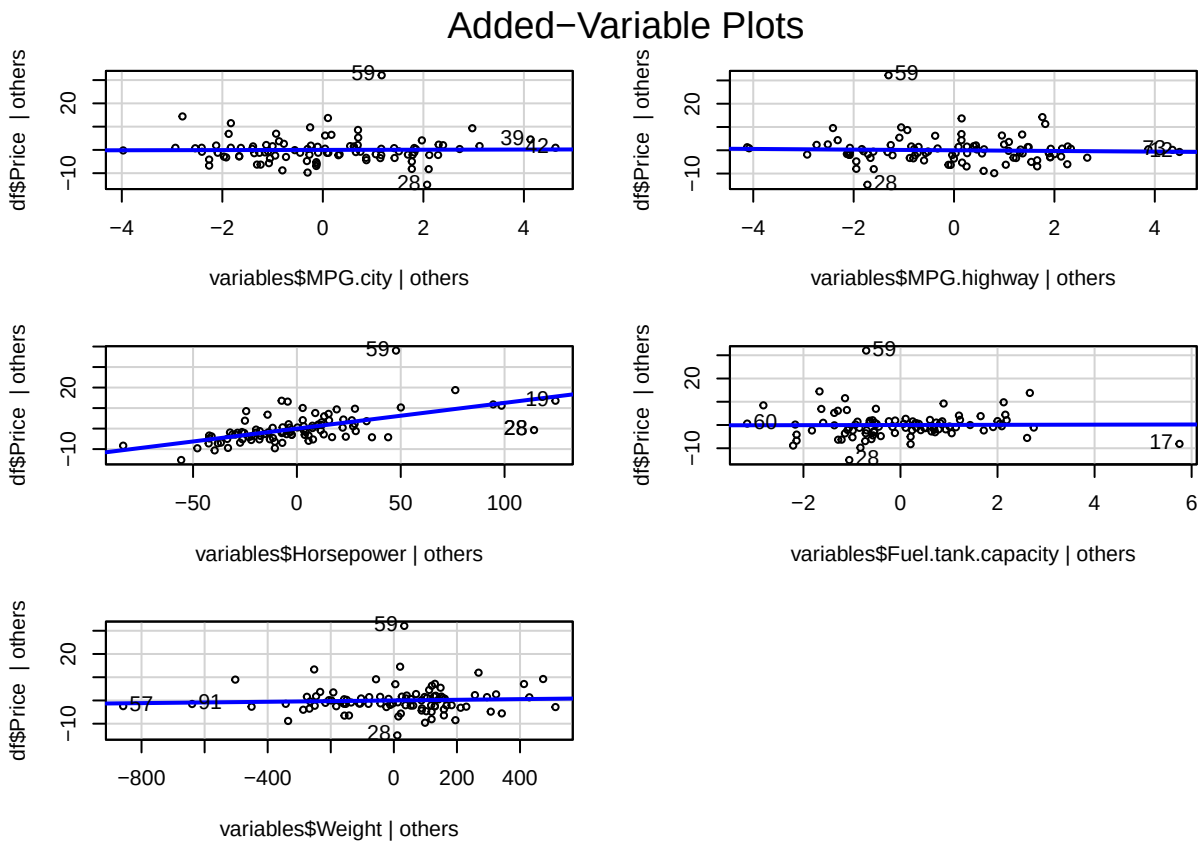
```
##    residuals_values rstudents_values hat_values cooks_values
## 1       -0.38946664      -0.38755986 0.03826500 1.005855e-03
## 2        1.04129739       1.04180812 0.03272143 6.113346e-03
## 3        0.88482093       0.88370625 0.02785555 3.738871e-03
## 4        2.33841031       2.40163126 0.05413240 5.215754e-02
## 5        0.27899177       0.27750790 0.07981546 1.125236e-03
```

```
## 6        0.17193951       0.17097755 0.04299253 2.213491e-04
## 7       -0.44505657      -0.44299596 0.03129362 1.066456e-03
## 8       -0.41739517      -0.41540555 0.08602247 2.732876e-03
## 9        0.44446491       0.44240570 0.01993450 6.696901e-04
## 10       1.18552682       1.18833151 0.05012634 1.236152e-02
## 11      -0.01454204      -0.01445824 0.13096519 5.311509e-06
## 12      -0.01286414      -0.01279000 0.10655412 3.289365e-06
## 13      -0.47514337      -0.47301890 0.04195495 1.647765e-03
## 14      -1.13735379      -1.13930004 0.05406970 1.232352e-02
## 15       0.08538300       0.08489443 0.03553587 4.476849e-05
## 16      -1.39783423      -1.40565194 0.04049715 1.374480e-02
## 17      -1.59967958      -1.61437892 0.25187471 1.435903e-01
## 18      -0.98706182      -0.98691431 0.09338865 1.672670e-02
## 19      -0.37562193      -0.37376014 0.17990122 5.158442e-03
## 20      -0.50681792      -0.50464227 0.02852789 1.257164e-03
## 21      -0.59304015      -0.59081742 0.02353031 1.412492e-03
## 22       1.47339011       1.48352371 0.07357856 2.873604e-02
## 23      -0.37730438      -0.37543699 0.04818875 1.201233e-03
## 24      -0.20213362      -0.20101578 0.03661216 2.587915e-04
## 25      -0.13991275      -0.13912198 0.03561707 1.204958e-04
## 26      -0.39181723      -0.38990306 0.07205523 1.986823e-03
## 27       0.22861308       0.22736371 0.03562189 3.217515e-04
## 28      -2.71457050      -2.82103825 0.15469627 2.247595e-01
## 29       0.12681911       0.12609981 0.04641693 1.304776e-04
## 30      -1.64889345      -1.66562255 0.04136479 1.955291e-02
## 31       0.04750456       0.04723137 0.12227271 5.239494e-05
## 32      -1.06875672      -1.06964166 0.04702109 9.393250e-03
## 33      -0.32713399      -0.32544870 0.05138815 9.662166e-04
## 34       0.27386059       0.27239957 0.02523607 3.236161e-04
## 35      -0.21124403      -0.21008035 0.01992452 1.511980e-04
## 36      -0.32654345      -0.32486048 0.07043173 1.346535e-03
## 37       0.17317841       0.17220994 0.04955771 2.606284e-04
## 38      -1.03218601      -1.03257876 0.04999730 9.345153e-03
## 39       0.85623783       0.85491247 0.28793255 4.940913e-02
## 40       0.24748721       0.24614742 0.03678305 3.898324e-04
## 41      -0.20087035      -0.19975891 0.02026900 1.391252e-04
## 42       0.14348729       0.14267715 0.26250485 1.221388e-03
## 43      -0.21412456      -0.21294652 0.01652595 1.284060e-04
## 44      -0.35763959      -0.35583992 0.05079231 1.140714e-03
## 45      -1.06345377      -1.06426424 0.04444228 8.766483e-03
## 46      -0.19302218      -0.19195076 0.05286412 3.465867e-04
## 47      -0.60604080      -0.60382366 0.03672872 2.334048e-03
## 48       1.62844945       1.64431766 0.12183997 6.132160e-02
## 49       0.34663774       0.34487805 0.02579268 5.302070e-04
## 50       0.69109126       0.68900181 0.06990025 5.982307e-03
## 51       1.96838299       2.00212770 0.04916173 3.338785e-02
## 52       1.11300391       1.11455225 0.06685350 1.479164e-02
## 53      -0.23126423      -0.23000199 0.04516147 4.216031e-04
## 54      -0.17204852      -0.17108599 0.04238083 2.183370e-04
## 55      -0.80907704      -0.80745720 0.03912474 4.442352e-03
## 56      -0.58126449      -0.57903969 0.03674035 2.147812e-03
## 57      -0.22661471      -0.22537509 0.21979830 2.411251e-03
## 58       2.47733220       2.55481241 0.05615262 6.085349e-02
## 59       5.43353957       6.64639801 0.04270604 2.195120e-01
```

```
## 60       0.12821994       0.12749296 0.15466142 5.013156e-04
## 61      -0.89171212      -0.89065200 0.03896764 5.373585e-03
## 62      -0.19610733      -0.19502013 0.04641693 3.119998e-04
## 63      -0.39008465      -0.38817592 0.03766008 9.924741e-04
## 64      -0.38559340      -0.38369895 0.04384379 1.136285e-03
## 65      -0.74569078      -0.74377351 0.01700906 1.603604e-03
## 66      -0.62307468      -0.62087026 0.09009002 6.406299e-03
## 67      -0.11493706      -0.11428327 0.03041320 6.906281e-05
## 68      -1.15823106      -1.16053755 0.01942226 4.428503e-03
## 69       0.26216320       0.26075518 0.02866518 3.380476e-04
## 70      -0.85563847      -0.85430895 0.04049715 5.150004e-03
## 71      -0.46191951      -0.45982133 0.03129362 1.148802e-03
## 72       0.36469292       0.36286840 0.04186133 9.684743e-04
## 73       0.14235196       0.14154797 0.10933341 4.145848e-04
## 74      -0.53259826      -0.53039388 0.04012564 1.976313e-03
## 75      -0.69367052      -0.69158755 0.05406970 4.584058e-03
## 76      -1.49506334      -1.50591716 0.05294711 2.082744e-02
## 77       0.15606565       0.15518785 0.03277880 1.375722e-04
## 78       1.66991416       1.68755511 0.07150271 3.579142e-02
## 79       0.17601401       0.17503068 0.08279575 4.661065e-04
## 80       0.04471662       0.04445939 0.08612982 3.140911e-05
## 81      -0.16412152      -0.16320084 0.05762171 2.744990e-04
## 82       0.30837651       0.30676681 0.01594099 2.567470e-04
## 83       0.27280468       0.27134839 0.12455407 1.764741e-03
## 84       0.07774907       0.07730363 0.04938317 5.233746e-05
## 85       0.08747612       0.08697575 0.01577803 2.044502e-05
## 86       0.07157201       0.07116159 0.03609817 3.197331e-05
## 87       0.33299961       0.33129149 0.07830230 1.570082e-03
## 88      -0.12185649      -0.12116448 0.06164321 1.625786e-04
## 89       0.38602664       0.38413080 0.13724948 3.951016e-03
## 90       0.33697806       0.33525467 0.05841657 1.174165e-03
## 91      -0.10664299      -0.10603526 0.10359664 2.190562e-04
## 92       1.17367218       1.17625667 0.02431665 5.721861e-03
## 93       0.63271605       0.63052158 0.03221907 2.221275e-03
```

Variable dependiente contra las variables predictoras

```
avPlots(modelo1)
```



E) Calcule:

Intervalos de confianza para los coeficientes de la regresión

```
level <- 1-0.05
confint(modelo1, level= level)
```

```
##                                   2.5 %       97.5 %
## (Intercept)                 -23.276866258 22.894335490
## variables$MPG.city           -0.709356294  0.786832047
## variables$MPG.highway        -0.863390515  0.573019138
## variables$Horsepower          0.089121198  0.161810032
## variables$Fuel.tank.capacity -0.852508278  0.923610190
## variables$Weight             -0.004029423  0.006905869
```

Intervalos de confianza para la respuesta media de la regresión

```
head(predict(modelo1 ,interval = "confidence", level=level),3)
```

```
##        fit      lwr      upr
## 1 18.20128 15.85865 20.54390
## 2 27.72949 25.56319 29.89578
## 3 23.84356 21.84482 25.84231
```

Intervalos de predicción

```
head(predict(modelo1, interval = "prediction", level=level),3)
```

```
## Warning in predict.lm(modelo1, interval = "prediction", level = level): predictions on current data
```

```
##        fit       lwr      upr
## 1 18.20128  5.998584 30.40397
## 2 27.72949 15.559416 39.89956
## 3 23.84356 11.702196 35.98493
```

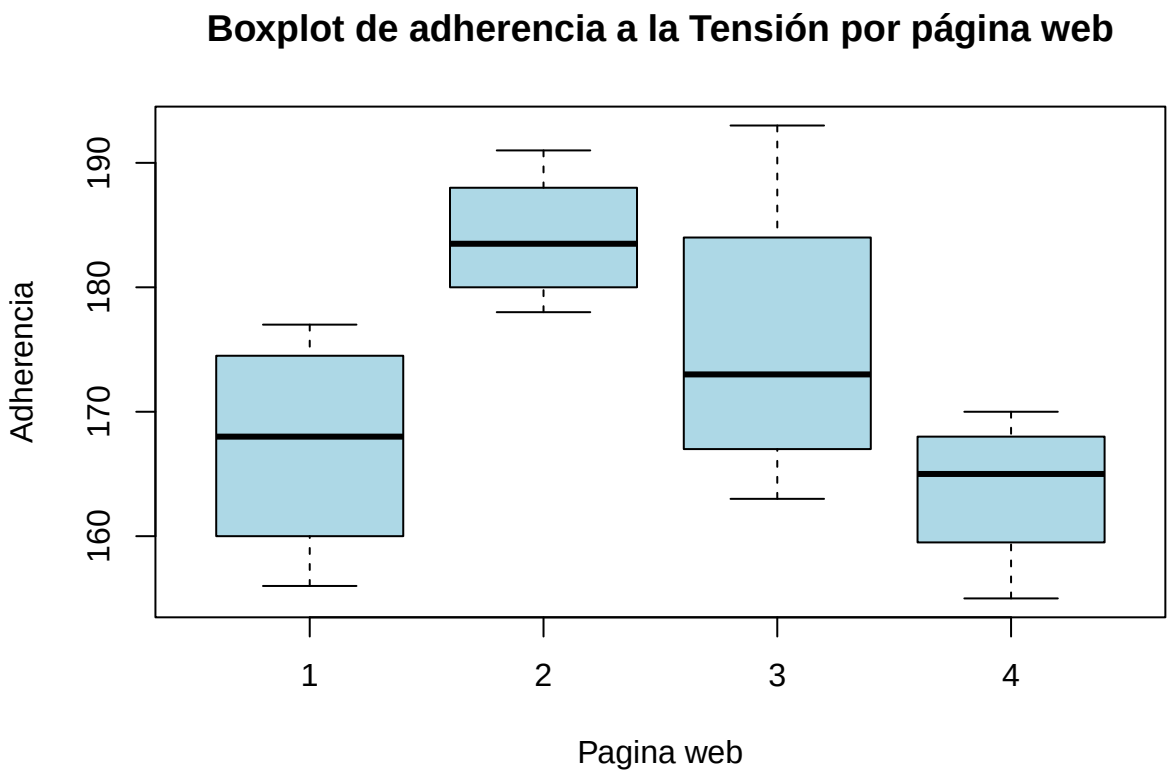
F) Interpreta los resultados desde la perspectiva estadística y en el contexto del problema

Problema 3

```
df <- data.frame(Adherencia=c(164,172,177,156,178,191,182,185,175,193,171,163,155,166,164,170) , Pagina=
```

A) Realice un gráfico de caja y bigotes

```
boxplot(Adherencia ~ Pagina, data = df,
        main = "Boxplot de adherencia a la Tensión por página web",
        xlab = "Pagina web",
        ylab = "Adherencia",
        col = "lightblue", border = "black")
```



B) Estime la media para la adherencia en cada sitio web

2) Hipótesis estadística

Hipótesis nula H_0: No hay diferencias significativas en la adherencia por página web

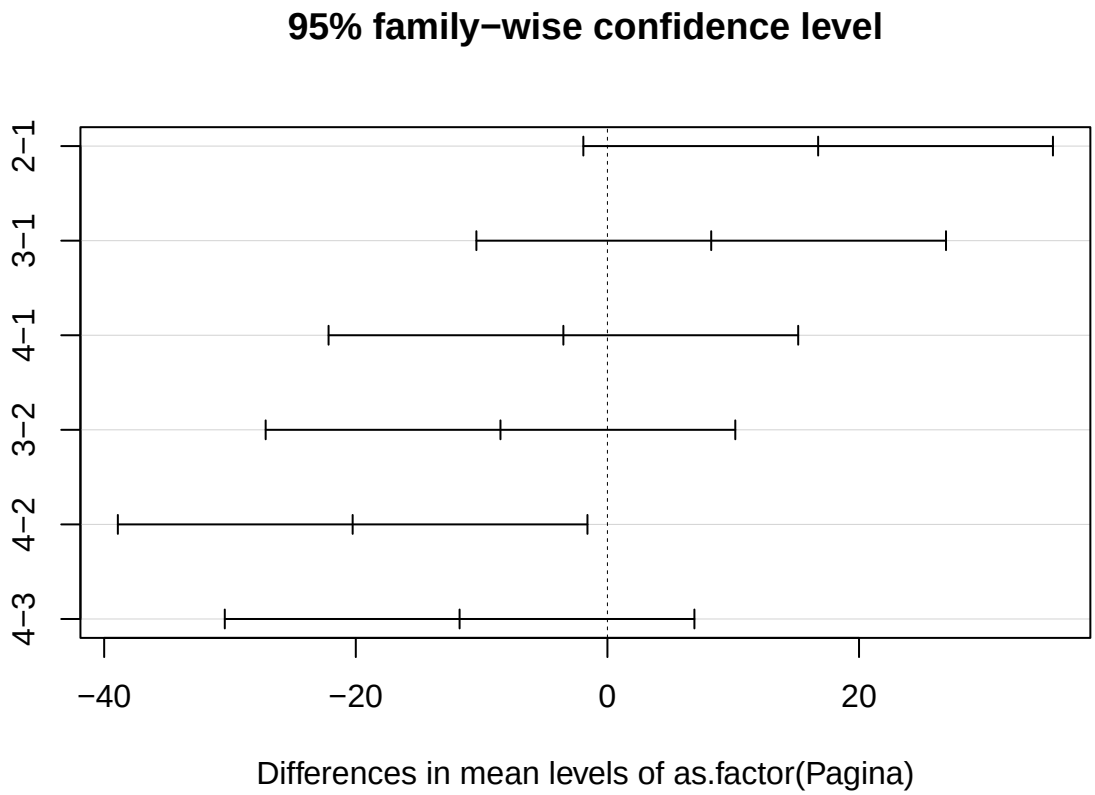
Hipótesis alternativa H_1: Existen diferencias significativas en la adherencia por página web entre al menos dos grupos.

```
modelo <- aov(Adherencia ~ as.factor(Pagina), data = df)
summary(modelo)
```

```
##                   Df Sum Sq Mean Sq F value Pr(>F)
## as.factor(Pagina)  3  981.2   327.1   4.138 0.0314 *
## Residuals         12  948.5    79.0
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

C) Obtenga los intervalos de confianza para la adherencia media en cada sitio

```
tukey_result <- TukeyHSD(modelo)


plot(tukey_result)
```



D) Realice el análisis de varianza con un nivel de significancia

```
anova_model <- anova(lm(Adherencia ~ as.factor(Pagina),data=df))
anova_model
```

```
## Analysis of Variance Table
##
## Response: Adherencia
##                   Df Sum Sq Mean Sq F value  Pr(>F)
## as.factor(Pagina)  3 981.25  327.08  4.1381 0.03141 *
## Residuals         12 948.50   79.04
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

E) Analiza la validez del modelo

Normalidad Hipótesis nula H_0: No existe normalidad en los residuos del modelo

Hipótesis alternativa H_1: La normalidad en los residuos son normales

```
# Test de Shapiro-Wilk para normalidad
shapiro_test <- shapiro.test(residuals(modelo))
print(shapiro_test)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(modelo)
## W = 0.98126, p-value = 0.9728
```

Homocedasticidad Hipótesis nula H_0: Existe homocedasticidad en los residuos

Hipótesis alternativa H_1: La varianza varia entre los grupos, heterocedasticidad

```
library(lmtest)
# Test de Bartlett para homocedasticidad
bartlett_test <- bartlett.test(Adherencia ~ as.factor(Pagina), data = df)
print(bartlett_test)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  Adherencia by as.factor(Pagina)
## Bartlett's K-squared = 2.2667, df = 3, p-value = 0.5189
```

Independencia Hipótesis nula H_0: Las observaciones se obtuvieron de manera independiente

Hipótesis alternativa H_1: Las observaciones tienen correlación entre ellas.

```
tabla <- table(df$Adherencia, df$Pagina)

chisq.test(tabla)
```

```
## Warning in chisq.test(tabla): Chi-squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  tabla
## X-squared = 44, df = 42, p-value = 0.3869
```

F) Interpreta el resultado desde la perspectiva estadística y en el contexto del problema

Problema 4

```
df <- wine
prop.table(table(df$Type))
```

```
##
##         1         2         3
## 0.3314607 0.3988764 0.2696629
```

A) Mediante un análisis discriminante realice una clasificación de la base de datos en los 3 diferentes grupos asociados los tipos de cultivares de vino.

```
lda.model = lda(Type~., data=df)
lda.model
```

```
## Call:
## lda(Type ~ ., data = df)
##
## Prior probabilities of groups:
##         1         2         3
## 0.3314607 0.3988764 0.2696629
##
## Group means:
##    Alcohol    Malic      Ash Alcalinity Magnesium  Phenols Flavanoids
## 1 13.74475 2.010678 2.455593   17.03729  106.3390 2.840169  2.9823729
## 2 12.27873 1.932676 2.244789   20.23803   94.5493 2.258873  2.0808451
## 3 13.15375 3.333750 2.437083   21.41667   99.3125 1.678750  0.7814583
##   Nonflavanoids Proanthocyanins    Color       Hue Dilution   Proline
## 1      0.290000        1.899322 5.528305 1.0620339 3.157797 1115.7119
## 2      0.363662        1.630282 3.086620 1.0562817 2.785352  519.5070
## 3      0.447500        1.153542 7.396250 0.6827083 1.683542  629.8958
##
## Coefficients of linear discriminants:
##                          LD1           LD2
## Alcohol         -0.403399781  0.8717930699
## Malic            0.165254596  0.3053797325
## Ash             -0.369075256  2.3458497486
## Alcalinity       0.154797889 -0.1463807654
## Magnesium       -0.002163496 -0.0004627565
## Phenols          0.618052068 -0.0322128171
## Flavanoids      -1.661191235 -0.4919980543
## Nonflavanoids   -1.495818440 -1.6309537953
## Proanthocyanins  0.134092628 -0.3070875776
## Color            0.355055710  0.2532306865
## Hue             -0.818036073 -1.5156344987
## Dilution        -1.157559376  0.0511839665
## Proline         -0.002691206  0.0028529846
##
## Proportion of trace:
##    LD1    LD2
## 0.6875 0.3125
```

B) Escriba las funciones discriminantes implementadas por el modelo y el porcentaje de clasificación asociado a cada una de éstas.

$LD1 = -0.4034 \cdot Alcohol + 0.1653 \cdot Malic - 0.3691 \cdot Ash + 0.1548 \cdot Alcalinity - 0.0022 \cdot Magnesium + 0.6181 \cdot Phenols - 1.6612 \cdot Flava$

$LD2 = 0.8718 \cdot Alcohol + 0.3054 \cdot Malic + 2.3458 \cdot Ash - 0.1464 \cdot Alcalinity - 0.0005 \cdot Magnesium - 0.0322 \cdot Phenols - 0.4920 \cdot Flavan$

```
# Predicción sobre el conjunto de datos original
lda_predictions <- predict(lda.model)

# Tabla de clasificación real vs. predicha
classification_table <- table(wine$Type, lda_predictions$class)
classification_table
```

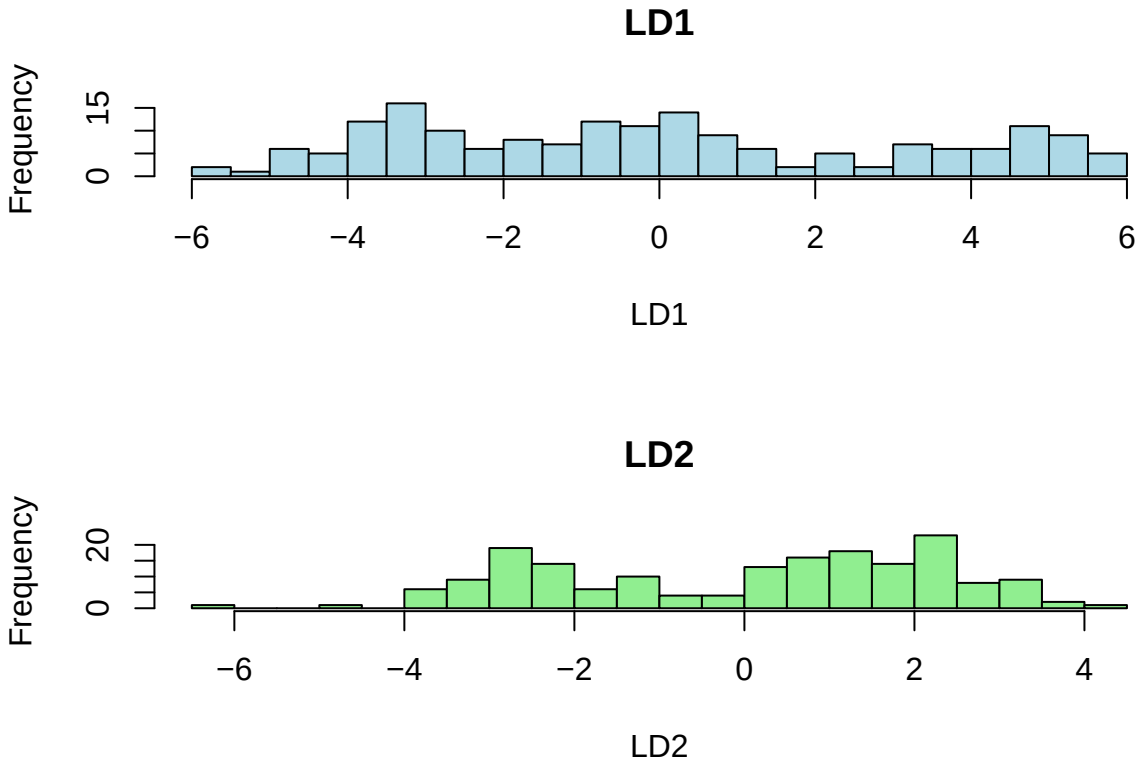
```
##
##      1  2  3
##   1 59  0  0
##   2  0 71  0
##   3  0  0 48
```

```
# Porcentaje de clasificación correcta
classification_percentage <- sum(diag(classification_table)) / sum(classification_table) * 100
classification_percentage
```

```
## [1] 100
```

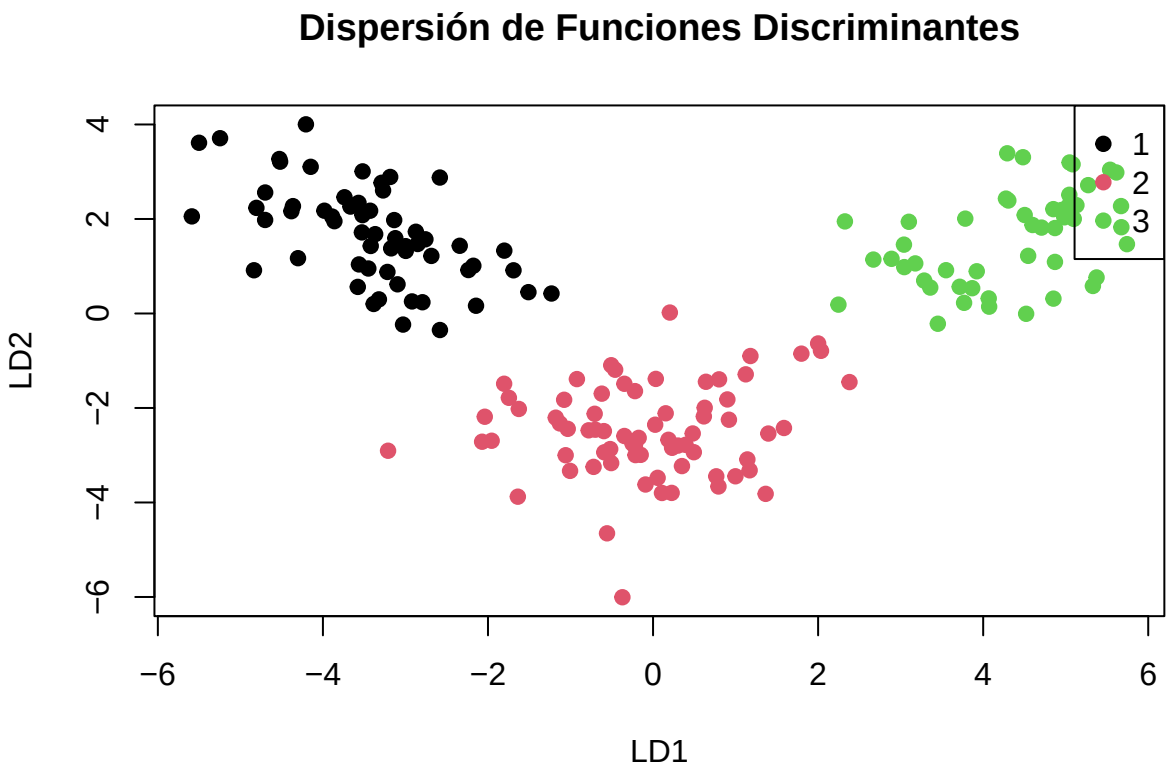
C) Represente con histogramas la distribución de los valores asociados por cada función discriminante en cada categoría.

```
# Valores discriminantes (LD1 y LD2)
lda_values <- lda_predictions$x

# Crear histogramas para cada función discriminante y grupo
par(mfrow = c(2, 1))  # Dos gráficos uno encima del otro

# Histograma de la primera función discriminante
hist(lda_values[, 1], breaks = 20, col = "lightblue", main = "LD1", xlab = "LD1")

# Histograma de la segunda función discriminante
hist(lda_values[, 2], breaks = 20, col = "lightgreen", main = "LD2", xlab = "LD2")
```



D) Represente visualmente sus resultados mediante un gráfico de dispersión con las funciones discriminantes.

```
# Gráfico de dispersión con LD1 y LD2
plot(lda_values[, 1], lda_values[, 2], col = wine$Type, pch = 19,
     xlab = "LD1", ylab = "LD2", main = "Dispersión de Funciones Discriminantes")
legend("topright", legend = levels(wine$Type), col = 1:3, pch = 19)
```



E) Determine la precisión del modelo.

```
classification_percentage
```

```
## [1] 100
```

Problema 5)

```
data("PimaIndiansDiabetes2")
```

A) Prepare la base de datos omitiendo los datos faltantes.

```
df <- na.omit(PimaIndiansDiabetes2)
df$diabetes <- ifelse(df$diabetes=="pos",1,0)
```

B) Divida el conjunto de datos en un conjunto de entrenamiento (80%) y un conjunto de prueba(20%)

```
target = "diabetes"
predictor = "glucose"

df$diabetes = as.factor(df$diabetes)
train_index = sample(nrow(df), 0.8 * nrow(df))
```

```r
train_dataset_2 = df[train_index,]
test_dataset_2 = df[-train_index,]
```

C) Considerando Diabetes como variable dependiente, formule un modelo de regresión logística con el cual predecir la probabilidad de que un paciente sea positivo para diabetes basado en la concentración de glucosa.

```r
#Ajuste del modelo
model = glm(diabetes ~ glucose, data = train_dataset_2, family=binomial)

#para la notación científica en el resumen
options(scipen=999)

#resumen del modelo
summary(model)
```

```
##
## Call:
## glm(formula = diabetes ~ glucose, family = binomial, data = train_dataset_2)
##
## Coefficients:
##              Estimate Std. Error z value             Pr(>|z|)
## (Intercept) -6.016674   0.695737  -8.648 < 0.0000000000000002 ***
## glucose      0.041770   0.005228   7.990  0.00000000000000135 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 399.38  on 312  degrees of freedom
## Residual deviance: 310.41  on 311  degrees of freedom
## AIC: 314.41
##
## Number of Fisher Scoring iterations: 4
```

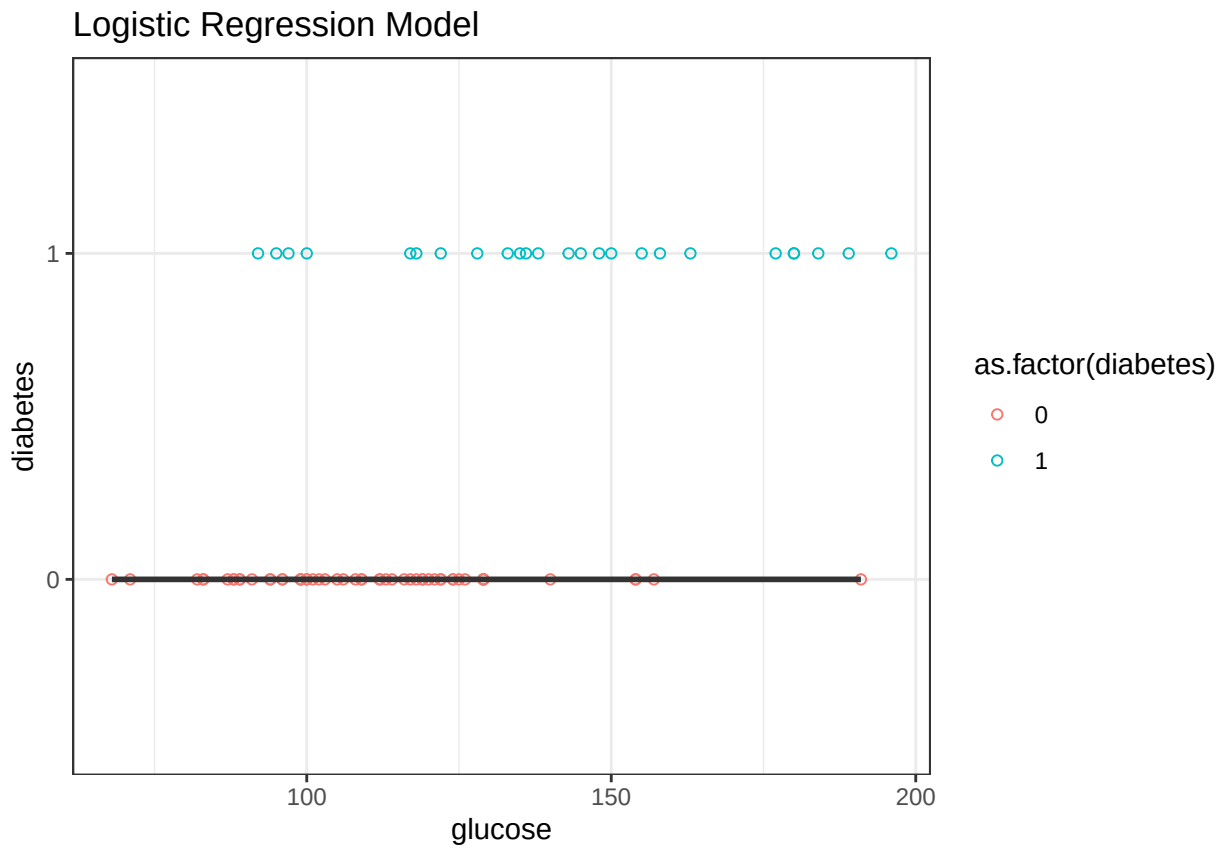
D) Grafique la curva de regresión logística

```r
test_dataset_2 %>%
  ggplot(aes(glucose, diabetes)) +
  geom_point(aes(color = as.factor(diabetes)), shape = 1) +
  geom_smooth(method = "glm", method.args = list(family = "binomial"),color = "gray20",
              se = FALSE) +
  theme_bw() +
  labs(
    title = "Logistic Regression Model",
    x = "glucose",
    y = "diabetes"
    )
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Failed to fit group 2.
## Caused by error:
```

```
## ! y values must be 0 <= y <= 1
```



E) Ajuste un modelo de regresión logística múltiple. Justifique la selección de las variables predictoras.

```
train_df <- train_dataset_2[c("glucose","pressure","mass","diabetes")]
test_df <- test_dataset_2[c("glucose","pressure","mass","diabetes")]
model_multiple <- glm(diabetes ~ glucose + pressure + mass, data =train_df, family = binomial)
summary(model_multiple)
```

```
##
## Call:
## glm(formula = diabetes ~ glucose + pressure + mass, family = binomial,
##     data = train_df)
##
## Coefficients:
##              Estimate Std. Error z value           Pr(>|z|)
## (Intercept) -8.348457   1.173018  -7.117 0.0000000000011024 ***
## glucose      0.040085   0.005323   7.531 0.0000000000000504 ***
## pressure     0.005999   0.011897   0.504            0.61406
## mass         0.062633   0.022269   2.813            0.00491 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 399.38  on 312  degrees of freedom
## Residual deviance: 300.13  on 309  degrees of freedom
## AIC: 308.13
##
## Number of Fisher Scoring iterations: 4
```

F) Evalúe el rendimiento del modelo sobre los individuos del conjunto de prueba.

```
prob_test = predict(model_multiple,test_df,type="response")
predicted.classes = ifelse(prob_test > 0.5, 1, 0)
```

Obtenga la matriz de confusión, identifique:

```
tabla_contingencia = table(Real = test_df$diabetes, Predicciones = predicted.classes)
print(tabla_contingencia)
```

```
##     Predicciones
## Real  0  1
##    0 51  3
##    1 12 13
```

Problema 6

```
data("swiss")
df <- swiss
```

1) Análisis Preliminar

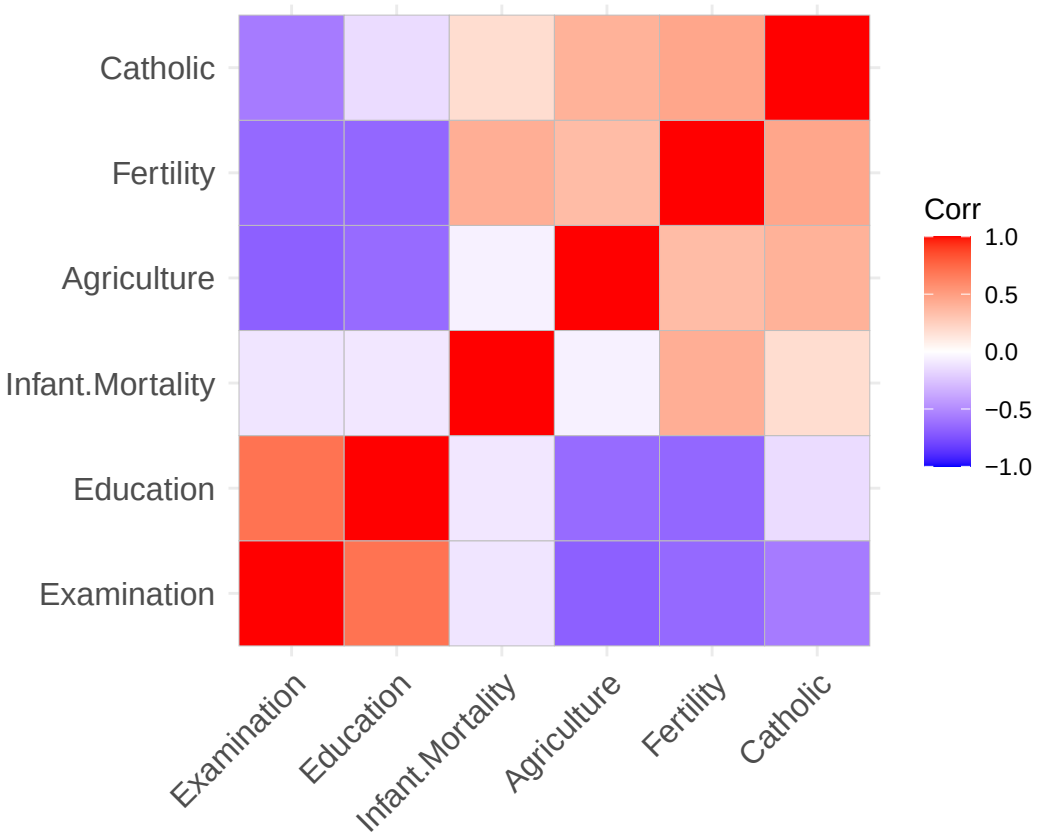
Explore la base de datos swiss para familiarizarse con las variables.

```
glimpse(df)
```

```
## Rows: 47
## Columns: 6
## $ Fertility        <dbl> 80.2, 83.1, 92.5, 85.8, 76.9, 76.1, 83.8, 92.4, 82.4,~
## $ Agriculture      <dbl> 17.0, 45.1, 39.7, 36.5, 43.5, 35.3, 70.2, 67.8, 53.3,~
## $ Examination      <int> 15, 6, 5, 12, 17, 9, 16, 14, 12, 16, 14, 21, 14, 19, ~
## $ Education        <int> 12, 9, 5, 7, 15, 7, 7, 8, 7, 13, 6, 12, 7, 12, 5, 2, ~
## $ Catholic         <dbl> 9.96, 84.84, 93.40, 33.77, 5.16, 90.57, 92.85, 97.16,~
## $ Infant.Mortality <dbl> 22.2, 22.2, 20.2, 20.3, 20.6, 26.6, 23.6, 24.9, 21.0,~
```

Realice un análisis de correlación entre Fertility, Catholic, Agriculture, y Examination.

```
mat_cor <- hetcor(df)$correlations
ggcorrplot(mat_cor,hc.order = T)
```



Observe los patrones de correlación entre estas variables y discútalos.

2) Propuesta del modelo

Proponga un modelo de regresión multivariada donde las variables dependientes sean Fertility y Catholic, y las variables independientes sean Agriculture y Examination.

$$Fertility = \beta_0 + \beta_1 Agriculture + \beta_2 Examination$$
$$Catholic = \alpha_0 + \alpha_1 Agriculture + \alpha_2 Examination$$

Especifique el modelo matemáticamente y ajuste el modelo de regresión multivariada.

```
modelo_fertility <- lm(Fertility ~ Agriculture + Examination, data = swiss)
modelo_catholic <- lm(Catholic ~ Agriculture + Examination, data = swiss)
```

3) Validación de supuestos del modelo

Realice la validación de los siguientes supuestos del modelo: Normalidad multivariante de los residuos.

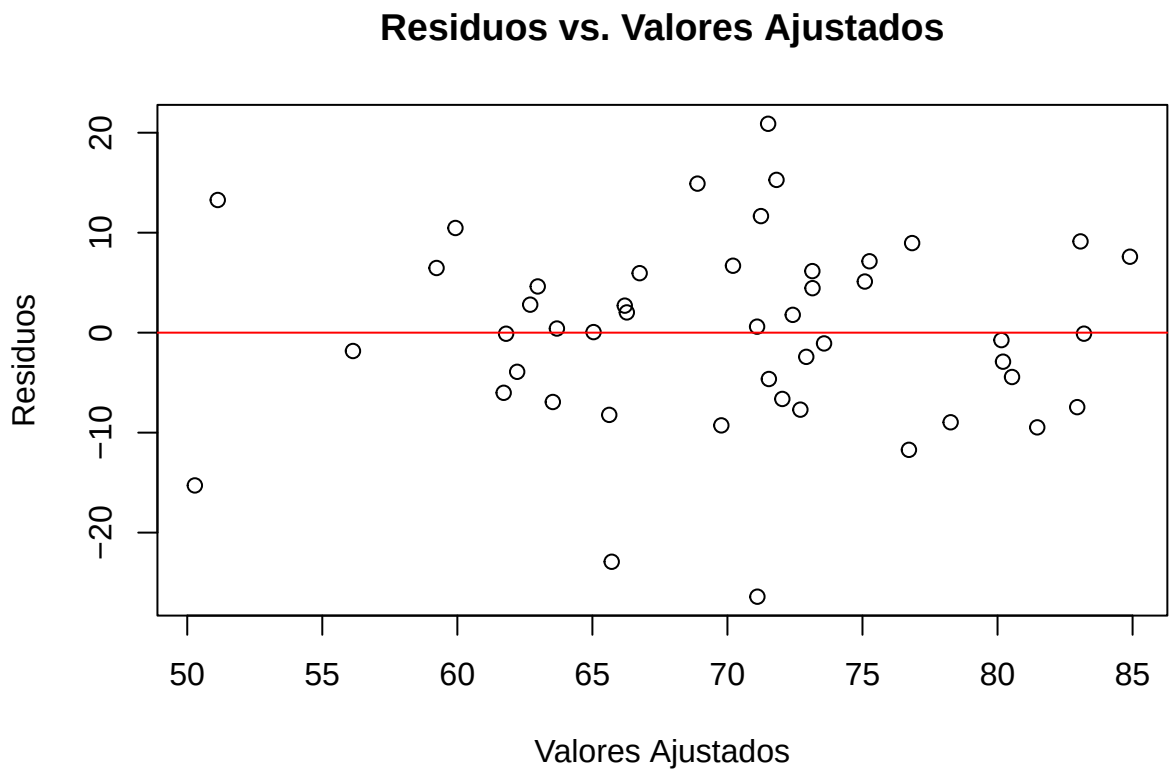
```
residuals_fer <- residuals(modelo_fertility)
residuals_cat <- residuals(modelo_catholic)
residuals_data <- data.frame(Fertility = residuals_fer, Catholic = residuals_cat)
mvn_result <- mvn(data = residuals_data, mvnTest = "mardia")
mvn_result
```

```
## $multivariateNormality
##              Test           Statistic             p value Result
```

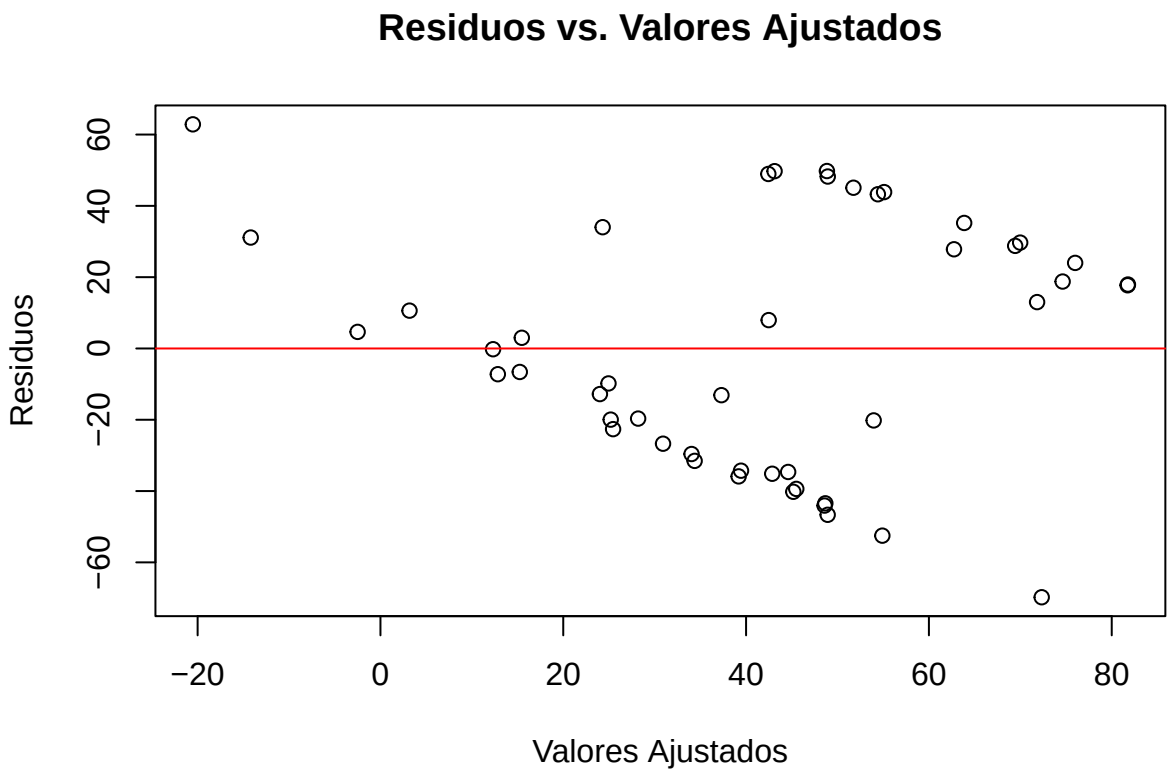
```
## 1 Mardia Skewness     13.823681461626 0.00787958322472484     NO
## 2 Mardia Kurtosis 0.00634856525902114   0.994934611822738    YES
## 3             MVN                <NA>                <NA>     NO
##
## $univariateNormality
##              Test  Variable Statistic   p value Normality
## 1 Anderson-Darling Fertility    0.2358    0.7773    YES
## 2 Anderson-Darling Catholic     0.6561    0.0815    YES
##
## $Descriptives
##            n                       Mean   Std.Dev      Median       Min      Max
## Fertility 47 -0.000000000000000002168035  9.409454  0.05767609 -26.40894 20.89375
## Catholic  47 -0.000000000000000007162563 34.183969 -0.21158602 -69.77964 62.86253
##                25th      75th        Skew   Kurtosis
## Fertility  -6.323385  6.313427 -0.40376303  0.4168935
## Catholic  -30.569602 29.238703  0.01739952 -1.2174036
```

Homoscedasticidad (varianza constante) entre las variables dependientes.

```
#Multinormalidad Test gráfico Q-Q Plot
plot(residuals_fer ~ fitted(modelo_fertility), main = "Residuos vs. Valores Ajustados", xlab = "Valores
abline(h = 0, col = "red")
```




```
plot(residuals_cat ~ fitted(modelo_catholic), main = "Residuos vs. Valores Ajustados", xlab = "Valores
abline(h = 0, col = "red")
```



No colinealidad excesiva entre las variables independientes.

```
#Se calcula el VIF
vif(modelo_catholic)
```

```
## Agriculture Examination
##    1.891576    1.891576
```

```
vif(modelo_fertility)
```

```
## Agriculture Examination
##    1.891576    1.891576
```