Actividad 2.4 Detección de datos influyentes

Franco Mendoza Muraira A01383399

2023-11-10

```
## Warning: package 'car' was built under R version 4.2.3

## Loading required package: carData

## Warning: package 'carData' was built under R version 4.2.2

## Warning: package 'lmtest' was built under R version 4.2.3

## Loading required package: zoo

## Warning: package 'zoo' was built under R version 4.2.3

##
## Attaching package: 'zoo'

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

```
df= read.csv(file = "datosRes.csv")
head(df,3)
```

```
##   Resistencia Longitud Altura.matriz Altura.poste Altura.amarre
## 1        9.95        2            50            1             1
## 2       24.45        8           110            1             1
## 3       31.75       11           120            2             1
```

```
model = lm(df$Resistencia~.,df)
```

1. Selección de variables por pasos

AIC:

```
step(model,direction = "both",trace=1)
```

```
## Start:  AIC=32.54
## df$Resistencia ~ Longitud + Altura.matriz + Altura.poste + Altura.amarre
##
##                 Df Sum of Sq     RSS     AIC
## - Altura.poste   1      1.81   63.41  31.269
## <none>                         61.60  32.543
## - Altura.amarre  1     28.92   90.52  40.167
## - Altura.matriz  1     40.50  102.10  43.176
## - Longitud       1   1568.75 1630.34 112.442
##
## Step:  AIC=31.27
## df$Resistencia ~ Longitud + Altura.matriz + Altura.amarre
##
##                 Df Sum of Sq     RSS     AIC
## <none>                         63.41  31.269
## + Altura.poste   1      1.81   61.60  32.543
## - Altura.matriz  1     40.21  103.62  41.546
## - Altura.amarre  1     51.76  115.17  44.189
## - Longitud       1   2552.49 2615.90 122.262

##
## Call:
## lm(formula = df$Resistencia ~ Longitud + Altura.matriz + Altura.amarre,
##     data = df)
##
## Coefficients:
##   (Intercept)       Longitud  Altura.matriz  Altura.amarre
##      1.367068       2.534919       0.008522       2.599278
```

BIC:

```
step(model,direction="both",trace=1,k=log(nrow(df)))
```

```
## Start:  AIC=38.64
## df$Resistencia ~ Longitud + Altura.matriz + Altura.poste + Altura.amarre
##
##                 Df Sum of Sq     RSS     AIC
## - Altura.poste   1      1.81   63.41  36.144
## <none>                         61.60  38.638
## - Altura.amarre  1     28.92   90.52  45.043
## - Altura.matriz  1     40.50  102.10  48.052
## - Longitud       1   1568.75 1630.34 117.317
##
## Step:  AIC=36.14
## df$Resistencia ~ Longitud + Altura.matriz + Altura.amarre
##
##                 Df Sum of Sq     RSS     AIC
## <none>                         63.41  36.144
## + Altura.poste   1      1.81   61.60  38.638
## - Altura.matriz  1     40.21  103.62  45.203
## - Altura.amarre  1     51.76  115.17  47.846
## - Longitud       1   2552.49 2615.90 125.919
```

```
##
## Call:
## lm(formula = df$Resistencia ~ Longitud + Altura.matriz + Altura.amarre,
##     data = df)
##
## Coefficients:
##   (Intercept)       Longitud  Altura.matriz  Altura.amarre
##      1.367068       2.534919       0.008522       2.599278
```

Según la evaluación del modelo AIC y BIC, nos quedaremos con las variables con estas evaluaciones más altas, y con el modelo que nos dan al final de la evaluación, "Longitud", "Altura.matriz" y "Altura.amarre".

2. Datos atípicos:

Identifica datos atípicos mediante el criterio de desviación estándar.

```
model = lm(formula = df$Resistencia ~ Longitud + Altura.matriz + Altura.amarre,
    data = df)
```

```
resuduals = rstandard(model)
resuduals
```

```
##           1           2           3           4           5           6
##  0.30995789 -0.44841839 -0.70704561 -0.97259115 -1.03667630  0.62805855
##           7           8           9          10          11          12
##  1.31710547  0.07663099 -2.03067946  0.41597748 -1.96463341 -0.50828735
##          13          14          15          16          17          18
## -1.21298733 -0.27131869  2.06875195  0.22750728  1.07912706 -0.82003000
##          19          20          21          22          23          24
##  0.66091424 -0.08090377  0.28009197  0.03204011  1.27689138  1.28159757
##          25
##  0.66393145
```

Identifica datos atípicos mediante el criterio de estandarización extrema.

```
rstud = rstudent(model)
rstud
```

```
##           1           2           3           4           5           6
##  0.30318224 -0.43972181 -0.69836840 -0.97127920 -1.03861768  0.61876125
##           7           8           9          10          11          12
##  1.34198733  0.07479465 -2.21063596  0.40763536 -2.12221013 -0.49911740
##          13          14          15          16          17          18
## -1.22753822 -0.26524523  2.26256901  0.22229849  1.08359296 -0.81339633
##          19          20          21          22          23          24
##  0.65180073 -0.07896631  0.27385378  0.03126871  1.29750549  1.30269262
##          25
##  0.65483998
```

Usamos los métodos de rstandard y de rstudent para encontrar datos atípicos dentro del dataset.

3. Datos Inflyentes

Por grado de Levarage

```
hatt = hatvalues(model)
hatt
```

```
##          1          2          3          4          5          6          7
## 0.17997067 0.11494030 0.16449688 0.10188969 0.06751594 0.07494729 0.11898344
##          8          9         10         11         12         13         14
## 0.17842872 0.13452586 0.06794985 0.15609899 0.05560348 0.19198635 0.11340056
##         15         16         17         18         19         20         21
## 0.20870226 0.09845327 0.42289094 0.30685227 0.22222579 0.14775554 0.26018831
##         22         23         24         25
## 0.17974960 0.23312954 0.10912925 0.09018521
```

Por distancia de Cook

```
cooks = cooks.distance(model)
cooks
```

```
##            1            2            3            4            5            6
## 5.271300e-03 6.528398e-03 2.460620e-02 2.682880e-02 1.945321e-02 7.989686e-03
##            7            8            9           10           11           12
## 5.857113e-02 3.188368e-04 1.602413e-01 3.153762e-03 1.784891e-01 3.802824e-03
##           13           14           15           16           17           18
## 8.739854e-02 2.353896e-03 2.821916e-01 1.413099e-03 2.133318e-01 7.442229e-02
##           19           20           21           22           23           24
## 3.120119e-02 2.836986e-04 6.897757e-03 5.624053e-05 1.239148e-01 5.030021e-02
##           25
## 1.092368e-02
```

Para identificar los datos influyentes en el dataset, usamos los métodos por distancia de Cook, y por grado de Leverage. Estos datos que sobresalen son los que cambian más al modelo.

4. Resumen de resultados

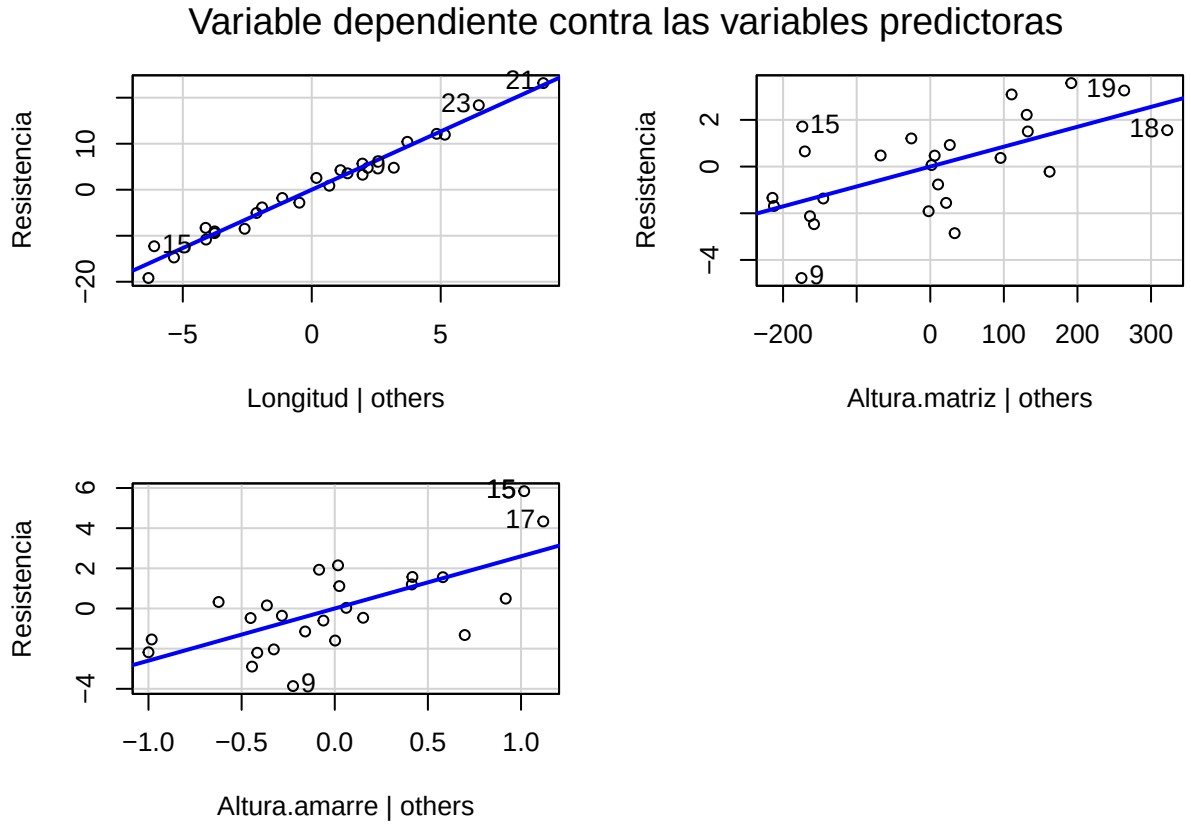
```
tabla = data.frame(residuals(model),resuduals,rstud,hatt,cooks)
head(tabla,5)
```

```
##   residuals.model.  resuduals      rstud       hatt       cooks
## 1        0.4877406  0.3099579  0.3031822 0.17997067 0.005271300
## 2       -0.7330629 -0.4484184 -0.4397218 0.11494030 0.006528398
## 3       -1.1230343 -0.7070456 -0.6983684 0.16449688 0.024606195
## 4       -1.6016467 -0.9725912 -0.9712792 0.10188969 0.026828797
## 5       -1.7395441 -1.0366763 -1.0386177 0.06751594 0.019453209
```

5. Gráficos complementarios

Variable dependiente contra las variables predictoras

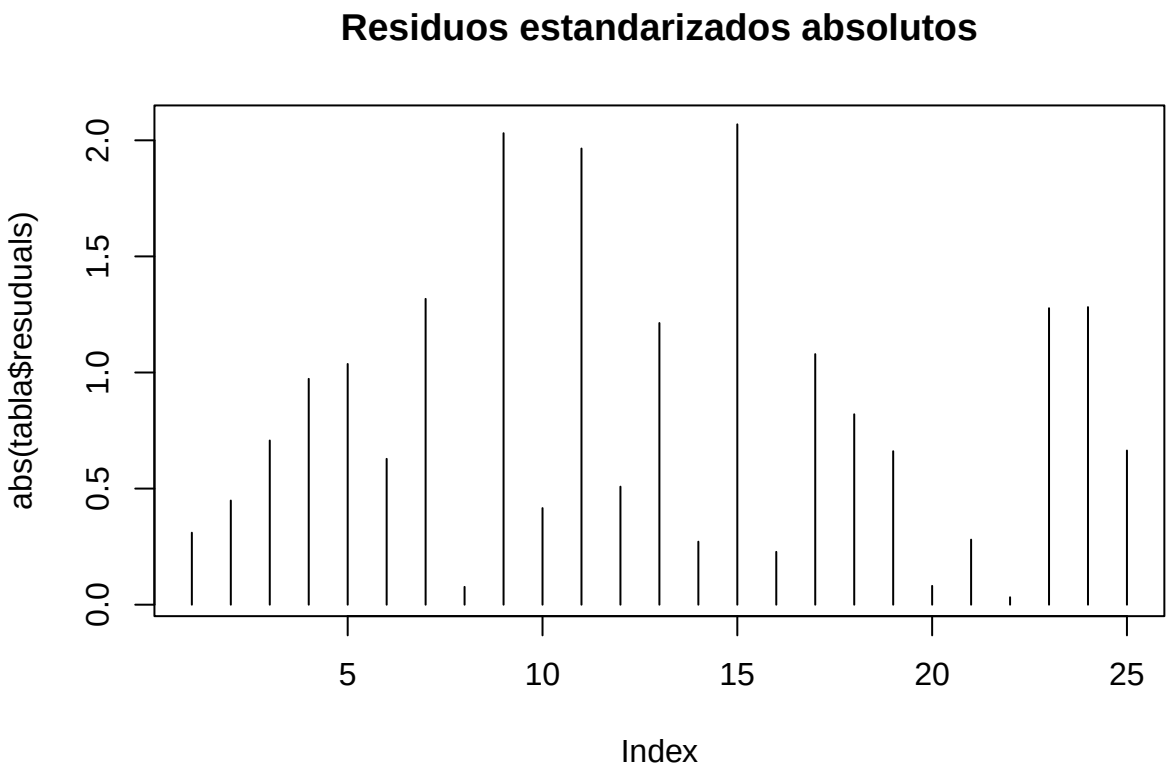
```
avPlots(model,main = paste("Variable dependiente contra las variables predictoras"),ylab = "Resistencia
```



Aquí podemos ver el modelo de Resistencia en función de cada variable independiente, y se puede ver qué observaciones son influyentes en el modelo de cada variable.

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

```
plot(abs(tabla$resuduals),type = "h",main = "Residuos estandarizados absolutos")
```



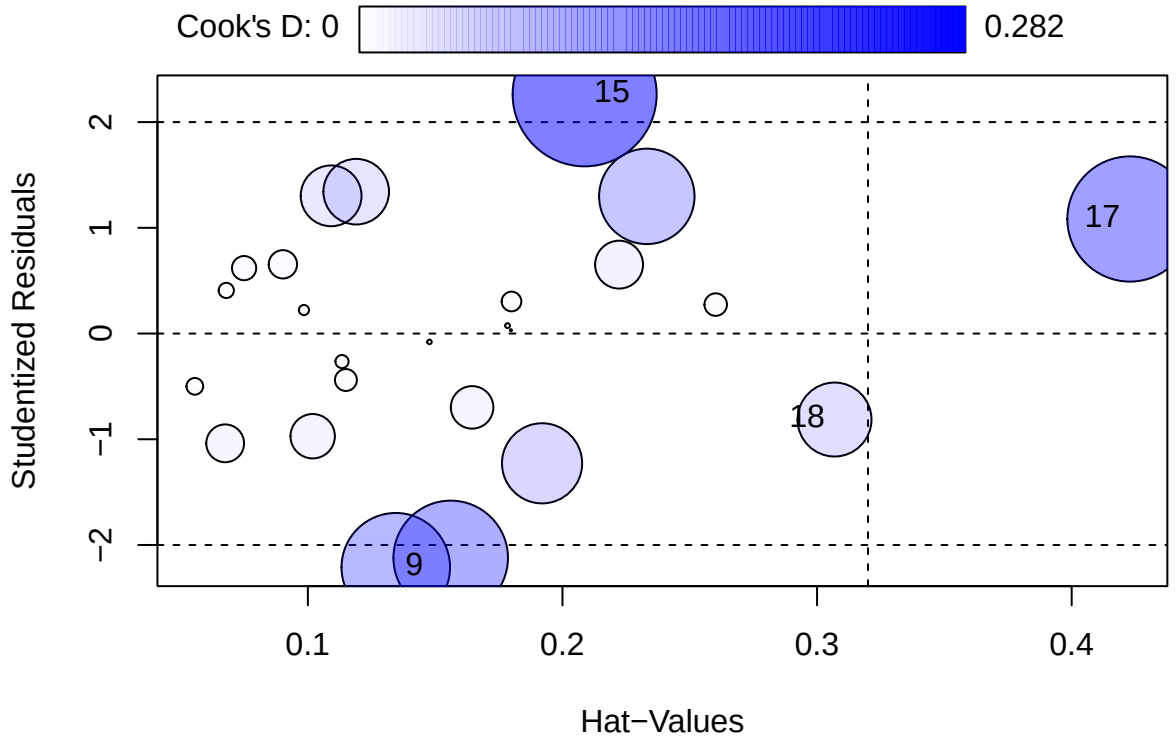

```
mayor3 = which(abs(tabla$resuduals)>3)
mayor3
```

```
## integer(0)
```

Encontramos ningún valor con valor absoluto mayor a 3 en los residuos estandarizados, podemos ver que los más altos llegan solo aproximadamente a 2.

Gráfico de influencia:

```
influencePlot(model,id=TRUE)
```




```
##       StudRes       Hat      CookD
## 9  -2.2106360 0.1345259 0.16024130
## 15  2.2625690 0.2087023 0.28219160
## 17  1.0835930 0.4228909 0.21333184
## 18 -0.8133963 0.3068523 0.07442229
```

Aquí encontramos los datos influyentes del dataset para el modelo, en este caso son las observaciones 9, 15, 17, y 18.

6. Ajustes al modelo

```
df2 = df[-c(9,15,17,18),]
model2 = lm(formula = df2$Resistencia ~ Longitud + Altura.matriz + Altura.amarre,
    data = df2)
summary(model2)
```

```
##
## Call:
## lm(formula = df2$Resistencia ~ Longitud + Altura.matriz + Altura.amarre,
##     data = df2)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -2.25032 -0.63733  0.06576  0.71273  2.00412
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.224033   0.755335   2.944  0.00907 **
## Longitud      2.598763   0.075790  34.289  < 2e-16 ***
## Altura.matriz 0.009871   0.002182   4.524  0.00030 ***
## Altura.amarre 1.322158   0.610545   2.166  0.04486 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.344 on 17 degrees of freedom
## Multiple R-squared:  0.9925, Adjusted R-squared:  0.9911
## F-statistic:   747 on 3 and 17 DF,  p-value: < 2.2e-16
```

```
summary(model)
```

```
##
## Call:
## lm(formula = df$Resistencia ~ Longitud + Altura.matriz + Altura.amarre,
##     data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2828 -1.1230  0.1207  1.0497  3.1978
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.367068   0.833704   1.640 0.115952
## Longitud      2.534919   0.087187  29.074  < 2e-16 ***
## Altura.matriz 0.008522   0.002335   3.649 0.001498 **
## Altura.amarre 2.599278   0.627791   4.140 0.000465 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.738 on 21 degrees of freedom
## Multiple R-squared:  0.9896, Adjusted R-squared:  0.9881
## F-statistic:   667 on 3 and 21 DF,  p-value: < 2.2e-16
```

Variabilidad explicada por el modelo/P valor significancia del modelo

H_0 : Todos los coeficientes de regresión son cero.

H_1 : No todos los coeficientes de regresión son cero.

Hacemos un nuevo modelo quitando primero las observaciones previamente mencionadas que eran influyentes en el modelo previo. Al comparar las 2 r^2, podemos ver que el modelo 2 mejora los resultados, aunque solo por poco debido a los valores de r^2 que ya teníamos previamente. El nuevo valor ajustado es de 0.9911, cuando el previo era de 0.9881. También podemos ver que en el nuevo modelo la significancia de la variable Altura.matriz aumenta, mientras que la de Altura.amarre baja, esto puede ser debido a que los datos influyentes afectaban más a esta variable.

En general podemos cocnluir que la significancia del modelo es muy alta debido a que las variables ecplican el 99% de la variabilidad de la Resistencia. Y con el p value tan bajo <2.2e-16, se tiene suficiente evidencia

para rechasar la hipótesis nula de que todos los coeficientes de regresión son iguales a cero, y podemos llegar a que el modelo si explica la variabilidad en la variable dependiente.

Supuestos del modelo

Normalidad de los residuos

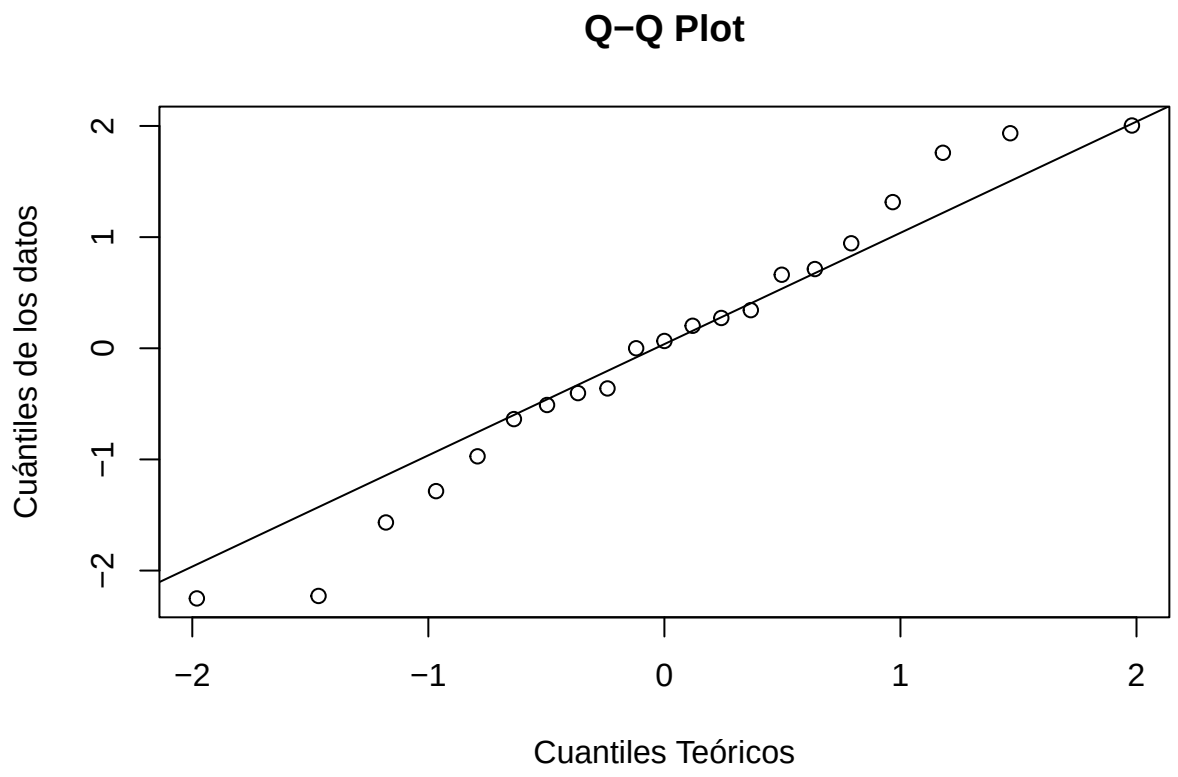
H_0 : Los residuos provienen de una distribución normal.

H_1 : Los residuos no provienen de una distribución normal.

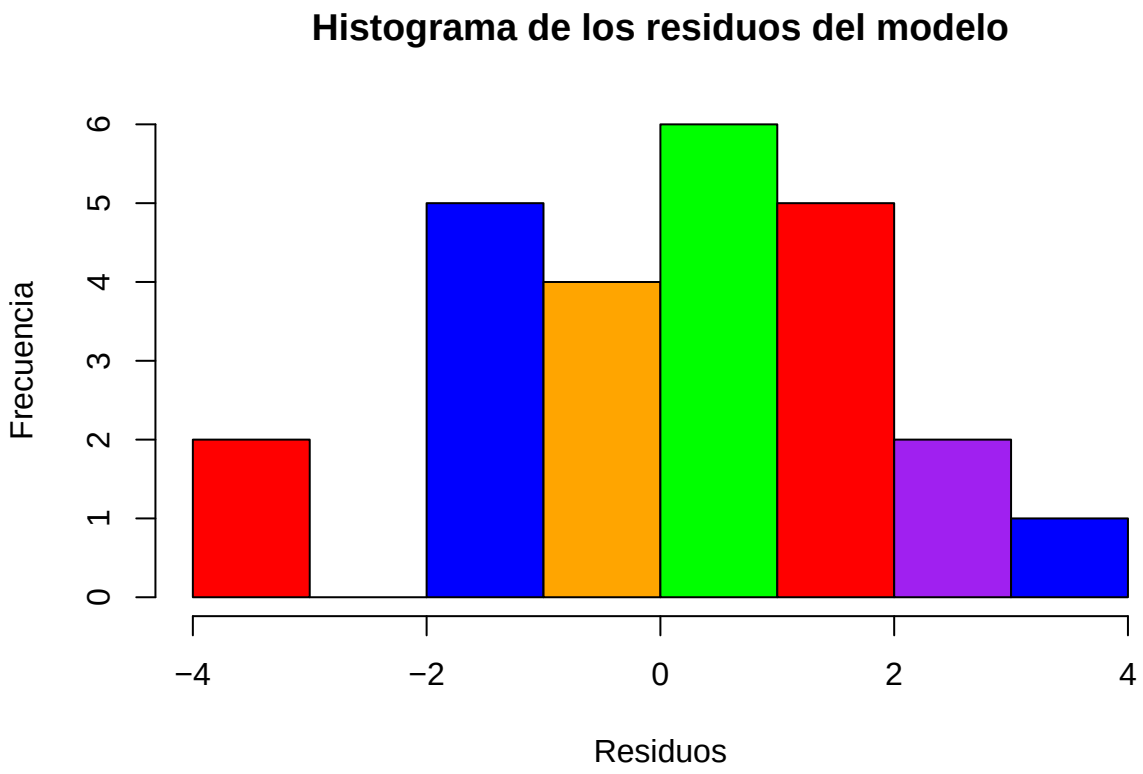
```
shapiro.test(residuals(model2))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(model2)
## W = 0.96967, p-value = 0.7256
```

```
qqnorm(model2$residuals,main = "Q-Q Plot",xlab = "Cuantiles Teóricos",ylab= "Cuántiles de los datos")
qqline(model2$residuals)
```




```
hist(model$residuals,main = "Histograma de los residuos del modelo",xlab = "Residuos",ylab = "Frecuencia
```



Viendo que el pvalue de la prueba de Shapiro es de 0.7256, y siendo mayor que el α de 0.05, no podemos rechazar la hipótesis nula, y se concluye que se presenta una distribución normal en los resiudos del modelo.

Verificación de media cero

H_0 : Media $= 0$.

H_1 : Media $\neq 0$.

```
t.test(residuals(model2))
```

```
##
##  One Sample t-test
##
## data:  residuals(model2)
## t = 3.4537e-17, df = 20, p-value = 1
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.5639406  0.5639406
## sample estimates:
##    mean of x
## 9.337046e-18
```

Debido a el pvalue de 1, no podemos rechazar la hipótesis nula, y concluimos que la media de los residuos si puede ser igual a 0.

Homocedasticidad

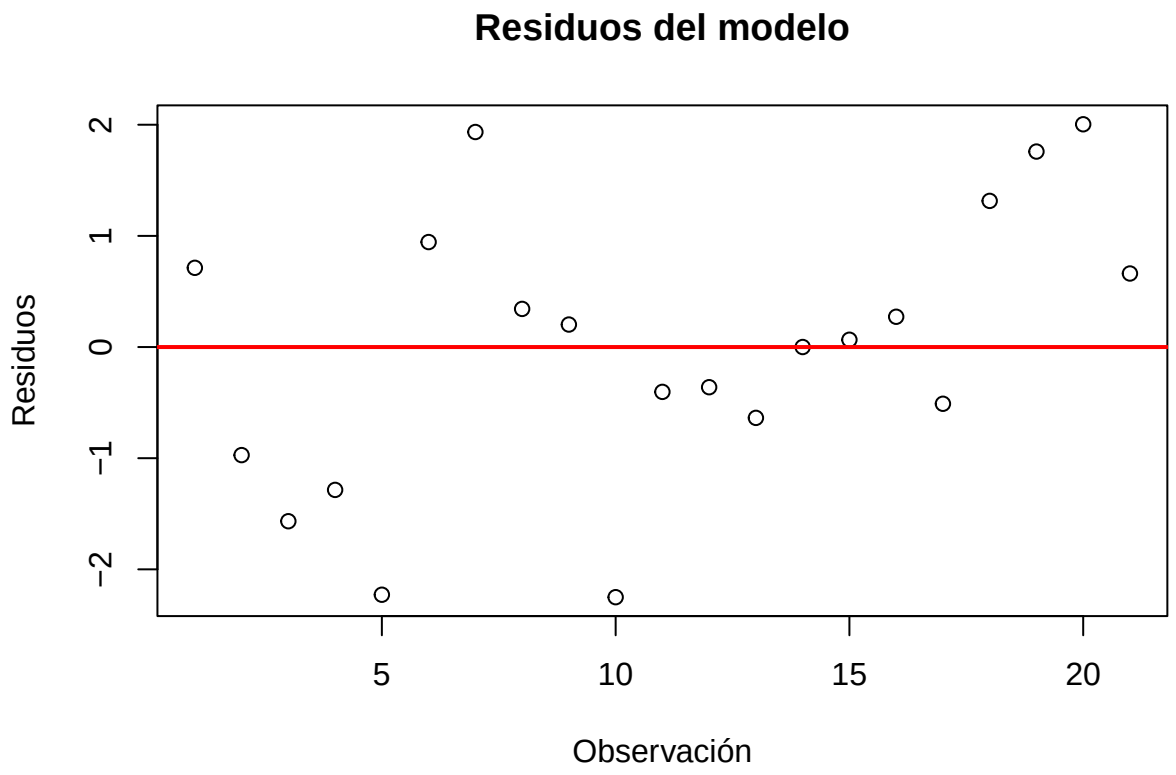
H_0 : La varianza de los errores es constante.

H_1 : La varianza de los errores no es constante.

```
bptest(model2)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  model2
## BP = 0.27837, df = 3, p-value = 0.964
```

```
plot(residuals(model2),main = "Residuos del modelo", xlab = "Observación",ylab= "Residuos")
abline(h=0,col="red",lwd=2)
```



Debido a que el pvalue es muy alto, con un valor de 0.964, no se puede rechazar la hipótesis nula, y se concluye que la varianza de los errores es constante, esto se puede ver en la gráfica superior.

Independencia

H_0 : Los residuos son independientes.

H_1 : Los residuos no son independientes.

```
dwtest(model2)
```

```
##
##  Durbin-Watson test
##
## data:  model2
## DW = 1.0986, p-value = 0.008179
## alternative hypothesis: true autocorrelation is greater than 0
```

Debido a que el p value es alto, con un valor de 0.008, y menor que 0.05, hay suficiente evidencia para rechazar la hipótesis nula, por lo que se concluye que los residuos no son independientes.

AIC:

```
step(model2,direction="both",  trace=1)
```

```
## Start:  AIC=15.97
## df2$Resistencia ~ Longitud + Altura.matriz + Altura.amarre
##
##                 Df Sum of Sq     RSS     AIC
## <none>                         30.70  15.973
## - Altura.amarre  1      8.47   39.17  19.089
## - Altura.matriz  1     36.95   67.65  30.566
## - Longitud       1   2123.04 2153.74 103.239
```

```
##
## Call:
## lm(formula = df2$Resistencia ~ Longitud + Altura.matriz + Altura.amarre,
##     data = df2)
##
## Coefficients:
##   (Intercept)       Longitud  Altura.matriz  Altura.amarre
##      2.224033       2.598763       0.009871       1.322158
```

BIC:

```
step(model2,direction="both",trace=1,k=log(nrow(df2)))
```

```
## Start:  AIC=20.15
## df2$Resistencia ~ Longitud + Altura.matriz + Altura.amarre
##
##                 Df Sum of Sq     RSS     AIC
## <none>                         30.70  20.151
## - Altura.amarre  1      8.47   39.17  22.222
## - Altura.matriz  1     36.95   67.65  33.699
## - Longitud       1   2123.04 2153.74 106.373
```

```
##
## Call:
## lm(formula = df2$Resistencia ~ Longitud + Altura.matriz + Altura.amarre,
##     data = df2)
##
## Coefficients:
##   (Intercept)       Longitud  Altura.matriz  Altura.amarre
##      2.224033       2.598763       0.009871       1.322158
```

Podemos ver que en las pruebas AIC y BIC se tienen las mismas variables, y no se retira ninguna del modelo.

7. Conclusiones

En esta actividad pudimos identificar datos atípicos e influyentes con diferentes métodos válidos en modelos de regresión de múltiples variables, y vimos como estos métodos eran diferentes y parecidos entre sí. Esto se hizo para observar cómo afectan a los modelos, al igual que ver cómo cambian estos modelos al retirar los datos influyentes, para ver como mejora el modelo sin estos datos.

También hicimos una selección de variables con el criterio de evaluación de modelo AIC y BIC, para encontrar las variables óptimas para el modelo que se buscaba crear.