A3-Regresión Múltiple: Detección datos atípicos

Oskar Arturo Gamboa Reyes

2024-09-24

Leer datos

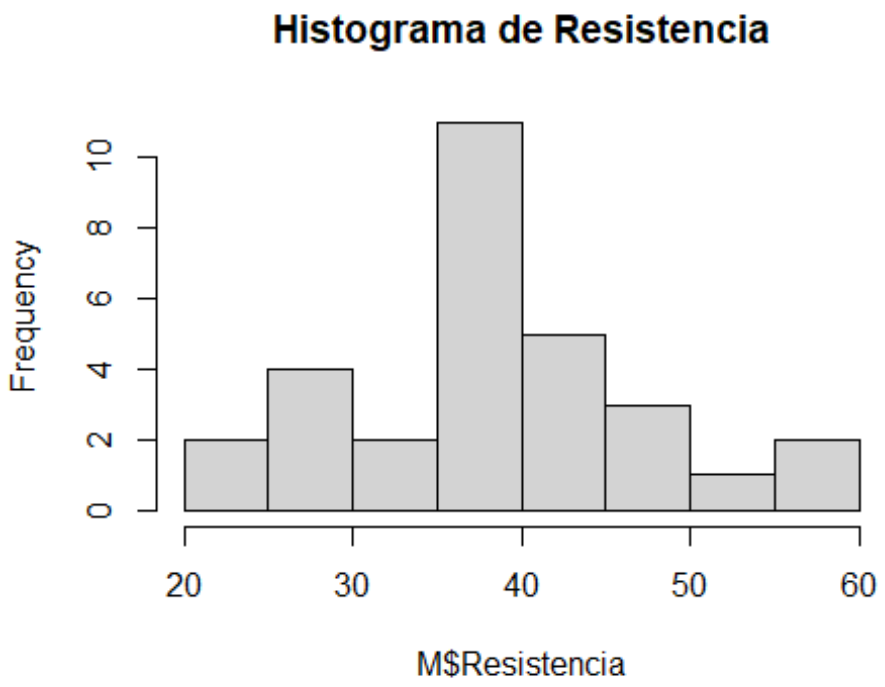
```
M = read.csv("AlCorte.csv")
summary(M)
```

```
##      Fuerza      Potencia    Temperatura      Tiempo    Resistencia
##  Min.   :25   Min.   : 45   Min.   :150   Min.   :10   Min.   :22.70
##  1st Qu.:30   1st Qu.: 60   1st Qu.:175   1st Qu.:15   1st Qu.:34.67
##  Median :35   Median : 75   Median :200   Median :20   Median :38.60
##  Mean   :35   Mean   : 75   Mean   :200   Mean   :20   Mean   :38.41
##  3rd Qu.:40   3rd Qu.: 90   3rd Qu.:225   3rd Qu.:25   3rd Qu.:42.70
##  Max.   :45   Max.   :105   Max.   :250   Max.   :30   Max.   :58.70
```

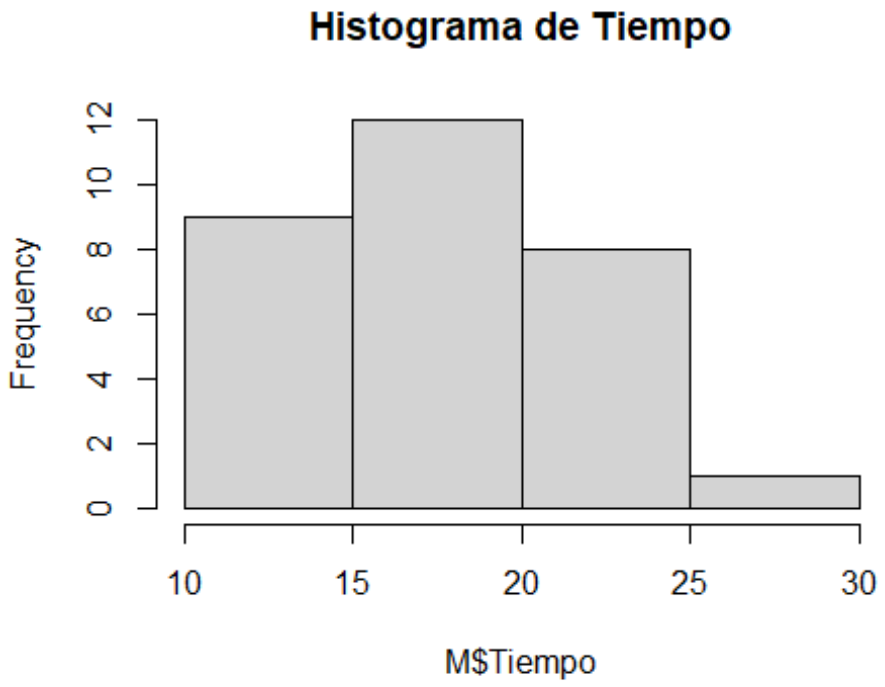
Análisis descriptivo

Histogramas

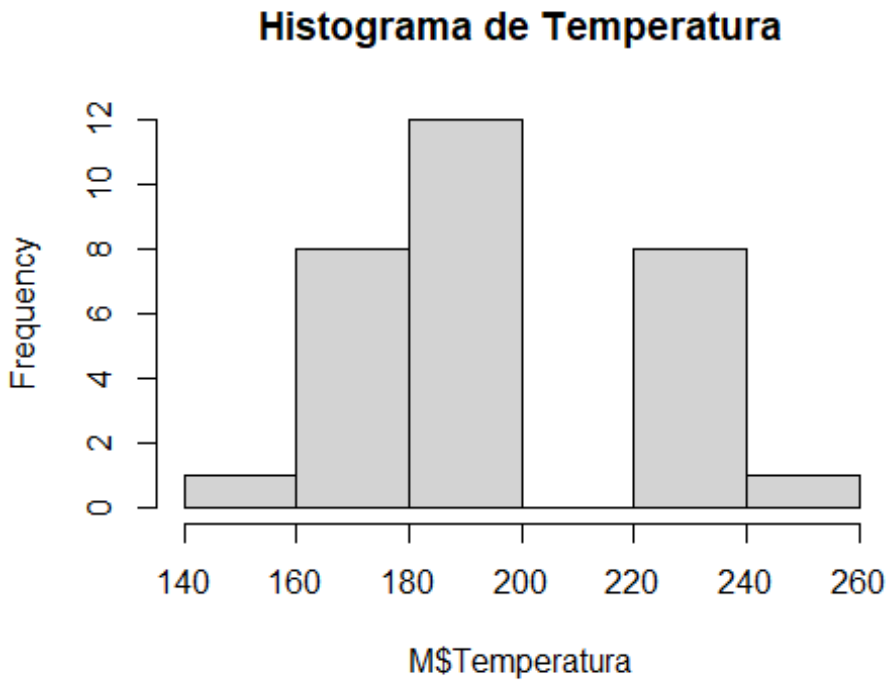
```
hist(M$Resistencia, main = "Histograma de Resistencia")
```



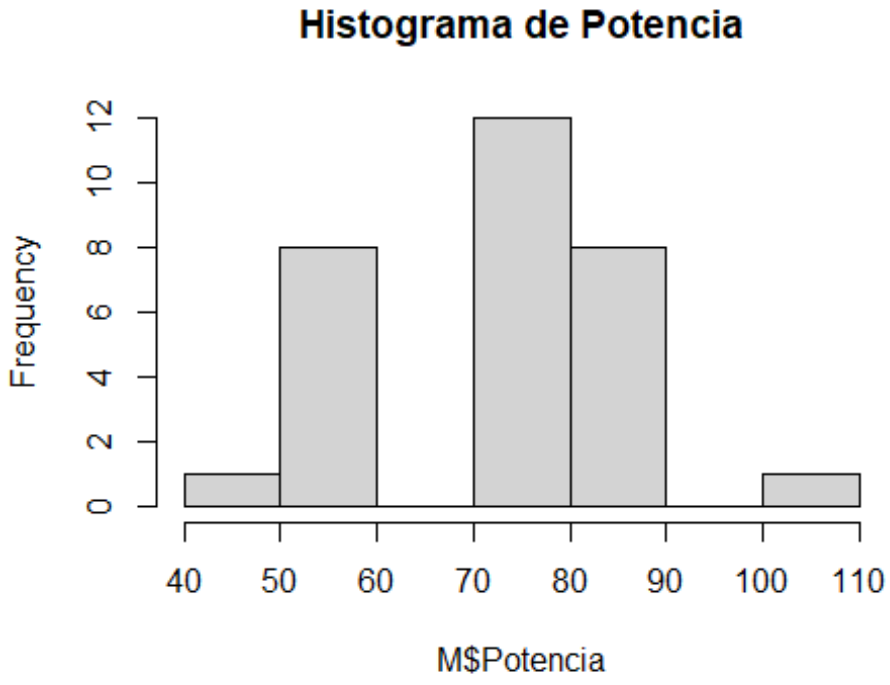

```r
hist(M$Tiempo, main = "Histograma de Tiempo")
```



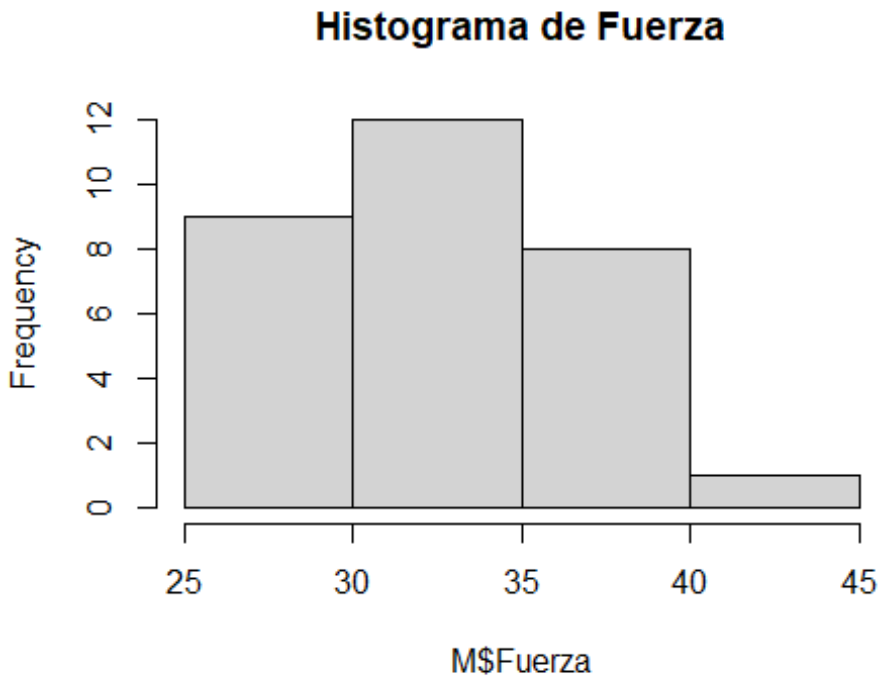

```r
hist(M$Temperatura, main = "Histograma de Temperatura")
```



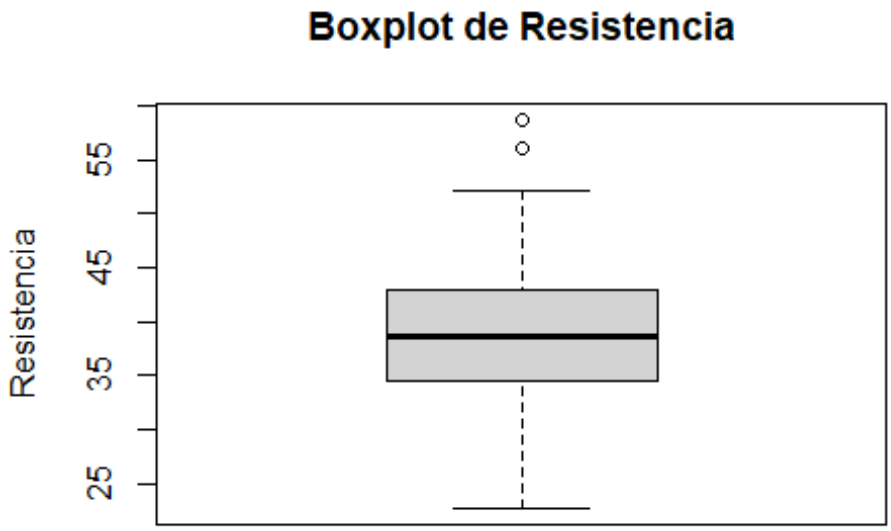

```r
hist(M$Potencia, main = "Histograma de Potencia")
```



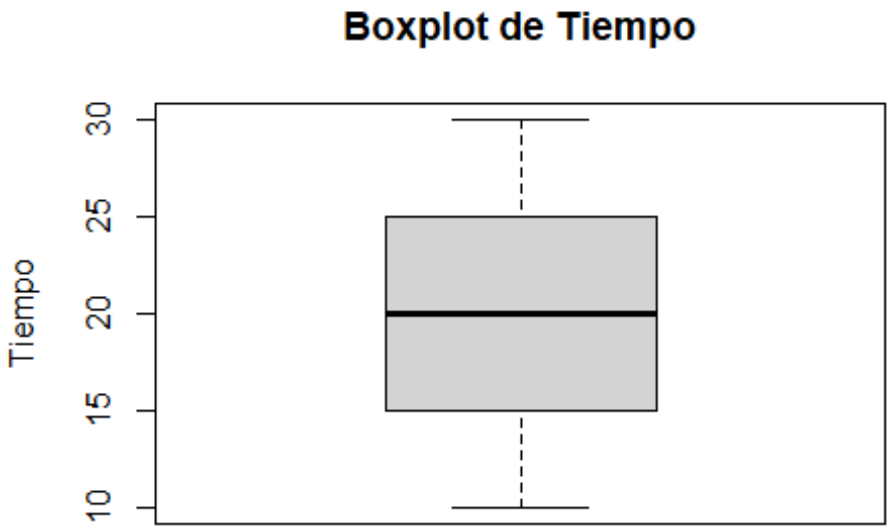

```r
hist(M$Fuerza, main = "Histograma de Fuerza")
```



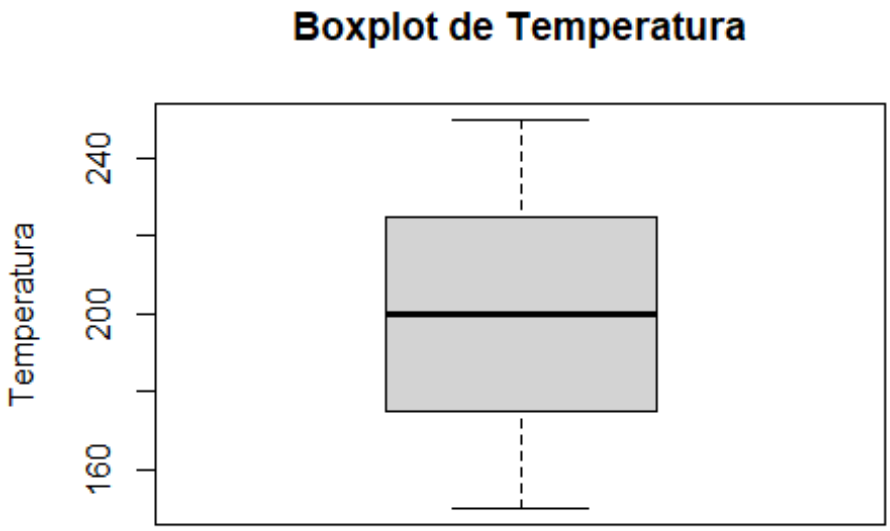

```r
boxplot(M$Resistencia, main = "Boxplot de Resistencia", ylab = "Resistencia")
```



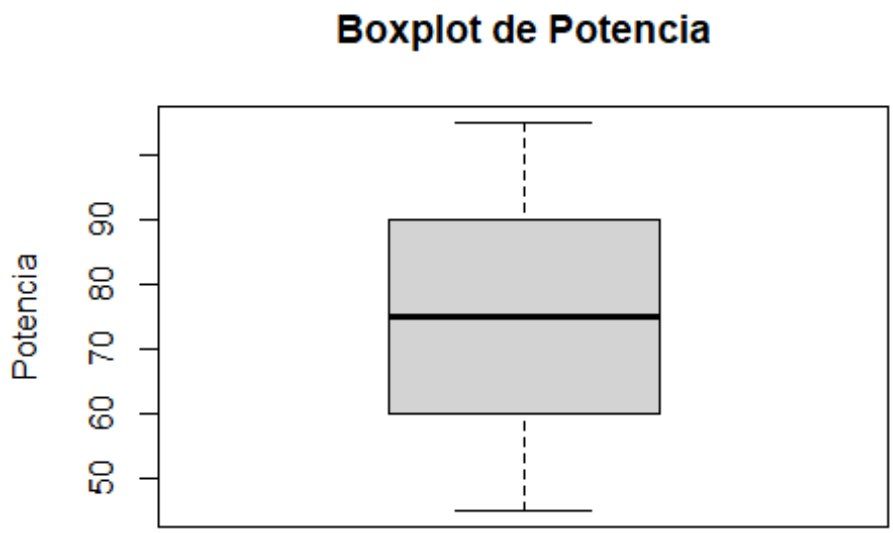

```r
boxplot(M$Tiempo, main = "Boxplot de Tiempo", ylab = "Tiempo")
```



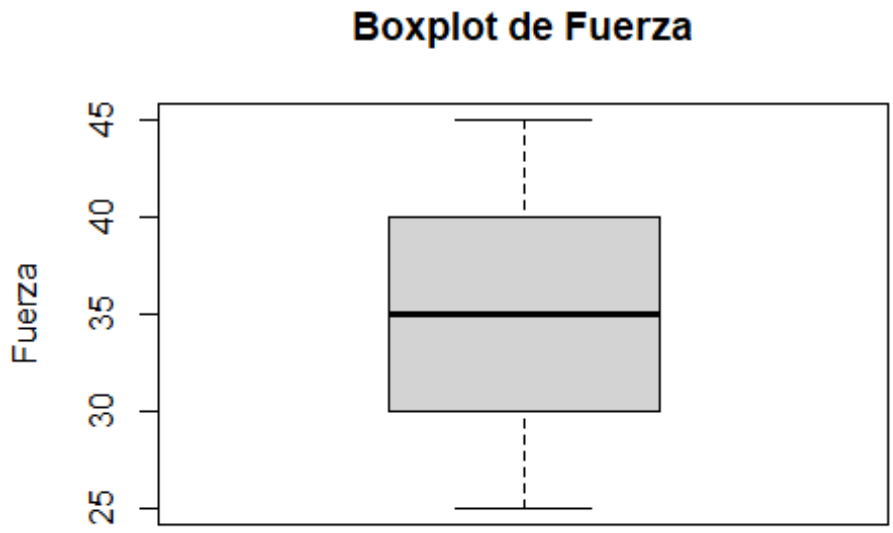

```r
boxplot(M$Temperatura, main = "Boxplot de Temperatura", ylab = "Temperatura")
```




```r
boxplot(M$Potencia, main = "Boxplot de Potencia", ylab = "Potencia")
```




```r
boxplot(M$Fuerza, main = "Boxplot de Fuerza", ylab = "Fuerza")
```



Encontrar modelo que explique la Resistencia

```
Modelo = lm(Resistencia~., data=M)

Pasos = step(Modelo, direction="both",trace=1)

## Start:  AIC=102.96
## Resistencia ~ Fuerza + Potencia + Temperatura + Tiempo
##
##               Df Sum of Sq     RSS    AIC
## - Fuerza       1     26.88  692.00 102.15
## - Tiempo       1     40.04  705.16 102.72
## <none>                      665.12 102.96
## - Temperatura  1    252.20  917.32 110.61
## - Potencia     1   1341.01 2006.13 134.08
##
## Step:  AIC=102.15
## Resistencia ~ Potencia + Temperatura + Tiempo
##
##               Df Sum of Sq     RSS    AIC
## - Tiempo       1     40.04  732.04 101.84
## <none>                      692.00 102.15
## + Fuerza       1     26.88  665.12 102.96
## - Temperatura  1    252.20  944.20 109.47
## - Potencia     1   1341.02 2033.02 132.48
##
## Step:  AIC=101.84
## Resistencia ~ Potencia + Temperatura
##
##               Df Sum of Sq     RSS    AIC
## <none>                      732.04 101.84
## + Tiempo       1     40.04  692.00 102.15
## + Fuerza       1     26.88  705.16 102.72
## - Temperatura  1    252.20  984.24 108.72
## - Potencia     1   1341.01 2073.06 131.07

summary(Pasos)

##
## Call:
## lm(formula = Resistencia ~ Potencia + Temperatura, data = M)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.3233  -2.8067  -0.8483   3.1892   9.4600
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -24.90167   10.07207  -2.472  0.02001 *
## Potencia      0.49833    0.07086   7.033 1.47e-07 ***
## Temperatura   0.12967    0.04251   3.050  0.00508 **
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.207 on 27 degrees of freedom
## Multiple R-squared:  0.6852, Adjusted R-squared:  0.6619
## F-statistic: 29.38 on 2 and 27 DF,  p-value: 1.674e-07
```

```
modelo_nulo = lm(Resistencia~1, data = M)
Pasos2 = step(modelo_nulo, scope = list(lower = modelo_nulo, upper=Modelo),
direction = "forward")
```

```
## Start:  AIC=132.51
## Resistencia ~ 1
##
##               Df Sum of Sq     RSS    AIC
## + Potencia     1   1341.01  984.24 108.72
## + Temperatura  1    252.20 2073.06 131.07
## <none>                     2325.26 132.51
## + Tiempo       1     40.04 2285.22 133.99
## + Fuerza       1     26.88 2298.38 134.16
##
## Step:  AIC=108.72
## Resistencia ~ Potencia
##
##               Df Sum of Sq    RSS    AIC
## + Temperatura  1   252.202 732.04 101.84
## <none>                     984.24 108.72
## + Tiempo       1    40.042 944.20 109.47
## + Fuerza       1    26.882 957.36 109.89
##
## Step:  AIC=101.84
## Resistencia ~ Potencia + Temperatura
##
##          Df Sum of Sq    RSS    AIC
## <none>                 732.04 101.84
## + Tiempo  1    40.042 692.00 102.15
## + Fuerza  1    26.882 705.16 102.72
```

```
summary(Pasos2)
```

```
##
## Call:
## lm(formula = Resistencia ~ Potencia + Temperatura, data = M)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.3233  -2.8067  -0.8483   3.1892   9.4600
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -24.90167   10.07207  -2.472  0.02001 *
```

```
## Potencia      0.49833    0.07086   7.033 1.47e-07 ***
## Temperatura   0.12967    0.04251   3.050  0.00508 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.207 on 27 degrees of freedom
## Multiple R-squared:  0.6852, Adjusted R-squared:  0.6619
## F-statistic: 29.38 on 2 and 27 DF,  p-value: 1.674e-07
```

```
n = length(M$Resistencia)
Pasos3 = step(Modelo, direction="both", k = log(n))
```

```
## Start:  AIC=109.97
## Resistencia ~ Fuerza + Potencia + Temperatura + Tiempo
##
##               Df Sum of Sq     RSS    AIC
## - Fuerza       1     26.88  692.00 107.76
## - Tiempo       1     40.04  705.16 108.32
## <none>                      665.12 109.97
## - Temperatura  1    252.20  917.32 116.21
## - Potencia     1   1341.01 2006.13 139.69
##
## Step:  AIC=107.76
## Resistencia ~ Potencia + Temperatura + Tiempo
##
##               Df Sum of Sq     RSS    AIC
## - Tiempo       1     40.04  732.04 106.04
## <none>                      692.00 107.76
## + Fuerza       1     26.88  665.12 109.97
## - Temperatura  1    252.20  944.20 113.68
## - Potencia     1   1341.02 2033.02 136.69
##
## Step:  AIC=106.04
## Resistencia ~ Potencia + Temperatura
##
##               Df Sum of Sq     RSS    AIC
## <none>                      732.04 106.04
## + Tiempo       1     40.04  692.00 107.76
## + Fuerza       1     26.88  705.16 108.32
## - Temperatura  1    252.20  984.24 111.52
## - Potencia     1   1341.01 2073.06 133.87
```

```
summary(Pasos3)
```

```
##
## Call:
## lm(formula = Resistencia ~ Potencia + Temperatura, data = M)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.3233  -2.8067  -0.8483   3.1892   9.4600
```

```
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -24.90167   10.07207  -2.472  0.02001 *
## Potencia      0.49833    0.07086   7.033 1.47e-07 ***
## Temperatura   0.12967    0.04251   3.050  0.00508 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.207 on 27 degrees of freedom
## Multiple R-squared:  0.6852, Adjusted R-squared:  0.6619
## F-statistic: 29.38 on 2 and 27 DF,  p-value: 1.674e-07
```

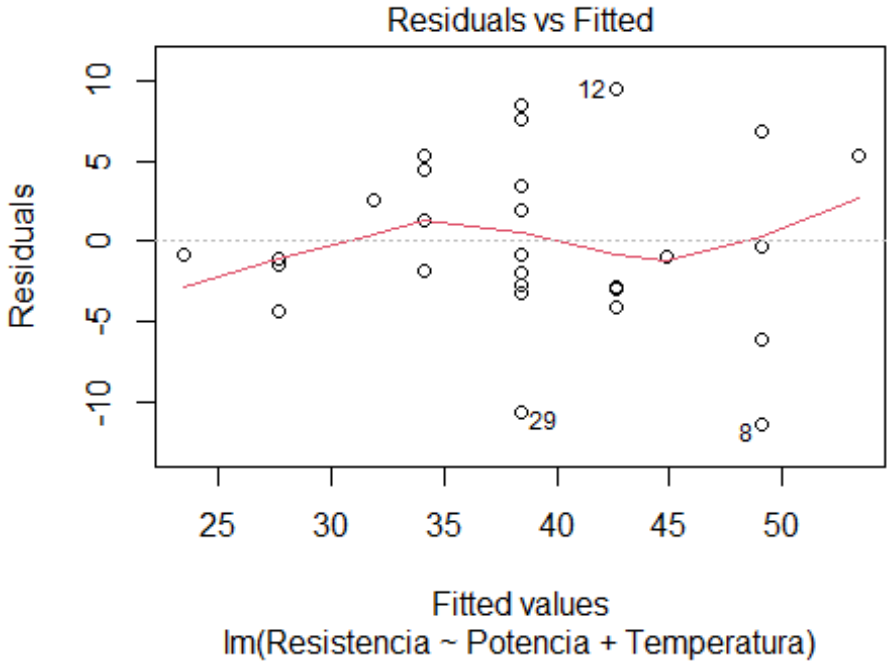
```
confint(Pasos)
```

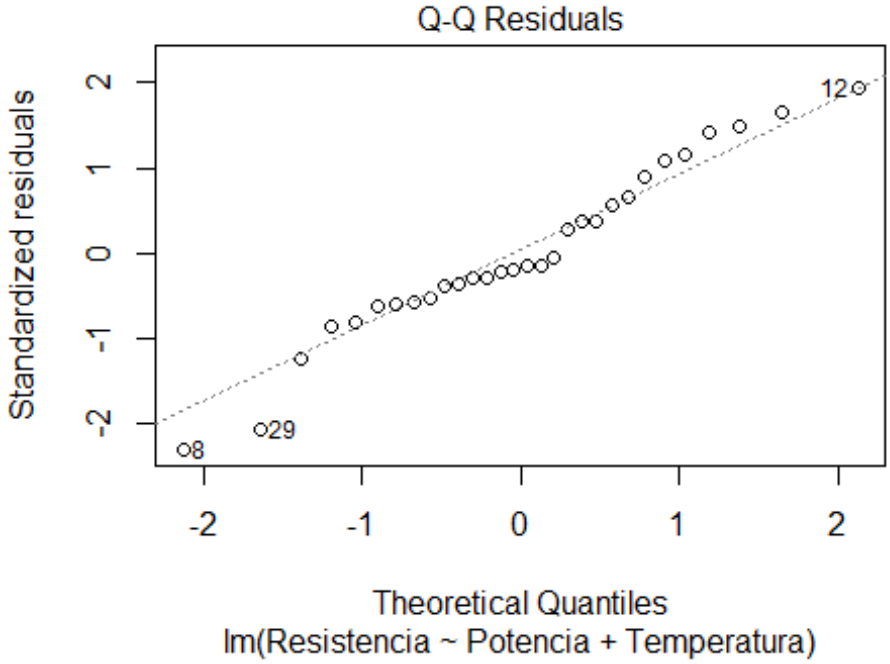
```
##                    2.5 %     97.5 %
## (Intercept) -45.56784390 -4.2354894
## Potencia      0.35294461  0.6437221
## Temperatura   0.04243343  0.2168999
```

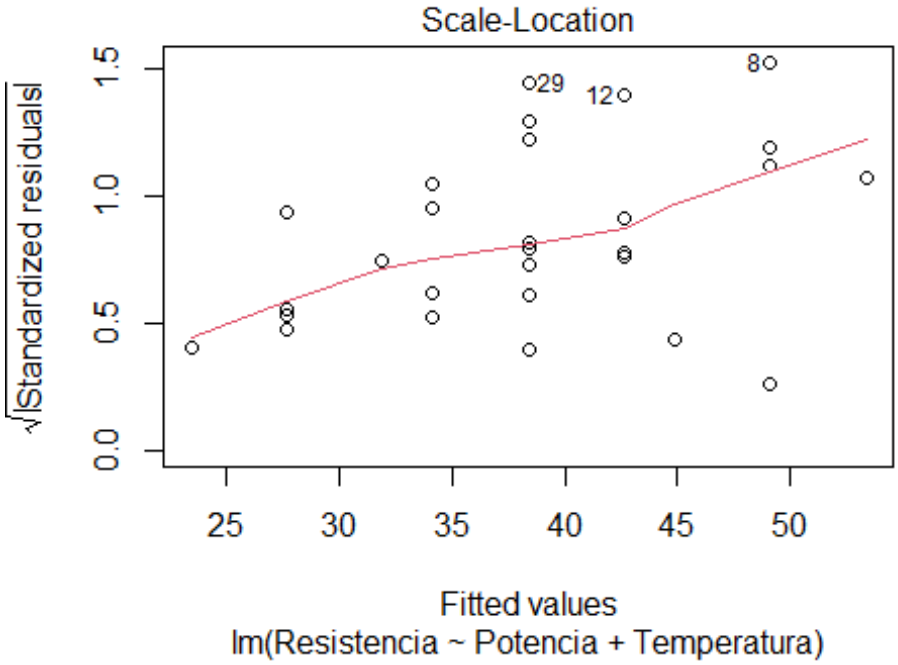
Como podemos ver los 3 metodos para encontrar una regresión multiple con buena economía resultan en los mismos 3 modelos, tomando en cuenta solamente potencia y temperatura, esto resulta en un r-squared de .68 de variación explicada por el modelo. Además por el análisis de cada variable podemos ver que la característica que más determina la resistencia es la potencia, seguido por la temperatura.

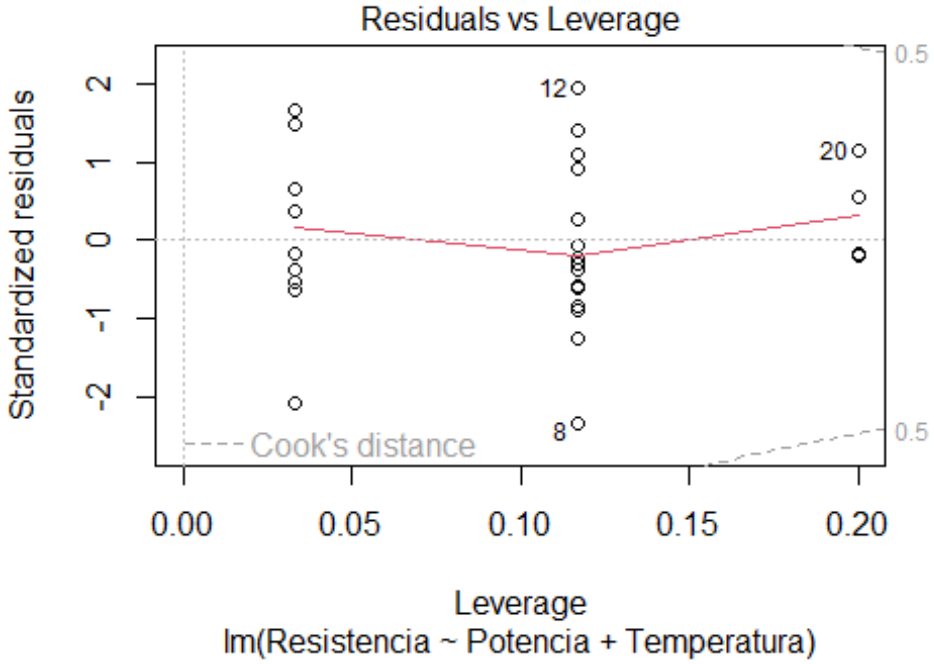
Análisis de residuos

```
plot(Pasos)
```


Residuals vs Fitted
Residuals
10
5
0
-5
-10
12
29
8
25
30
35
40
45
50
Fitted values
lm(Resistencia ~ Potencia + Temperatura)

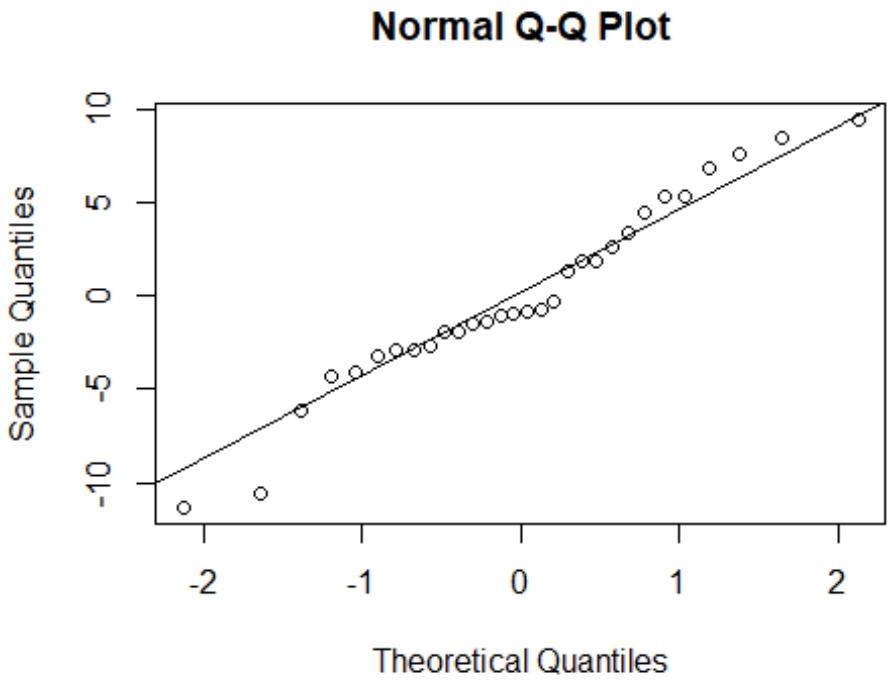

Q-Q Residuals
Standardized residuals
2
1
0
-1
-2
12
29
8
-2
-1
0
1
2
Theoretical Quantiles
lm(Resistencia ~ Potencia + Temperatura)





QQPlot

```
qqnorm(Pasos$residuals)
qqline(Pasos$residuals)
```



Homocedasticidad

```
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

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

```
bptest(Pasos)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  Pasos
## BP = 4.0043, df = 2, p-value = 0.135
```

```
gqtest(Pasos)
```

```
##
##  Goldfeld-Quandt test
##
## data:  Pasos
```

```
## GQ = 0.9753, df1 = 12, df2 = 12, p-value = 0.5169
## alternative hypothesis: variance increases from segment 1 to 2
```

Independencia

```
dwtest(Pasos)

##
##  Durbin-Watson test
##
## data:  Pasos
## DW = 2.3511, p-value = 0.8267
## alternative hypothesis: true autocorrelation is greater than 0

bgtest(Pasos)

##
##  Breusch-Godfrey test for serial correlation of order up to 1
##
## data:  Pasos
## LM test = 1.1371, df = 1, p-value = 0.2863
```

Linealidad

```
library(lmtest)
resettest(Pasos)

##
##  RESET test
##
## data:  Pasos
## RESET = 0.79035, df1 = 2, df2 = 25, p-value = 0.4647
```

Multicolinealidad

```
library(car)

## Loading required package: carData

vif(Pasos)

##    Potencia Temperatura
##           1           1
```

Conclusión

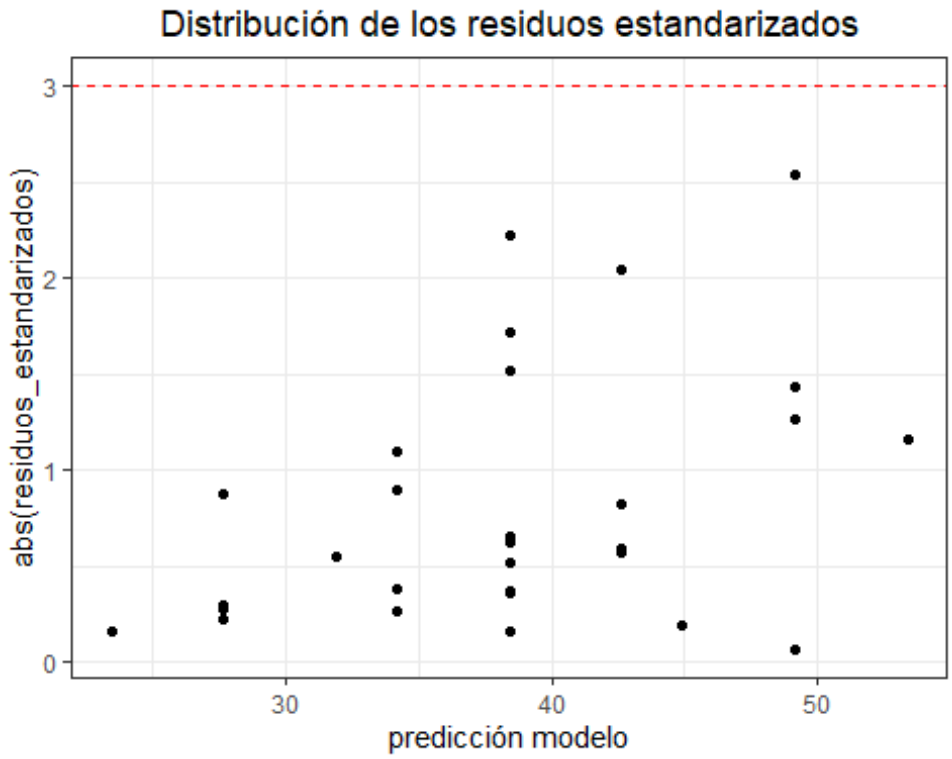
Podemos ver que es un modelo adecuado para poder predecir la Resistencia, sin embargo todavia no tiene resultados de residuos perfectos, existen datos atípicos que no son explicados por el modelo encontrado, lo que causa que no tenga mucha homocedasticidad, se puede ver claramente que con datos más pequeños el modelo es mas preciso que con datos arriba de 35. Las dos variables no estan correlacionadas, esto lo podemos comprobar

con los resultados de independencia y de multicolinealidad. Finalmente la gráfica de qqplot es muy útil y nos deja ver que los residuos siguen una curva muy parecidad a la normal, lo que indica que nuestro modelo es suficientemente acertado.

Análisis datos atípicos e influyentes

Datos atípicos (en y)

```
library(dplyr)

##
## Attaching package: 'dplyr'

## The following object is masked from 'package:car':
##
##     recode

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

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

library(ggplot2)
M$residuos_estandarizados <- rstudent(Pasos)
#Introduce una columna en Datos con los residuos estandarizados de los n
datos

ggplot(data = M, aes(x = predict(Pasos), y = abs(residuos_estandarizados))) +
geom_hline(yintercept = 3, color = "red", linetype = "dashed") +
# se identifican en rojo observaciones con residuos estandarizados absolutos
> 3
geom_point(aes(color = ifelse(abs(residuos_estandarizados) > 3, 'red',
'black'))) +
scale_color_identity() +
labs(title = "Distribución de los residuos estandarizados", x = "predicción
modelo") +
theme_bw() + theme(plot.title = element_text(hjust = 0.5))
```

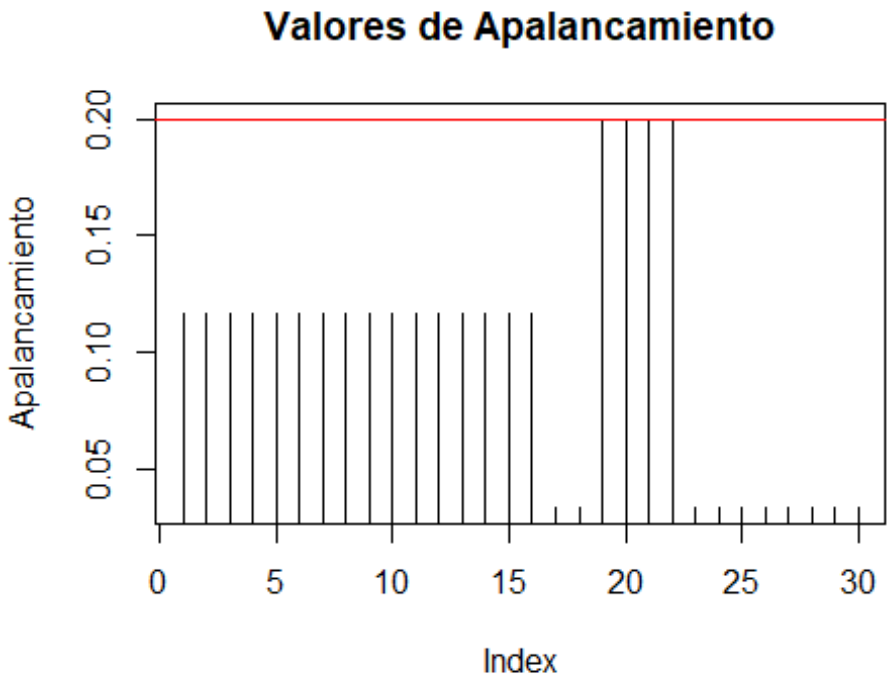



```
Atipicos = which(abs(M$residuos_estandarizados)>3)
M[Atipicos, ]
```

```
## [1] Fuerza                  Potencia                Temperatura
## [4] Tiempo                  Resistencia
residuos_estandarizados
## <0 rows> (or 0-length row.names)
```

Datos Atípicos (en x)

```
leverage = hatvalues(Pasos)
#Calcula el leverage de los n datos

plot(leverage, type="h", main="Valores de Apalancamiento",
ylab="Apalancamiento")
abline(h = 2*mean(leverage), col="red") # Límite comúnmente usado
```

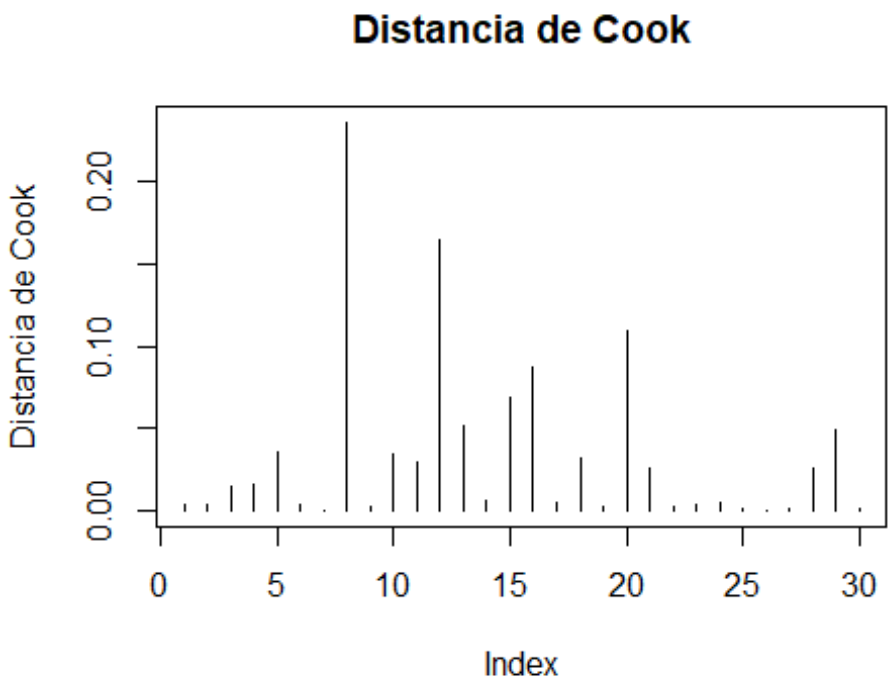



```
high_leverage_points = which(leverage > 2*mean(leverage))

M[high_leverage_points, ]
```

```
##    Fuerza Potencia Temperatura Tiempo Resistencia residuos_estandarizados
## 19     35       45         200     20        22.7                -0.159511
## 20     35      105         200     20        58.7                 1.154355
```

Datos Influyentes

```
cooksdistance <- cooks.distance(Pasos)
#Calcula la distancia de Cook de los n datos

plot(cooksdistance, type="h", main="Distancia de Cook", ylab="Distancia de
Cook")
abline(h = 1, col="red") # Límite comúnmente usado
```

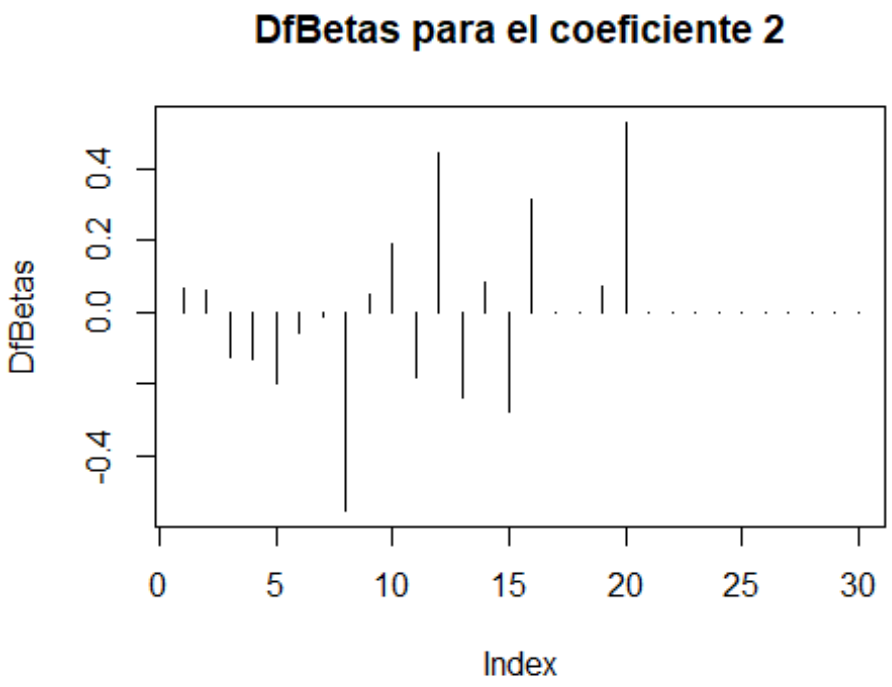



```
puntos_influyentes = which(cooksdistance > 1)

M[puntos_influyentes, ]
```

```
## [1] Fuerza                  Potencia                Temperatura
## [4] Tiempo                  Resistencia
residuos_estandarizados
## <0 rows> (or 0-length row.names)
```

Datos Influyentes en Betas

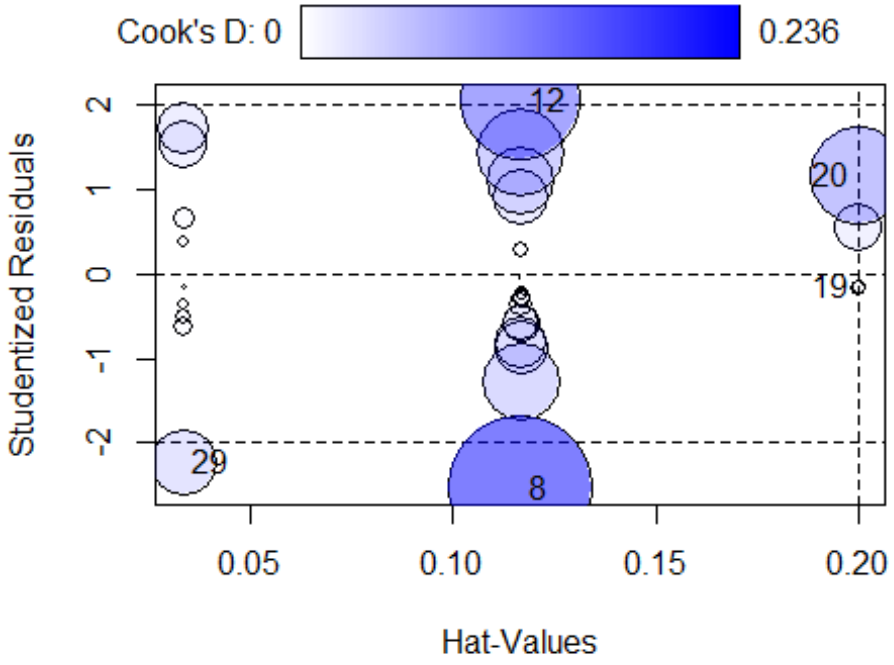
```
dfbetas_values = dfbetas(Pasos)
#Calcula la DfBeta de los n datos para cada βj

plot(dfbetas_values[,2], type="h", main="DfBetas para el coeficiente 2",
ylab="DfBetas")
abline(h = c(-1, 1), col="red") # Límites comunes
```



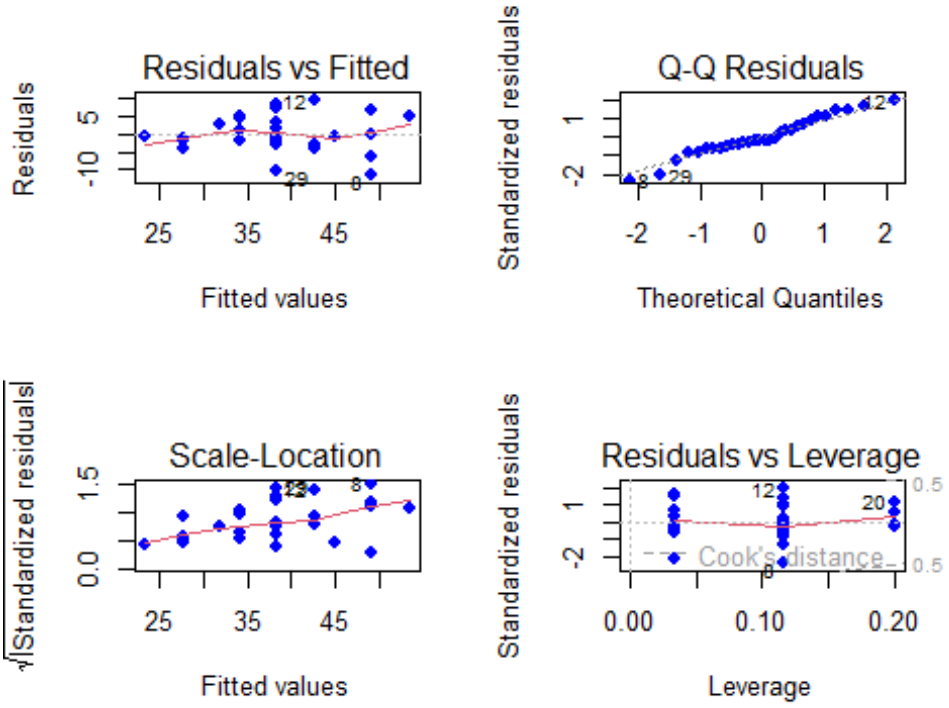

```
puntos_influyentes = which(abs(dfbetas_values[, 2]) > 1)
```

Resumen datos influyentes

```
influencia = influence.measures(Pasos)
#Calcula las medidas de los n datos

summary(influencia)

## Potentially influential observations of
##   lm(formula = Resistencia ~ Potencia + Temperatura, data = M) :
##
##    dfb.1_ dfb.Ptnc dfb.Tmpr dffit cov.r   cook.d hat
## 8   0.71  -0.55    -0.55    -0.92  0.65_*  0.24   0.12
## 19 -0.04   0.07     0.00    -0.08  1.40_*  0.00   0.20
## 21  0.22   0.00    -0.25     0.27  1.35_*  0.03   0.20
## 22  0.07   0.00    -0.09    -0.09  1.39_*  0.00   0.20

# Detecta los datos con posible influencia

library(car)
influencePlot(Pasos)
```




```
##      StudRes        Hat       CookD
## 8  -2.535832 0.11666667 0.235696235
## 12  2.043589 0.11666667 0.164507739
## 19 -0.159511 0.20000000 0.002199712
## 20  1.154355 0.20000000 0.109693544
## 29 -2.216952 0.03333333 0.049338917

par(mfrow=c(2, 2))
plot(Pasos, col="blue", pch=19)
```



Conclusiones

Como podemos ver el modelo explica muy bien los datos, por el análisis de datos atípicos e influyentes podemos ver que solo tiene 2 datos atípicos en el eje de las x, mientras que no muestra tener datos influyentes en ninguna parte del modelo.