Actividad 3: Regresión Múltiple - Detección de Datos Atípicos

Daniela Jiménez Téllez

2024-09-24

Problema

En la base de datos Al corte se describe un experimento realizado para evaluar el impacto de las variables: fuerza, potencia, temperatura y tiempo sobre la resistencia al corte. Indica cuál es la mejor relación entre estas variables que describen la resistencia al corte.

Importación de librerías

```
library(lmtest)
```

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

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

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

```
library(car)
```

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

```
library(dplyr)
```

```
##
## Adjuntando el paquete: '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)
```

Importación de datos

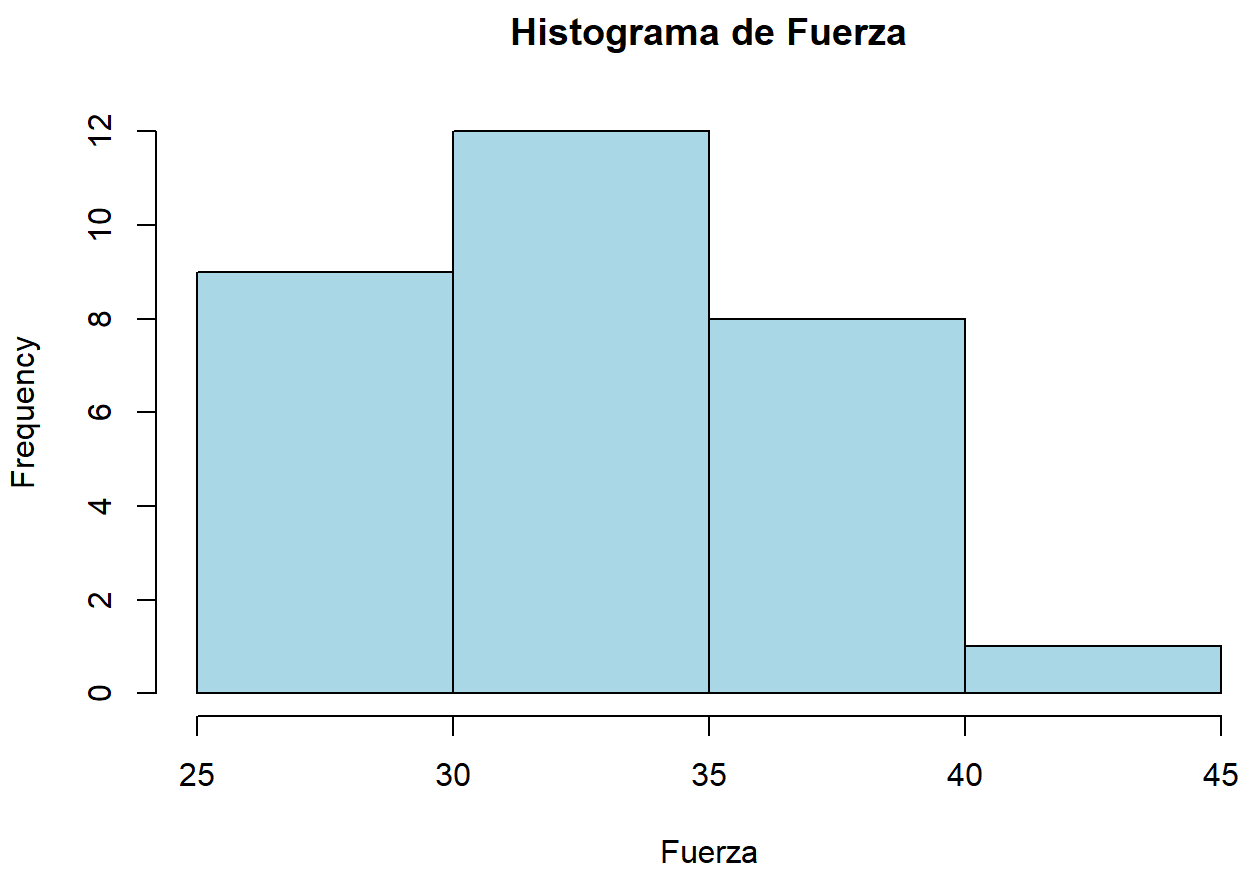
```
datos <- read.csv("AlCorte.csv")
```

1. Haz un análisis descriptivo de los datos: medidas principales y gráficos (ya lo hiciste en la actividad A2).

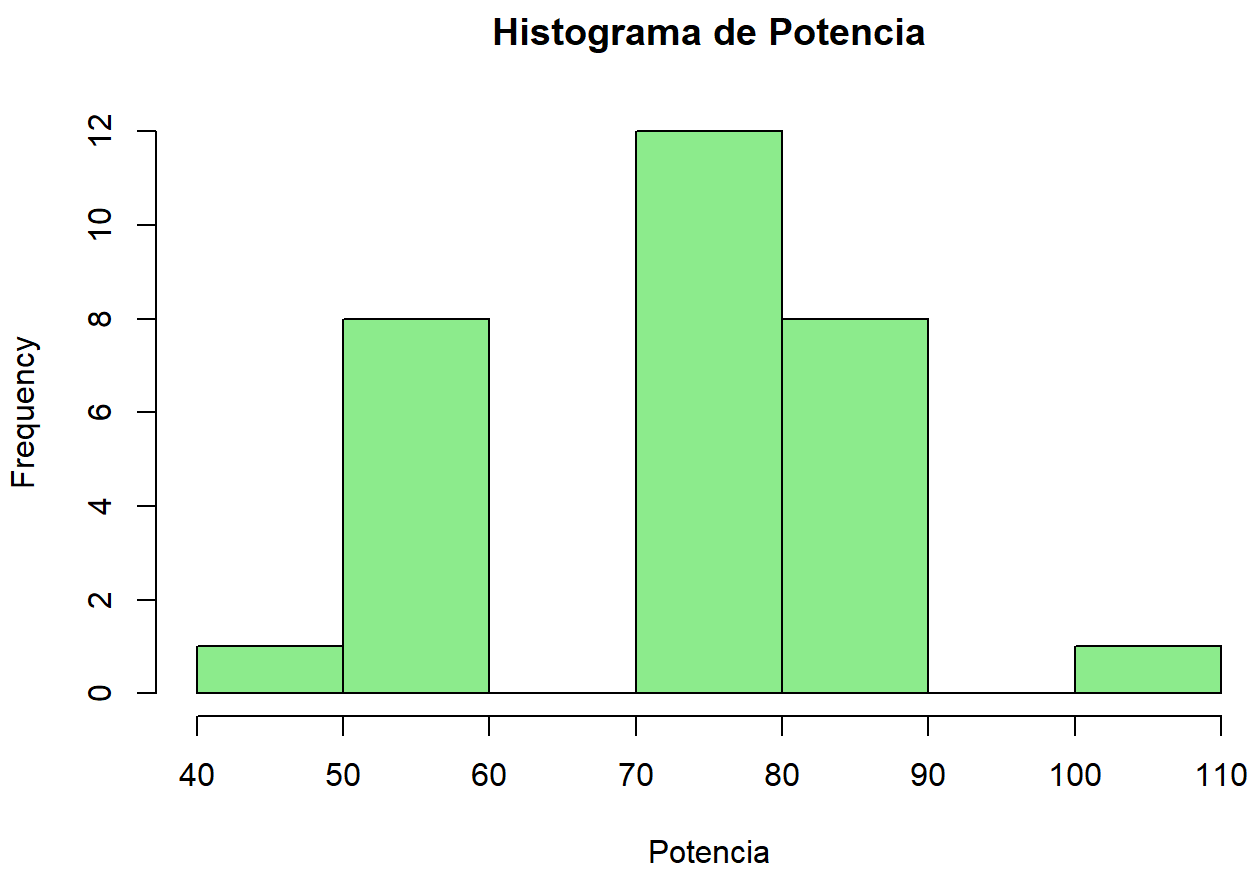
```
# Resumen de medidas

print(summary(datos))
```

```
##      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
```

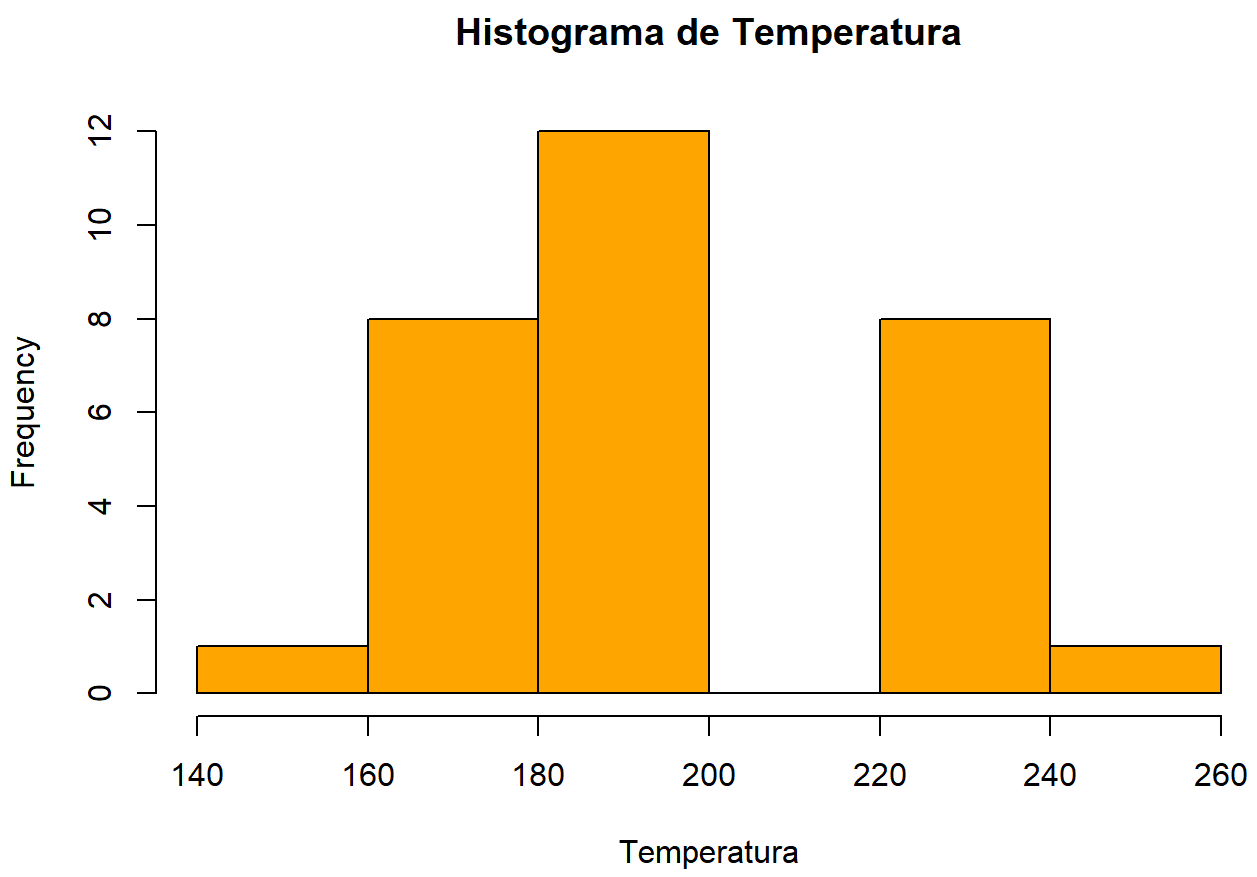
```
# Histogramas

hist(datos$Fuerza, main = "Histograma de Fuerza", xlab = "Fuerza", col = "lightblue")
```



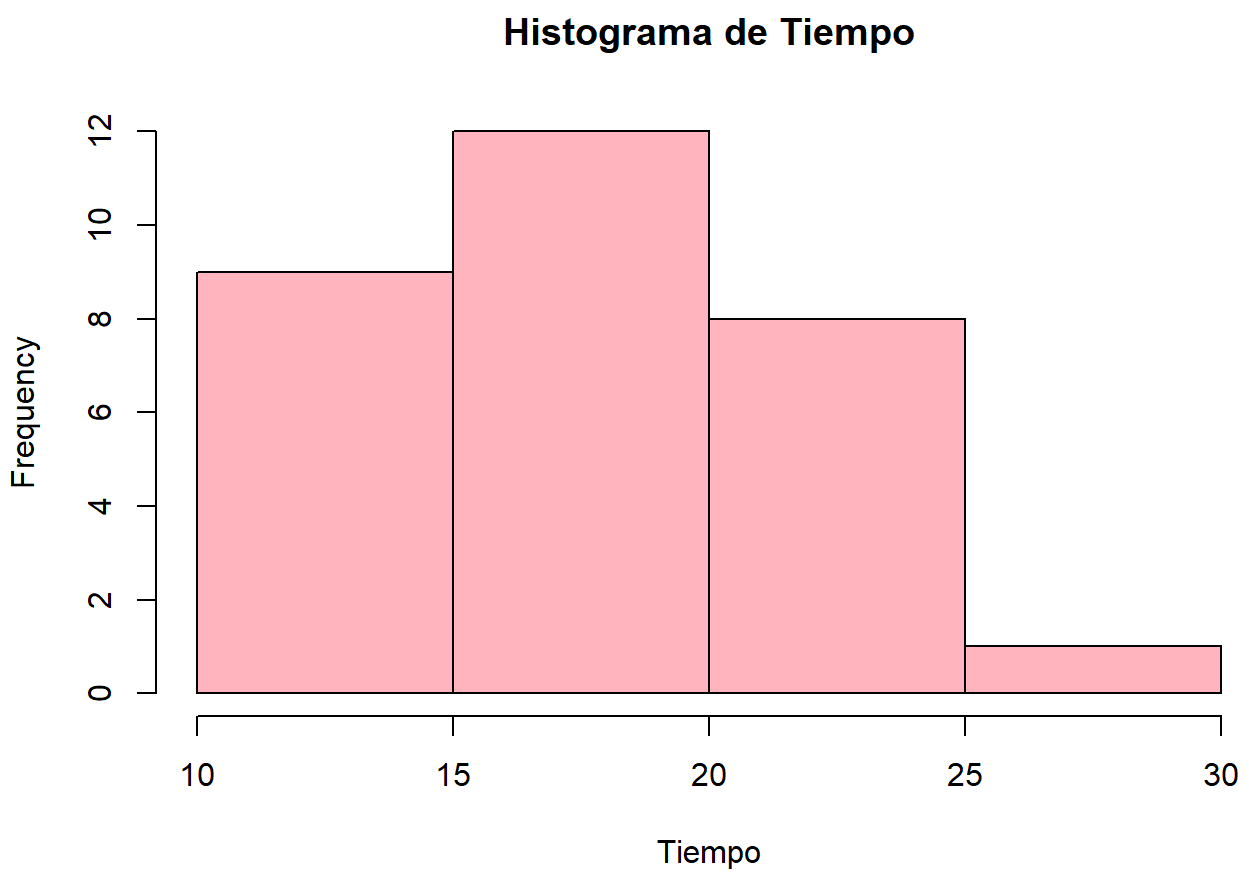

```
hist(datos$Potencia, main = "Histograma de Potencia", xlab = "Potencia", col = "lightgreen")
```



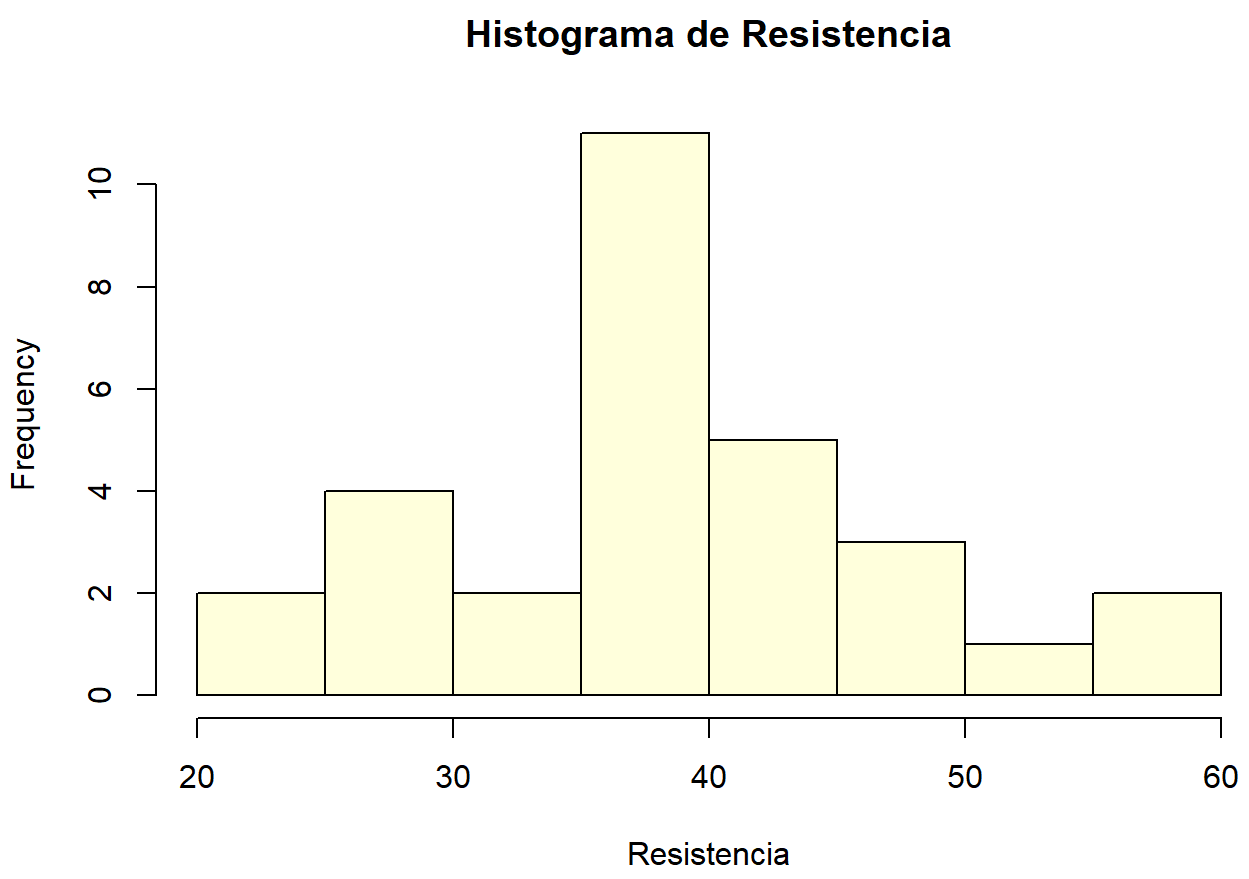

```
hist(datos$Temperatura, main = "Histograma de Temperatura", xlab = "Temperatura", col = "orang
e")
```



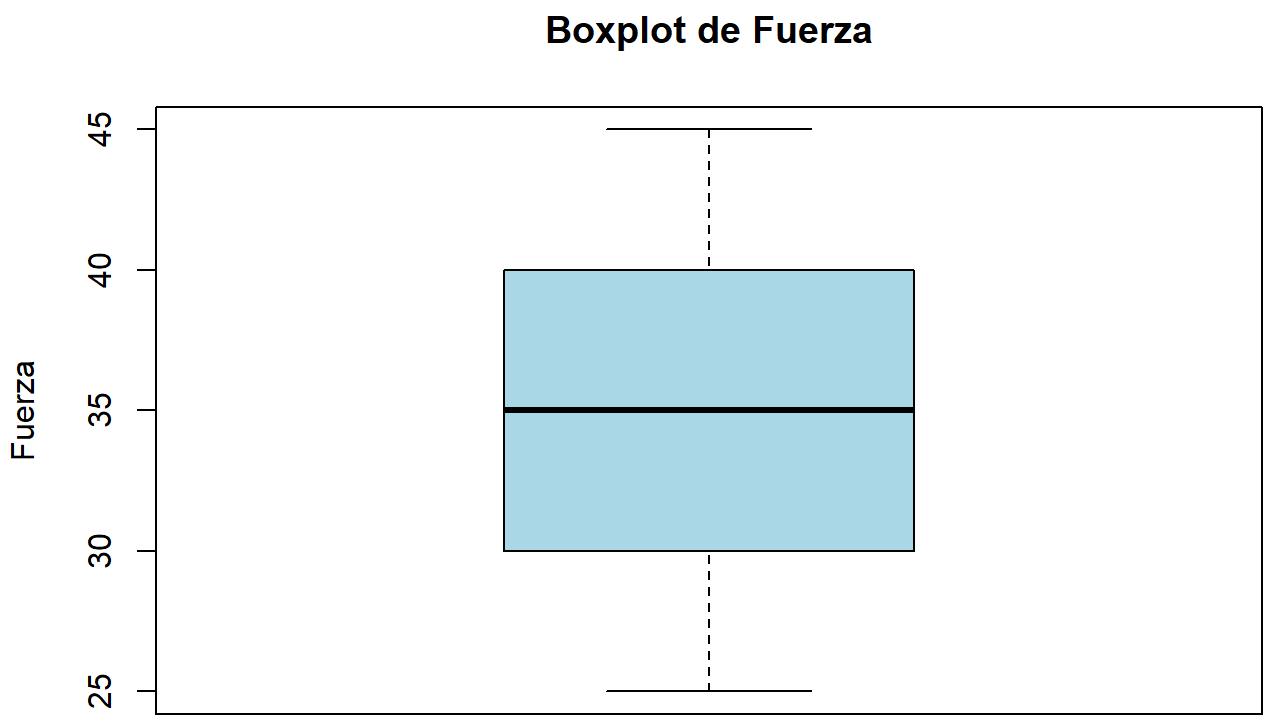

```
hist(datos$Tiempo, main = "Histograma de Tiempo", xlab = "Tiempo", col = "lightpink")
```



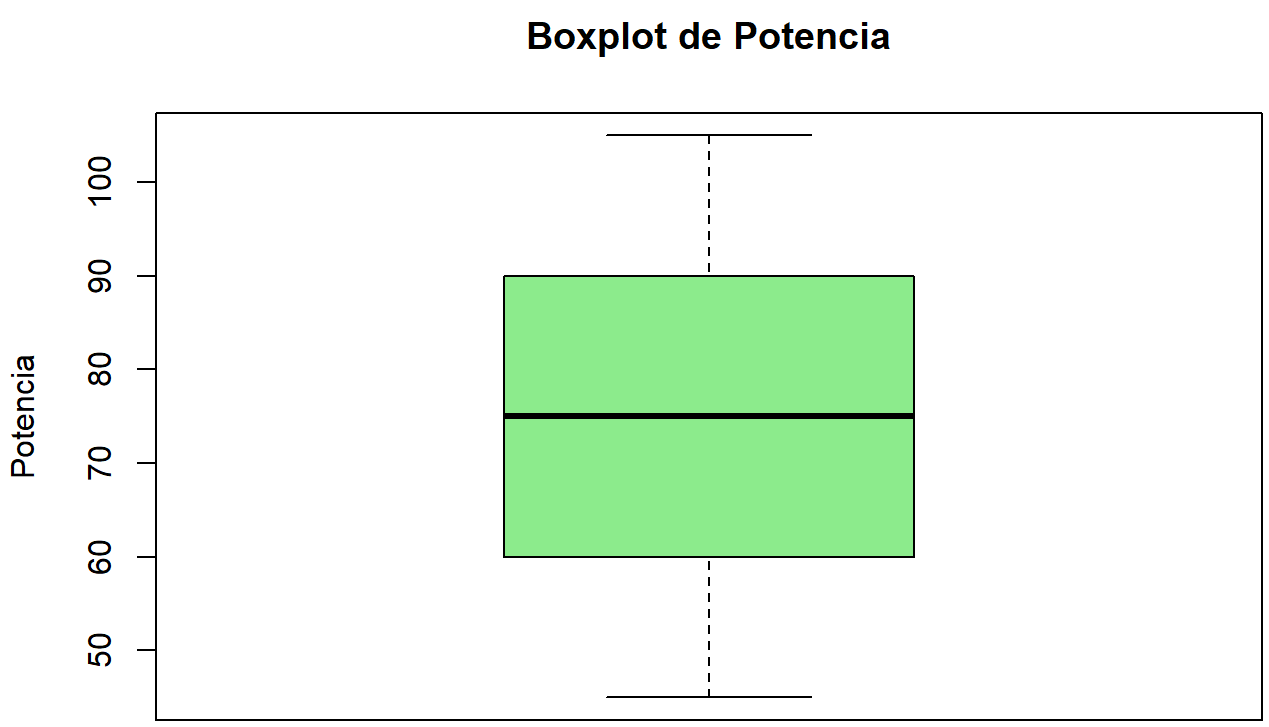

```
hist(datos$Resistencia, main = "Histograma de Resistencia", xlab = "Resistencia", col = "lightye
llow")
```



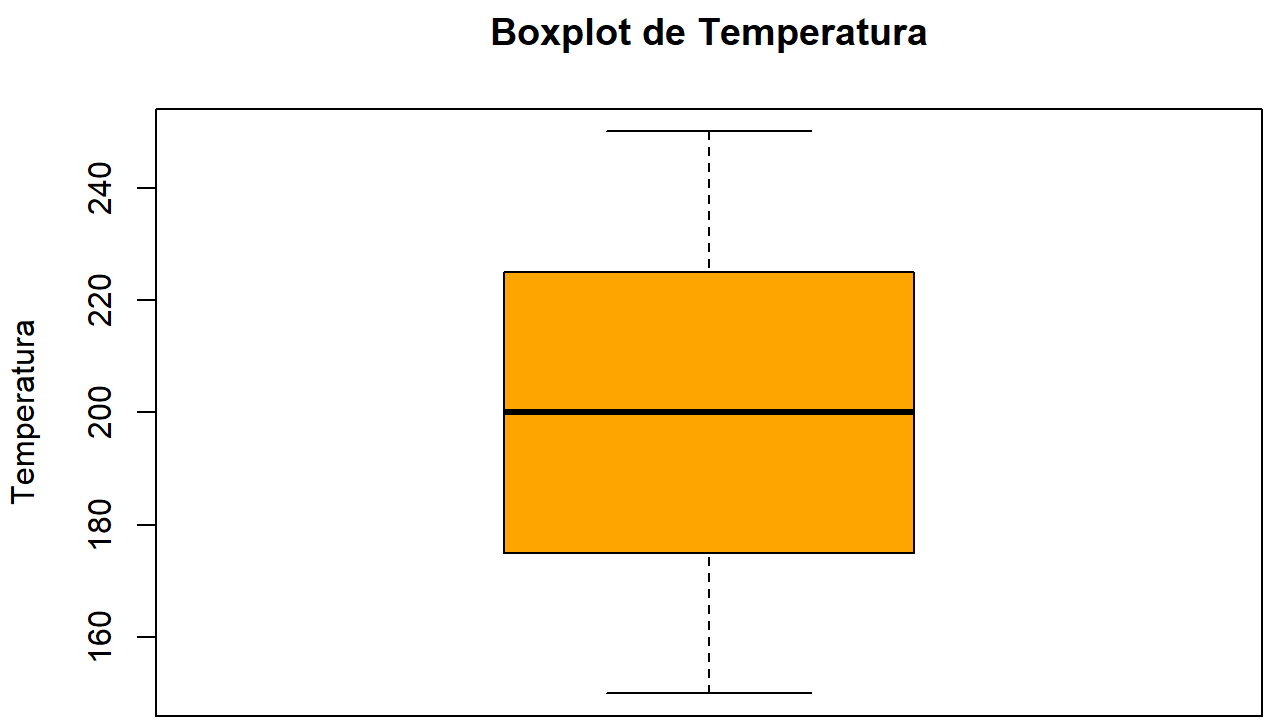

```
# Boxplots

boxplot(datos$Fuerza, main = "Boxplot de Fuerza", ylab = "Fuerza", col = "lightblue")
```



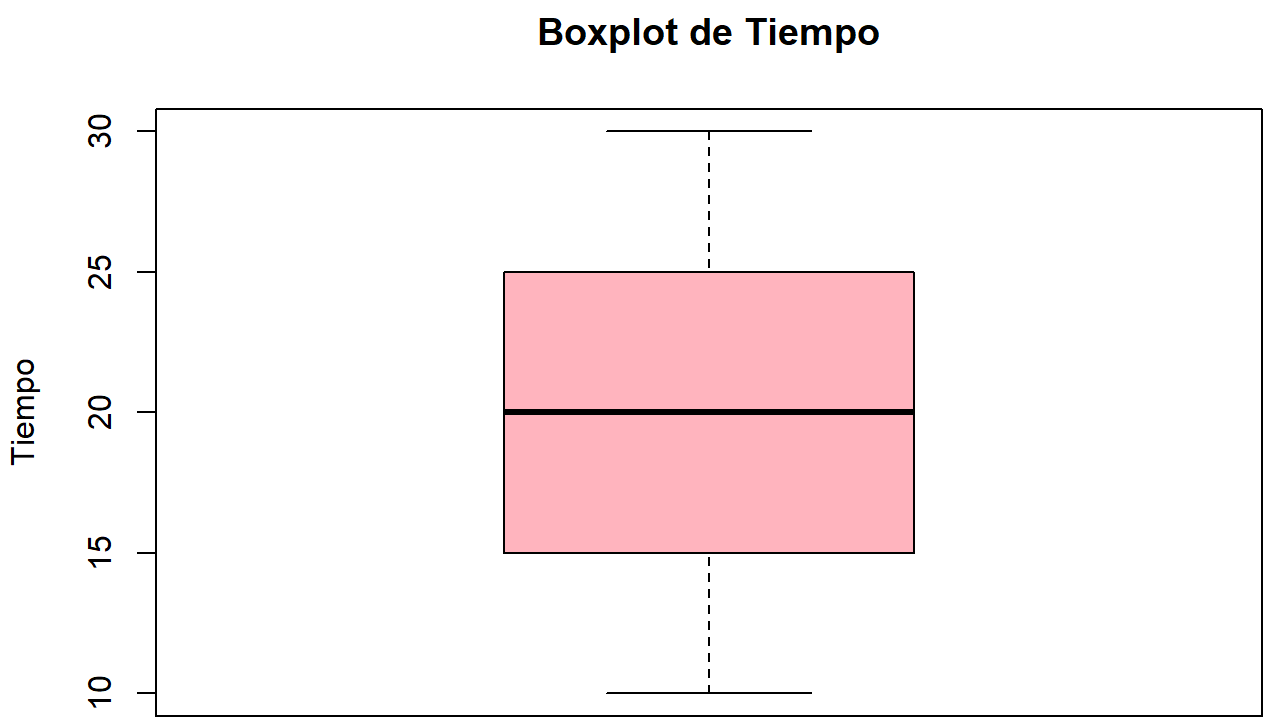

```
boxplot(datos$Potencia, main = "Boxplot de Potencia", ylab = "Potencia", col = "lightgreen")
```



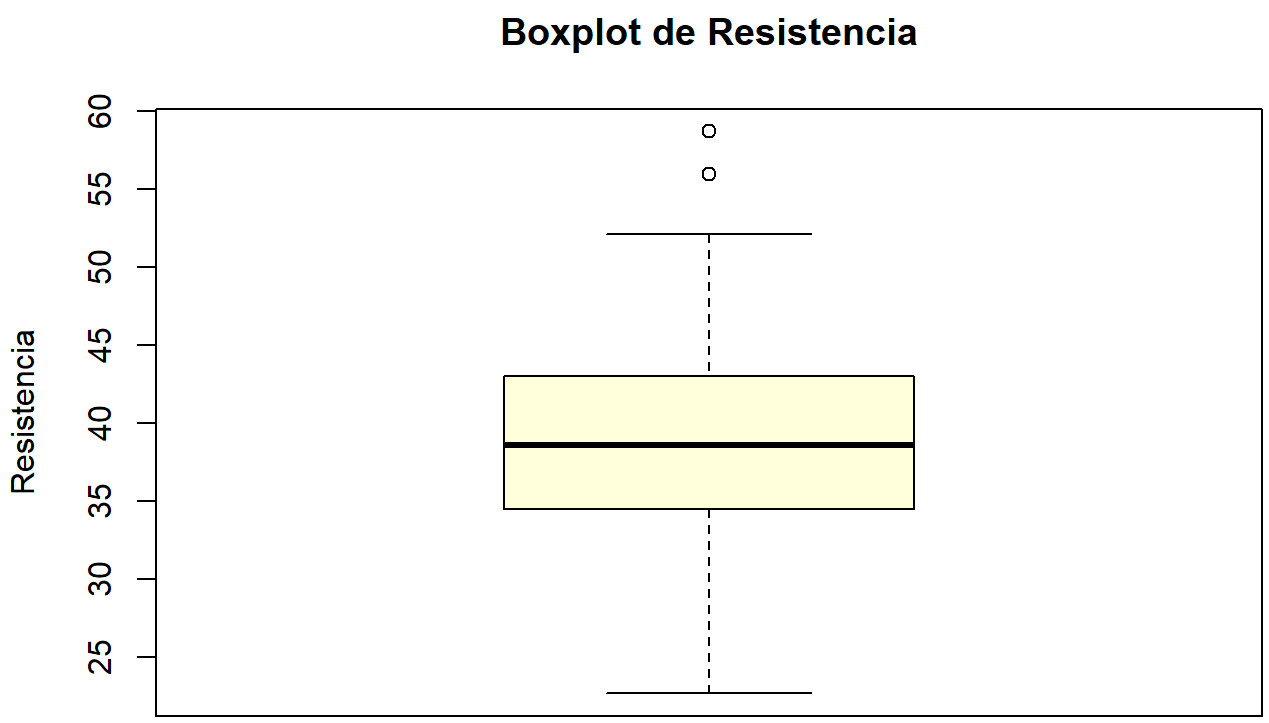

```
boxplot(datos$Temperatura, main = "Boxplot de Temperatura", ylab = "Temperatura", col = "orang
e")
```




```
boxplot(datos$Tiempo, main = "Boxplot de Tiempo", ylab = "Tiempo", col = "lightpink")
```




```
boxplot(datos$Resistencia, main = "Boxplot de Resistencia", ylab = "Resistencia", col = "lightye
llow")
```



2. Encuentra el mejor modelo de regresión que explique la variable Resistencia (ya lo hiciste en la actividad A2).

Como se mencionó en la actividad pasada, el mejor modelo de regresión que explica la variable Resistencia es el que contiene las variables Potencia y Temperatura.

```
modelo <- lm(Resistencia ~ Potencia + Temperatura, data = datos)
```

3. Analiza la validez del modelo encontrado (ya lo hiciste en la actividad A2).

```
# Significancia global

resumen <- summary(modelo)

# Significancia individual

sigIndividual <- confint(modelo)

# Variación

r2 <- resumen$adj.r.squared

# Prints

cat("Summary: \n\n")
```

```
## Summary:
```

```
print(resumen)
```

```
##
## Call:
## lm(formula = Resistencia ~ Potencia + Temperatura, data = datos)
##
## 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
```

```
cat("\n\n")
```

```
cat("Significancia Individual: \n\n")
```

```
## Significancia Individual:
```

```
print(sigIndividual)
```

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

```
cat("\n\n")
```

```
cat("Variación del modelo: \n\n")
```

```
## Variación del modelo:
```

```
cat("R² ajustado =", r2, "\n")
```

```
## R² ajustado = 0.6618581
```

Validez del modelo

- Análisis de residuos (homocedasticidad, independencia, etc)
- No multicolinealidad de X_i

```
# Media cero

mediaCero <- t.test(resid(modelo))

# Normalidad

norm <- shapiro.test(resid(modelo))

# Homocedasticidad

hom1 <- bptest(modelo)
hom2 <- gqtest(modelo)

# Independencia

ind1 <- dwtest(modelo)
ind2 <- bgtest(modelo)

# No multicolinealidad

multi <- vif(modelo)

# Prints

cat("Media Cero en los Residuos del Modelo:\n\n")
```

```
## Media Cero en los Residuos del Modelo:
```

```
print(mediaCero)
```

```
##
##  One Sample t-test
##
## data:  resid(modelo)
## t = 8.8667e-17, df = 29, p-value = 1
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -1.876076  1.876076
## sample estimates:
##    mean of x
## 8.133323e-17
```

```
cat("\n\n")
```

```
cat("Prueba de Normalidad (Shapiro-Wilk) para los Residuos del Modelo:\n")
```

```
## Prueba de Normalidad (Shapiro-Wilk) para los Residuos del Modelo:
```

```
print(norm)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(modelo)
## W = 0.96588, p-value = 0.4333
```

```
cat("\n\n")
```

```
cat("Prueba de Homocedasticidad Breusch-Pagan para el Modelo:\n")
```

```
## Prueba de Homocedasticidad Breusch-Pagan para el Modelo:
```

```
print(hom1)
```

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

```
cat("\n")
```

```
cat("Prueba de Homocedasticidad Goldfeld-Quandt para el Modelo:\n")
```

```
## Prueba de Homocedasticidad Goldfeld-Quandt para el Modelo:
```

```
print(hom2)
```

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

```
cat("\n\n")
```

```
cat("Prueba de Independencia Durbin-Watson para el Modelo:\n")
```

```
## Prueba de Independencia Durbin-Watson para el Modelo:
```

```
print(ind1)
```

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

```
cat("\n")
```

```
cat("Prueba de Independencia Breusch-Godfrey para el Modelo:\n")
```

```
## Prueba de Independencia Breusch-Godfrey para el Modelo:
```

```
print(ind2)
```

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

```
cat("\n\n")
```

```
cat("Verificación de Multicolinealidad (VIF) para el Modelo:\n")
```

```
## Verificación de Multicolinealidad (VIF) para el Modelo:
```

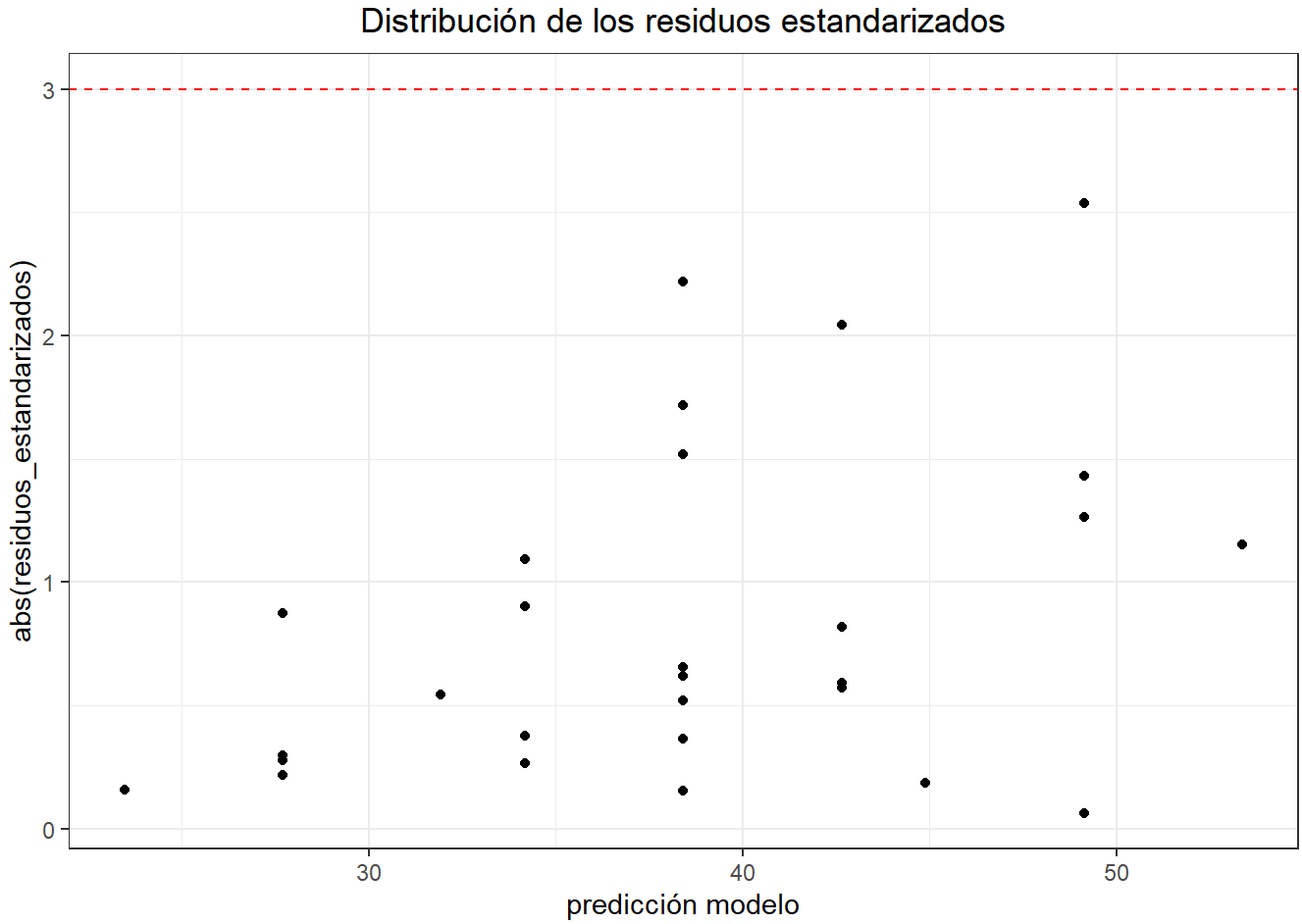
```
print(multi)
```

```
##    Potencia Temperatura
##           1           1
```

```
cat("\n\n")
```

4. Haz el análisis de datos atípicos e influyentes del mejor modelo encontrado.

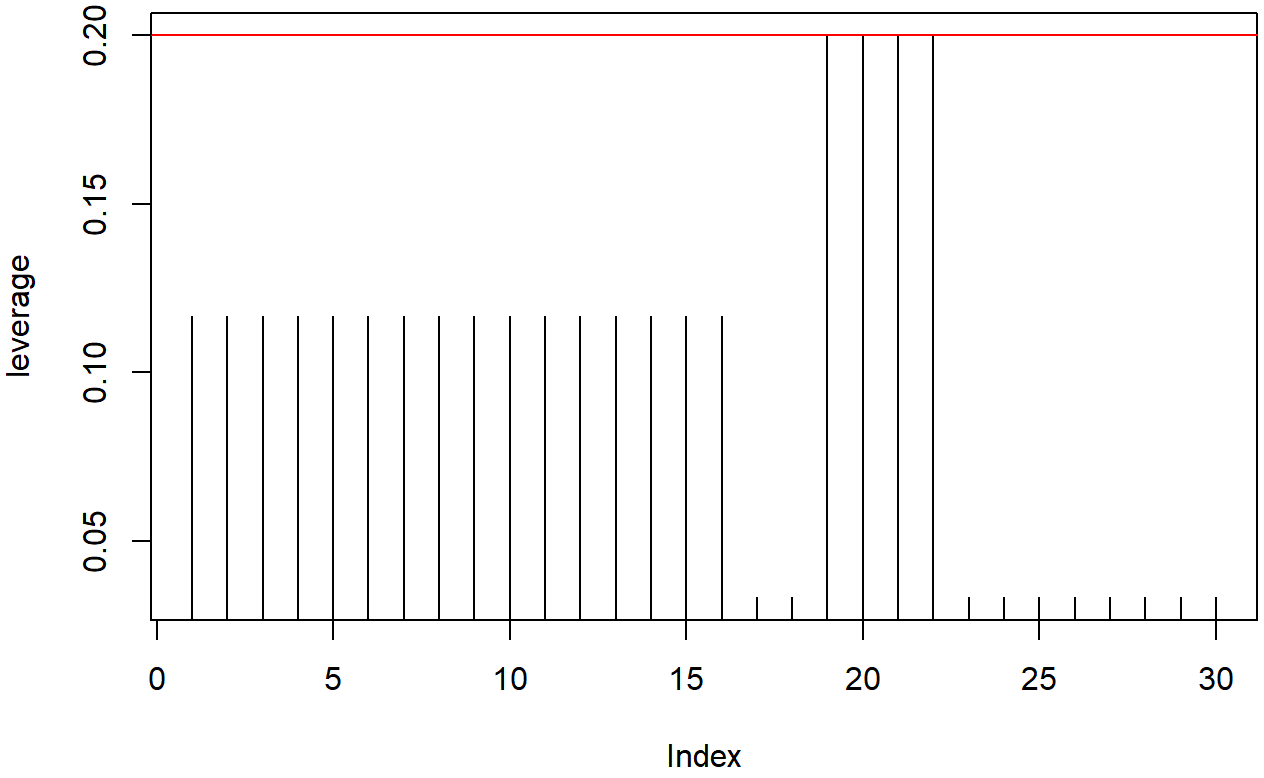
```
## DATOS ATÍPICOS

# Estandarización extrema de los residuos

datos$residuos_estandarizados <- rstudent(modelo)

ggplot(data = datos, aes(x = predict(modelo), y = abs(residuos_estandarizados))) +
  geom_hline(yintercept = 3, color = "red", linetype = "dashed") +

  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(datos$residuos_estandarizados) > 3)

datos[Atipicos, ]
```

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

```
# Distancia de Leverage

leverage <- hatvalues(modelo)

plot(leverage, type = "h")
abline(h = 2*mean(leverage), col="red")
```



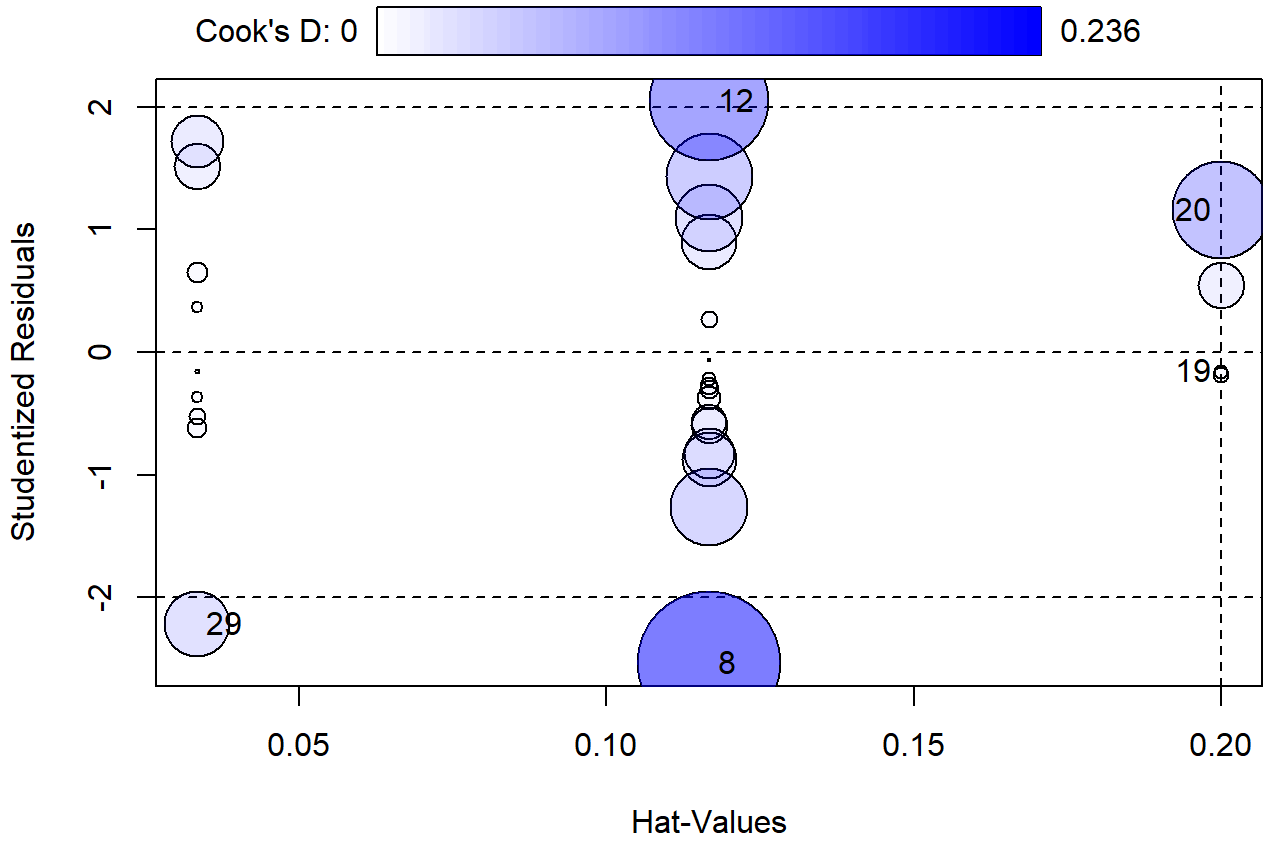

```
high_leverage_points <- which(leverage > 2*mean(leverage))

datos[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

I <- influence.measures(modelo)
summary(I)
```

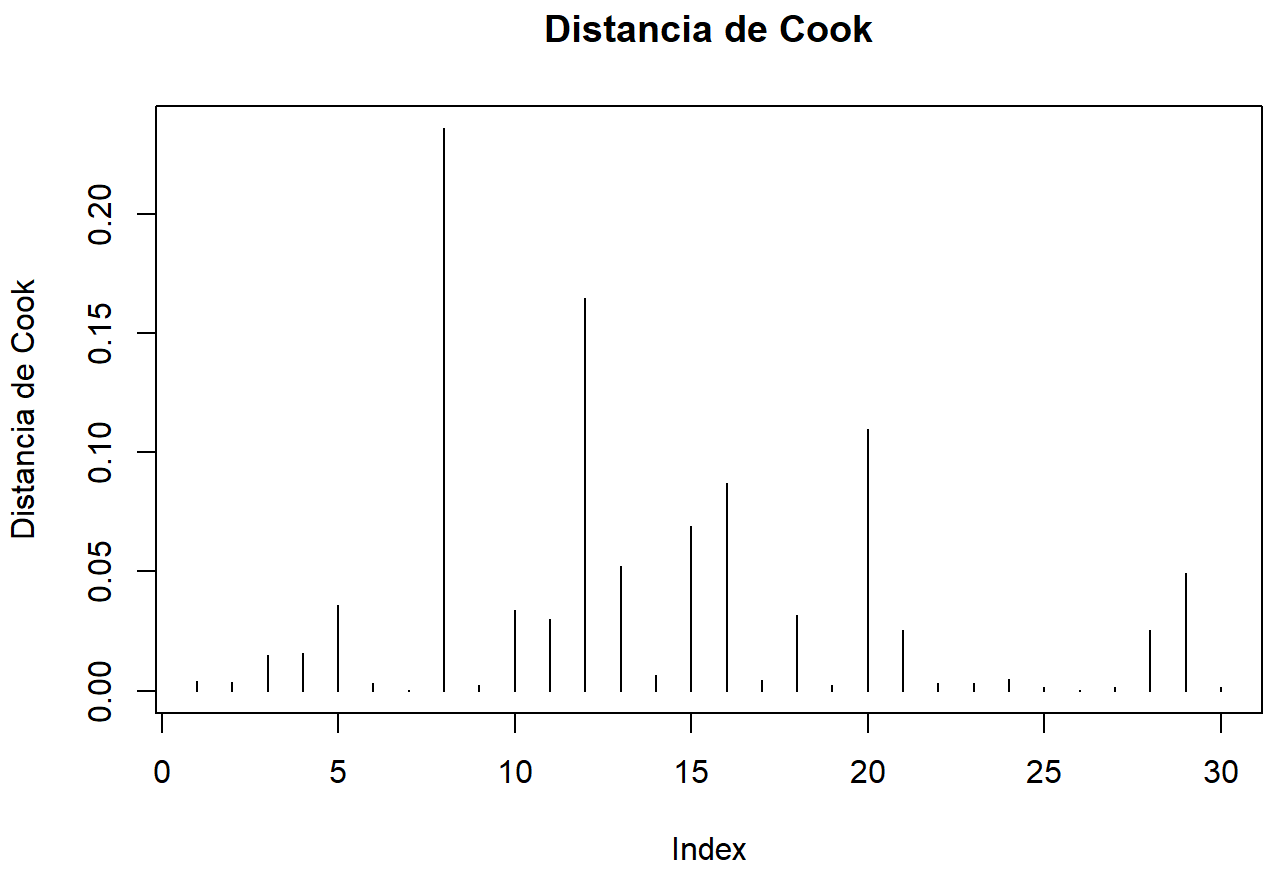
```
## Potentially influential observations of
##   lm(formula = Resistencia ~ Potencia + Temperatura, data = datos) :
##
##    dfb.1_ dfb.Ptnc dfb.Tmpr dffit cov.r   cook.d hat
## 8   0.71  -0.55    -0.55    -0.92  0.65_*  0.24   0.12
## 19 -0.04   0.07     0.00    -0.08  1.40_*  0.00   0.20
## 21 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

```
influencePlot(modelo)
```




```
##      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
```

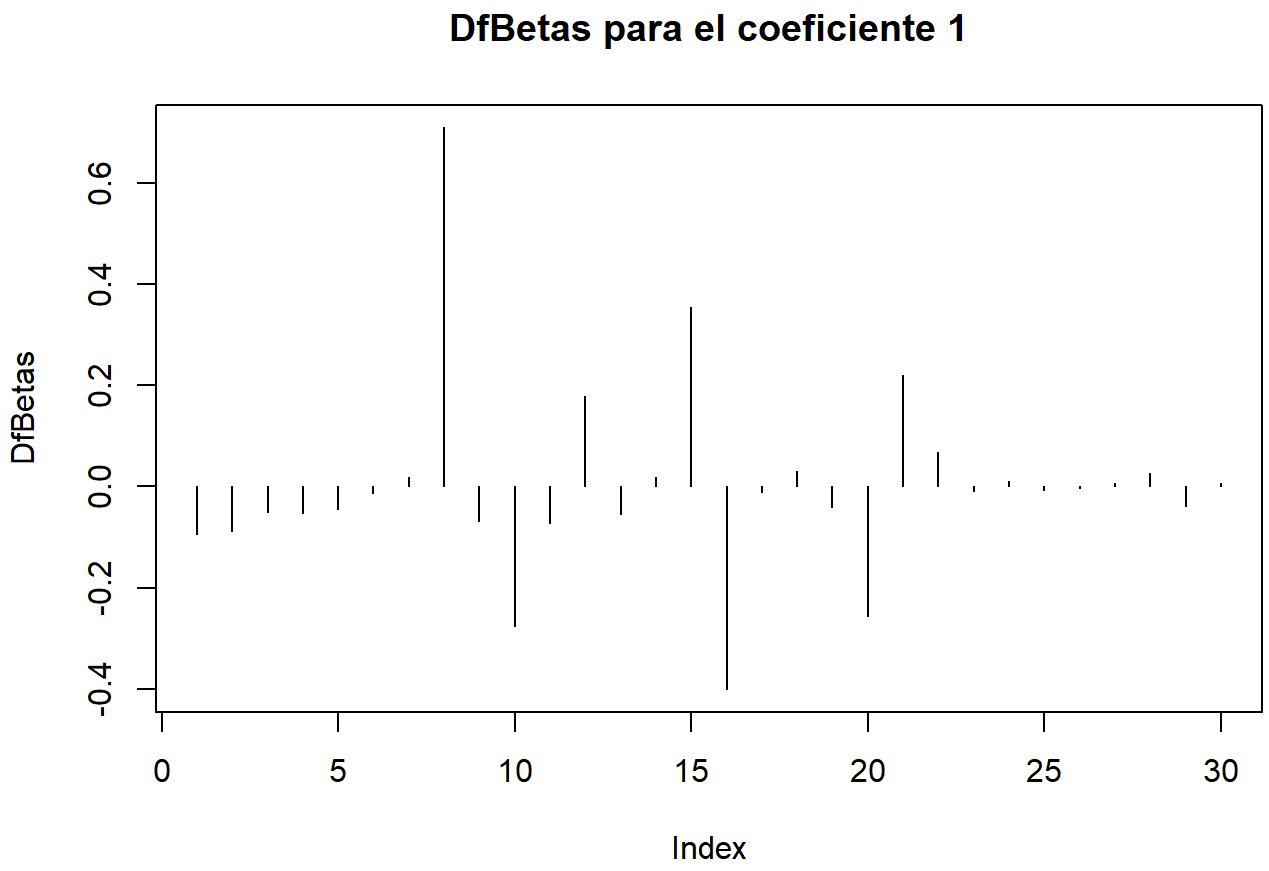
```
# Distancia de cook

cooks_dist <- cooks.distance(modelo)

plot(cooks_dist, type = "h", main = "Distancia de Cook", ylab = "Distancia de Cook")
abline(h = 1, col = "red")
```

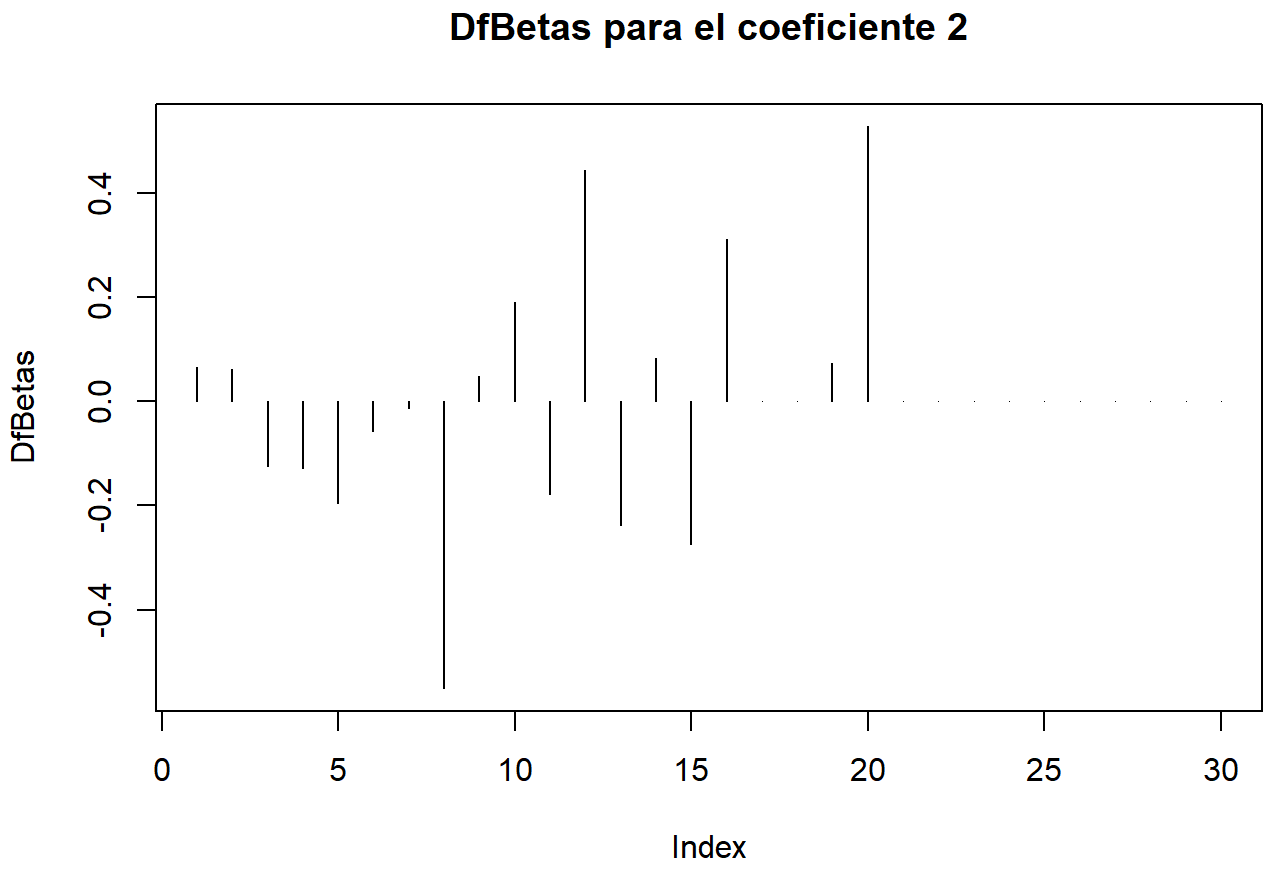


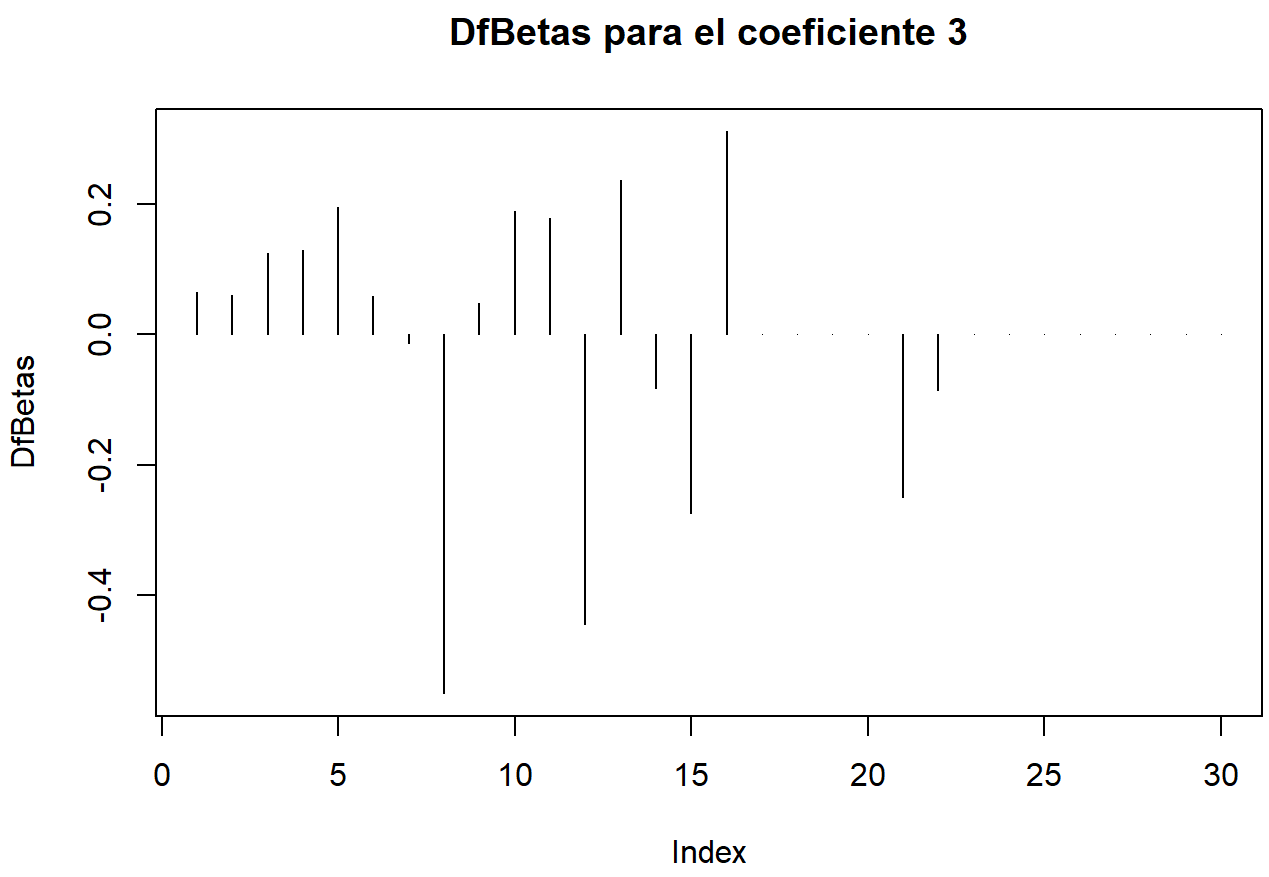

```
influyentes_cooks <- which(cooks_dist > 1)

datos[influyentes_cooks, ]
```

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

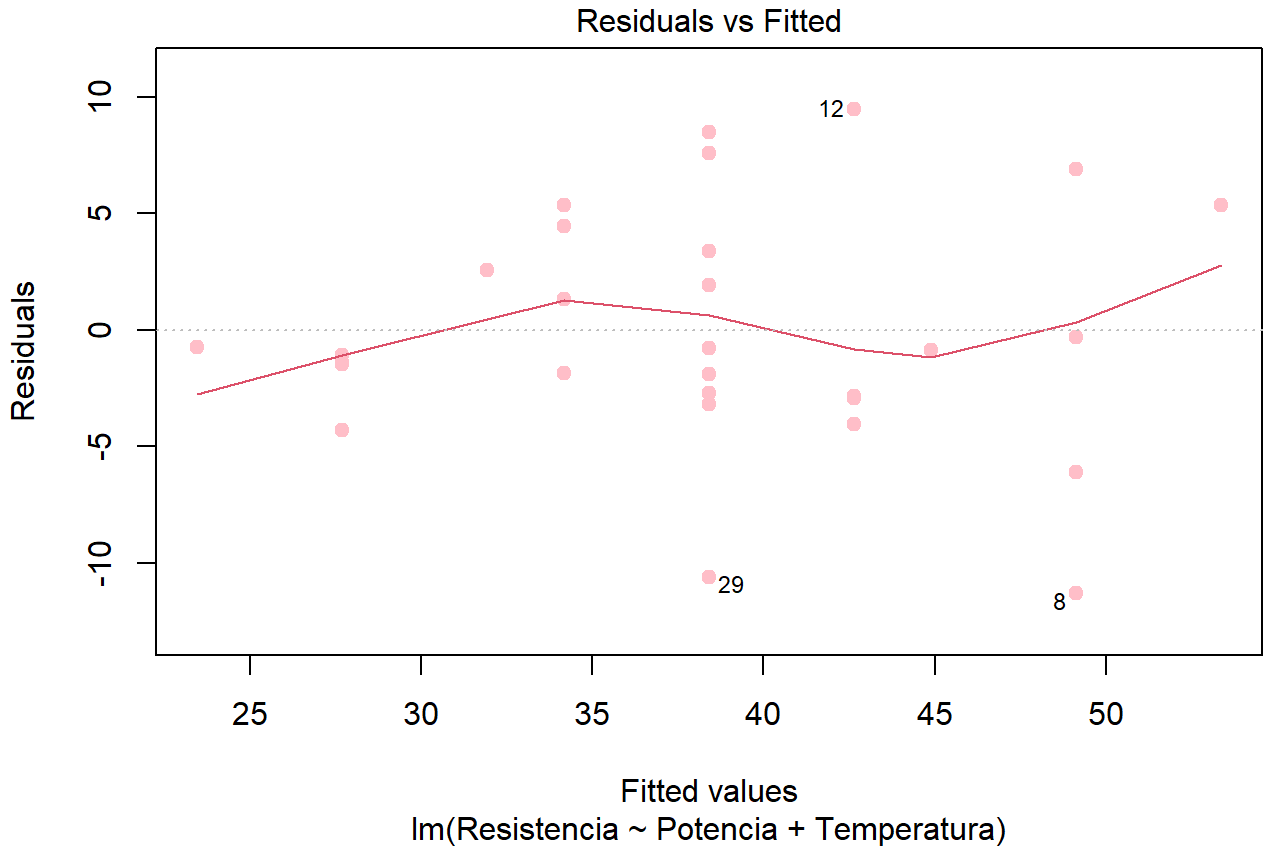
```
# DFBetas

dfbet <- dfbetas(modelo)

num_coef <- ncol(dfbet)

for (i in 1:num_coef) {

  plot(dfbet[, i], type = "h", main = paste("DfBetas para el coeficiente", i), ylab = "DfBetas")
  abline(h = c(-1, 1), col = "red")

  influyentes_dfb <- which(abs(dfbet[, i]) > 1)

  if (length(influyentes_dfb) > 0) {
    print(paste("Observaciones influyentes para el coeficiente", i, ":"))
    print(influyentes_dfb)
  }
}
```

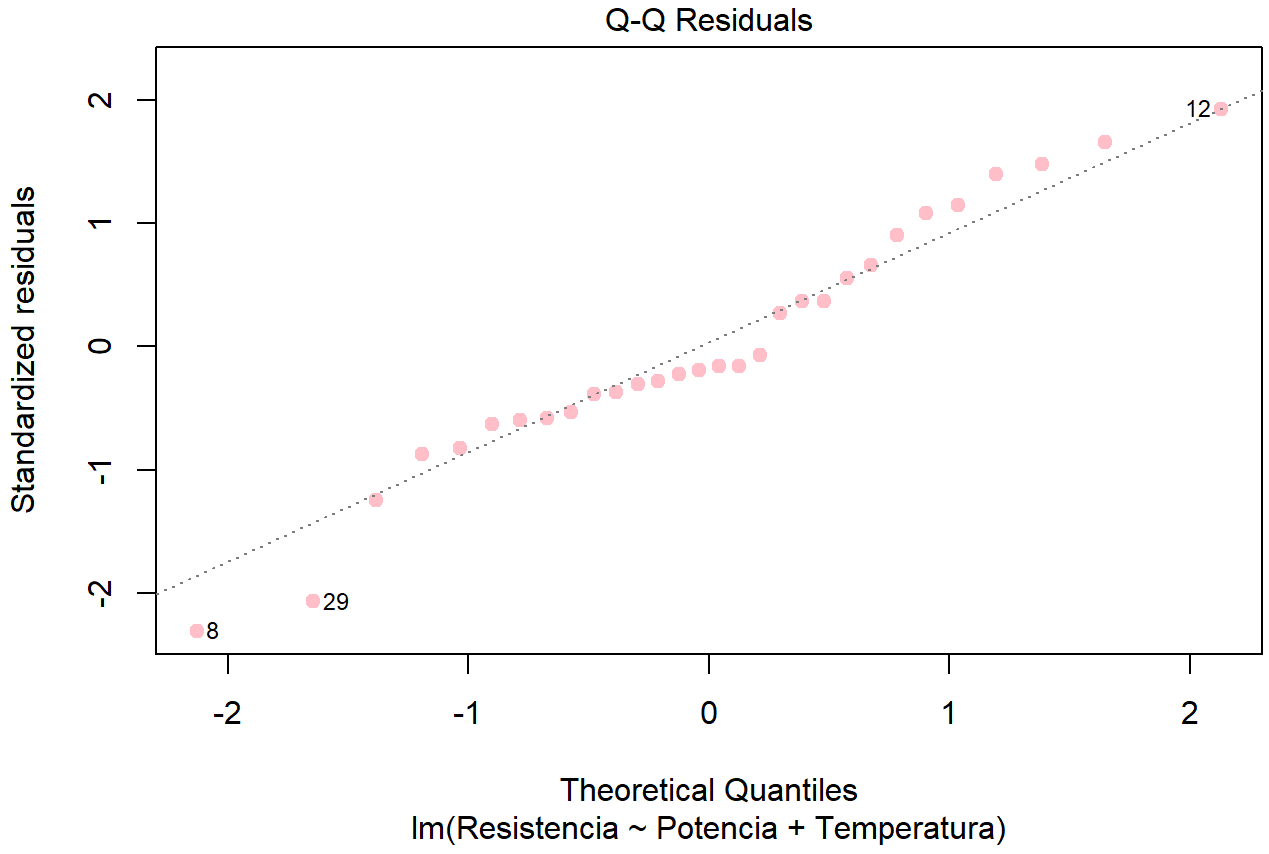

DfBetas para el coeficiente 1
DfBetas
-0.4
-0.2
0.0
0.2
0.4
0.6
0
5
10
15
20
25
30
Index

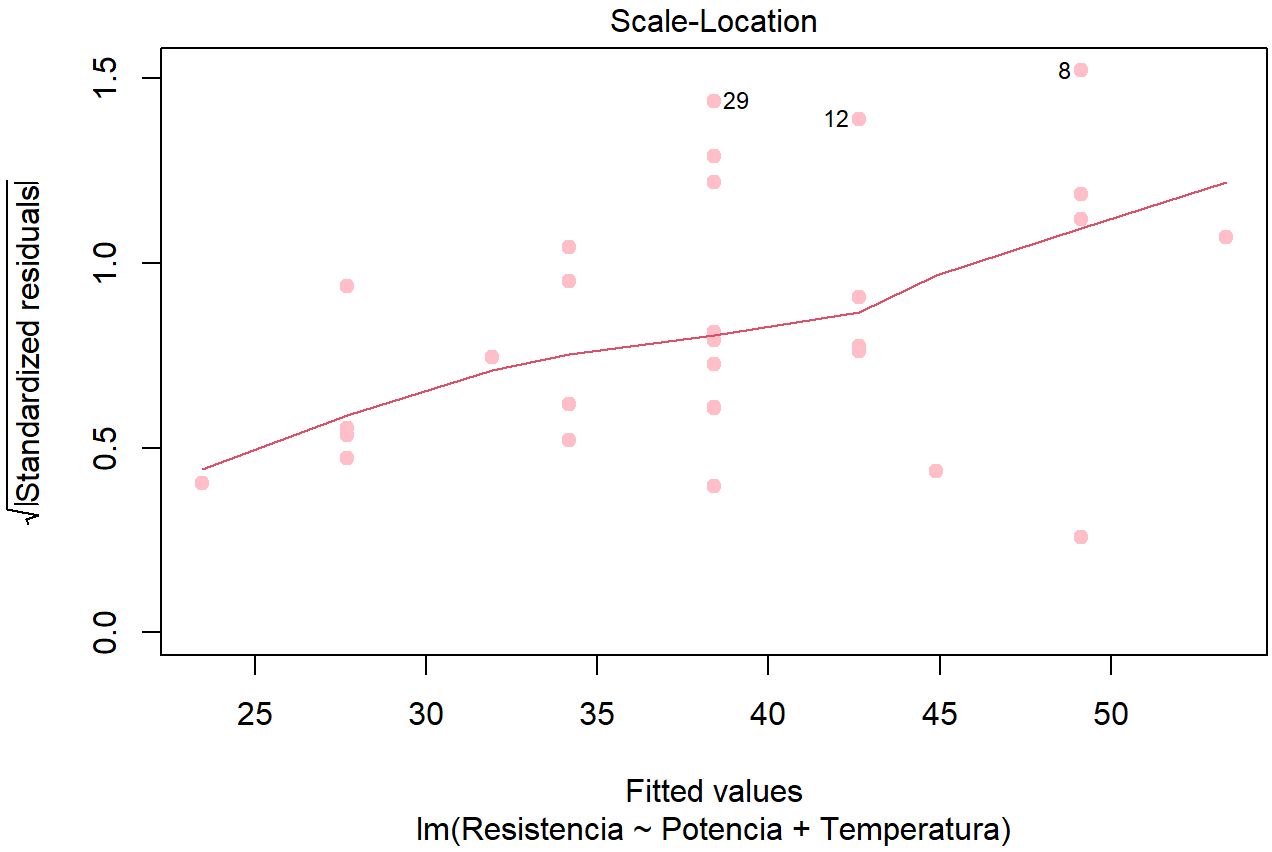

DfBetas para el coeficiente 2
DfBetas
-0.4
-0.2
0.0
0.2
0.4
0
5
10
15
20
25
30
Index

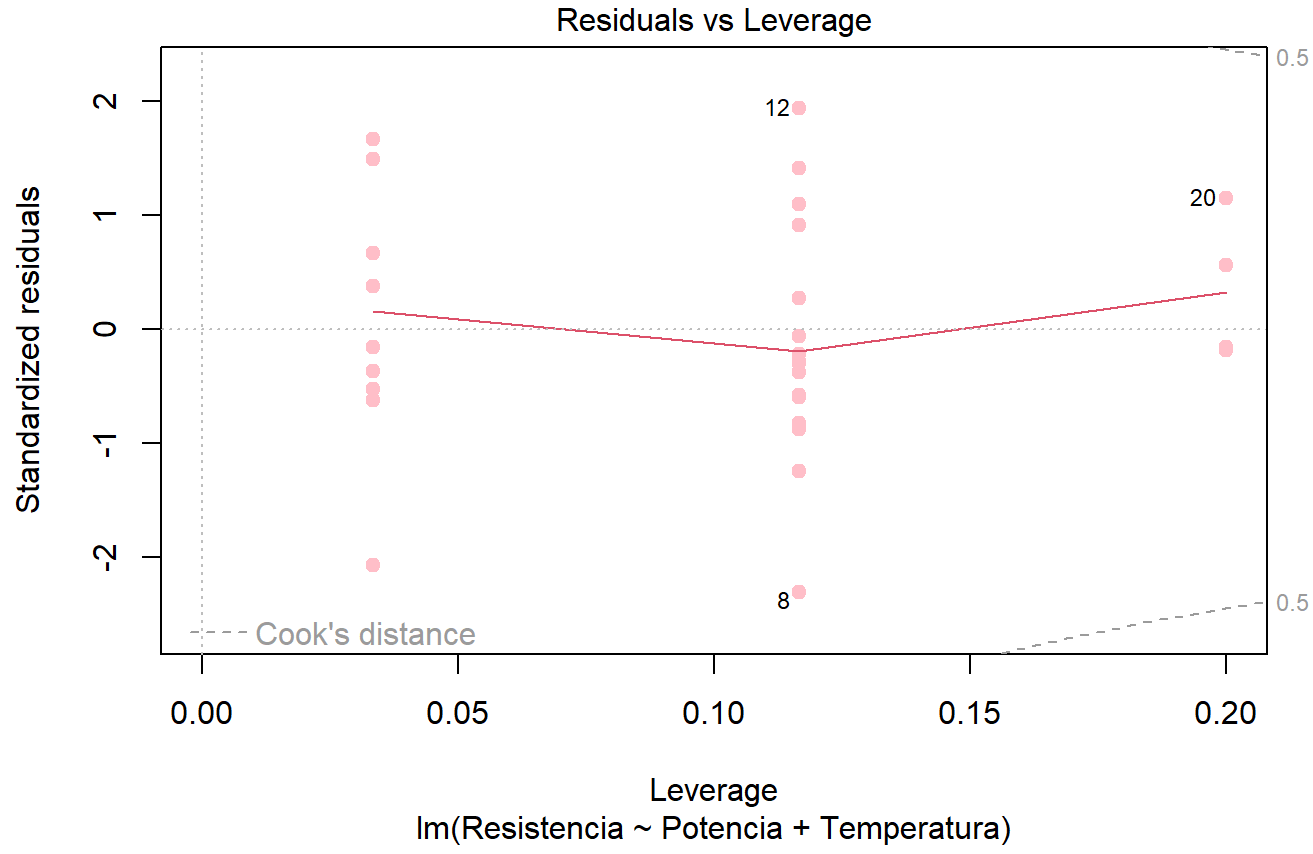



```
plot(modelo, col = "pink", pch = 19)
```


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)


Scale-Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
25
30
35
40
45
50
29
12
8
Fitted values
lm(Resistencia ~ Potencia + Temperatura)


Residuals vs Leverage
Standardized residuals
-2
-1
0
1
2
0.00
0.05
0.10
0.15
0.20
12
20
8
0.5
0.5
Cook's distance
Leverage
lm(Resistencia ~ Potencia + Temperatura)