A3. Regresión Múltiple - Detección de Datos Atípicos

Oscar Gutierrez

2024-09-24

Cargar los datos

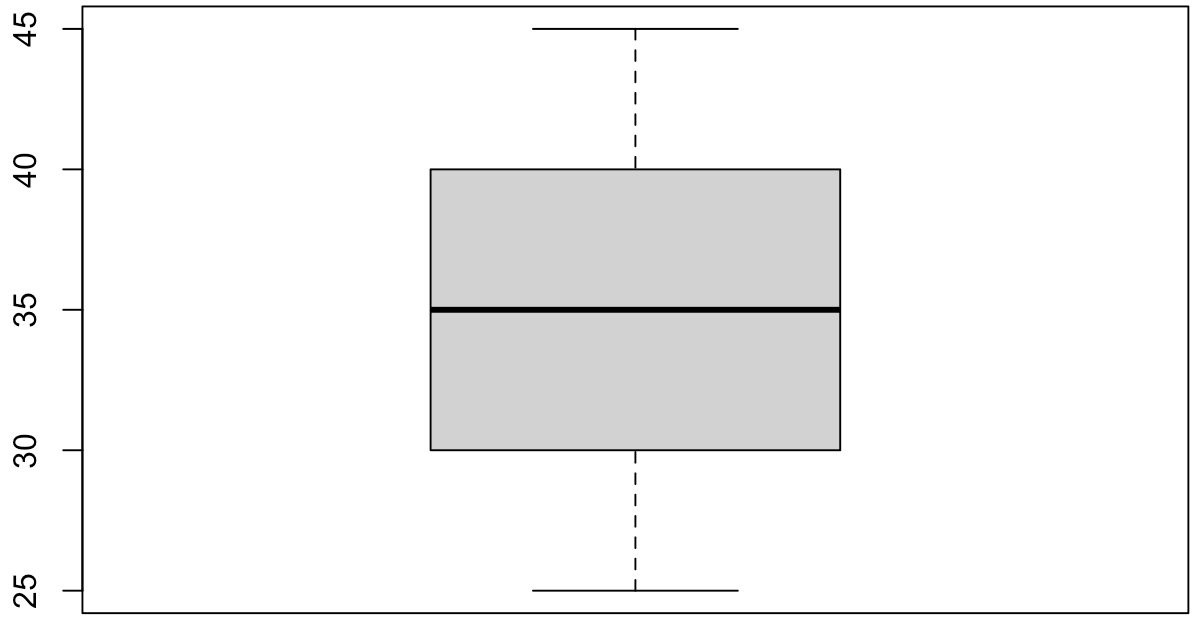
```
D = read.csv('AlCorte.csv')
```

Análisis descriptivo

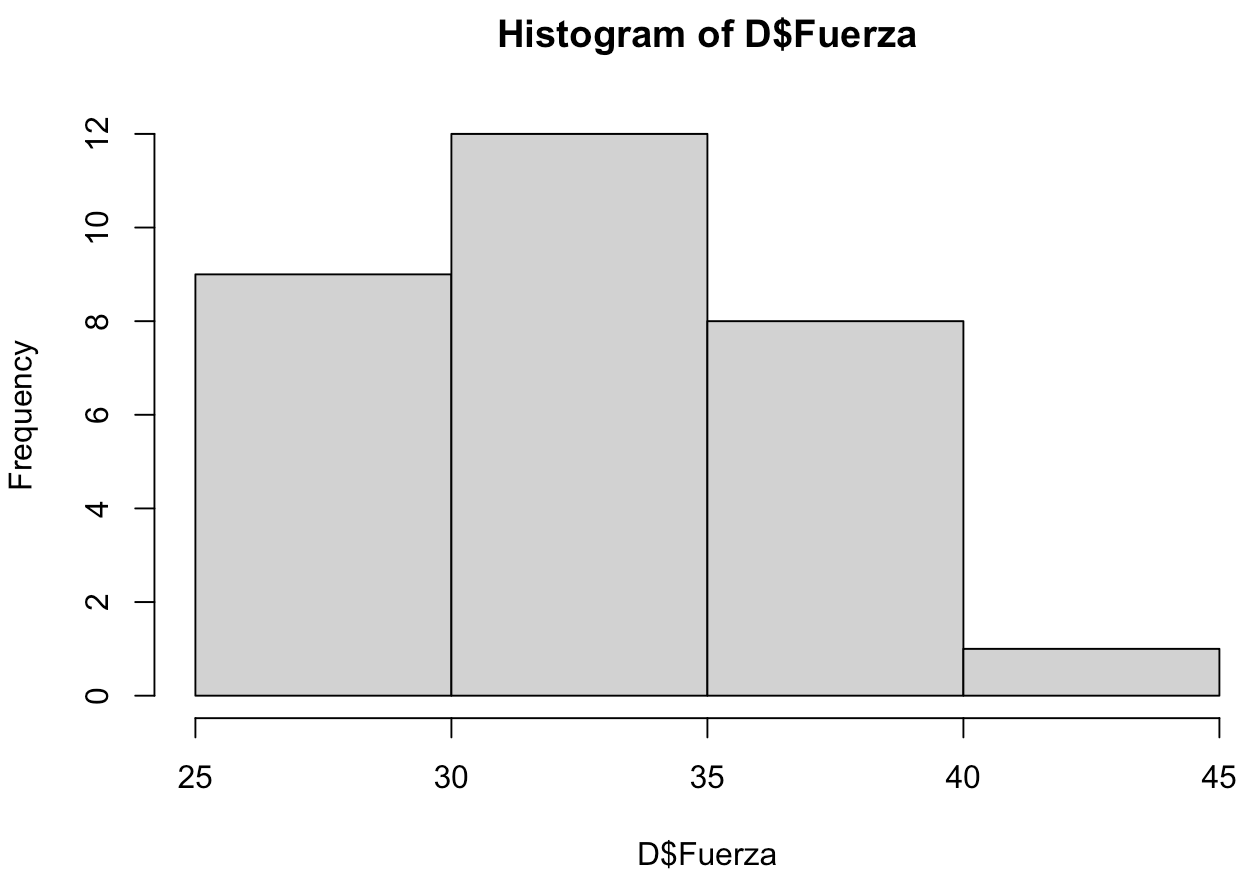
```
n=5 #número de variables

d=matrix(NA,ncol=7,nrow=n)
for(i in 1:n){
  d[i,]<-c(as.numeric(summary(D[,i])),sd(D[ ,i]))
}
m=as.data.frame(d)

row.names(m)=c('fuerza', 'potencia','temperatura', 'tiempo','resistencia')
names(m)=c("Minimo","Q1","Mediana","Media","Q3","Máximo","Desv Est")
m
```

```
##             Minimo      Q1 Mediana     Media    Q3 Máximo  Desv Est
## fuerza        25.0  30.000    35.0  35.00000  40.0   45.0  4.548588
## potencia      45.0  60.000    75.0  75.00000  90.0  105.0 13.645765
## temperatura  150.0 175.000   200.0 200.00000 225.0  250.0 22.742941
## tiempo        10.0  15.000    20.0  20.00000  25.0   30.0  4.548588
## resistencia   22.7  34.675    38.6  38.40667  42.7   58.7  8.954403
```

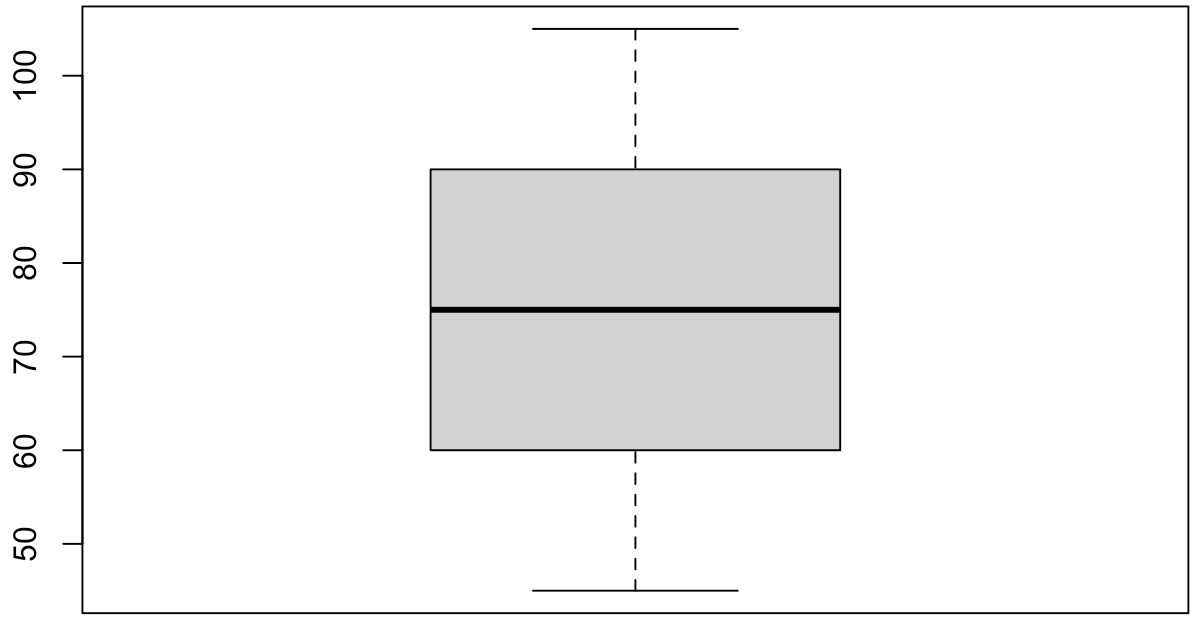
```
boxplot(D$Fuerza)
```



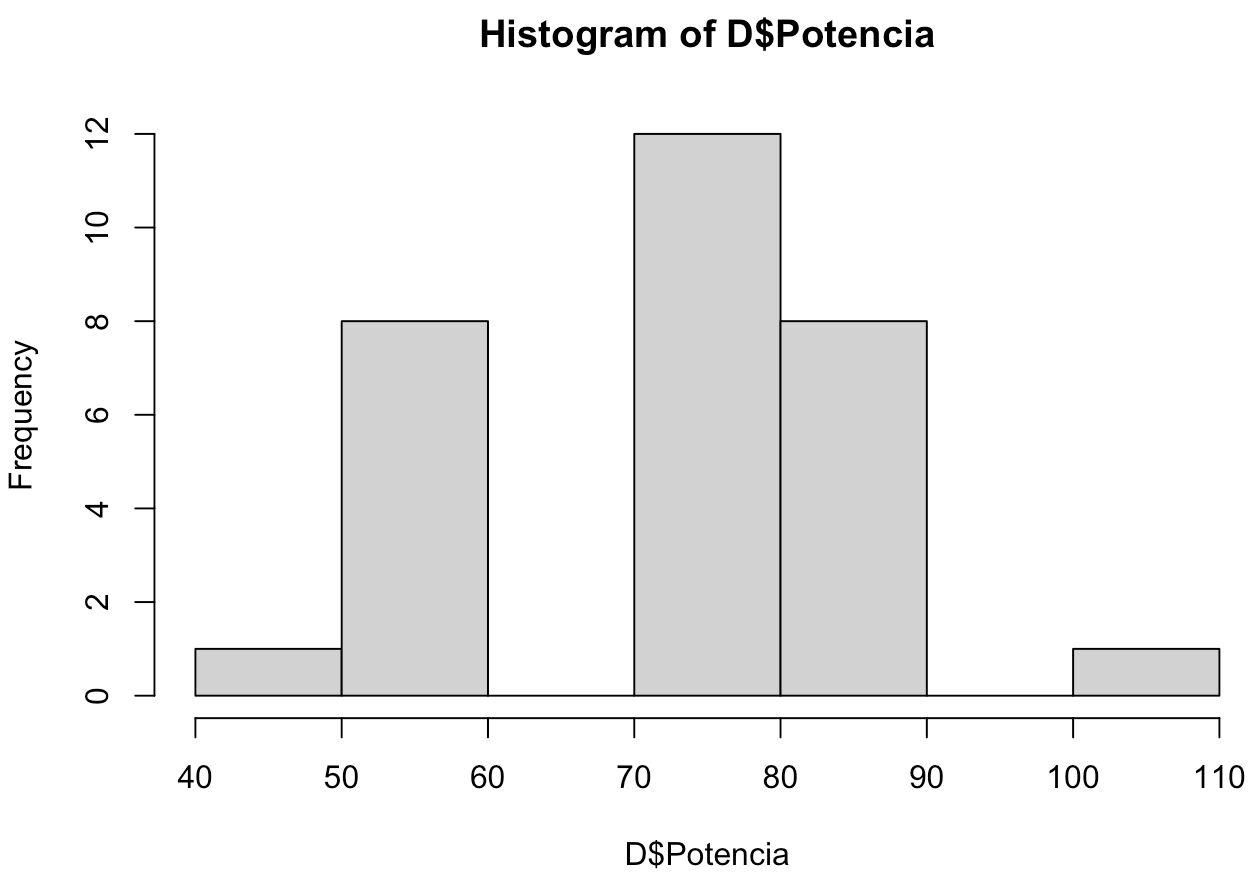

```
hist(D$Fuerza)
```



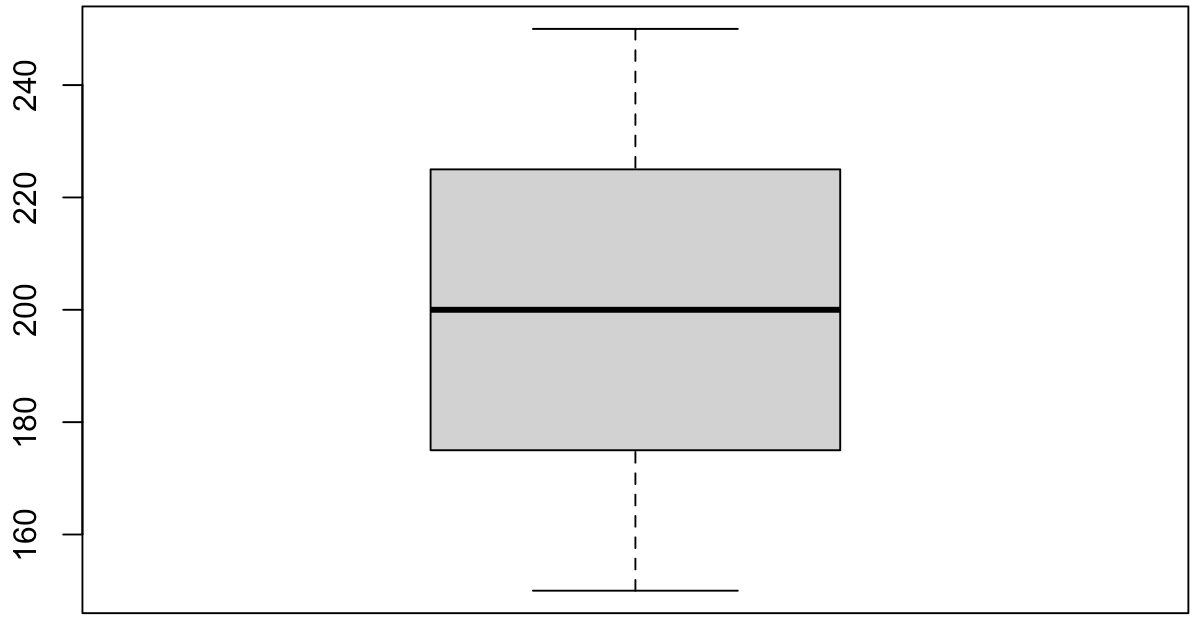

```
boxplot(D$Potencia)
```



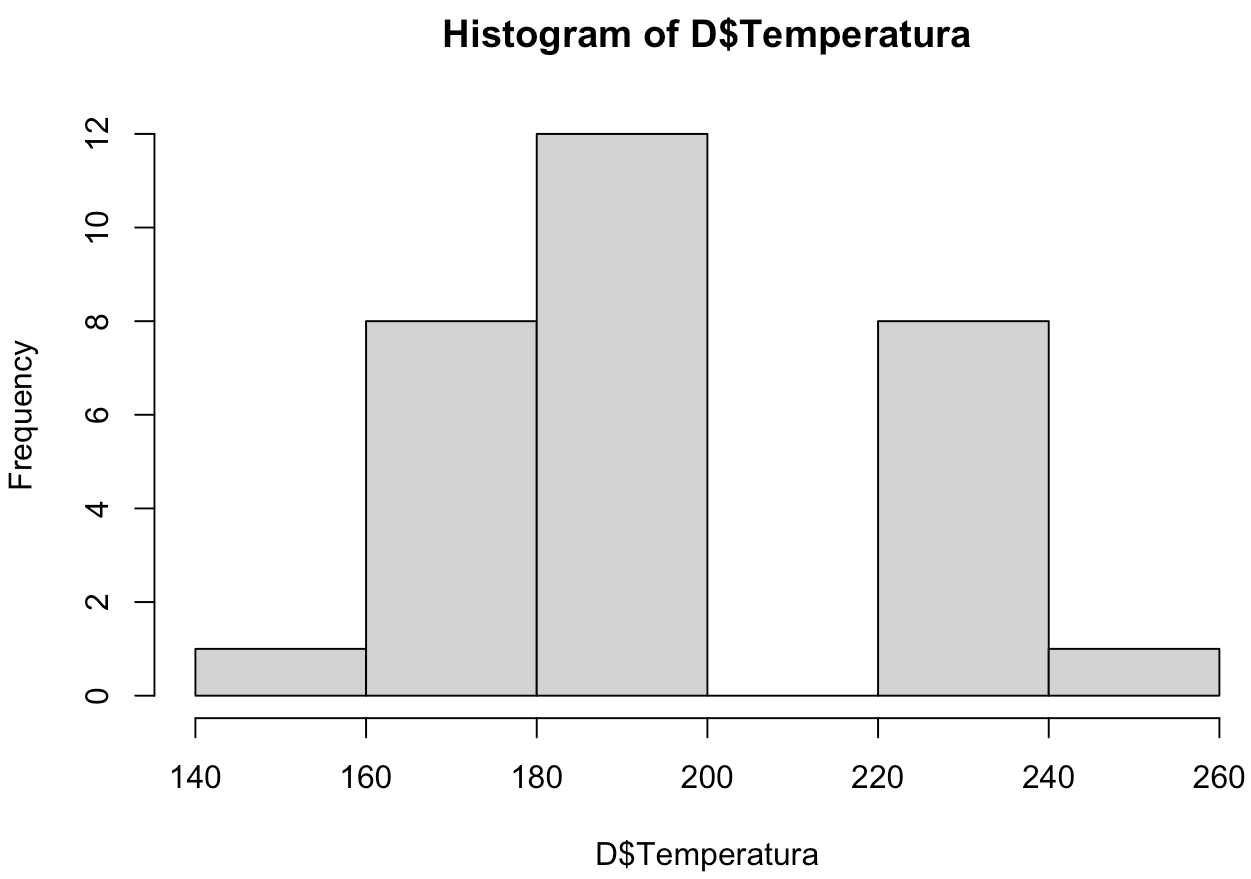

```
hist(D$Potencia)
```



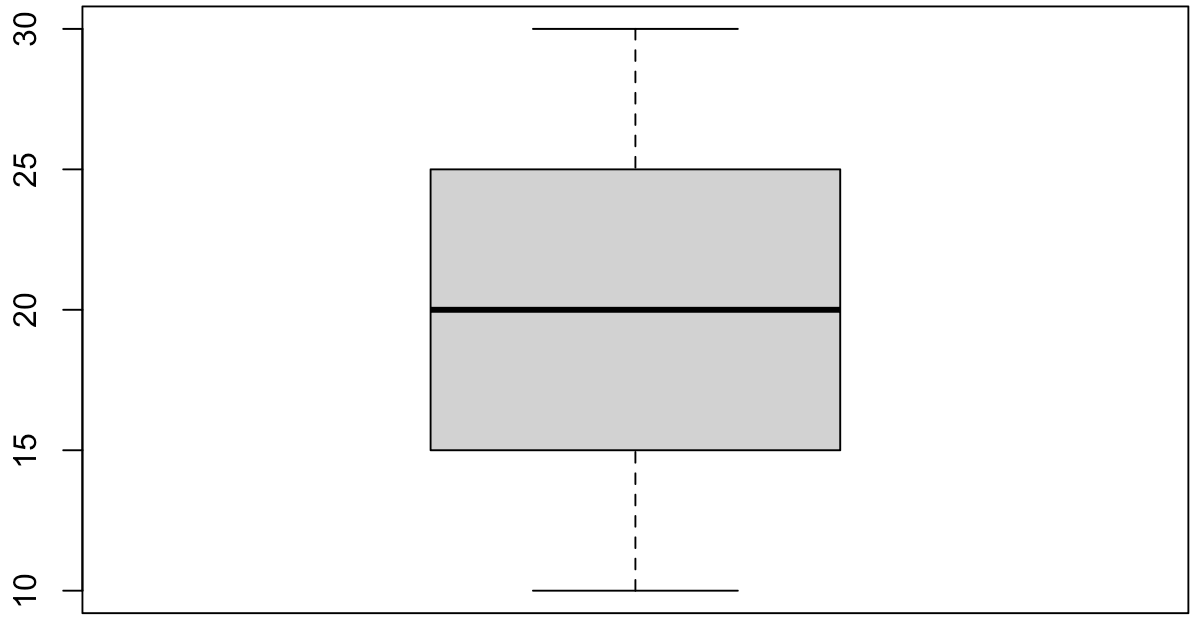

```
boxplot(D$Temperatura)
```



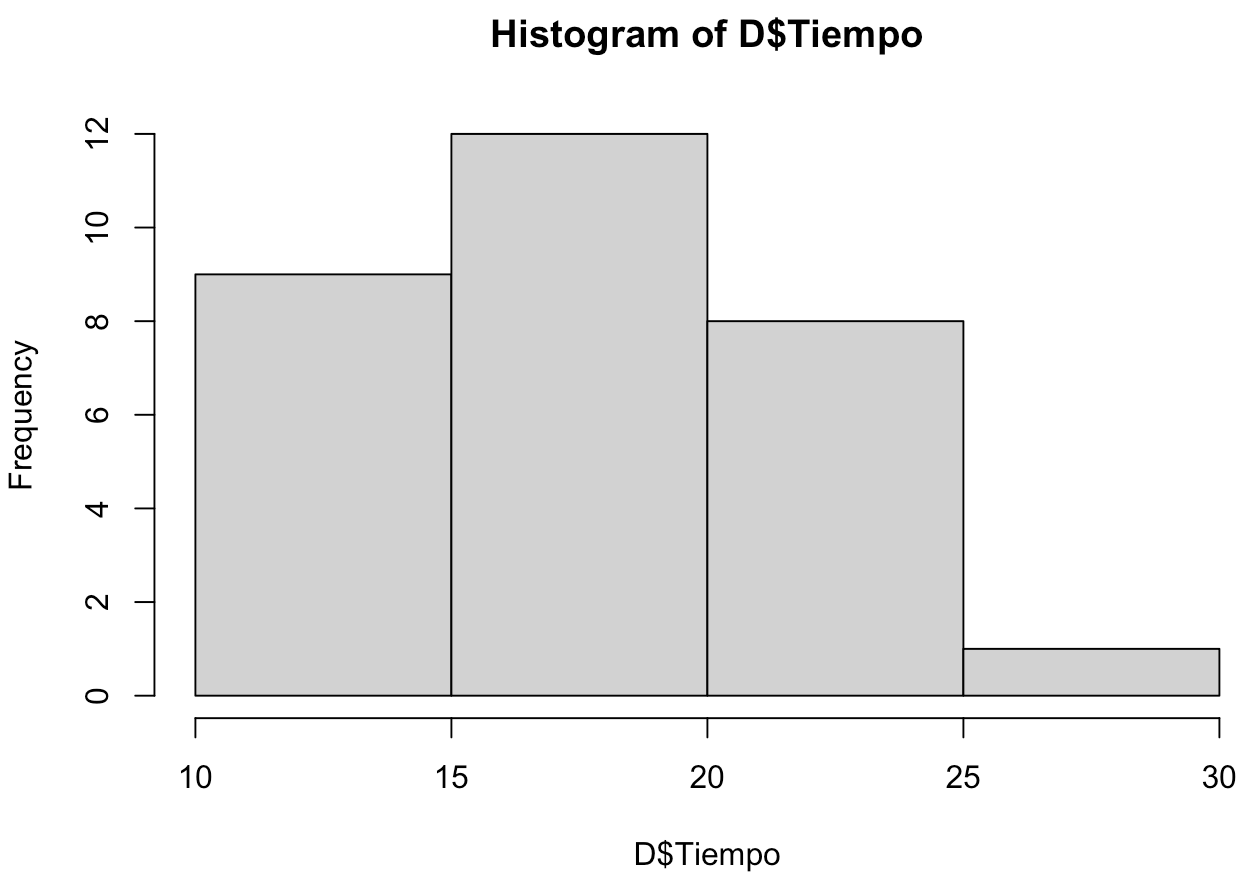

```
hist(D$Temperatura)
```



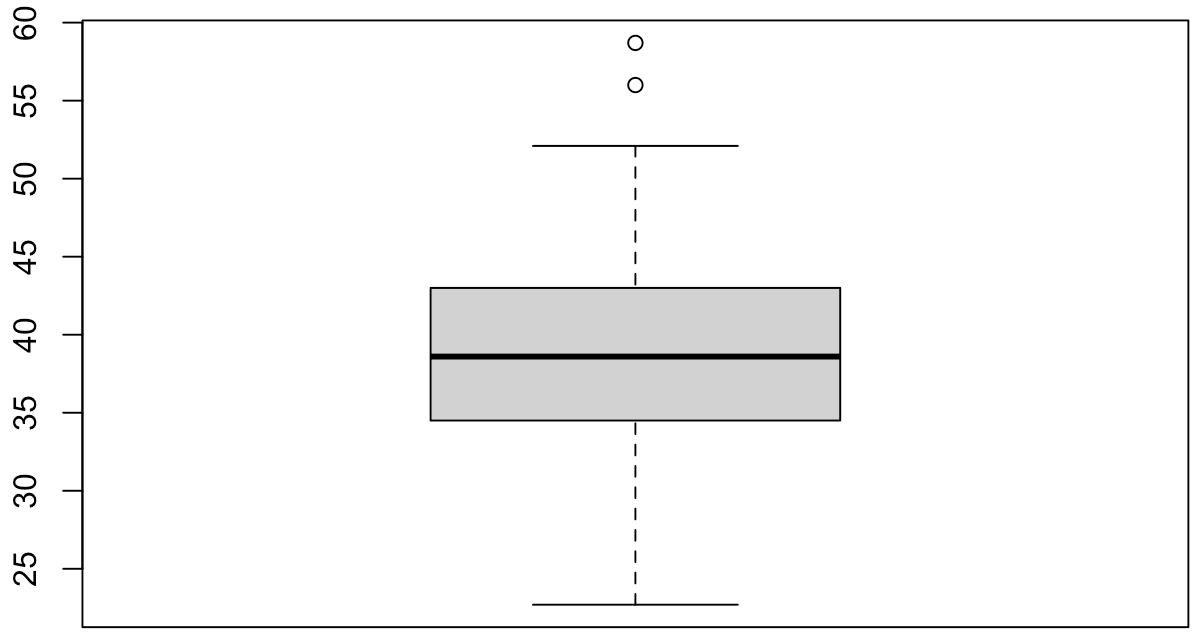

```
boxplot(D$Tiempo)
```



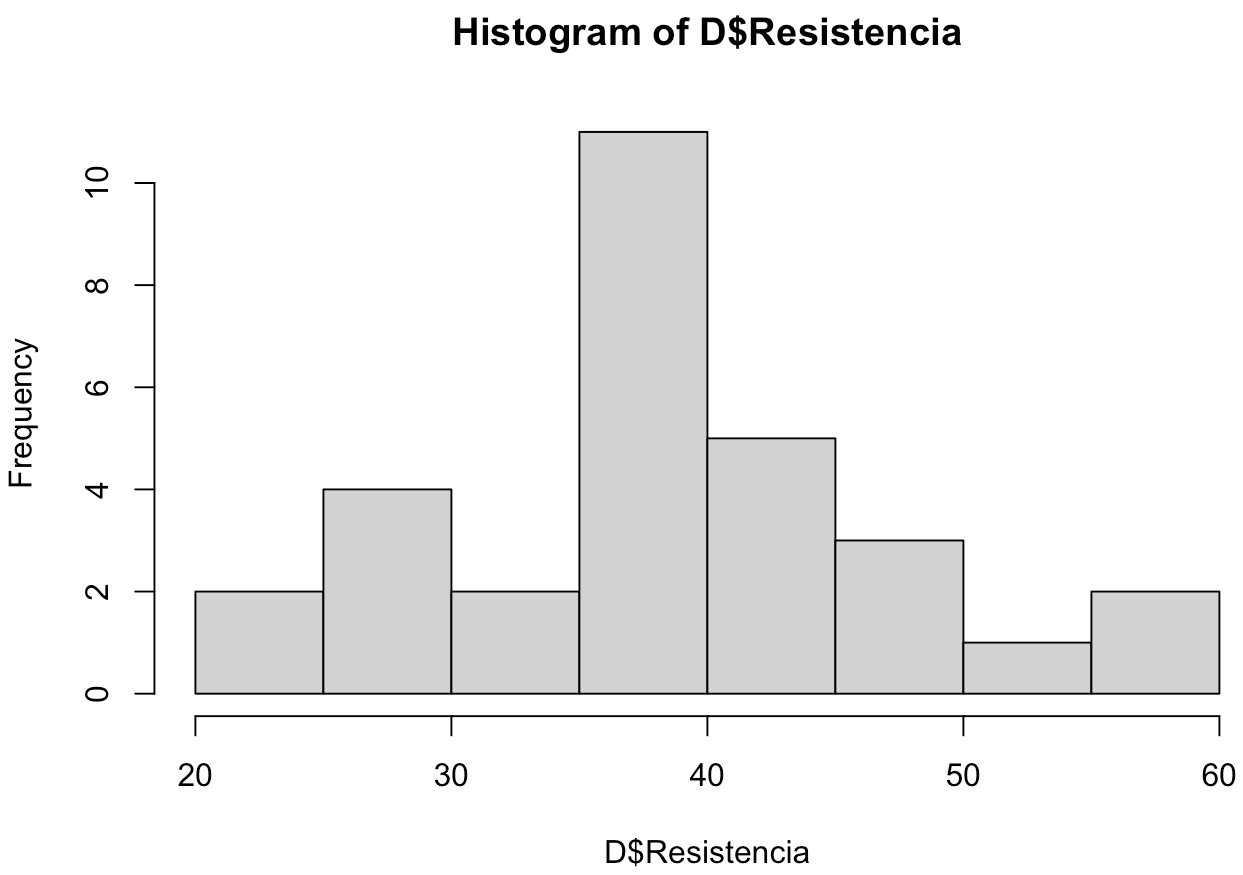

```
hist(D$Tiempo)
```




```
boxplot(D$Resistencia)
```




```
hist(D$Resistencia)
```



No se tienen distribuciones normales de los datos, solo hay 2 valores atípicos, estos se encuentran en la variable Resistencia.

Realizar modelos

Se realizan modelo utilizando las 3 técnicas, forward, backward y both.

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

```
Pasos1 = 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   1341.01 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(Pasos1)
```

```
##
## Call:
## lm(formula = Resistencia ~ Potencia + Temperatura, data = D)
##
## 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 = D)
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 = D)
##
## 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(D$Resistencia)
Pasos3 = step(Modelo, direction= 'both', trace = 1, 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   1341.01 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 = D)
##
## 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
```

Significancia del modelo

Los 3 procesos dieron como resultado el mismo modelo.

El modelo resultante utiliza solamente 2 de 4 variables, por lo que es un modelo más económico comparado con uno que utiliza las 4. Esto tiene beneficios puesto que reduce la carga computacional además de eliminar las variables que no son significativas.

Para analizar la significacncia de las variables y del modelo se realizan pruebas de hipótesis donde las hipótesis son las siguientes:

Hipótesis para las variables $H_0 : \beta_i = 0 \; H_1 : \beta_i \neq 0$

Con $\alpha = 0.05$

Hipótesis para el modelo
$$H_0 : \beta_1 = \beta_2 = \cdots = \beta_p = 0$$
$$H_1 : \text{Al menos un } \beta_i \neq 0 \text{ para alguna } i$$

De acuerdo con los resultados obtenidos por el modelo, las variables Potencia, Temperatura y el Intercept son significativos ya que el valor p es menor que el alpha seleccionado. Lo mismo puede decirse para el modelo completo. Por lo tanto, se tiene evidencia para rechazar las hipótesis nulas.

El modelo logra explicar una parte significativa de la variación en la variable de interés, obteniendo un coeficiente de determinación de 0.6619 lo que equivale a que logra explicar un 66.19% de la variación.

Validación del modelo

Normalidad de residuos

H_0 : Los residuos se distribuyen normalmente H_1 : Los residuos no se distribuyen normalmente

```
Modelo = Pasos1

library(nortest)
library(moments)
ad.test(Modelo$residuals)
```

```
##
##  Anderson-Darling normality test
##
## data:  Modelo$residuals
## A = 0.41149, p-value = 0.3204
```

```
jarque.test(Modelo$residuals)
```

```
##
##  Jarque-Bera Normality Test
##
## data:  Modelo$residuals
## JB = 0.085643, p-value = 0.9581
## alternative hypothesis: greater
```

De acuerdo con las pruebas de hipótesis, los residuos se distribuyen normalmente considerando un alpha de 0.05.

Comprobar media = 0

$H_0 : \mu = 0 \, H_1 : \mu \neq 0$

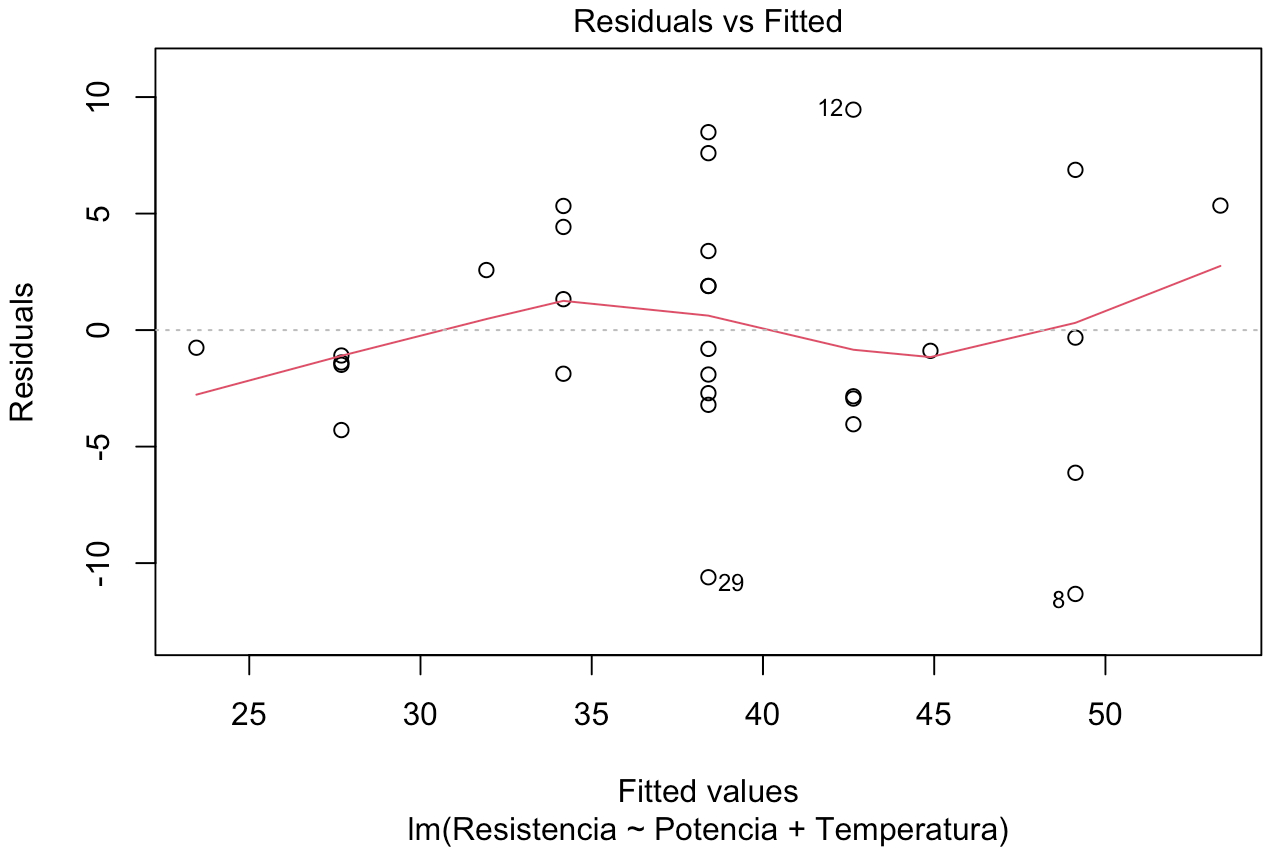
```
t.test(Modelo$residuals)
```

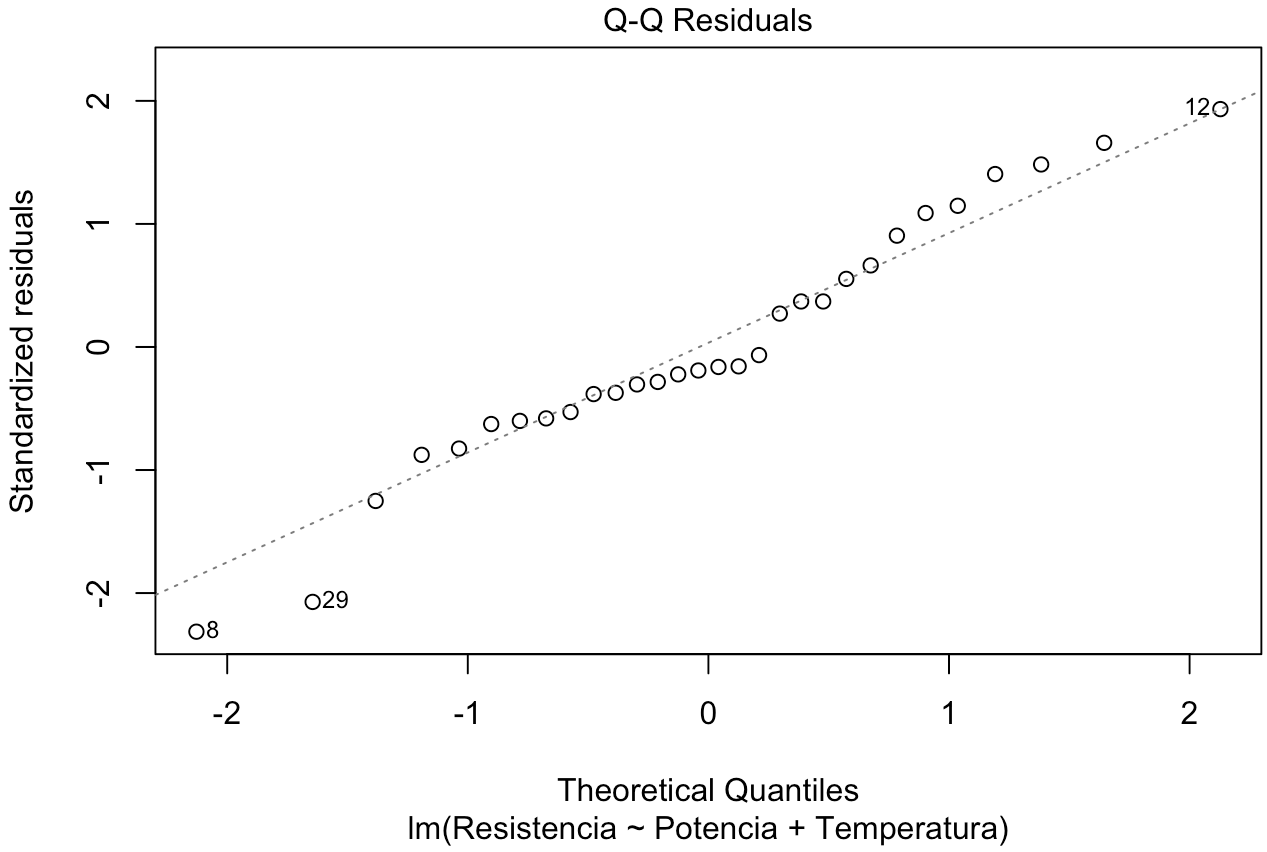
```
##
##  One Sample t-test
##
## data:  Modelo$residuals
## t = 2.6627e-16, df = 29, p-value = 1
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -1.876076  1.876076
## sample estimates:
##    mean of x
## 2.442491e-16
```

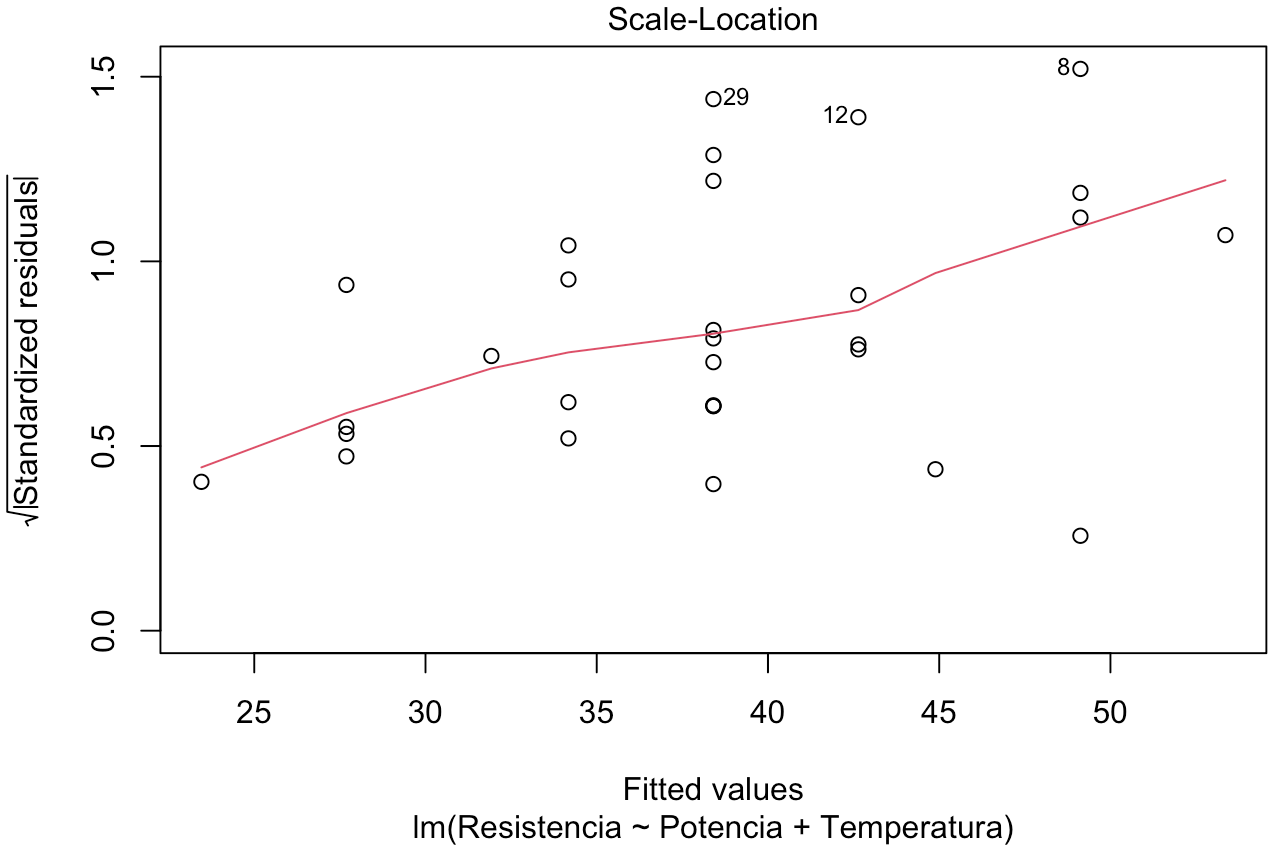
De acuerdo con la prueba de hipótesis, los residuos sí cuentan con una media de 0.

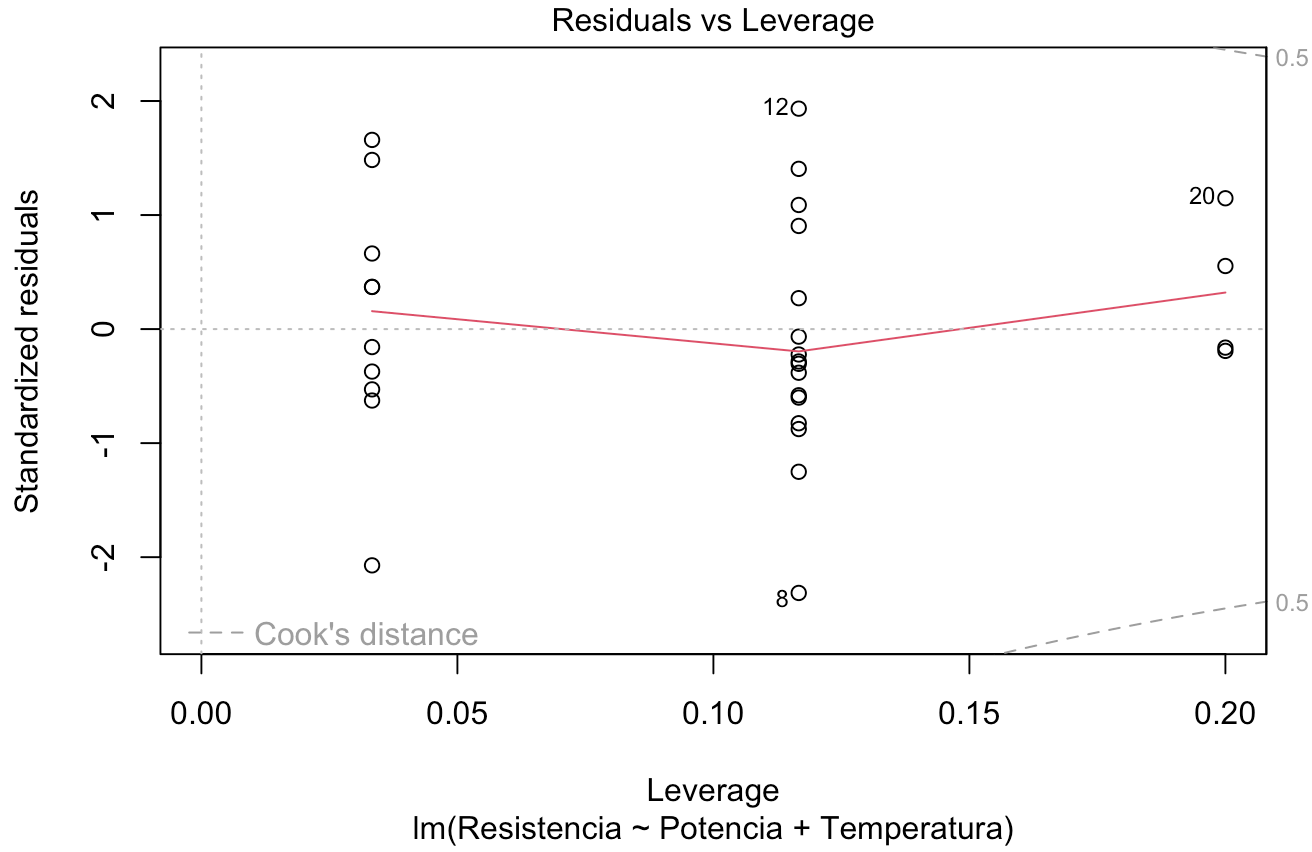
Homocedasticidad e independencia

```
plot(Modelo)
```


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
8
29
12
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)

En las gráficas de los errores, parece haber alguna relación que no es explicada por el modelo.

Prueba de independencia

H_0: Los errores no están autocorrelacionados. H_1: Los errores están autocorrelacionados.

```
library(lmtest)
```

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

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

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

```
dwtest(Modelo)
```

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

```
bgtest(Modelo)
```

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

De acuerdo con las pruebas de hipótesis, los errores no están autocorrelacionados.

Prueba de homocedasticidad

H_0: La varianza de los errores es constante (homocedasticidad) H_1: La varianza de los errores no es constante (heterocedasticidad)

```
gqtest(Modelo)
```

```
##
##  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
```

De acuerdo con la prueba de hipótesis, la varianza en los errores es constante, hay homocedasticidad.

Multicolinealidad

```
library(car)
```

```
## Loading required package: carData
```

```
vif(Modelo)
```

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

De acuerdo con el Variance Inflation Factor (VIF), no hay multicolinealidad entre variables.

Conclusiones

El modelo obtenido logra explicar una cantidad considerable de la variación en la variable de interés, aproximadamente un 66.19%. Además, el modelo cumple con todas las suposiciones de la regresión lineal ya que cuenta con una distribución normal de residuos, varianza constante, homocedasticidad e independencia.

El modelo final obtenido fue $resistencia = -24.9 + 0.498 * potencia + 0.12 * temperatura$. En el contexto del problema esto significa que por cada incremento de una unidad en la potencia, se aumenta la resistencia por 0.498, mientras que por cada incremento de una unidad en la temperatura se aumenta por 0.12. Como se puede observar en la ecuación, mientras más potencia y mas temperatura se tiene, la resistencia será mayor ya que ambos coeficientes (pendientes) son positivas. El intercept no tiene un significado en sí dentro del contexto del problema, simplemente es la intersección de la recta con el eje y.

Datos atípicos e influyentes

```
library(dplyr)
```

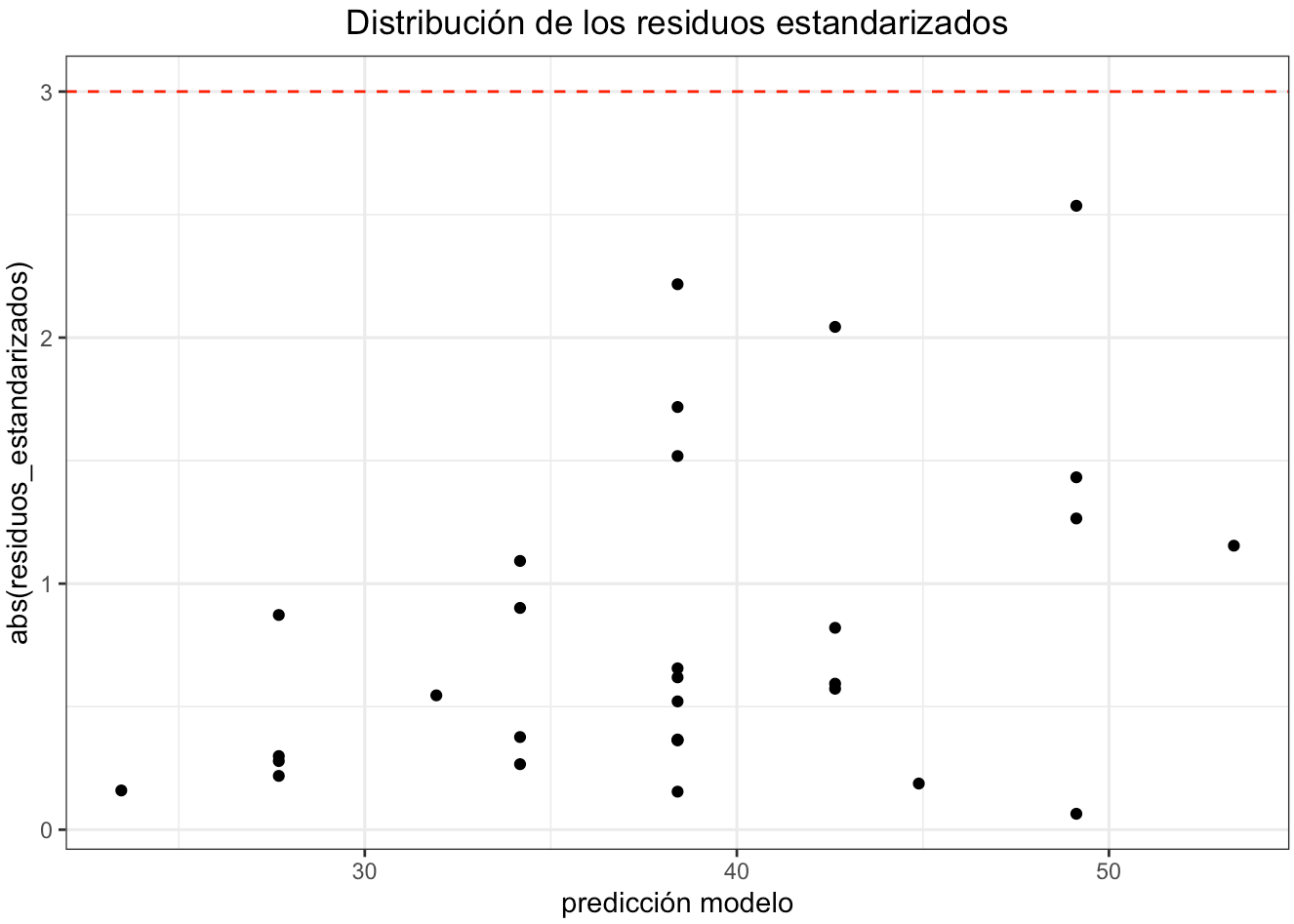
```
##
## Attaching package: 'dplyr'
```

```
## 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)
D$residuos_estandarizados <- rstudent(Modelo)
#Introduce una columna en Datos con los residuos estandarizados de los n datos

ggplot(data = D, aes(x = predict(Modelo), 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))
```

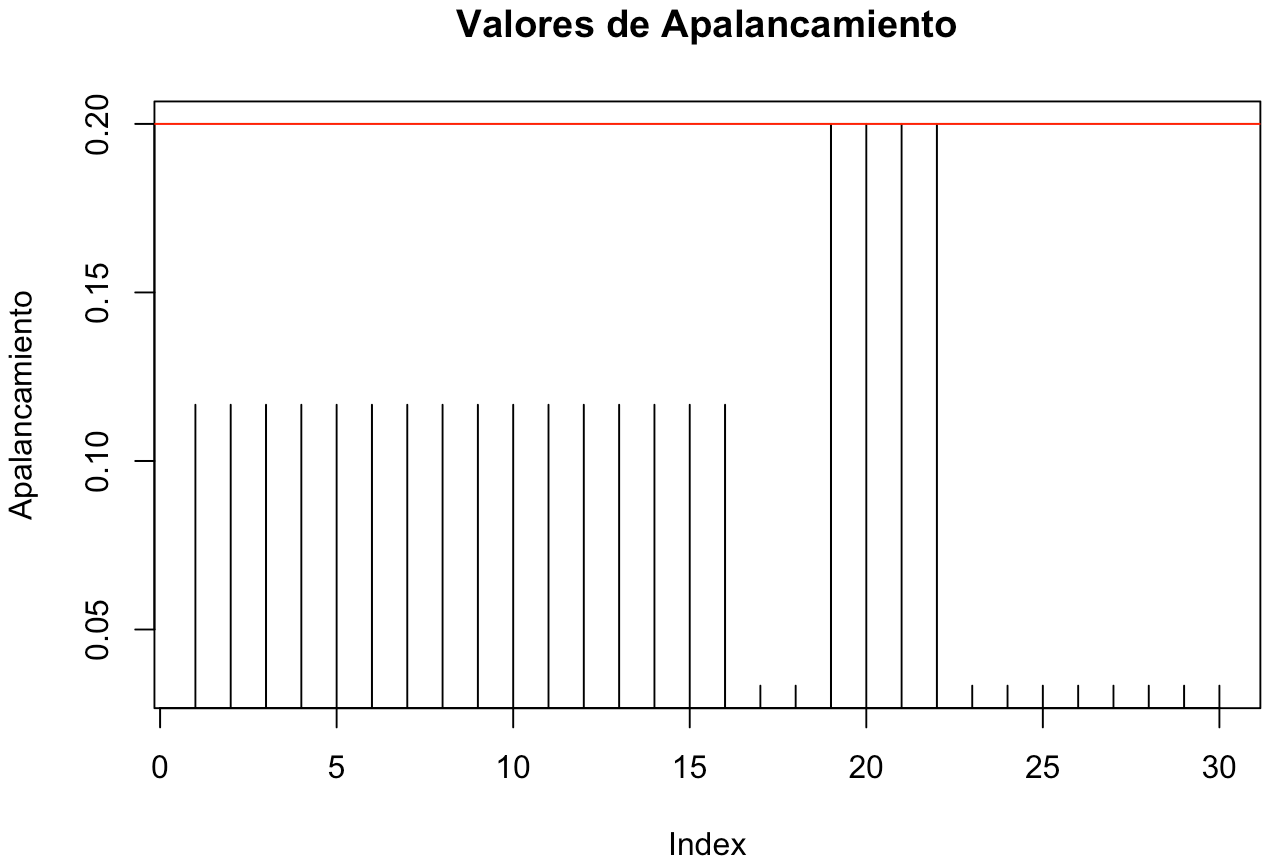



```
Atipicos = which(abs(D$residuos_estandarizados)>3)

D[Atipicos, ]
```

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

De acuerdo con la gráfica de residuos estandarizados, no hay datos atípicos debido a que ninguno de los datos rebasa 3 residuos estandarizados. Sin embargo, esto no significa que no haya datos influyentes.

```
leverage = hatvalues(Modelo)
#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
```

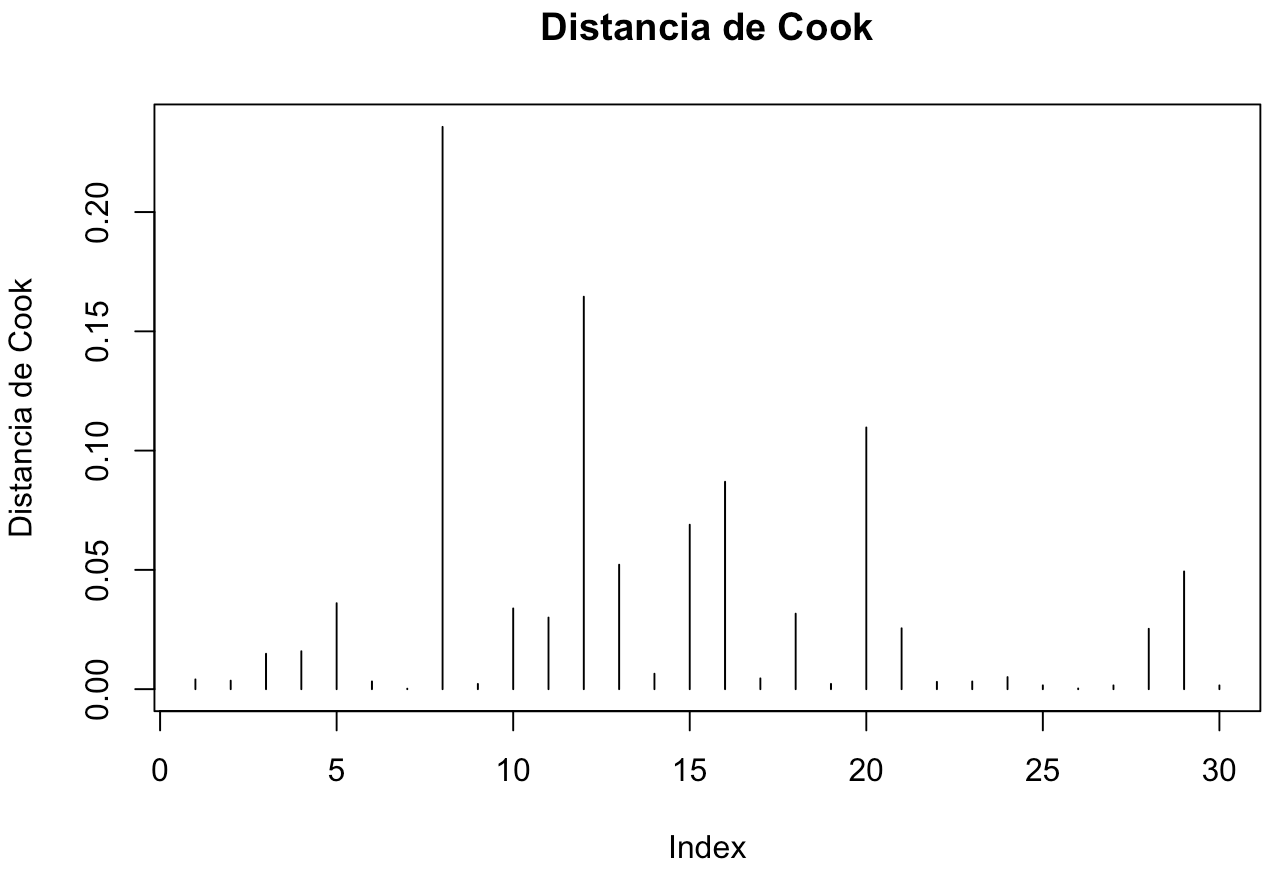



```
high_leverage_points = which(leverage > 2*mean(leverage))

D[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
```

Se puede observar que solamente hay dos puntos que tienen un leverage alto, considerando alto un leverage que es mayor a 2 veces el promedio de leverage. Los puntos con leverage alto son el 19 y el 20.

```
cooksdistance <- cooks.distance(Modelo)
#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
```

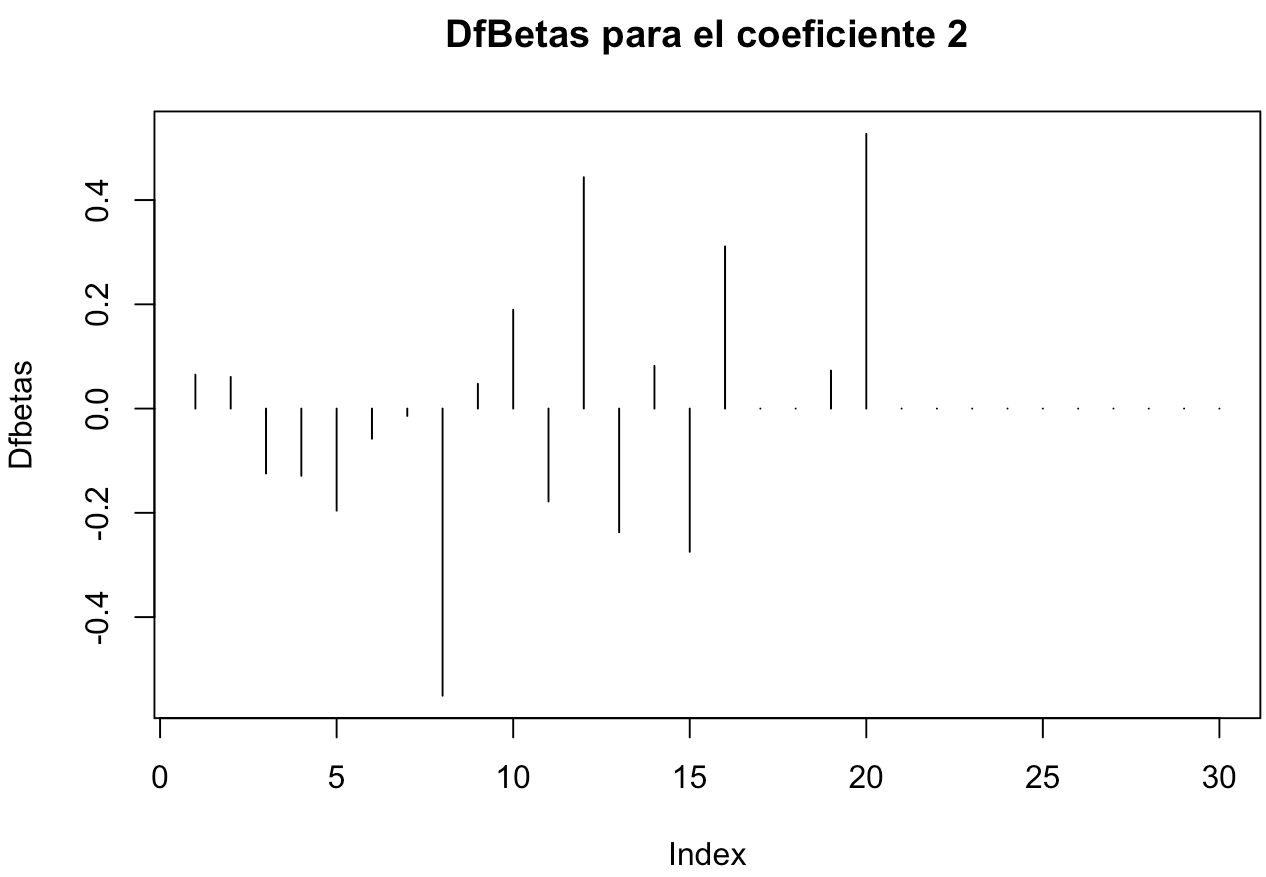



```
puntos_influyentes = which(cooksdistance > 1)

D[puntos_influyentes, ]
```

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

Considerando la distancia de Cook, no hay ningún dato influyente.

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

plot(dfbetas_values[, 2], type="h", main="DfBetas para el coeficiente 2", ylab='Dfbeta
s')

abline(h = c(-1, 1), col="red") # Límites comunes
```




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

D[puntos_influyentes, ]
```

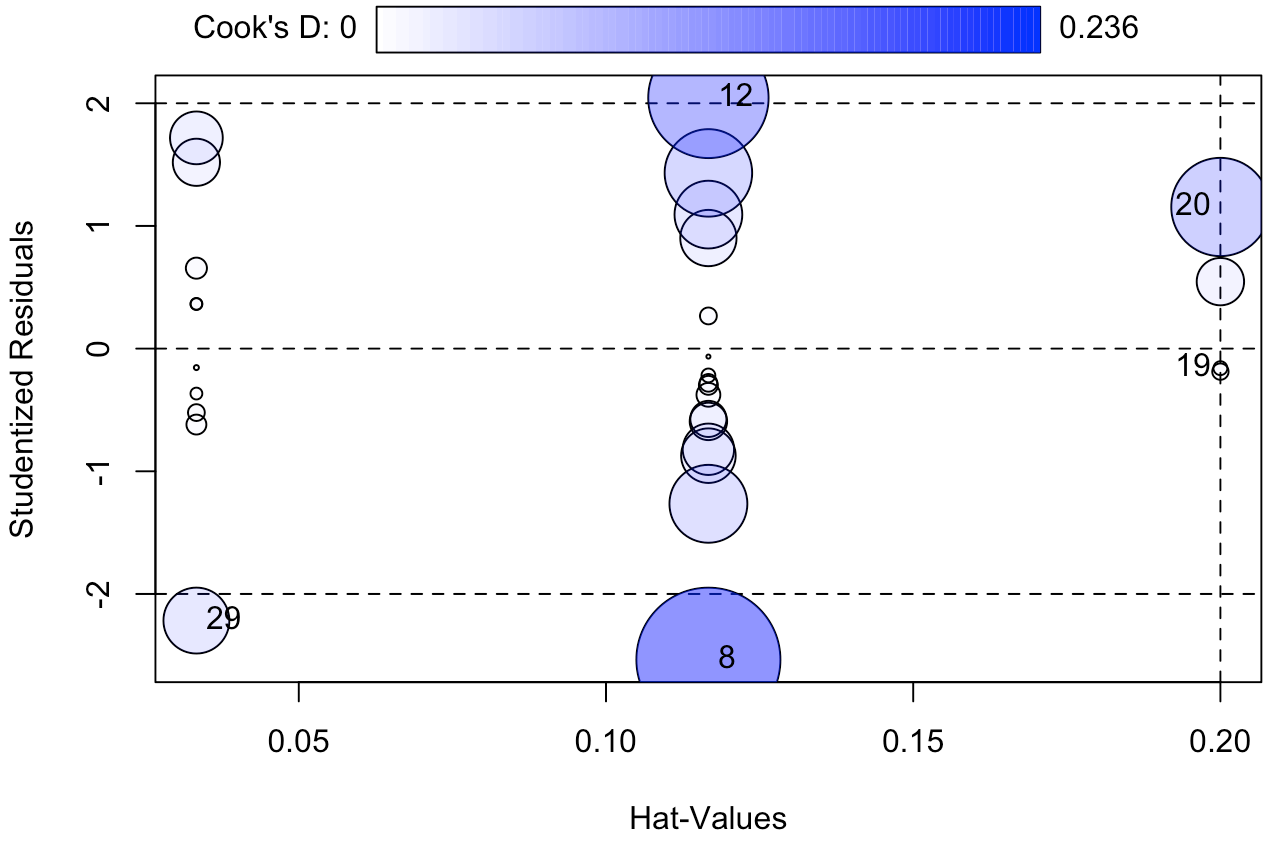
```
## [1] Fuerza                  Potencia                Temperatura
## [4] Tiempo                  Resistencia             residuos_estandarizados
## <0 rows> (or 0-length row.names)
```

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

summary(influencia)
```

```
## Potentially influential observations of
##   lm(formula = Resistencia ~ Potencia + Temperatura, data = D) :
##
##    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
```

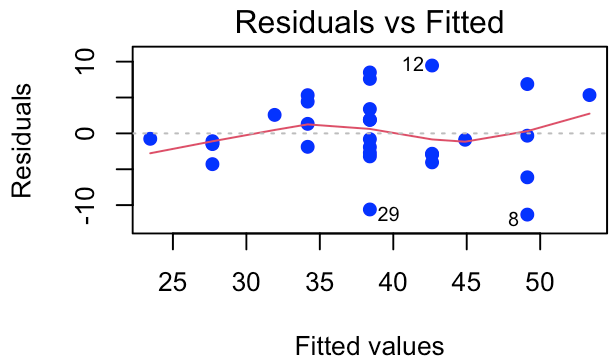
```
library(car)
influencePlot(Modelo)
```

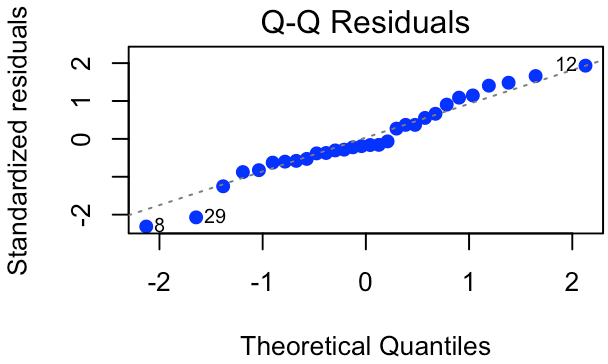


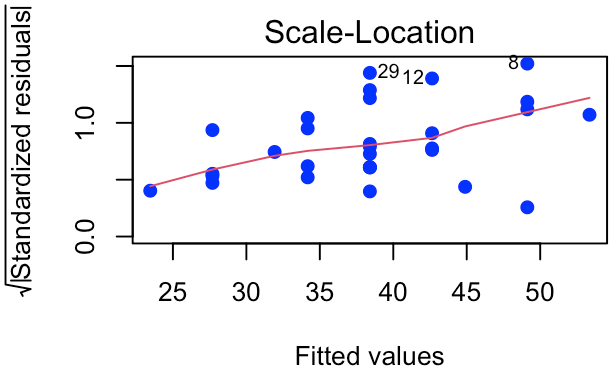

```
##      StudRes        Hat       CookD
## 8  -2.535832 0.11666667 0.235696235
## 12  2.043589 0.11666667 0.164507739
## 19 -0.159511 0.20000000 0.002199712
## 20  1.154355 0.20000000 0.109693544
## 29 -2.216952 0.03333333 0.049338917
```

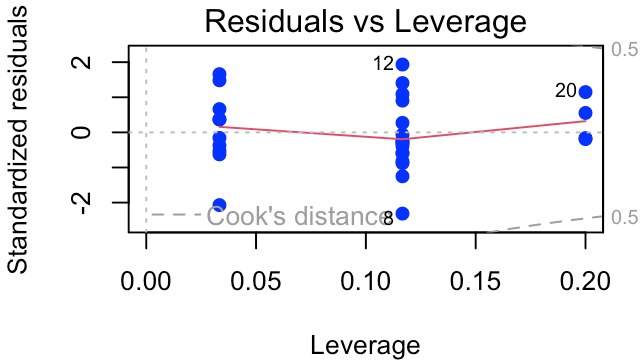
```
# grafica los residuos con estandarización extrema, el laverage y la distancia de cook
```

```
par(mfrow=c(2, 2))
plot(Modelo, col='blue', pch=19)
```


Residuals vs Fitted
Residuals
Fitted values
12
29
8


Q-Q Residuals
Standardized residuals
Theoretical Quantiles
12
29
8


Scale-Location
√|Standardized residuals|
Fitted values
29
12
8


Residuals vs Leverage
Standardized residuals
Leverage
Cook's distance
12
20
8