

# RegMultiple

Facundo Colasurdo Caldironi

2024-09-17

```
M=read.csv("file:///Users/facundocolasurdocaldironi/Downloads/AlCorte.csv") #leer la base de datos
head(M)
```

```
##   Fuerza Potencia Temperatura Tiempo Resistencia
## 1     30      60         175      15         26.2
## 2     40      60         175      15         26.3
## 3     30      90         175      15         39.8
## 4     40      90         175      15         39.7
## 5     30      60         225      15         38.6
## 6     40      60         225      15         35.5
```

## Haz un análisis descriptivo de los datos: medidas principales y gráficos caja bigotes de cada variable y la desviación estándar

```
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
```

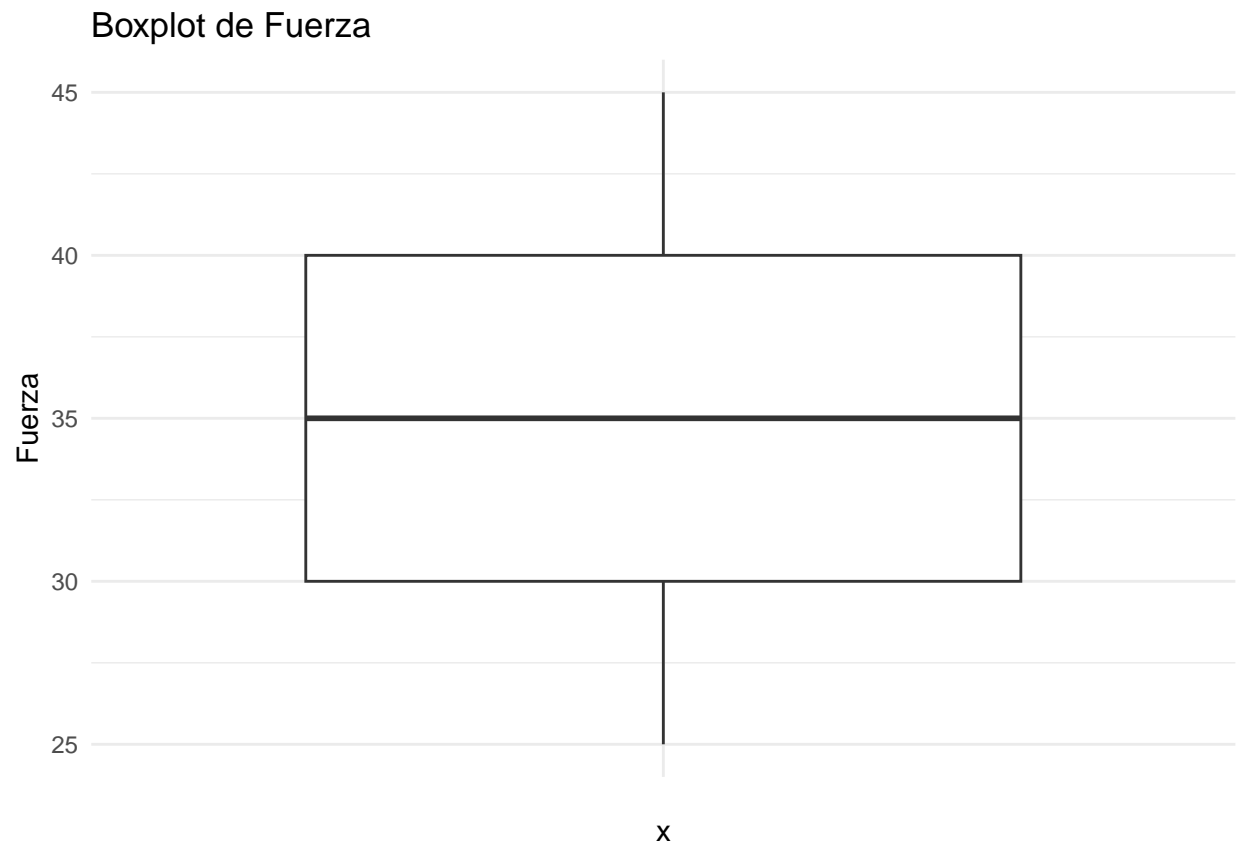
```
sapply(M, sd, na.rm = TRUE)
```

```
##      Fuerza      Potencia      Temperatura      Tiempo      Resistencia
## 4.548588   13.645765   22.742941    4.548588    8.954403
```

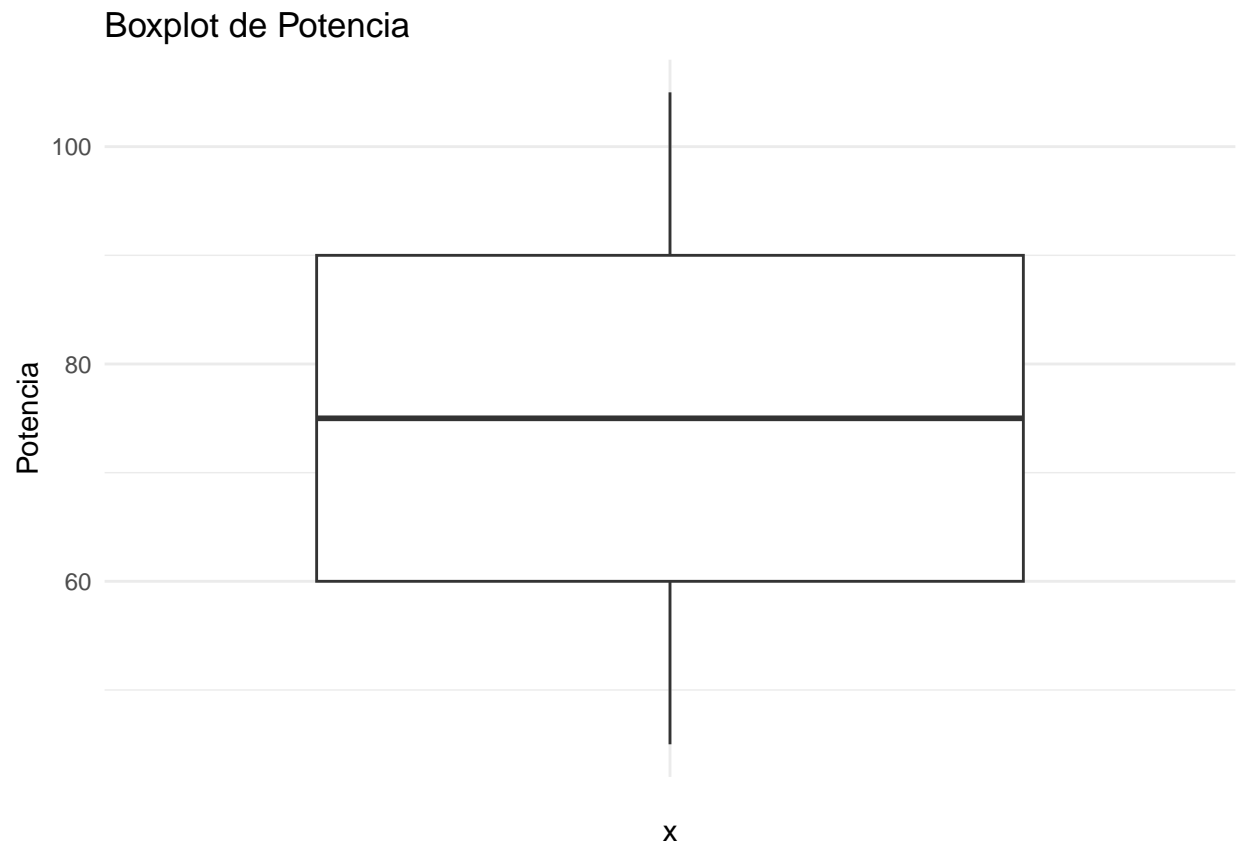
```
library(ggplot2)
```

```
library(ggplot2)
```

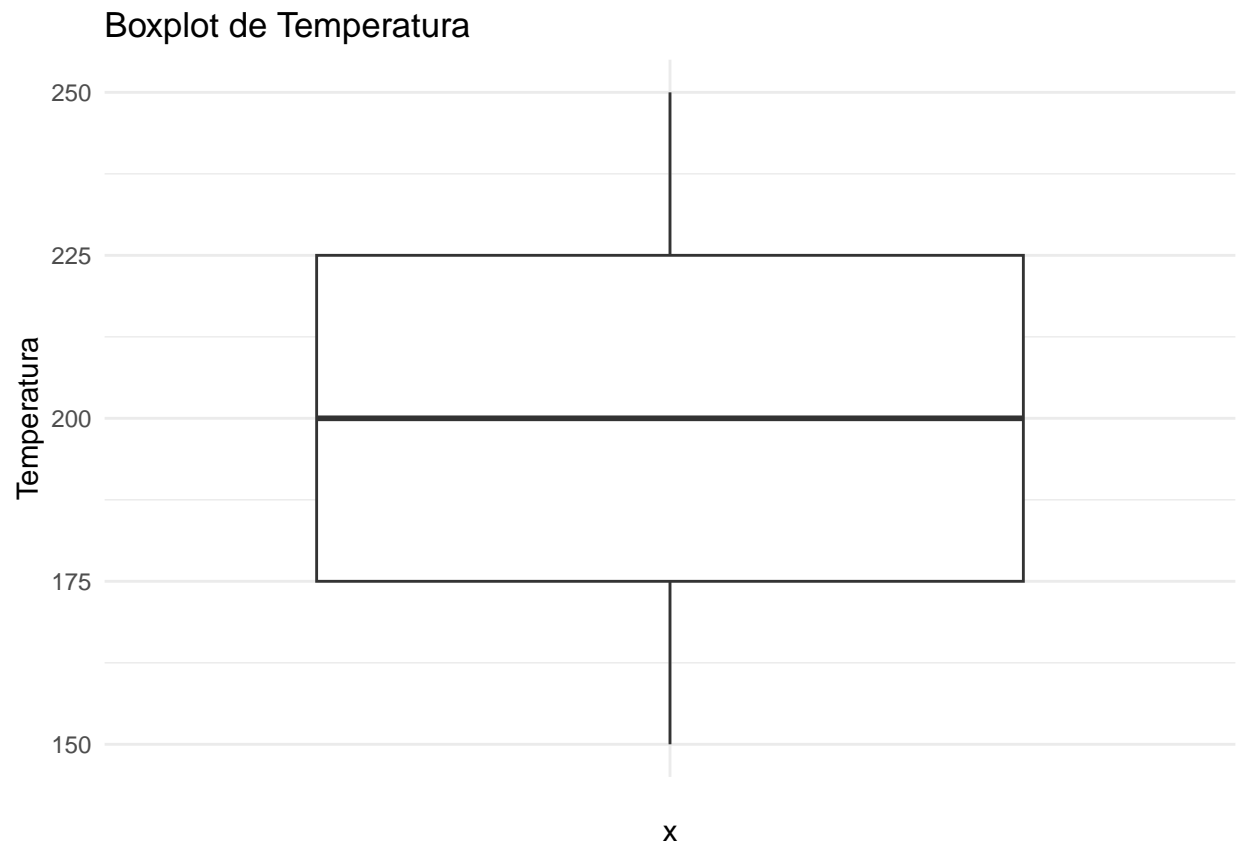
```
# Crear boxplots individuales para cada variable
ggplot(M, aes(x = "", y = Fuerza)) +
  geom_boxplot() +
  labs(title = "Boxplot de Fuerza", y = "Fuerza") +
  theme_minimal()
```



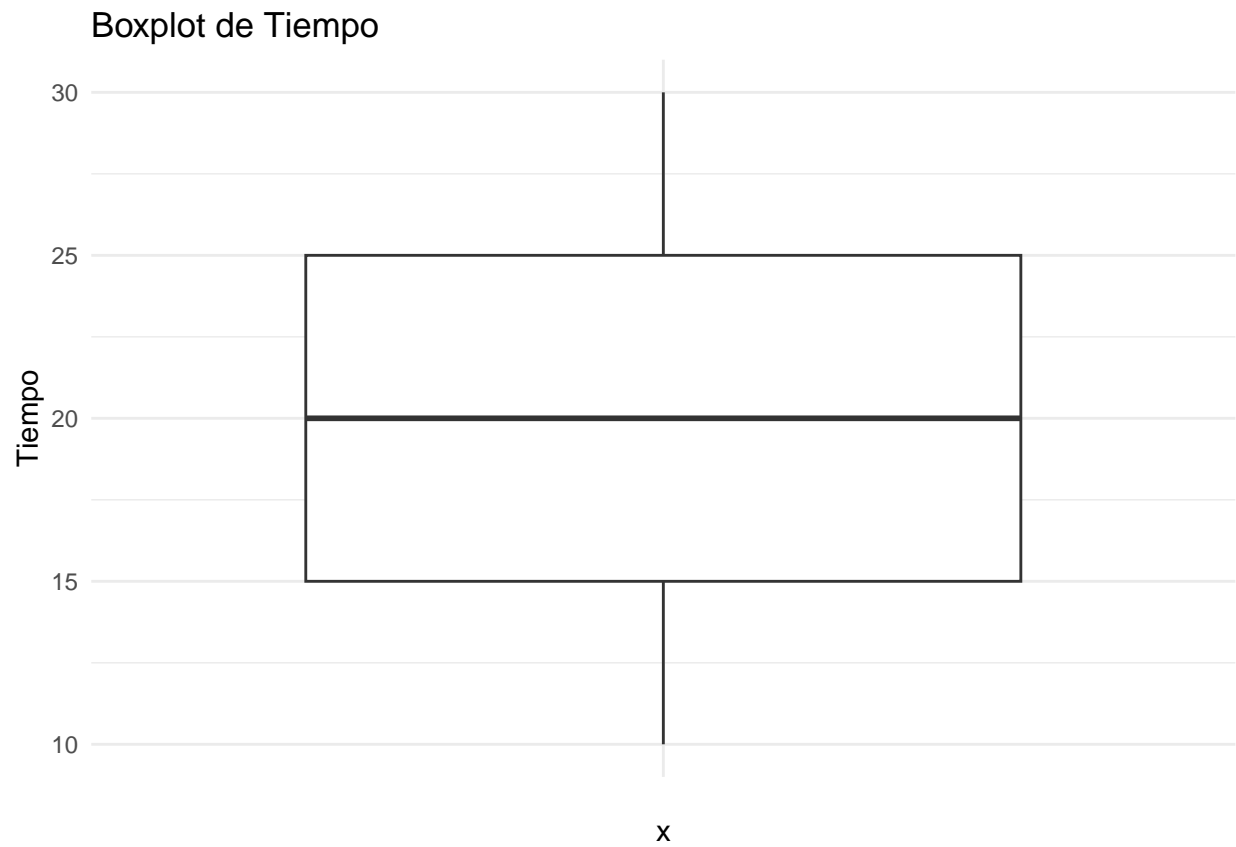
```
ggplot(M, aes(x = "", y = Potencia)) +  
  geom_boxplot() +  
  labs(title = "Boxplot de Potencia", y = "Potencia") +  
  theme_minimal()
```



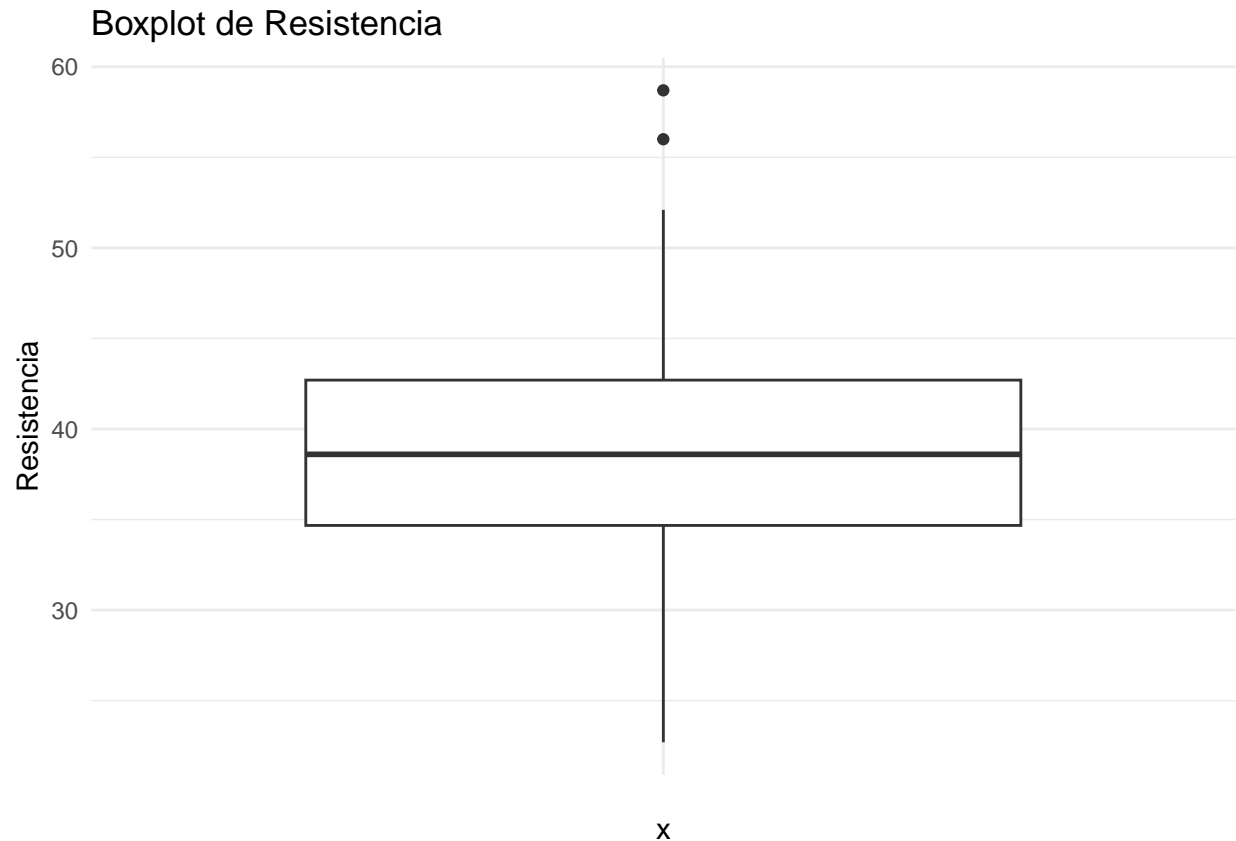
```
ggplot(M, aes(x = "", y = Temperatura)) +  
  geom_boxplot() +  
  labs(title = "Boxplot de Temperatura", y = "Temperatura") +  
  theme_minimal()
```



```
ggplot(M, aes(x = "", y = Tiempo)) +  
  geom_boxplot() +  
  labs(title = "Boxplot de Tiempo", y = "Tiempo") +  
  theme_minimal()
```



```
ggplot(M, aes(x = "", y = Resistencia)) +  
  geom_boxplot() +  
  labs(title = "Boxplot de Resistencia", y = "Resistencia") +  
  theme_minimal()
```



## Encuentra el mejor modelo de regresión que explique la variable Resistencia. Analiza el modelo basándote en: Significancia del modelo: Economía de las variables Significación global (Prueba para el modelo) Significación individual (Prueba para cada Bi) Variación explicada por el modelo

```
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
```

```
Modelo6 =lm(Resistencia ~ Potencia + Temperatura, data=M)
```

```
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
```

```
Pasos3 = step(Modelo, direction = "backward")
```

```
## 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
## - 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
## - Temperatura  1      252.2  984.24 108.72
## - Potencia     1     1341.0 2073.06 131.07
```

`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
```

EL MEJOR RESULTADO ES RESISTENCIA CONTRA TEMPERATURA (POTENCIA + TEMPERATURA)

`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
```

BIC

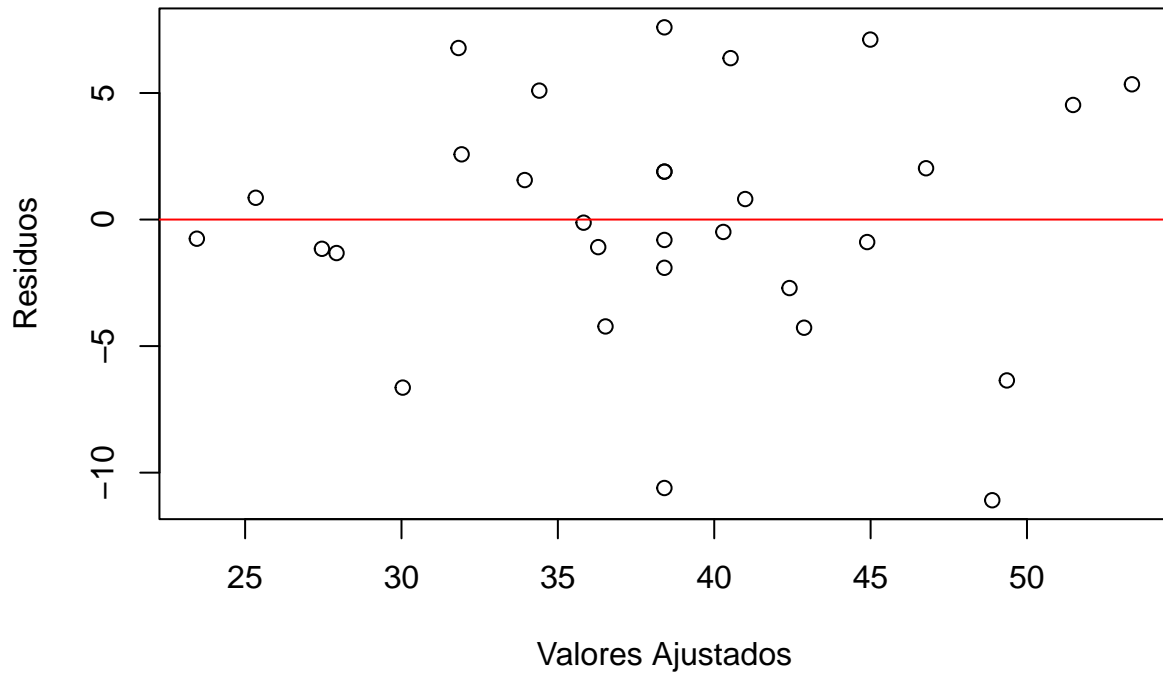
```
n= length(M$Resistencia)
Pasos = step(Modelo, direction="both", k=log(n))
```

```
## 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
```

## Analiza la validez del modelo encontrado: Análisis de residuos (homocedasticidad, independencia, etc)  
No multicolinealidad de Xi

```
plot(Modelo$fitted.values, residuals(Modelo), main = "Residuos vs Valores Ajustados", xlab = "Valores Ajustados", ylab = "Residuos",
abline(h = 0, col = "red"))
```

## Residuos vs Valores Ajustados



```
library(lmtest)
```

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

```
##
```

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

```
## The following objects are masked from 'package:base':
```

```
##
```

```
## as.Date, as.Date.numeric
```

```
bptest(Modelo)
```

```
##
```

```
## studentized Breusch-Pagan test
```

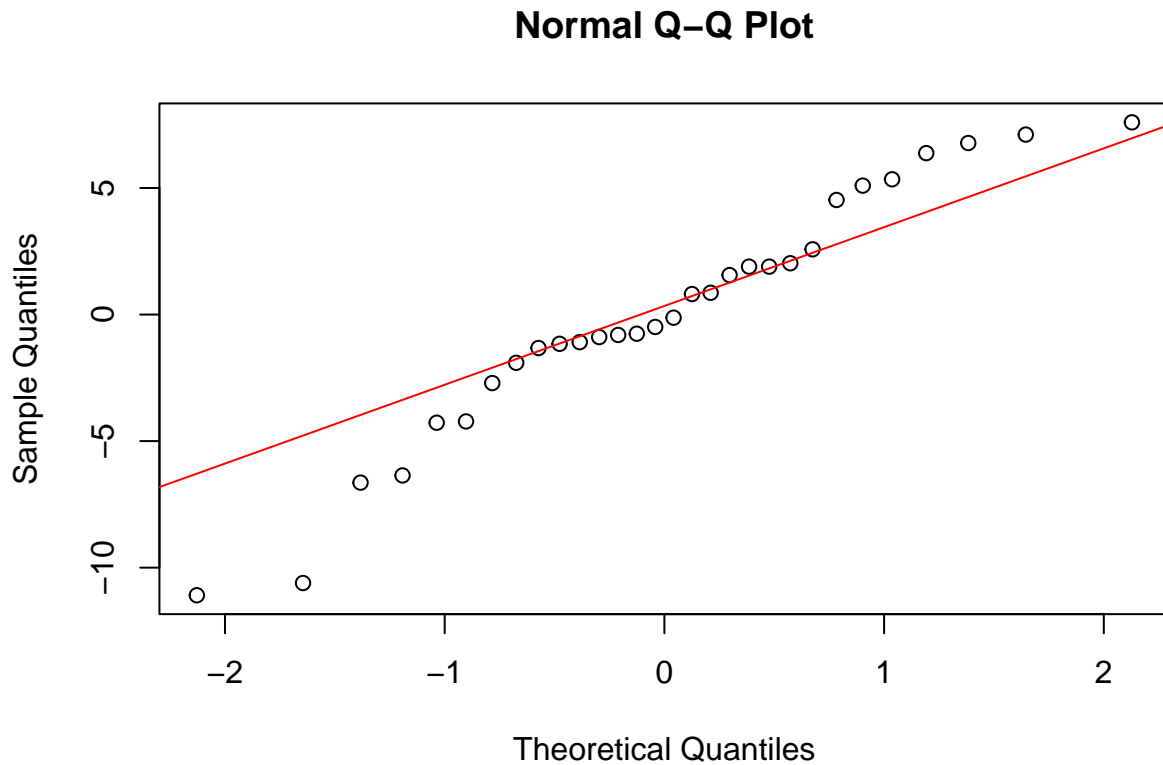
```
##
```

```
## data: Modelo
```

```
## BP = 4.2293, df = 4, p-value = 0.3759
```

```
qqnorm(residuals(Modelo))
```

```
qqline(residuals(Modelo), col = "red")
```



```
library(lmtest)
dwtest(Modelo)
```

```
##
## Durbin-Watson test
##
## data: Modelo
## DW = 2.2611, p-value = 0.7917
## alternative hypothesis: true autocorrelation is greater than 0
```

```
library(car)
```

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

```
vif(Modelo)
```

```
##      Fuerza      Potencia Temperatura      Tiempo
##          1          1          1          1
```

##Emite conclusiones sobre el modelo final encontrado e interpreta en el contexto del problema el efecto de las variables predictoras en la variable respuesta

En conclusión, el modelo Resistencia ~ Potencia + Temperatura, la cual se obtuvo con el modelo Pasos, según el análisis de AIC y la significancia estadística de los coeficientes, lo que se da a entender que La potencia

tiene un efecto notable en el aumento de la resistencia, mientras que la temperatura también contribuye, aunque en menor medida.

##A3-Regresión Múltiple-Detección datos atípicos En la base de datos Al corte Download Al cortese 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.

Haz un análisis descriptivo de los datos: medidas principales y gráficos (ya lo hiciste en la actividad A2) Encuentra el mejor modelo de regresión que explique la variable Resistencia (ya lo hiciste en la actividad A2) Analiza la validez del modelo encontrado (ya lo hiciste en la actividad A2)

Haz el análisis de datos atípicos e incluyentes del mejor modelo encontrado Consulta los apoyos sobre Detección de datos atípicos Download Detección de datos atípicos para revisar códigos

```
modeloFinal = lm(Resistencia ~ Potencia + Temperatura, data = M)
summary(modeloFinal)
```

```
##
## 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
```

```
library(dplyr)
```

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

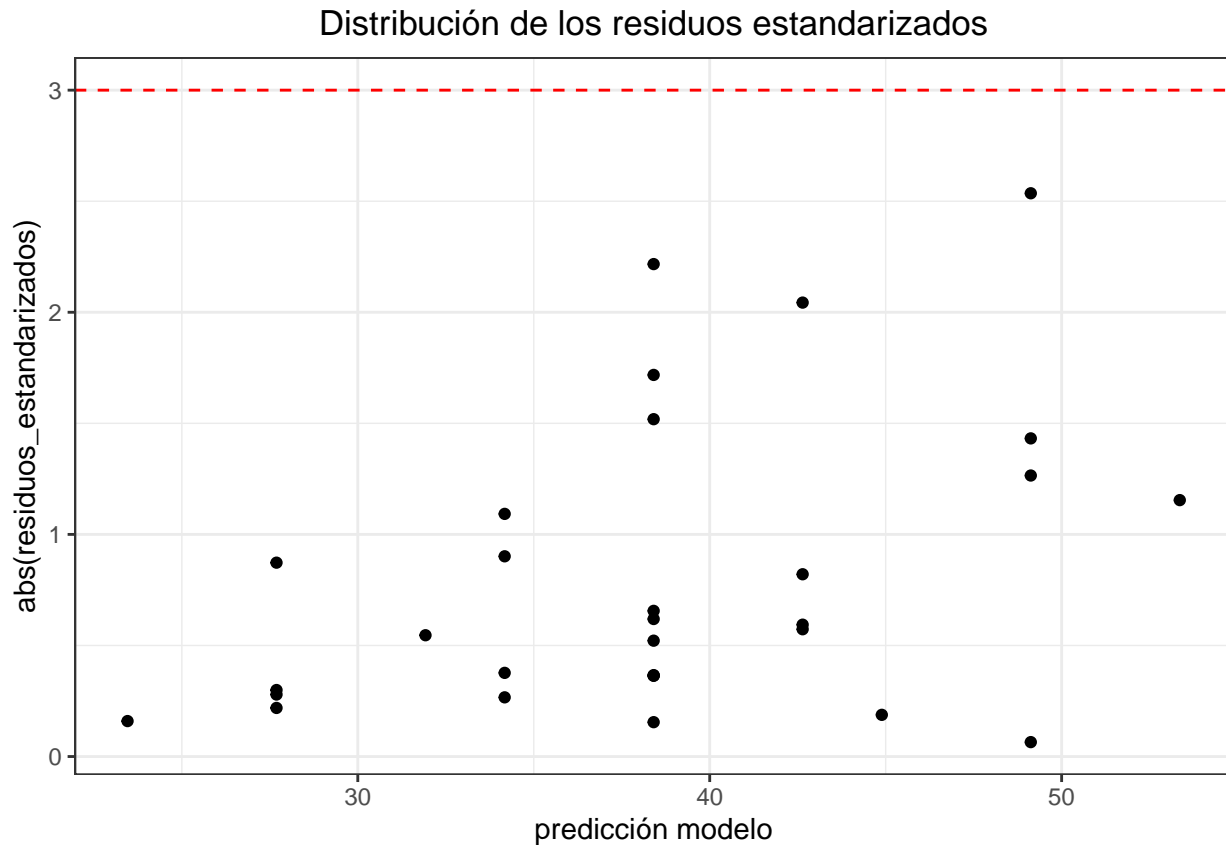
## The following object is masked from 'package:car':
##
##      recode

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

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

```
M$residuos_estandarizados <- rstudent(modeloFinal)

ggplot(data = M, aes(x = predict(modeloFinal), 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))
```

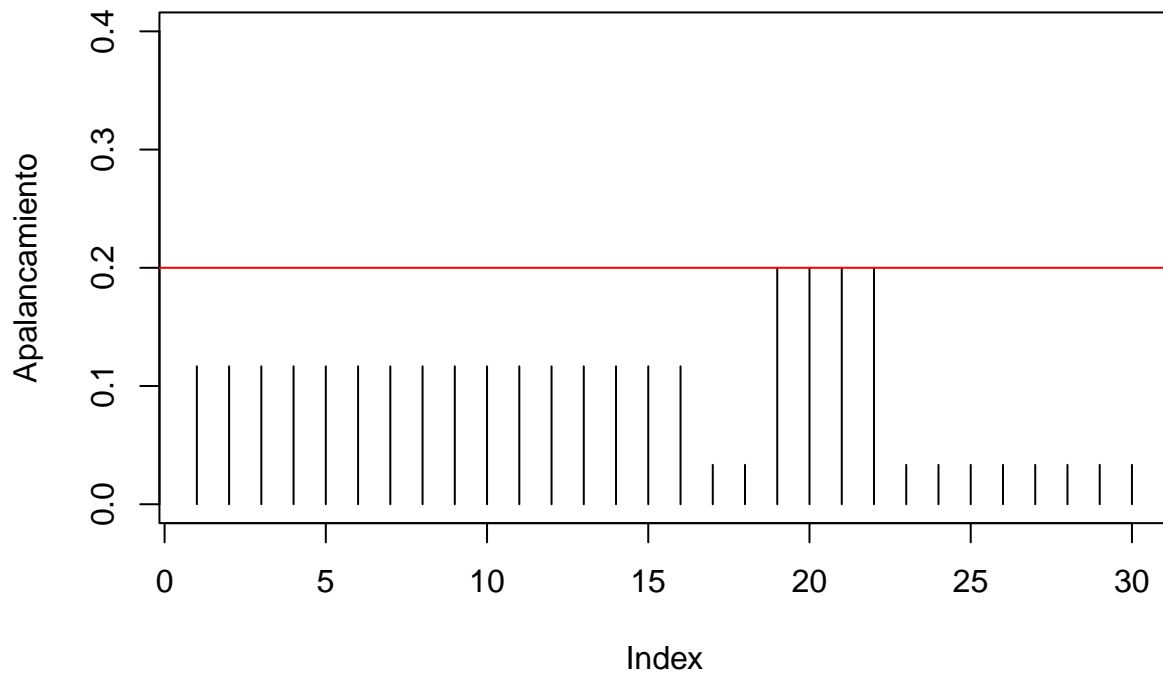


```
Atipicos = which(abs(M$residuos_estandarizados)>3)
M[Atipicos,]
```

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

```
leverage = hatvalues(modeloFinal)
#Calcula el leverage de los n datos
plot(leverage, type="h", main="Valores de Apalancamiento", ylab="Apalancamiento", ylim=c(0,4*mean(leverage)))
abline(h = 2*mean(leverage), col="red") # Límite comúnmente usado
```

## Valores de Apalancamiento

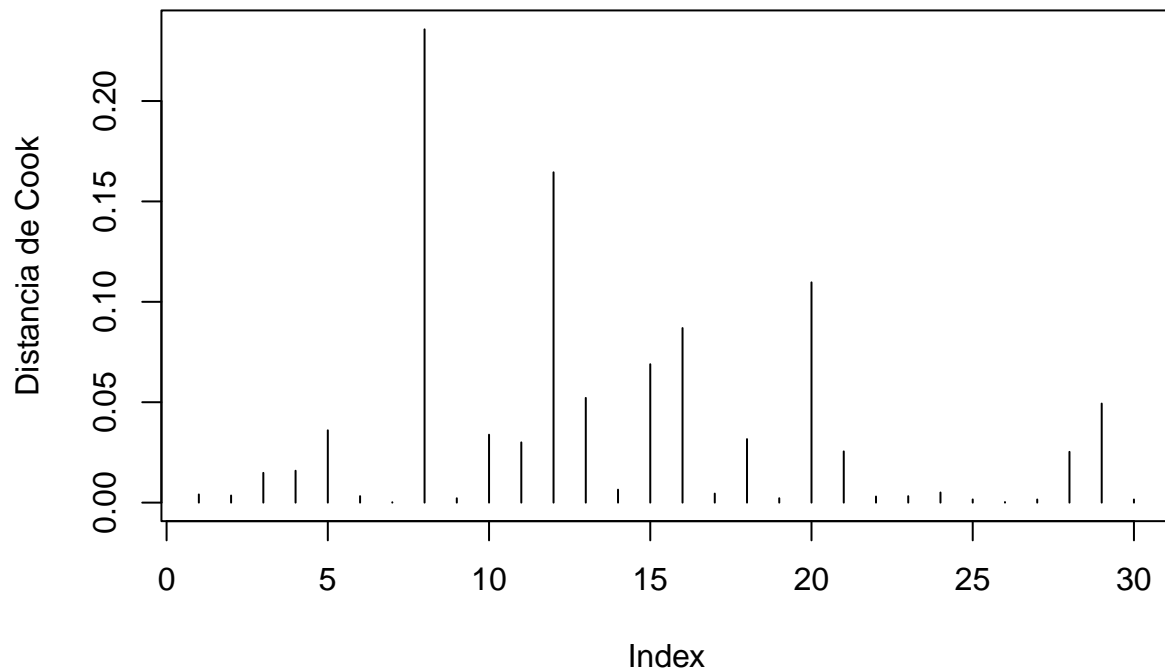


```
high_leverage_points = which(leverage > 2*mean(leverage))
M[high_leverage_points, ]
```

```
##      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
```

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

## Distancia de Cook



```
puntos_influyentes = which(cooksdistance > 1)
M[puntos_influyentes, ]
```

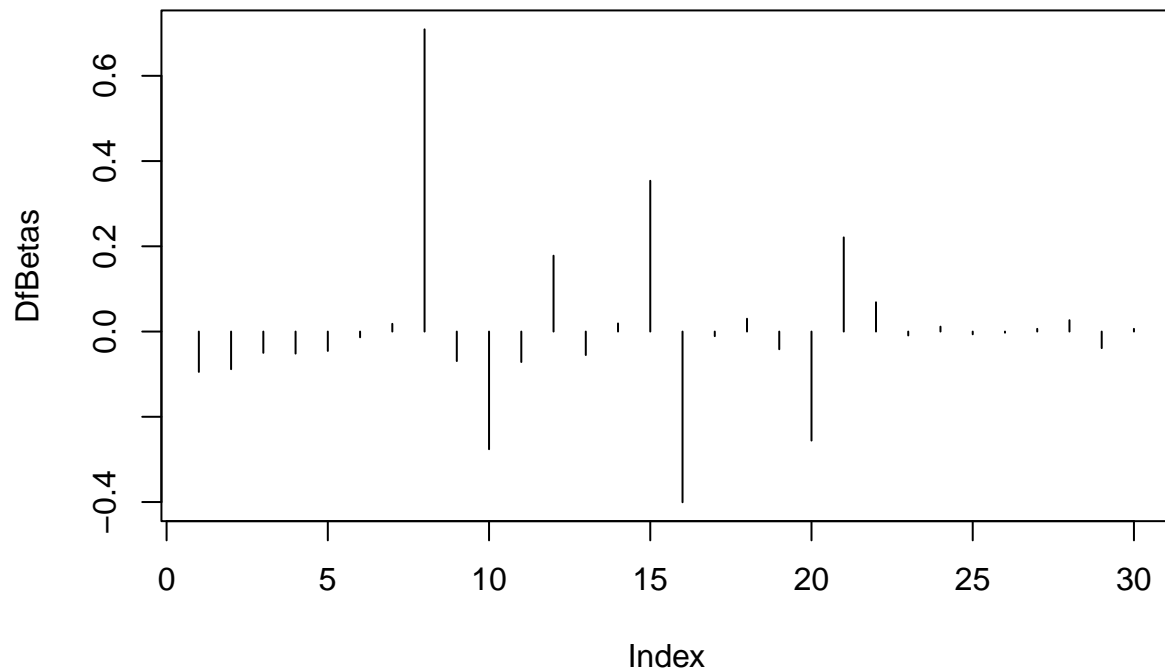
```
## [1] Fuerza          Potencia          Temperatura
## [4] Tiempo          Resistencia       residuos_estandarizados
## <0 rows> (or 0-length row.names)
```

```
dfbetas_values = dfbetas(modeloFinal)
```

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



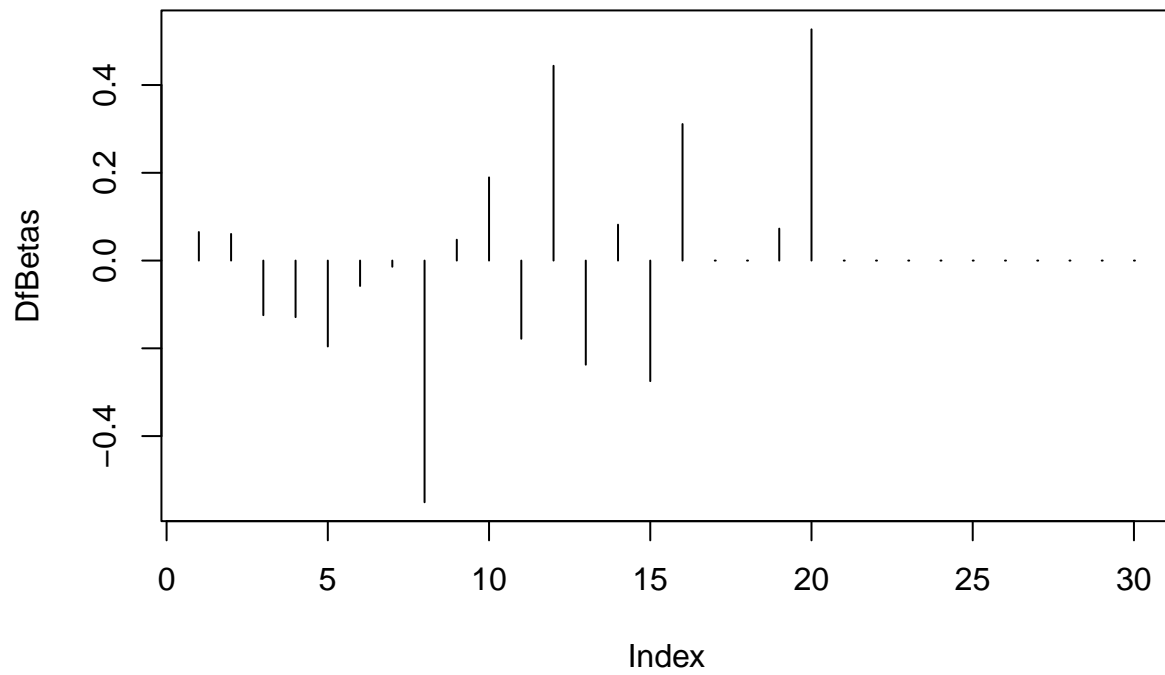
## DfBetas para el coeficiente 1



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

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

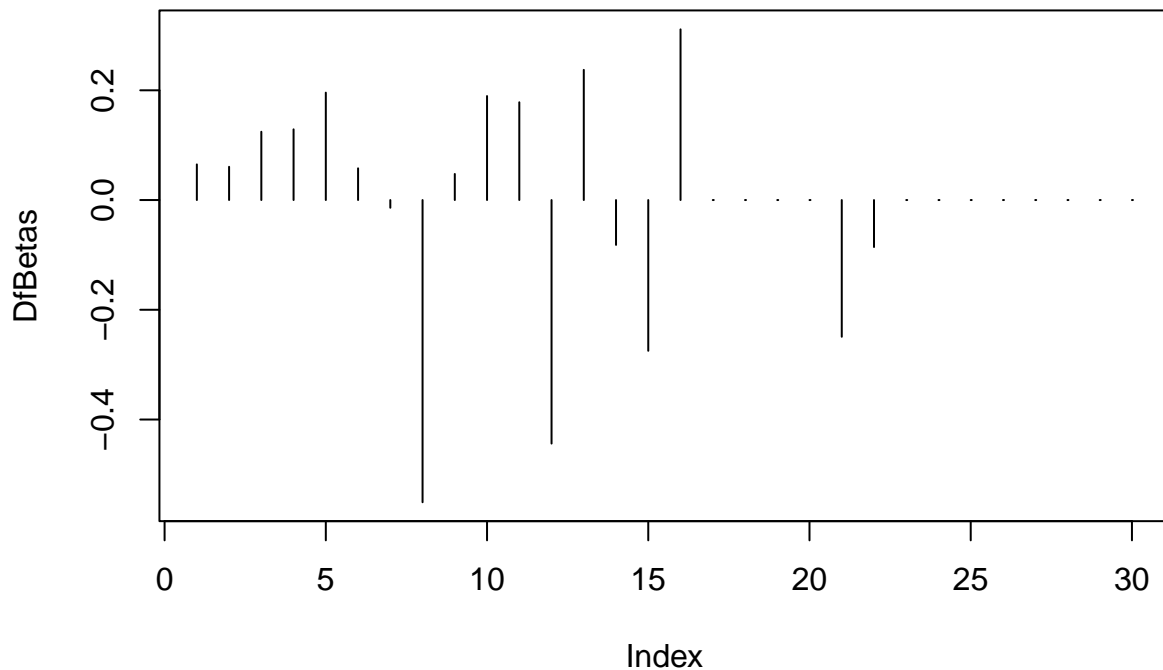
## DfBetas para el coeficiente 2



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

plot(dfbetas_values[, 3], type="h", main="DfBetas para el coeficiente 3",ylab="DfBetas")
abline(h = c(-1, 1), col="red")
```

### DfBetas para el coeficiente 3

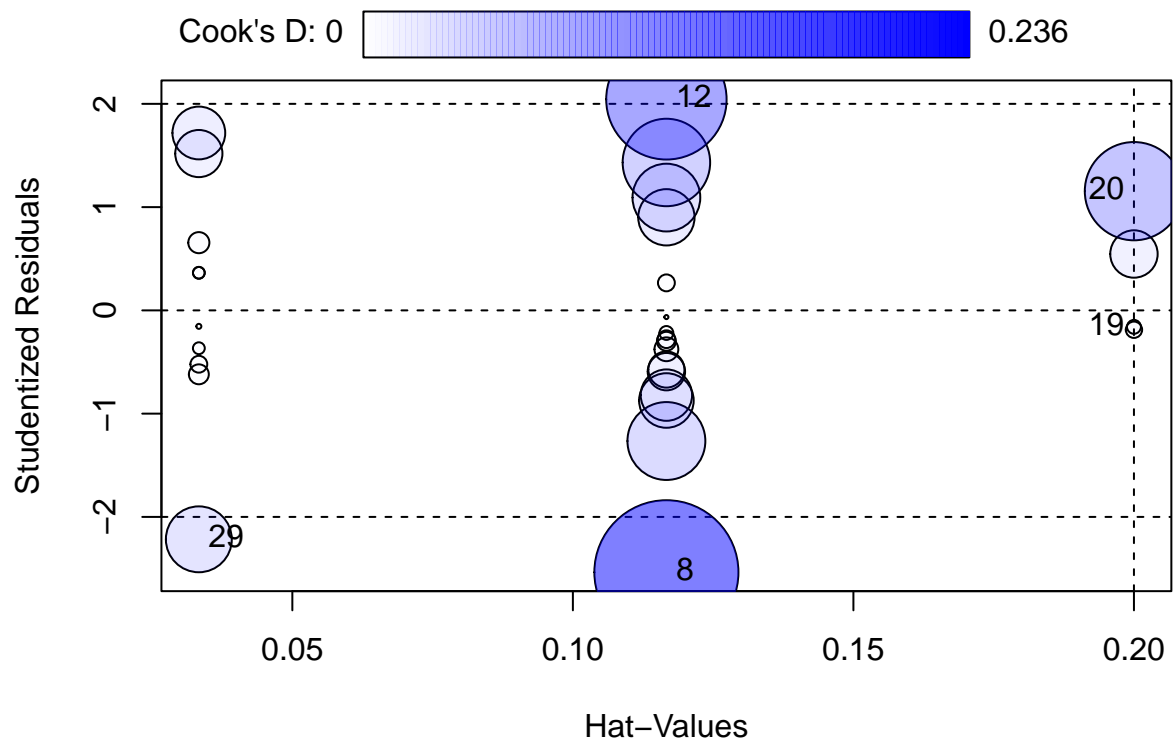


```
puntos_influyentes = which(abs(dfbetas_values[, 3]) > 1)
```

```
influencia = influence.measures(modeloFinal)
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
```

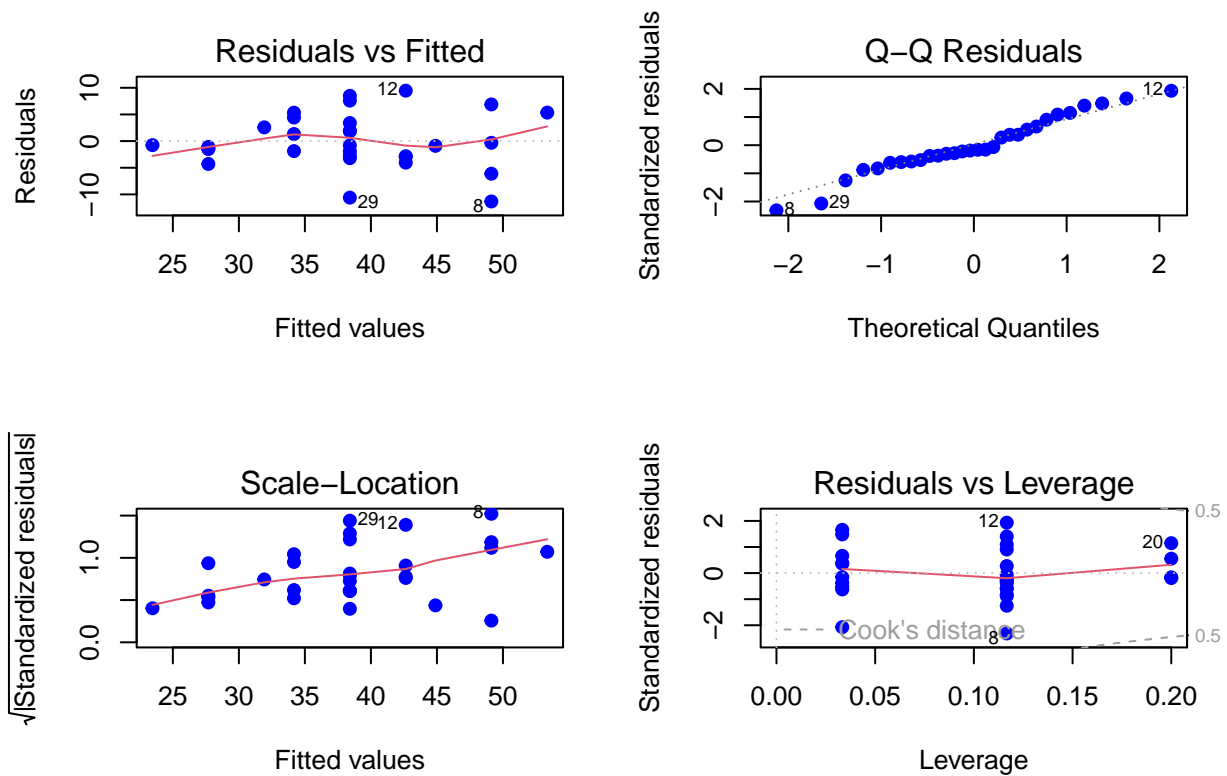
```
library(car)
influencePlot(modeloFinal)
```



```
##      StudRes      Hat      CookD
## 8  -2.535832 0.1166667 0.235696235
## 12  2.043589 0.1166667 0.164507739
## 19 -0.159511 0.2000000 0.002199712
## 20  1.154355 0.2000000 0.109693544
## 29 -2.216952 0.0333333 0.049338917
```

*# grafica los residuos con estandarización extrema, el laverage y la distancia de cook Muestra las obse*

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



En conclusión, podemos observar que existen datos atípicos en los datos presentados, tal como se pueden ver en las gráficas anteriores:

En la gráfica de Residuals vs Fitted se puede observar que existe curvatura en el abline, y se denotan los valores 29, 12 y 8 como valores muy grandes, lo cual nos puede indicar que existe una relación no lineal en las variables.

En la segunda gráfica de Q-Q residuals, se puede observar los casos extremos en ambas puntas de las gráficas, lo cual nos indica que aunque su mayoría si siguen la distribución, esta no es una distribución perfecta.

En la gráfica de Scale-Location, es posible vislumbrar un incremento positivo, lo cual nos dice que los residuos no tienen una varianza constante

En la cuarta gráfica de Residuales vs Leverage, se observa que aunque no se superen los límites dados por cook, los valores 12, 8 y 20 muestran influencia debido a su alto apalancamiento