Regresión Múltiple

Erika Martínez Meneses

2024-09-17

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.

Lectura de Datos

```
file.choose()
```

```
## [1] "C:\\Users\\erika\\Downloads\\AlCorte.csv"
```

```
data <- read.csv("C:\\Users\\erika\\Downloads\\AlCorte.csv")
```

Analisis desciptivo

```
head(data)
```

```
##   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
```

Medidas estadísticas

```
summary(data)
```

```
##      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
```

Correlación

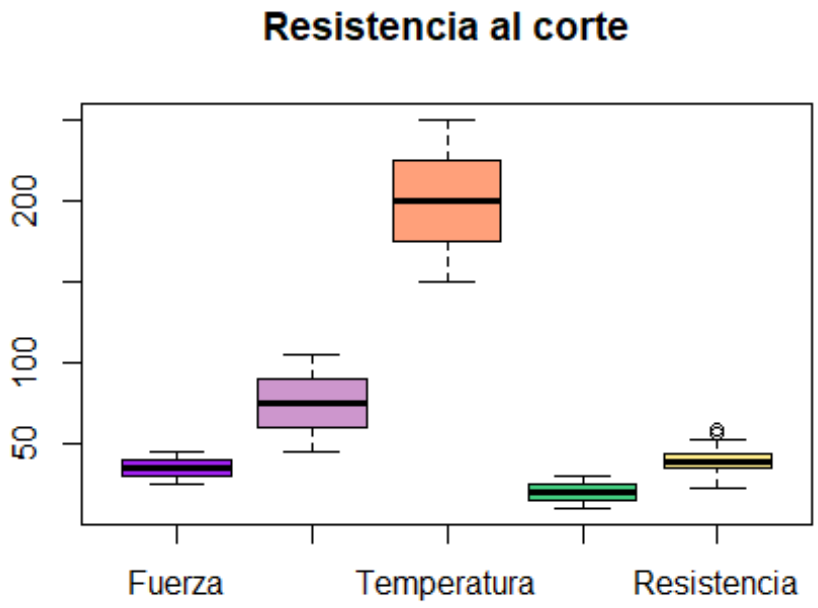
```
cor(data)
```

```
##                Fuerza  Potencia Temperatura    Tiempo Resistencia
## Fuerza      1.0000000 0.0000000   0.0000000 0.0000000   0.1075208
## Potencia    0.0000000 1.0000000   0.0000000 0.0000000   0.7594185
```

```
## Temperatura 0.0000000 0.0000000   1.0000000 0.0000000   0.3293353
## Tiempo      0.0000000 0.0000000   0.0000000 1.0000000   0.1312262
## Resistencia 0.1075208 0.7594185   0.3293353 0.1312262   1.0000000
```

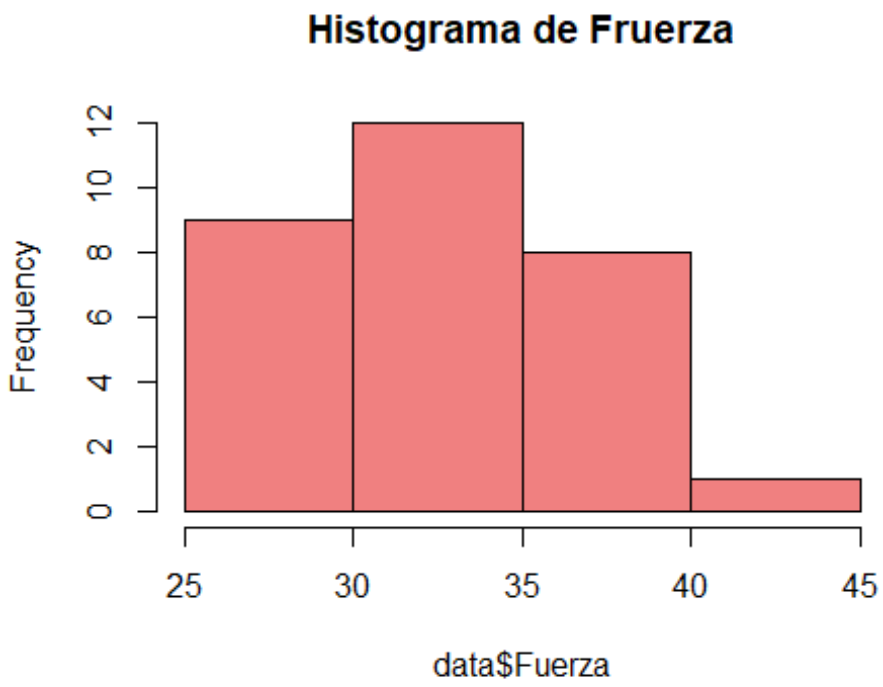
Boxplot

```
boxplot(data, main = "Resistencia al corte", col=c("purple","#CD96CD",
"#FFA07A", "#43CD80", "#FFEC8B"))
```

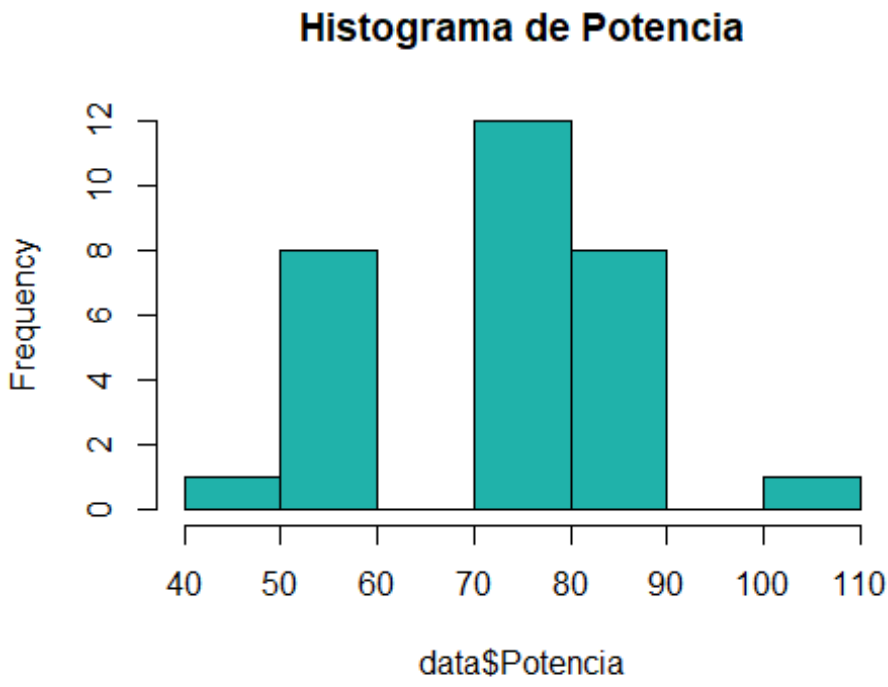


Histograma

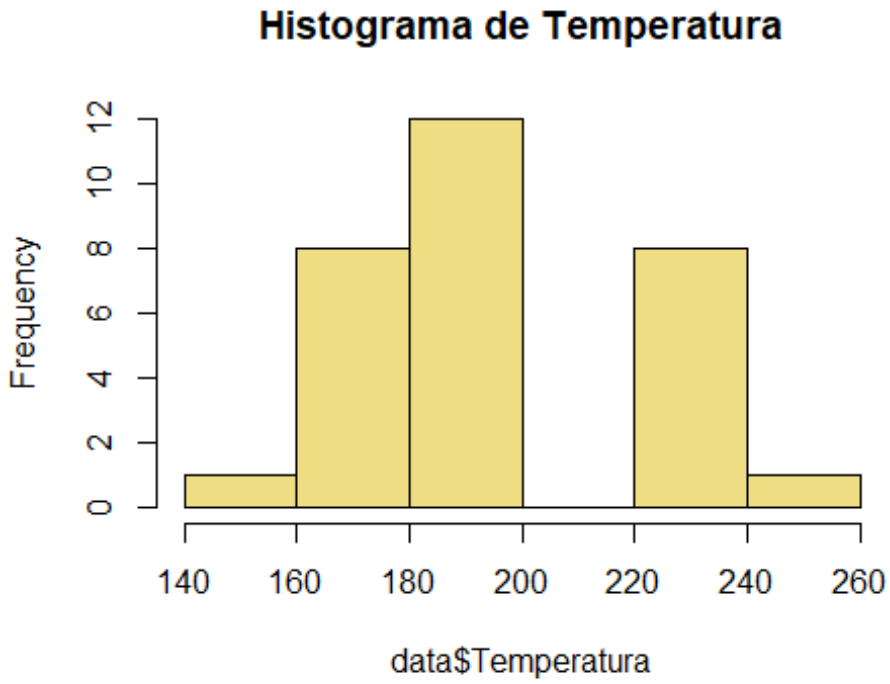
```
hist(data$Fuerza, col = "lightcoral", main = "Histograma de Fruerza")
```



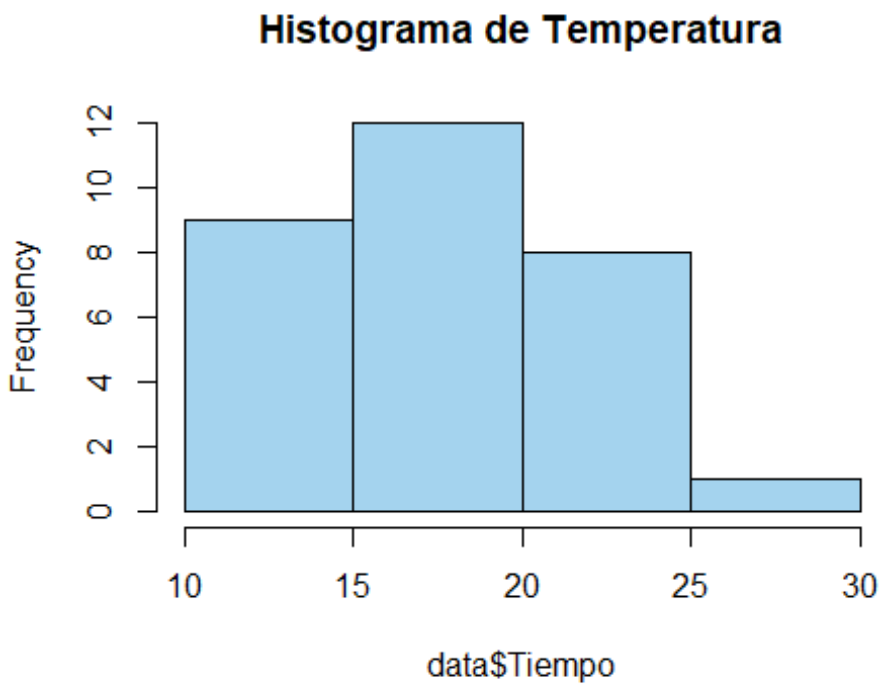

```r
hist(data$Potencia, col = "lightseagreen", main = "Histograma de Potencia")
```



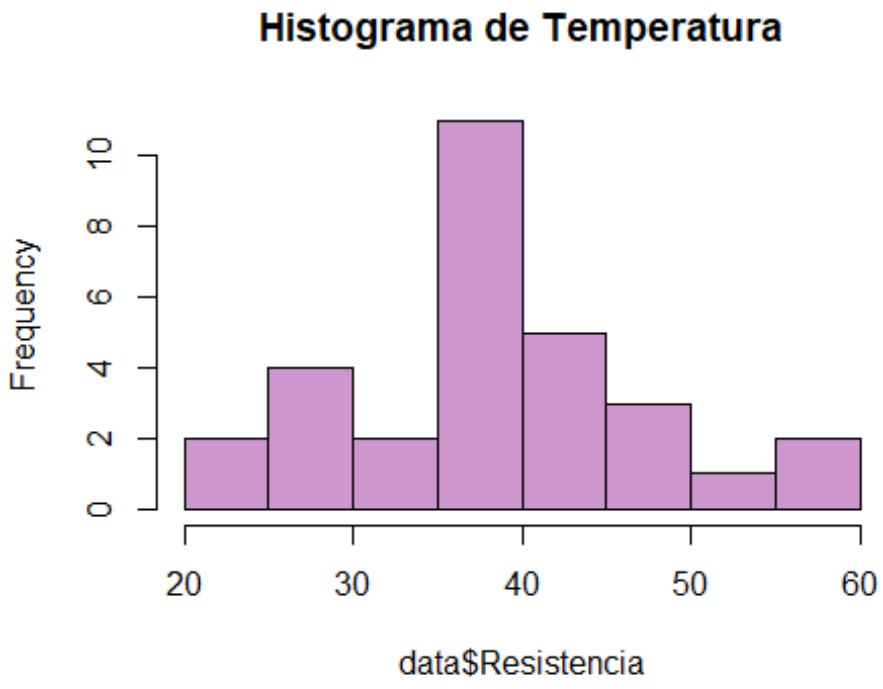
```r
hist(data$Temperatura, col = "lightgoldenrod", main = "Histograma de
Temperatura")
```




```r
hist(data$Tiempo, col = "lightskyblue2", main = "Histograma de
Temperatura")
```




```r
hist(data$Resistencia, col = "#CD96CD", main = "Histograma de
Temperatura")
```



Encuentra el mejor modelo de regresión que explique la variable Resistencia. Analiza el modelo basándote en Significancia del modelo (Significación global, Significación individual, Variación explicada por el modelo)

Significancia del modelo:

Economía de las variables

Criterio de información de Akaike (AIC)

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

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   1341.02 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(Pasos1)
```

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

```
## 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 = data)
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) # Te regresa el mejor modelo
```

```
##
## Call:
```

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

```
#Modelo2 = lm(Resistencia~Potencia+Temperatura, data= data)
```

```
Pasos3 = step(Modelo, direction="backward", 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
## - 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
```

Criterio Shwarz o de información Bayesiano (BIC)

```
n = length(data$Resistencia)
Pasos1 = 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(Pasos1)
```

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

```
Pasos2 = step(modelo_nulo, scope = list(lower = modelo_nulo, upper =
Modelo), direction = "forward", k=log(n))
```

```
## Start:  AIC=133.91
## Resistencia ~ 1
##
##               Df Sum of Sq     RSS    AIC
## + Potencia     1   1341.01  984.24 111.52
## + Temperatura  1    252.20 2073.06 133.87
## <none>                     2325.26 133.91
## + Tiempo       1     40.04 2285.22 136.79
## + Fuerza       1     26.88 2298.38 136.97
##
## Step:  AIC=111.52
## Resistencia ~ Potencia
##
##               Df Sum of Sq    RSS    AIC
## + Temperatura  1   252.202 732.04 106.04
## <none>                     984.24 111.52
## + Tiempo       1    40.042 944.20 113.68
## + Fuerza       1    26.882 957.36 114.09
##
## Step:  AIC=106.04
## Resistencia ~ Potencia + Temperatura
##
##          Df Sum of Sq    RSS    AIC
## <none>                 732.04 106.04
## + Tiempo  1    40.042 692.00 107.76
## + Fuerza  1    26.882 705.16 108.32
```

```
Pasos3 = step(Modelo, direction="backward", 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
## - 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
## - Temperatura  1     252.2  984.24 111.52
## - Potencia     1    1341.0 2073.06 133.87
```

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

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

Significación global (Prueba para el modelo)

Valida la significancia del modelo con un alfa de 0.05 (incluye las hipótesis que pruebas y el valor frontera)

- $H_0: \beta_1 = 0$
- $H_1: \beta_1 \neq 0$

```
if (summary(Best_model)$fstatistic[1] > qf(1-0.05, df1 =
summary(Best_model)$fstatistic[2], df2 =
summary(Best_model)$fstatistic[3])) {
  print("El modelo es significativo con alfa de 0.05.")
} else {
  print("El modelo no es significativo con alfa de 0.05.")
}
```

```
## [1] "El modelo es significativo con alfa de 0.05."
```

Significación individual (Prueba para cada β_i)

Hipótesis * $H_0: \beta_i = 0$ * $H_1: \exists \beta_i \neq 0$

```
coef(summary(Best_model))[, 4] < 0.05 # Devuelve TRUE si los coeficientes son significativos
```

```
## (Intercept)    Potencia Temperatura
##        TRUE        TRUE        TRUE
```

##Variación explicada por el modelo

```
paste("El modelo explica el", round(summary(Best_model)$r.squared * 100, 2), "% de la variabilidad del precio")
```

```
## [1] "El modelo explica el 68.52 % de la variabilidad del precio"
```

Analiza la validez del modelo encontrado:

#Análisis de residuos

Normalidad de los residuos

Prueba de Hipótesis

- H_0 = La muestra proviene de una distribución normal
- H_1 = La muestra no proviene de una distribución normal

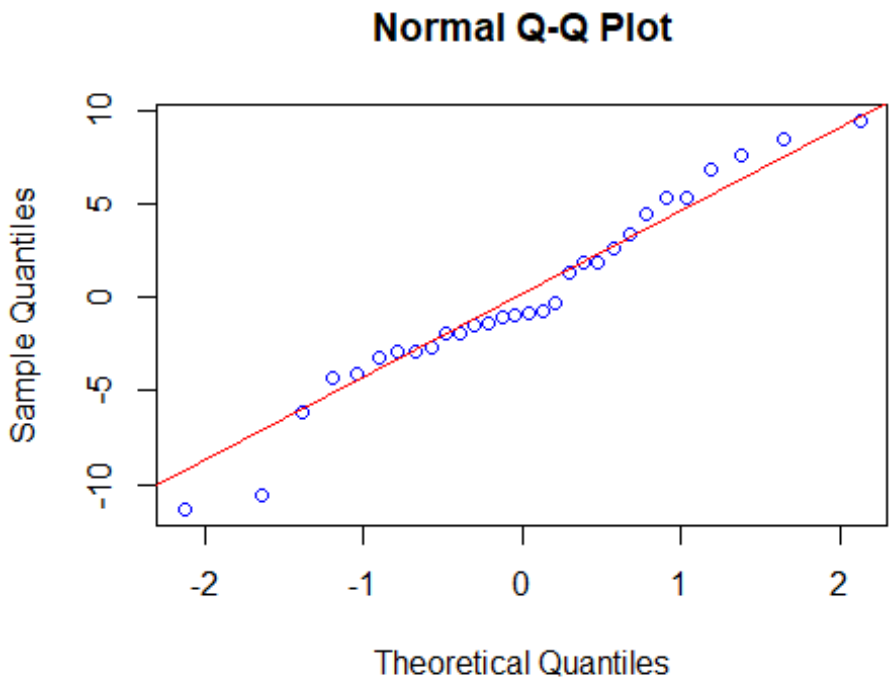
Regla de decisión: Se rechaza H_0 si valor p < α

Modelo 1

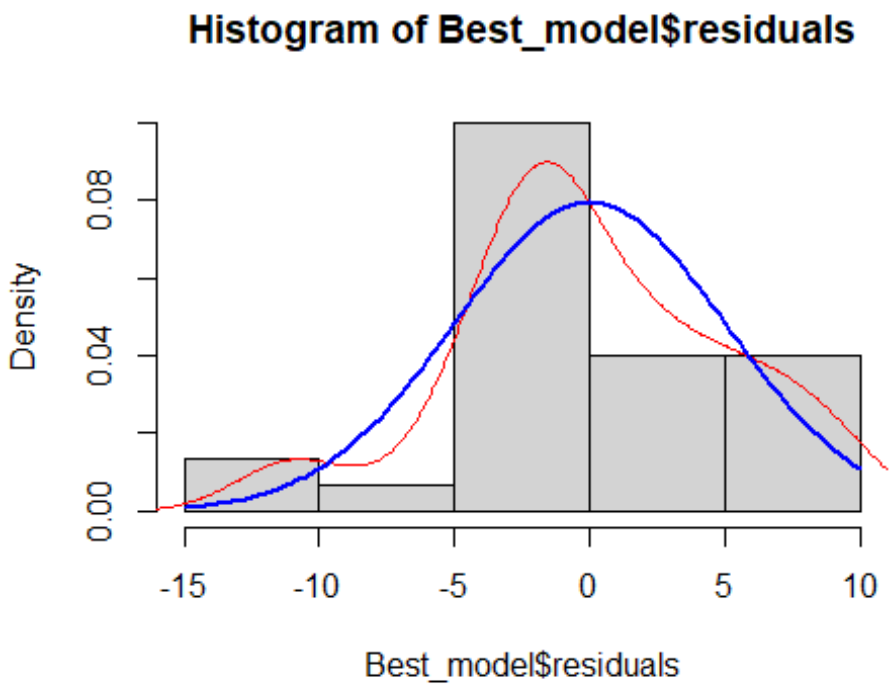
```
library(nortest)
ad.test(Best_model$residuals)
```

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

```
qqnorm(Best_model$residuals, col = "blue")
qqline(Best_model$residuals, col = "red")
```




```
hist(Best_model$residuals,freq=FALSE)
lines(density(Best_model$residual),col="red")
curve(dnorm(x,mean=mean(Best_model$residuals),sd=sd(Best_model$residuals)
), add=TRUE, col="blue",lwd=2)
```



Aceptamos H_0 ya que el valor p = 0.3204 > $\alpha = 0.05$ por lo que podemos decir que nuestros datos provienen de una distribución normal

Verificación de media cero

Prueba de Hipótesis

- $H_0: \mu_e = 0$
- $H_1: \mu_e \neq 0$

Regla de decisión * Se rechaza si valor p < α

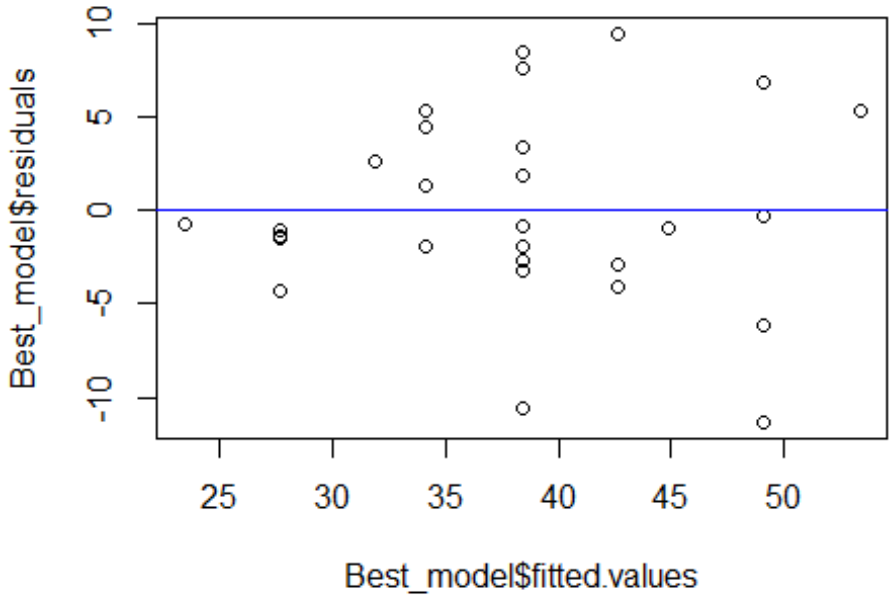
```
t.test(Best_model$residuals)
```

```
##
##  One Sample t-test
##
## data:  Best_model$residuals
## 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
```

Aceptamos H_0 ya que nuestro valor p = 1 > $\alpha = 0.05$ entonces podemos concluir que $\mu_e = 0$. Los residuos tienen media cero: el modelo es bueno.

Homocedasticidad, linealidad e independencia

```
plot(Best_model$fitted.values,Best_model$residuals)
abline(h=0, col="blue")
```



Pruebas de hipótesis para independencia

Test de Durbin-Watson y Prueba Breusch-Godfrey

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

Regla de decisión: Se rechaza H_0 si valor p < α

```
library(lmtest)

## Loading required package: zoo

##
## Attaching package: 'zoo'

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

dwtest(Best_model)
```

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

```
bgtest(Best_model)
```

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

Aceptamos H_0 ya que el valor p = 0.8267 y 0.2863 para Durbin-Watson test y Breusch-Godfrey test respectivamente siendo mayores que $\alpha = 0.05$ lo que significa que los errores no están autocorrelacionados.

Pruebas de hipótesis para homocedasticidad

Prueba de Breusch-Pagan y White

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

Regla de decisión: Se rechaza H_0 si valor p < α

Modelo 1

```
library(lmtest)
bptest(Best_model)
```

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

```
gqtest(Best_model)
```

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

Aceptamos H_0 ya que el valor p = 0.135 y 0.5169 para Breusch-Pagan test y Goldfeld-Quandt test respectivamente siendo mayores que $\alpha = 0.05$ lo que significa que La varianza de los errores es constante (hay homocedasticidad).

Pruebas de hipótesis para linealidad

- H_0: No hay términos omitidos que indican linealidad
- H_1: Hay una especificación errónea en el modelo que indica no linealidad

Regla de decisión: Se rechaza H_0 si valor p < α

Modelo 1

```
resettest(Best_model)
```

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

Aceptamos H_0 ya que p-value = 0.4647 A $\alpha = 0.05$ lo que indica que no hay términos omitidos que indican linealidad.

No multicolinealidad de X_i

Matriz de correlación

```
cor(data)
```

```
##                Fuerza  Potencia Temperatura    Tiempo Resistencia
## Fuerza      1.0000000 0.0000000   0.0000000 0.0000000   0.1075208
## Potencia    0.0000000 1.0000000   0.0000000 0.0000000   0.7594185
## Temperatura 0.0000000 0.0000000   1.0000000 0.0000000   0.3293353
## Tiempo      0.0000000 0.0000000   0.0000000 1.0000000   0.1312262
## Resistencia 0.1075208 0.7594185   0.3293353 0.1312262   1.0000000
```

Factor de inflación de la varianza (VIF)

```
library(car)
```

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

```
vif(Best_model)
```

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

Tenemos un valor bajo de VIF, valor de 1, lo que nos indica que hay baja multicolinealidad.

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

Modelo final: El mejor modelo incluye las variables Potencia y Temperatura como predictoras significativas de la Resistencia al corte.

Potencia: Tiene un efecto positivo y significativo sobre la resistencia, con un coeficiente de 0.498. Un aumento en la potencia incrementa la resistencia.

Temperatura: También es significativa, con un coeficiente de 0.130, indicando que un aumento en la temperatura incrementa la resistencia, pero con un efecto menor que la potencia.

Calidad del modelo:

- Significación global: El modelo es significativo, lo que indica que las variables seleccionadas explican una porción importante de la variabilidad en la resistencia.
- R^2 ajustado: El modelo explica el 66.19% de la variabilidad en la resistencia al corte, lo que indica un buen ajuste.
- Análisis de residuos: Los residuos cumplen con las suposiciones de normalidad, homocedasticidad, independencia, y media cero, lo que sugiere que el modelo es válido.

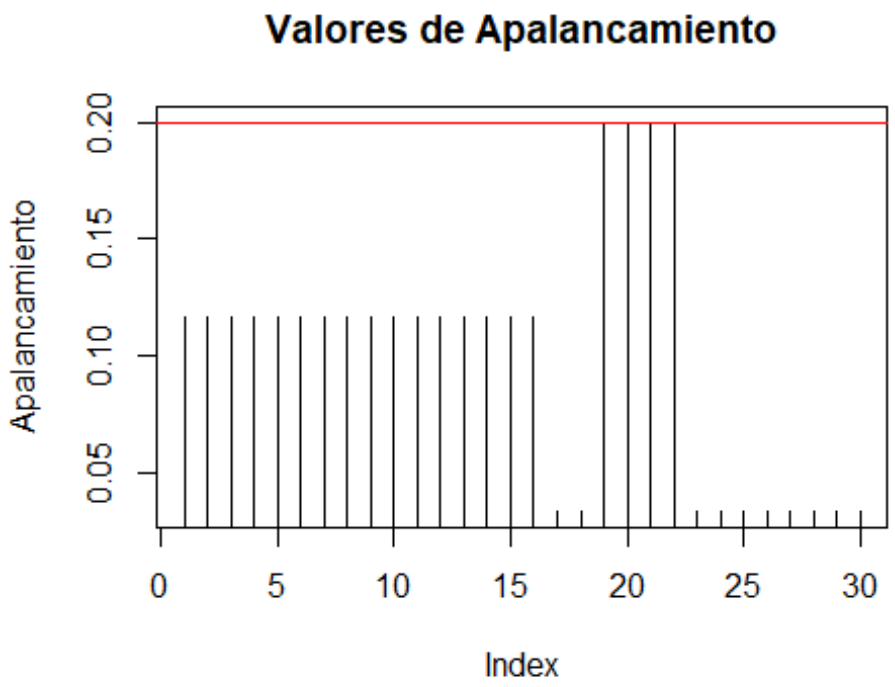
En conclusión el mejor modelo que explica la resistencia al corte incluye las variables Potencia y Temperatura. La potencia tiene un mayor impacto, y ambas variables contribuyen significativamente a explicar la variabilidad en la resistencia. El modelo es estadísticamente sólido, ya que cumple con las principales suposiciones de la regresión lineal.

Detección datos atípicos

Distancia de Leverage

Para detectar datos atípicos en X

```
leverage = hatvalues(Best_model)
#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
```



Cuenta e identifica cuántos datos atípicos hay:

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

Muestra las observaciones con alto leverage

```
data[high_leverage_points, ]
```

```
##    Fuerza Potencia Temperatura Tiempo Resistencia
## 19     35       45         200     20        22.7
## 20     35      105         200     20        58.7
```

Identificamos que existen 2 datos atípicos en X, estos son los datos 19 y 20, esto lo podemos observar también en la gráfica de Valores de Apalancamiento en donde vemos que las líneas negras que representan a estos valores alcanzan la línea roja, y a pesar que en apariencia en la gráfica parezca que el 21 y 22 también alcanzan la línea roja no alcanzan a ser datos atípicos, seguramente por una diferencia de decimales que evita que alcancen el criterio para ser dato atípico.

Estandarización extrema de los residuos

Para detectar datos atípicos en Y

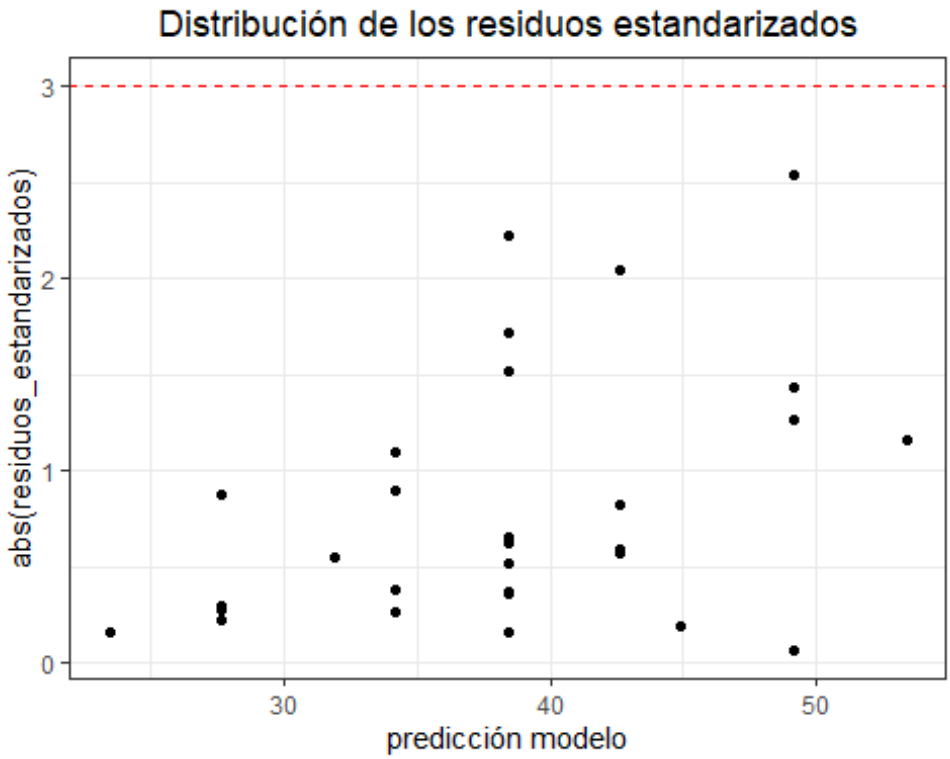
Se detectan residuos estandarizados que sean mayor a 3

```
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

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

Gráfico auxiliar:

```
library(ggplot2)
ggplot(data = data, aes(x = predict(Best_model), 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))
```



Cuenta e identifica cuántos datos atípicos hay:

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

Muestra las observaciones con altos residuos estandarizados

```
data[Atipicos, ]
```

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

En el caso de Y, el análisis de estandarización extrema de los residuos nos indica que no existen ningún dato atípico, esto lo confirmamos con la gráfica en donde todos los valores se encuentran por debajo de la línea roja.

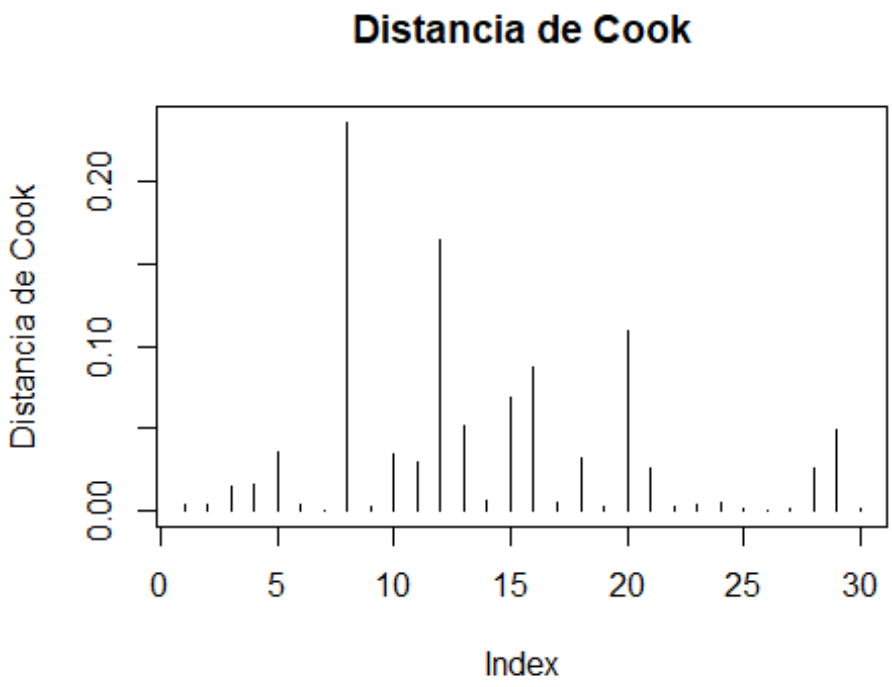
Detección de datos influyentes

Distancia de Cook

Mide cuánto cambian todos los valores ajustados en el modelo cuando se elimina el i-ésimo punto de datos

Se detectan distancias de Cook mayores a 1

```r
cooksdistance <- cooks.distance(Best_model)
#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
```




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

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

No se encontraron datos influyentes sobre las y mediante la distancia de cook. Un valor grande para la distancia indica un valor fuertemente influyente por lo que podemos deducir que nuestros valores son pequeños.

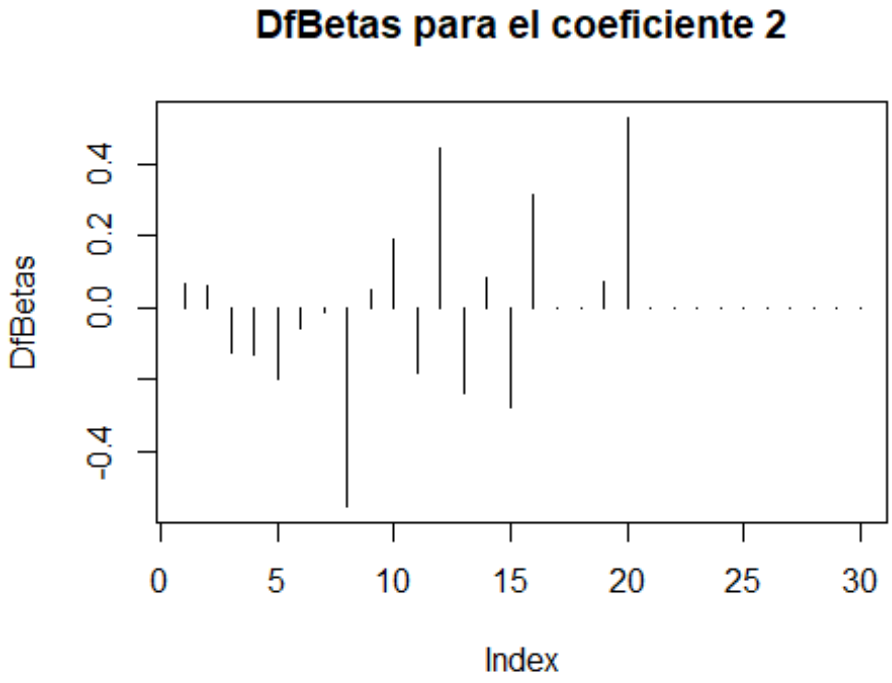
DfBetas

Mide cuánto cambia β_j cuando se elimina el i-ésimo dato.

Se detectan DfBetas mayores a |1|

```r
dfbetas_values = dfbetas(Best_model)
#Calcula la DfBeta de los n datos para cada $\beta_j$
```

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




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

## data frame with 0 columns and 30 rows
```

El análisis de DfBetas nos indica que no existen datos influyentes. Para que el i-ésimo dato tenga influencia alta en el coeficiente necesita tener un valor grande para la DfBeta.

Influence.measures

El hecho de que un valor sea atípico o con alto grado de leverage no implica que sea influyente en el conjunto del modelo. Sin embargo, si un valor es influyente, suele ser o atípico o de alto leverage

Calculan: * Distancia de leverange (h_{ii}) * Distancia de Cook * DfBetas

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

Resumen de datos influyentes:

```r
summary(influencia)
```

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

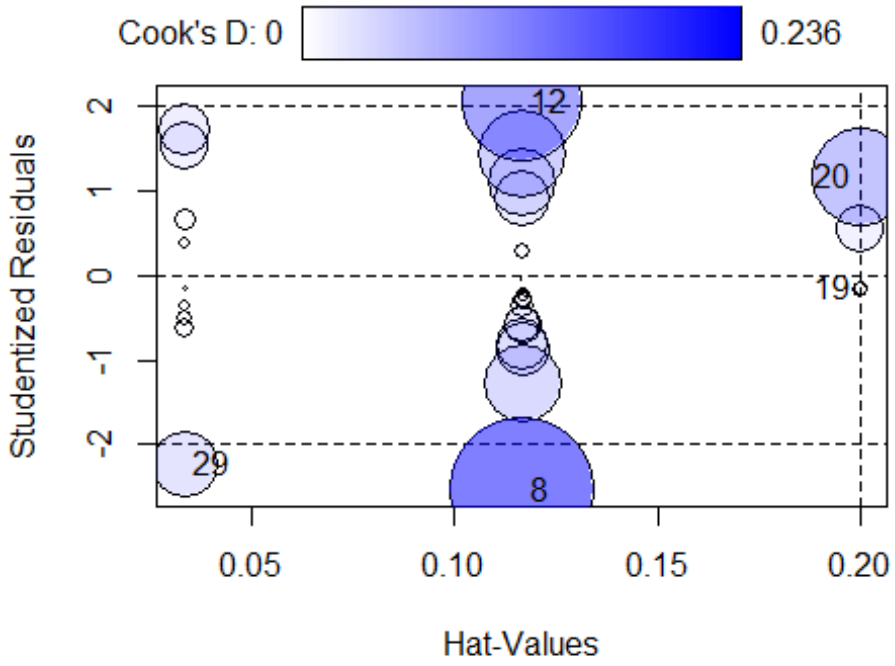
Mediante Influence.measures podemos observar datos a los que nos recomienda prestar atención sin embargo no significa que todos los datos debamos considerarlos como atípicos, se necesita un análisis más profundo sobre sus resultados. En este caso nos da los valores de los datos 8, 19, 21 y 22, recordemos que el 19 ya lo habíamos contemplado como atípico anteriormente sin embargo los demás datos no,por lo que se reafirma que nos arroja los datos a los que debemos prestar atención según cierto criterio, en este caso cov.r, sin embargo también podemos observar otros resultados como son Distancia de Cook y Leverages (hat) para un análisis más profundo.

influencePlot

Calcula: * Distancia de leverange (h_{ii}) * Distancia de Cook * Residuos estandarizados

Grafica los residuos con estandarización extrema, el laverage y la distancia de cook Muestra las observaciones influyentes

```
library(car)
influencePlot(Best_model)
```



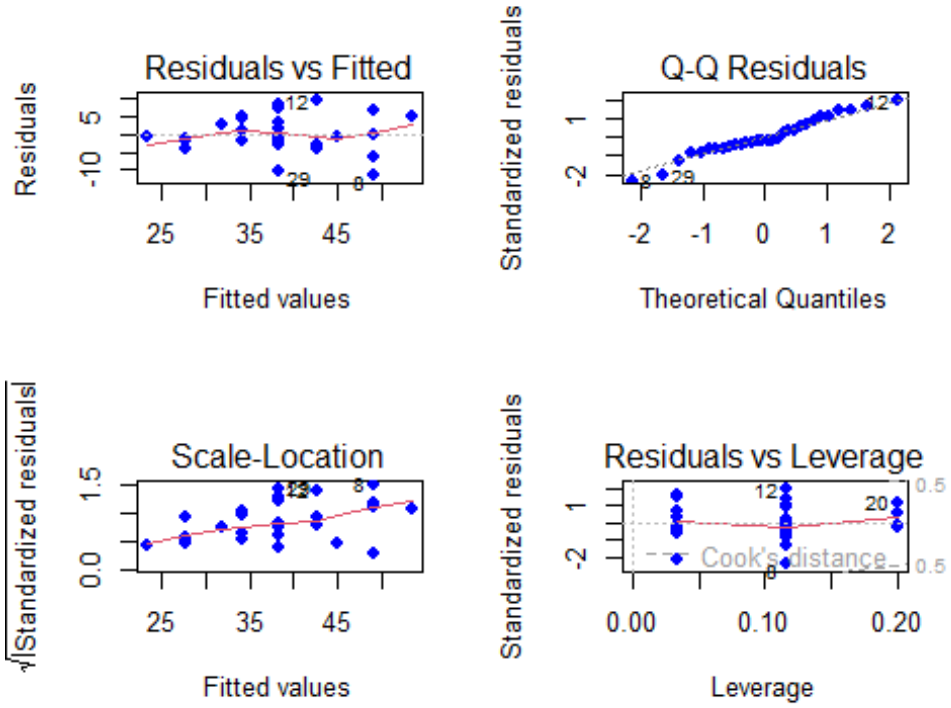

```
##      StudRes        Hat       CookD
## 8  -2.535832 0.11666667 0.235696235
## 12  2.043589 0.11666667 0.164507739
## 19 -0.159511 0.20000000 0.002199712
## 20  1.154355 0.20000000 0.109693544
## 29 -2.216952 0.03333333 0.049338917
```

Mediante el influencePlot podemos ver nuevamente datos que tienen comportamientos importantes para analizar, sin embargo no sinifica que todos sean datos atípicos, aquí podemos ver que nos arroja los valores de los datos 8, 12, 19, 20 y 29 y que nuevamente aparecen los 2 datos (19 y 20) que ya habíamos determinado como atípico. En la gráfica podemos reafirmar estos datos.

Plot del modelo

Gráfica y detecta atípicos o influyentes en los gráficos: * Residuos vs valores ajustados * Qqplot de los residuos * Residuos estandarizados vs valores ajustados * Residuos estandarizados vs Distancia de Leverage y de Cook

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



Mediante estas graficas podemos analizar el comportamiento en general de los datos, este análisis ya se hizo anteriormente y podemos nuevamente concluir que nuestros datos son normales y en el caso de la gráfica Residual vs Leverage podemos obeservar que nos grafica la distancia de Cook como una ligera curva.

Conclusión

Mediante la distancia de Leverage pudimos identificar que existen 2 valores atipicos para el eje X, mientras que para el eje y la estandarización extrema de los residuos nos indica que no existen datos atipicos con referencia al eje y. Sin embargo, debido al análisis de datos influyentes determinamos que no es necesario quitar los datos atípicos ya que nuestros datos atípicos no resultan ser influyentes a nuestro modelo. Unicamente quitamos los datos atípicos cuando resultan ser influyentes.