A3-Regresión Múltiple-Detección datos atípicos

Catherine Rojas

2024-09-24

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.

```
library(readr)

## Warning: package 'readr' was built under R version 4.3.3

data <- read_csv("AlCorte.csv")

## Rows: 30 Columns: 5
## ── Column specification
────────────────────────────────────────────────────────
## Delimiter: ","
## dbl (5): Fuerza, Potencia, Temperatura, Tiempo, Resistencia
##
## ℹ Use `spec()` to retrieve the full column specification for this
data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet
this message.

head(data)

## # A tibble: 6 × 5
##   Fuerza Potencia Temperatura Tiempo Resistencia
##    <dbl>    <dbl>       <dbl>  <dbl>       <dbl>
## 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
```

1. Haz un análisis descriptivo de los datos: medidas principales y gráficos

```
# Cargar las librerías necesarias
library(readr)
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.3.3
```

```r
library(dplyr)

## Warning: package 'dplyr' was built under R version 4.3.3

##
## Attaching package: 'dplyr'

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

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

# Medidas descriptivas básicas
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
```

Observaciones

- Todas las variables parecen tener distribuciones relativamente simétricas, con medias y medianas muy cercanas, lo que indica que probablemente no hay grandes sesgos en la distribución de las observaciones.

```r
# Cálculo de la desviación estándar para cada variable
sapply(data, sd, na.rm = TRUE)

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

Observaciones

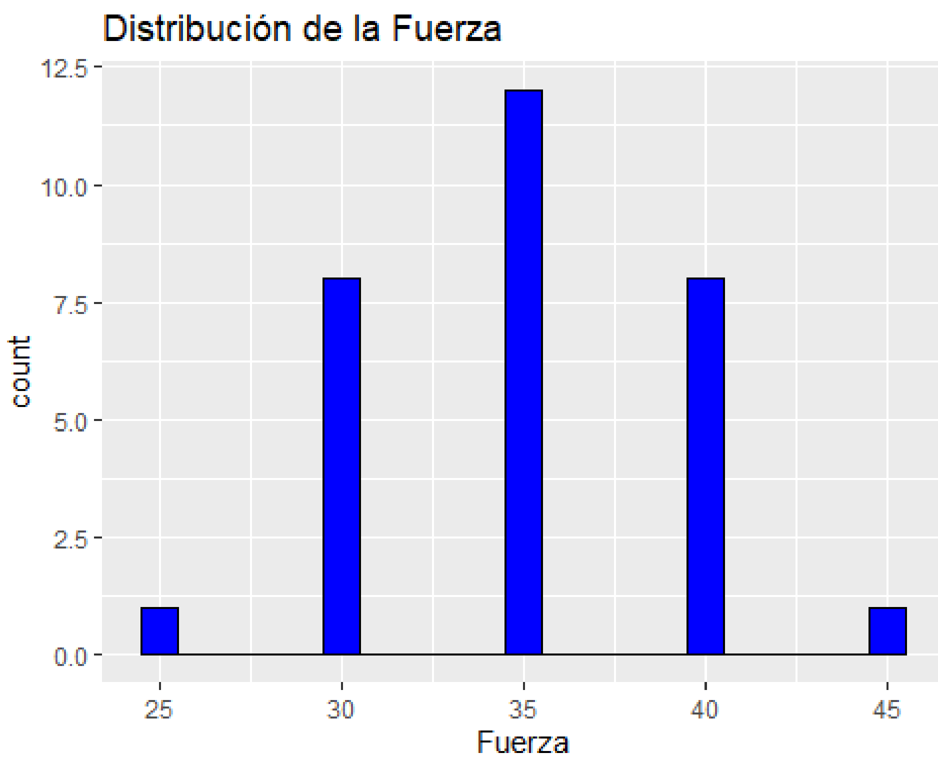
- Las variables potencia y temperatura tienen las mayores desviaciones estándar, lo que indica que estas dos variables muestran la mayor variabilidad en el experimento.

- Fuerza y tiempo tienen desviaciones estándar más bajas, lo que sugiere que estos valores están más agrupados cerca de sus respectivas medias.

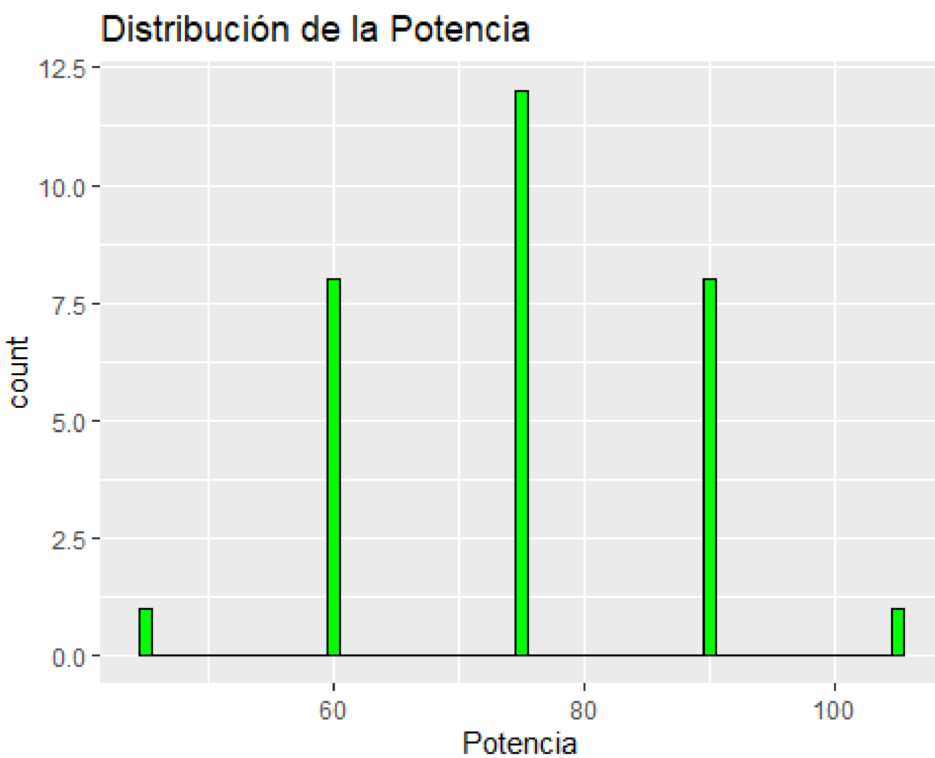
- Resistencia al corte presenta una desviación estándar moderada, lo cual es razonable dado que la resistencia está influenciada por múltiples factores, pero no es extremadamente dispersa.

```r
# Histogramas para visualizar la distribución de las variables
ggplot(data, aes(x = Fuerza)) + geom_histogram(binwidth = 1, fill =
```

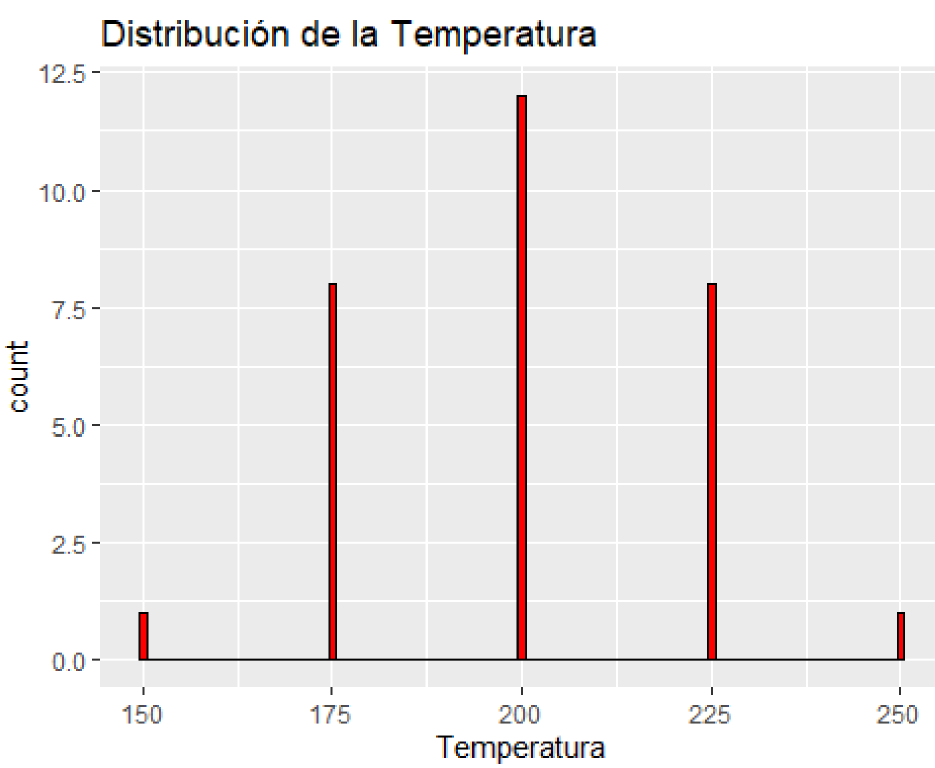
```
"blue", color = "black") +
  ggtitle("Distribución de la Fuerza")
```



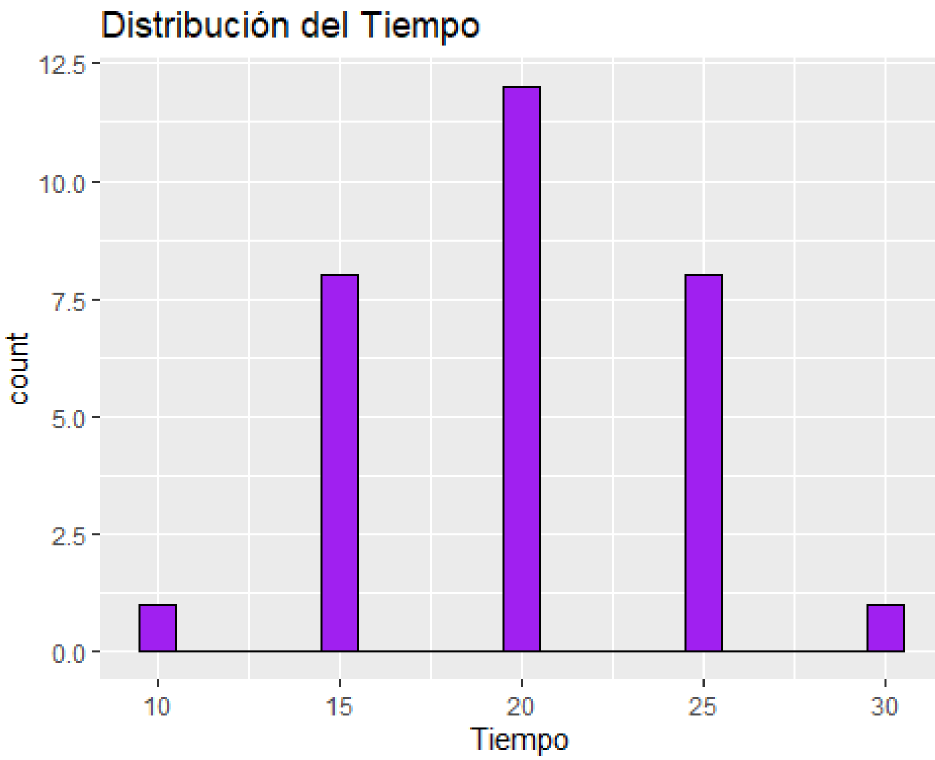

```
ggplot(data, aes(x = Potencia)) + geom_histogram(binwidth = 1, fill =
"green", color = "black") +
  ggtitle("Distribución de la Potencia")
```



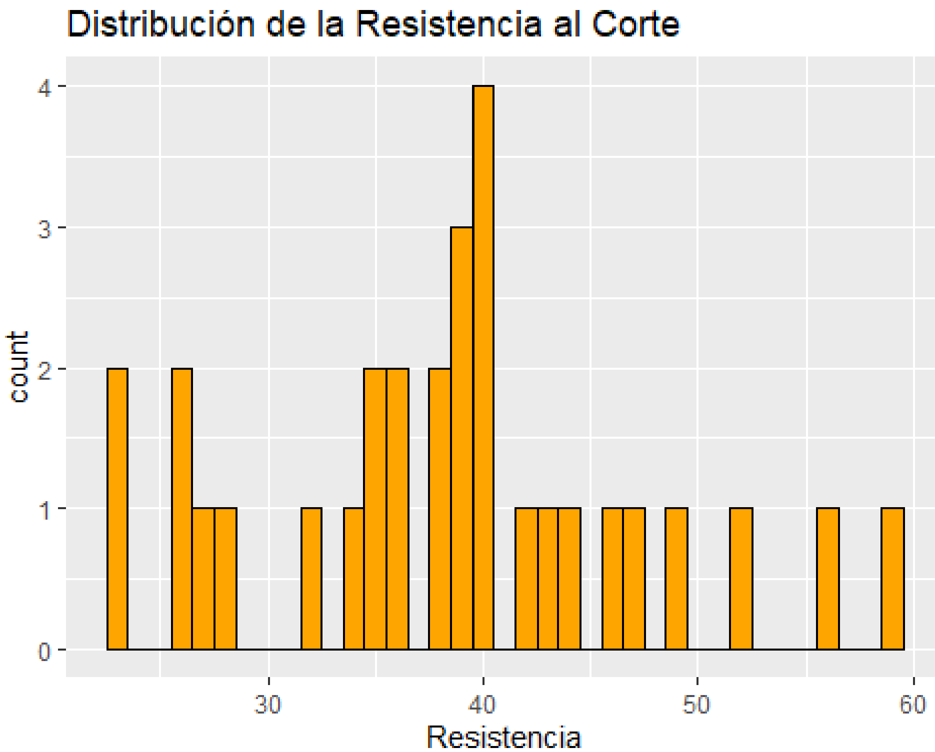

```r
ggplot(data, aes(x = Temperatura)) + geom_histogram(binwidth = 1, fill =
"red", color = "black") +
  ggtitle("Distribución de la Temperatura")
```



```r
ggplot(data, aes(x = Tiempo)) + geom_histogram(binwidth = 1, fill =
"purple", color = "black") +
  ggtitle("Distribución del Tiempo")
```



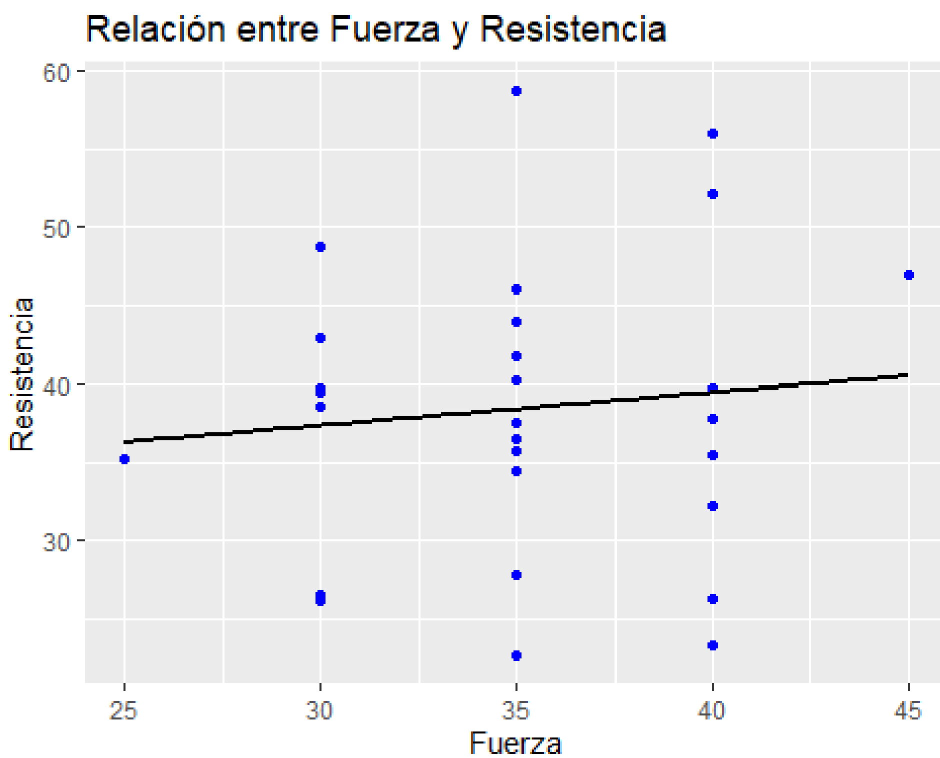

```r
ggplot(data, aes(x = Resistencia)) + geom_histogram(binwidth = 1, fill =
"orange", color = "black") +
  ggtitle("Distribución de la Resistencia al Corte")
```



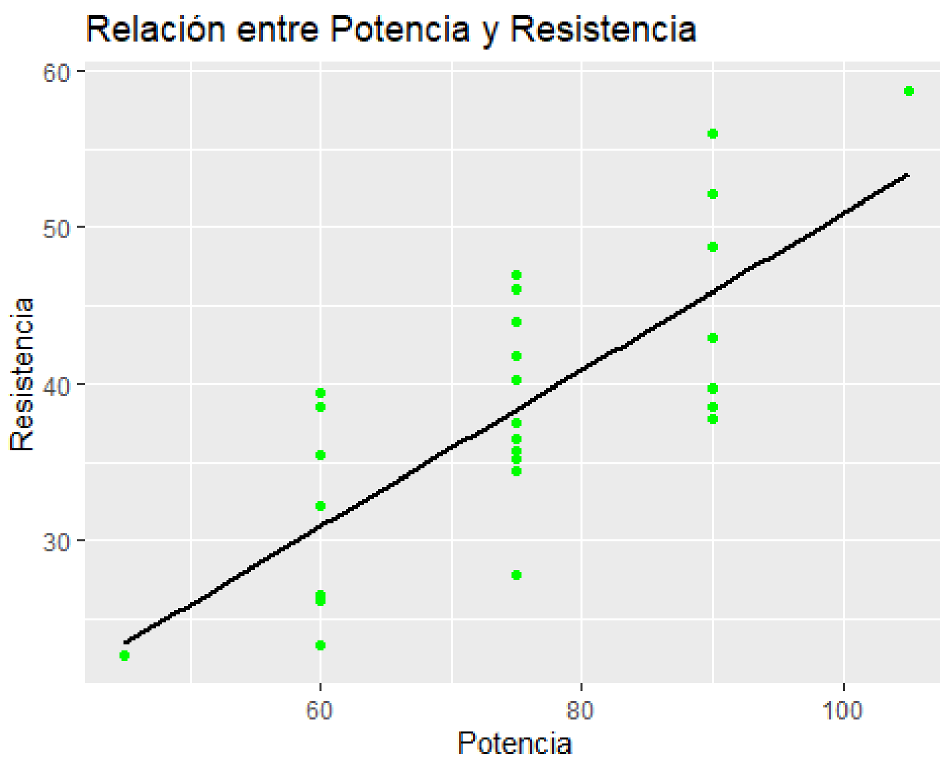
Observaciones

- Las variables fuerza, potencia, temperatura y tiempo tienen una distribución discreta, concentrándose en valores específicos, lo que podría reflejar las condiciones del experimento. En cambio, la resistencia al corte muestra una distribución más dispersa, lo que indica que está influenciada de manera más compleja por las otras variables.

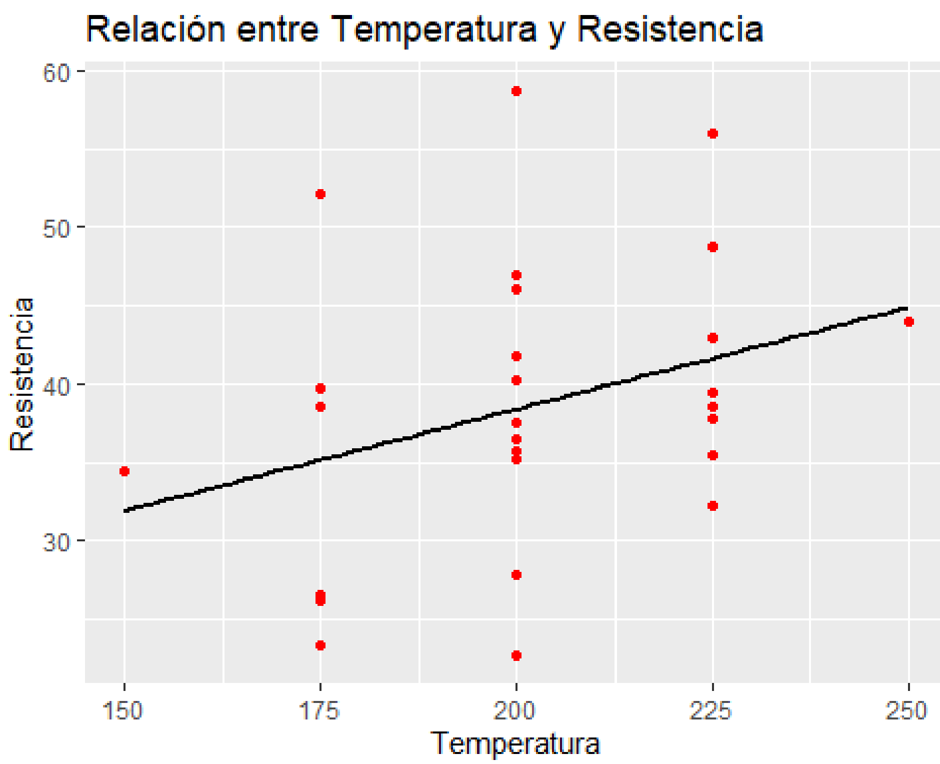
```r
# Gráficos de dispersión para ver relaciones entre variables
ggplot(data, aes(x = Fuerza, y = Resistencia)) +
  geom_point(color = "blue") +
  geom_smooth(method = "lm", color = "black", se = FALSE) +
  ggtitle("Relación entre Fuerza y Resistencia")

## `geom_smooth()` using formula = 'y ~ x'
```



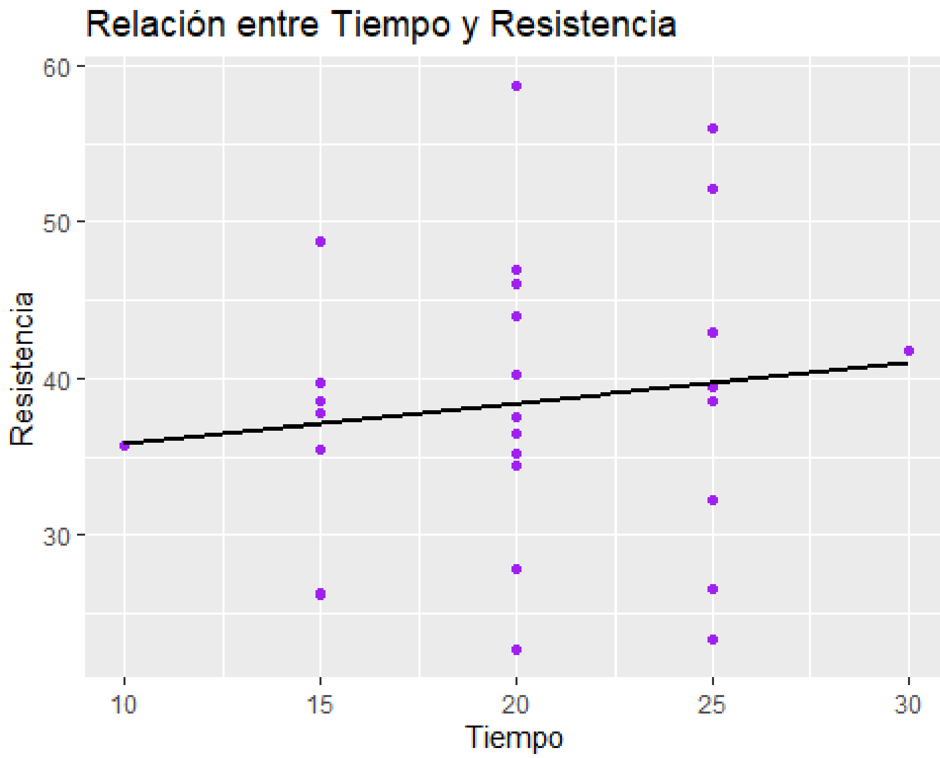

```
ggplot(data, aes(x = Potencia, y = Resistencia)) +
  geom_point(color = "green") +
  geom_smooth(method = "lm", color = "black", se = FALSE) +
  ggtitle("Relación entre Potencia y Resistencia")

## `geom_smooth()` using formula = 'y ~ x'
```




```r
ggplot(data, aes(x = Temperatura, y = Resistencia)) +
  geom_point(color = "red") +
  geom_smooth(method = "lm", color = "black", se = FALSE) +
  ggtitle("Relación entre Temperatura y Resistencia")

## `geom_smooth()` using formula = 'y ~ x'
```




```r
ggplot(data, aes(x = Tiempo, y = Resistencia)) +
  geom_point(color = "purple") +
  geom_smooth(method = "lm", color = "black", se = FALSE) +
  ggtitle("Relación entre Tiempo y Resistencia")

## `geom_smooth()` using formula = 'y ~ x'
```



Observaciones * Potencia parece ser la variable que más influencia tiene sobre la resistencia al corte, ya que muestra una correlación positiva clara.

- Temperatura también muestra una correlación positiva con la resistencia, pero no es tan fuerte ni tan consistente.
- Fuerza y tiempo tienen una relación mucho más débil con la resistencia, lo que sugiere que tienen un menor impacto en los resultados del experimento.

```
# Correlaciones entre las variables
cor(data, use = "complete.obs")
```

```
##                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
```

Observaciones

- Potencia es la variable con la mayor correlación con la resistencia al corte (0.7594), lo que indica que es un factor determinante en el experimento.
- Temperatura tiene una correlación moderada con la resistencia (0.3293), lo que sugiere que también afecta, aunque en menor medida.

- Fuerza y tiempo tienen correlaciones muy débiles con la resistencia, lo que indica que estos factores tienen poca influencia.
- No existen correlaciones entre las variables predictoras (fuerza, potencia, temperatura, tiempo), lo que implica que pueden considerarse independientes entre sí en este conjunto de datos.

```
# Definir las variables
x1 <- data$Fuerza
x2 <- data$Potencia
x3 <- data$Temperatura
x4 <- data$Tiempo

y <- data$Resistencia

# Crear un data frame con las variables
df <- data.frame(Fuerza = x1, Potencia = x2, Temperatura = x3, Tiempo =
x4, Resistencia = y)
```

2. 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 □□) Variación explicada por el modelo

```
# Modelo completo
model <- lm(y~x1+x2+x3+x4)
summary(model)

##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x4)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.0900  -1.7608  -0.3067   2.4392   7.5933
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -37.47667   13.09964  -2.861  0.00841 **
## x1            0.21167    0.21057   1.005  0.32444
## x2            0.49833    0.07019   7.100 1.93e-07 ***
## x3            0.12967    0.04211   3.079  0.00499 **
## x4            0.25833    0.21057   1.227  0.23132
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.158 on 25 degrees of freedom
## Multiple R-squared:  0.714,  Adjusted R-squared:  0.6682
## F-statistic:  15.6 on 4 and 25 DF,  p-value: 1.592e-06
```

Interpretación

- Economía de las variables: Potencia y temperatura son las variables más relevantes en este modelo para predecir la resistencia al corte, ya que son las únicas con coeficientes estadísticamente significativos.

- Significación global (Prueba para el modelo): El valor p del modelo completo es 1.592e-06, lo que indica que el modelo globalmente es significativo. Esto sugiere que, al menos una de las variables explicativas (x1, x2, x3, x4) tiene un efecto significativo sobre la resistencia.

- Significación individual (Prueba para cada βi): En este caso, fuerza y tiempo no parecen ser relevantes, ya que sus coeficientes no son estadísticamente significativos.

- Variación explicada por el modelo: El modelo en su conjunto es significativo, explicando el 66% de la variabilidad en la resistencia.

```
library(car)

## Warning: package 'car' was built under R version 4.3.3

## Loading required package: carData

## Warning: package 'carData' was built under R version 4.3.3

##
## Attaching package: 'car'

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

# Calcular el VIF
vif_values <- vif(model)
vif_values

## x1 x2 x3 x4
##  1  1  1  1

# Identificar si hay multicolinealidad (VIF > 10)
if (any(vif_values > 10)) {
  cat("Advertencia: Se detectó multicolinealidad en algunas
variables.\n")
} else {
```

```r
  cat("No se detectó multicolinealidad preocupante.\n")
}

## No se detectó multicolinealidad preocupante.
```

Interpretación

- Los valores de VIF menores a 10 indican que no hay evidencia de multicolinealidad preocupante en las variables del modelo. En este caso, los valores son exactamente 1 para todas las variables, lo que significa que no existe colinealidad entre las variables explicativas.
- La multicolinealidad ocurre cuando una o más variables independientes están altamente correlacionadas entre sí, lo que puede inflar los errores estándar y afectar las estimaciones de los coeficientes. Sin embargo, este no es el caso, por lo que se puede confiar en las estimaciones de los coeficientes.

Se aplican diferentes métodos para determinar que variables son las efectivas para realizar un modelo y se evalua en AIC y BIC en cada modelo

```r
# Selección hacia atrás (Elimina variables menos significativas)
model_backward <- step(model, direction = "backward", trace = 1) # AIC

## Start:  AIC=102.96
## y ~ x1 + x2 + x3 + x4
##
##        Df Sum of Sq     RSS    AIC
## - x1    1     26.88  692.00 102.15
## - x4    1     40.04  705.16 102.72
## <none>               665.12 102.96
## - x3    1    252.20  917.32 110.61
## - x2    1   1341.01 2006.13 134.08
##
## Step:  AIC=102.15
## y ~ x2 + x3 + x4
##
##        Df Sum of Sq     RSS    AIC
## - x4    1     40.04  732.04 101.84
## <none>               692.00 102.15
## - x3    1    252.20  944.20 109.47
## - x2    1   1341.02 2033.02 132.48
##
## Step:  AIC=101.84
## y ~ x2 + x3
##
##        Df Sum of Sq     RSS    AIC
## <none>               732.04 101.84
```

```
## - x3    1     252.2  984.24 108.72
## - x2    1    1341.0 2073.06 131.07
```

```r
summary(model_backward)
```

```
##
## Call:
## lm(formula = y ~ x2 + x3)
##
## 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 *
## x2            0.49833    0.07086   7.033 1.47e-07 ***
## x3            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
```

```r
# Selección hacia atrás (Elimina variables menos significativas)
n <- nrow(data)

model_backward <- step(model, direction = "backward", k=log(n)) # BIC
```

```
## Start:  AIC=109.97
## y ~ x1 + x2 + x3 + x4
##
##        Df Sum of Sq     RSS    AIC
## - x1    1     26.88  692.00 107.76
## - x4    1     40.04  705.16 108.32
## <none>               665.12 109.97
## - x3    1    252.20  917.32 116.21
## - x2    1   1341.01 2006.13 139.69
##
## Step:  AIC=107.76
## y ~ x2 + x3 + x4
##
##        Df Sum of Sq     RSS    AIC
## - x4    1     40.04  732.04 106.04
## <none>               692.00 107.76
## - x3    1    252.20  944.20 113.68
## - x2    1   1341.02 2033.02 136.69
##
## Step:  AIC=106.04
## y ~ x2 + x3
```

```
##
##        Df Sum of Sq     RSS    AIC
## <none>               732.04 106.04
## - x3    1     252.2  984.24 111.52
## - x2    1    1341.0 2073.06 133.87
```

```r
summary(model_backward)
```

```
##
## Call:
## lm(formula = y ~ x2 + x3)
##
## 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 *
## x2            0.49833    0.07086   7.033 1.47e-07 ***
## x3            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
```

```r
# Selección hacia adelante (Agrega variables una por una)
model_null <- lm(Resistencia ~ 1, data = df)  # Modelo nulo (sin
predictores)
model_forward <- step(model_null, direction = "forward", scope =
formula(model), trace = 1)
```

```
## Start:  AIC=132.51
## Resistencia ~ 1
##
##        Df Sum of Sq     RSS    AIC
## + x2    1   1341.01  984.24 108.72
## + x3    1    252.20 2073.06 131.07
## <none>              2325.26 132.51
## + x4    1     40.04 2285.22 133.99
## + x1    1     26.88 2298.38 134.16
##
## Step:  AIC=108.72
## Resistencia ~ x2
##
##        Df Sum of Sq    RSS    AIC
## + x3    1   252.202 732.04 101.84
## <none>              984.24 108.72
## + x4    1    40.042 944.20 109.47
```

```
## + x1    1    26.882 957.36 109.89
##
## Step:  AIC=101.84
## Resistencia ~ x2 + x3
##
##        Df Sum of Sq    RSS    AIC
## <none>              732.04 101.84
## + x4    1    40.042 692.00 102.15
## + x1    1    26.882 705.16 102.72
```

```
summary(model_forward)
```

```
##
## Call:
## lm(formula = Resistencia ~ x2 + x3, data = df)
##
## 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 *
## x2            0.49833    0.07086   7.033 1.47e-07 ***
## x3            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
```

```
# Selección hacia adelante (Agrega variables una por una)
model_forward <- step(model_null, direction = "forward", scope =
formula(model), k=log(n)) #BIC
```

```
## Start:  AIC=133.91
## Resistencia ~ 1
##
##        Df Sum of Sq     RSS    AIC
## + x2    1   1341.01  984.24 111.52
## + x3    1    252.20 2073.06 133.87
## <none>              2325.26 133.91
## + x4    1     40.04 2285.22 136.79
## + x1    1     26.88 2298.38 136.97
##
## Step:  AIC=111.52
## Resistencia ~ x2
##
##        Df Sum of Sq    RSS    AIC
## + x3    1   252.202 732.04 106.04
```

```
## <none>              984.24 111.52
## + x4    1    40.042 944.20 113.68
## + x1    1    26.882 957.36 114.09
##
## Step:  AIC=106.04
## Resistencia ~ x2 + x3
##
##        Df Sum of Sq    RSS    AIC
## <none>              732.04 106.04
## + x4    1    40.042 692.00 107.76
## + x1    1    26.882 705.16 108.32
```

```r
summary(model_forward)
```

```
##
## Call:
## lm(formula = Resistencia ~ x2 + x3, data = df)
##
## 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 *
## x2            0.49833    0.07086   7.033 1.47e-07 ***
## x3            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
```

```r
# Selección mixta (Combina la selección hacia adelante y hacia atrás)
model_both <- step(model, direction = "both", trace = 1) #AIC
```

```
## Start:  AIC=102.96
## y ~ x1 + x2 + x3 + x4
##
##        Df Sum of Sq     RSS    AIC
## - x1    1     26.88  692.00 102.15
## - x4    1     40.04  705.16 102.72
## <none>               665.12 102.96
## - x3    1    252.20  917.32 110.61
## - x2    1   1341.01 2006.13 134.08
##
## Step:  AIC=102.15
## y ~ x2 + x3 + x4
##
##        Df Sum of Sq     RSS    AIC
```

```
## - x4    1     40.04  732.04 101.84
## <none>              692.00 102.15
## + x1    1     26.88  665.12 102.96
## - x3    1    252.20  944.20 109.47
## - x2    1   1341.02 2033.02 132.48
##
## Step:  AIC=101.84
## y ~ x2 + x3
##
##        Df Sum of Sq     RSS    AIC
## <none>              732.04 101.84
## + x4    1     40.04  692.00 102.15
## + x1    1     26.88  705.16 102.72
## - x3    1    252.20  984.24 108.72
## - x2    1   1341.01 2073.06 131.07
```

```r
summary(model_both)
```

```
##
## Call:
## lm(formula = y ~ x2 + x3)
##
## 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 *
## x2            0.49833    0.07086   7.033 1.47e-07 ***
## x3            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 modelo encontrado

```r
# Selección mixta (Combina la selección hacia adelante y hacia atrás)
model_both <- step(model, direction = "both", k=log(n)) #BIC
```

```
## Start:  AIC=109.97
## y ~ x1 + x2 + x3 + x4
##
##        Df Sum of Sq     RSS    AIC
## - x1    1     26.88  692.00 107.76
## - x4    1     40.04  705.16 108.32
## <none>              665.12 109.97
## - x3    1    252.20  917.32 116.21
```

```
## - x2    1   1341.01 2006.13 139.69
##
## Step:  AIC=107.76
## y ~ x2 + x3 + x4
##
##        Df Sum of Sq     RSS    AIC
## - x4    1     40.04  732.04 106.04
## <none>               692.00 107.76
## + x1    1     26.88  665.12 109.97
## - x3    1    252.20  944.20 113.68
## - x2    1   1341.02 2033.02 136.69
##
## Step:  AIC=106.04
## y ~ x2 + x3
##
##        Df Sum of Sq     RSS    AIC
## <none>               732.04 106.04
## + x4    1     40.04  692.00 107.76
## + x1    1     26.88  705.16 108.32
## - x3    1    252.20  984.24 111.52
## - x2    1   1341.01 2073.06 133.87
```

```r
summary(model_both)
```

```
##
## Call:
## lm(formula = y ~ x2 + x3)
##
## 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 *
## x2            0.49833    0.07086   7.033 1.47e-07 ***
## x3            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
```

```r
model_sig <- lm(y~x2+x3)
summary(model_sig)
```

```
##
## Call:
## lm(formula = y ~ x2 + x3)
##
```

```
## 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 *  
## x2            0.49833    0.07086   7.033 1.47e-07 ***
## x3            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
```

Interpretación

- Economía de las variables: El modelo final incluye solo dos variables: **potencia (x2)** y **temperatura (x3)**. Cada uno de los métodos aplicados evaluando el AIC y el BIC determinaron que estas dos variables son suficientes para explicar una parte significativa de la variación en la resistencia, eliminando las variables menos importantes, en este caso, fuerza y tiempo.

- Significación global (Prueba para el modelo): El valor **p** global del modelo es **1.674e-07**, lo que indica que el modelo es altamente significativo en su conjunto. Esto sugiere que las variables seleccionadas (potencia y temperatura) explican de manera significativa la variabilidad en la resistencia al corte.

- Significación individual (Prueba para cada βi): El coeficiente de potencia es **0.49833**, con un valor p extremadamente bajo (**1.47e-07**), lo que indica que es una variable predictora muy significativa. Por otro lado, temperatura tiene un coeficiente de **0.12967**, con un valor p de **0.00508**, lo que también indica que es una variable predictora significativa.

- Variación explicada por el modelo: El $\mathbf{R^2}$ **ajustado** es **0.6619**, lo que lo que indica que el 66.1% de la variabilidad en la resistencia es explicada por la potencia y la temperatura. Esto muestra que el modelo tiene un buen ajuste a los datos.

3. Analiza la validez del modelo encontrado

Para todas las hipótesis se utiliza un alpha de 0.05.

Residuos con media cero

$$H_0\text{:La media de los residuos es cero } (\mu = 0)$$

$$H_1\text{:La media de los residuos no es cero } (\mu \neq 0)$$

```
# Obtener los residuos del modelo
residuos <- residuals(model_sig)

# Prueba t para la media de los residuos
t_test_residuos <- t.test(residuos, mu = 0)
t_test_residuos

##
##  One Sample t-test
##
## data:  residuos
## 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

# Calcular la media de los residuos
media_residuos <- mean(residuos)
cat("Media de los residuos:", media_residuos, "\n")

## Media de los residuos: 8.133323e-17
```

Interpretación

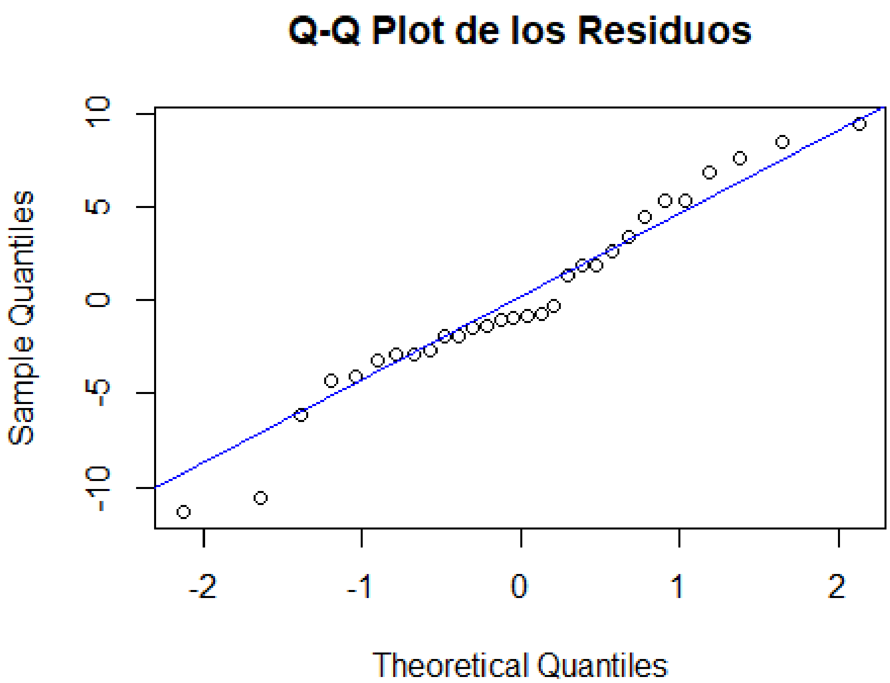
- El One Sample t-test no rechaza la hipótesis nula, lo que significa que la media de los residuos no es significativamente diferente de 0.

Normalidad de los residuos

$$H_0\text{:Los residuos siguen una distribución normal}$$

$$H_1\text{:Los residuos no siguen una distribución normal}$$

```
# Cargar las librerías necesarias
library(nortest)
library(tseries)

## Warning: package 'tseries' was built under R version 4.3.3

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo

# Gráfico Q-Q para los residuos
qqnorm(residuos, main = "Q-Q Plot de los Residuos")
qqline(residuos, col = "blue")
```




```r
# Anderson-Darling Test para los residuos
ad_test_residuos <- ad.test(residuos)
ad_test_residuos

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

Interpretación

- El Q-Q plot sugiere que los residuos se aproximan a una distribución normal, aunque hay algunas desviaciones en los valores extremos (tanto en los cuantiles más bajos como en los más altos). Estas desviaciones no parecen lo suficientemente grandes como para invalidar el modelo, pero podría ser útil revisar la presencia de valores atípicos o heterocedasticidad.

Homocedasticidad, independencia y linealidad.

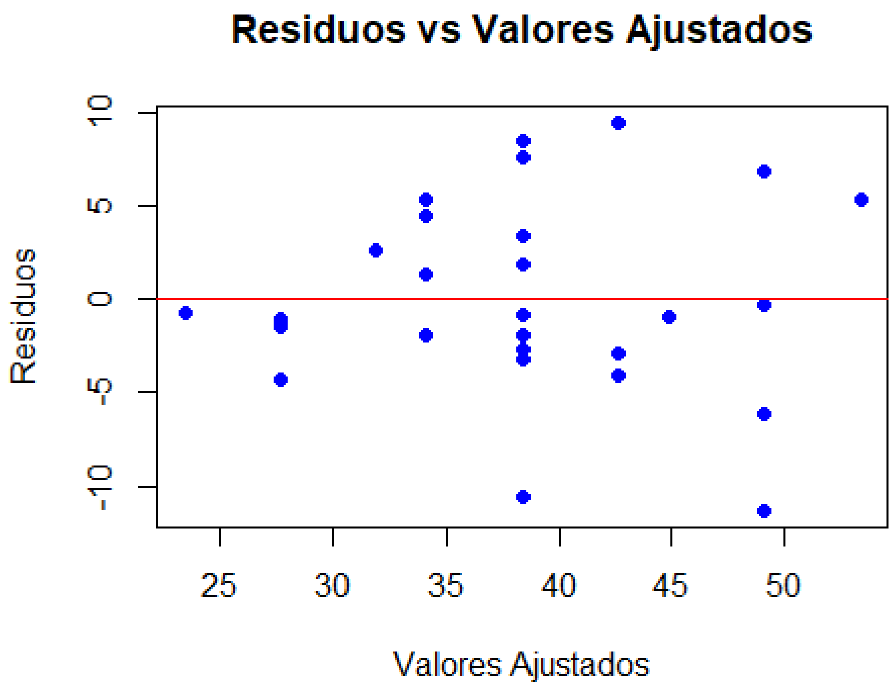
Homocedasticidad y Linealidad

H_0:Los residuos tienen una varianza constante (homocedasticidad)

H_1:Los residuos no tienen una varianza constante (heterocedasticidad)

```r
# Obtener los valores ajustados y los residuos
fitted_values <- fitted(model_sig)
```

```r
residuos <- residuals(model_sig)

# Graficar los residuos vs valores ajustados
plot(fitted_values, residuos,
     main = "Residuos vs Valores Ajustados",
     xlab = "Valores Ajustados",
     ylab = "Residuos",
     pch = 16, col = "blue")

# Añadir una línea horizontal en y = 0
abline(h = 0, col = "red")
```



Interpretación

- El gráfico de residuos vs valores ajustados no muestra patrones claros, lo que indica que no hay problemas evidentes de heterocedasticidad o de falta de linealidad en el modelo. Aunque algunos residuos están más alejados de la línea de 0, no se observa un problema importante de valores atípicos o influencia de los mismos. El modelo parece cumplir con los supuestos de homocedasticidad y linealidad.

```r
library(lmtest)

## Warning: package 'lmtest' was built under R version 4.3.3

## Loading required package: zoo

## Warning: package 'zoo' was built under R version 4.3.3
```

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

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

# Prueba de Breusch-Pagan (homocedasticidad)
bptest(model_sig)

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

# Prueba de Goldfeld-Quandt (homocedasticidad)
gqtest(model_sig)

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

Observaciones

- Ambas pruebas, tanto Breusch-Pagan como Goldfeld-Quandt, muestran valores p mayores a 0.05, lo que indica que no hay evidencia de heterocedasticidad en el modelo. Los residuos parecen tener varianza constante, esto es una señal de que el modelo es adecuado en términos de homocedasticidad.

```
# Prueba de Ramsey RESET (linealidad)
resettest(model_sig)

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

Observaciones

- El modelo está correctamente especificado y no hay necesidad de agregar variables no lineales o de transformar las variables incluidas en el modelo.

Independencia

$$H_0\text{:No hay autocorrelación en los residuos}$$

$$H_1\text{:Hay autocorrelación en los residuos}$$

```r
library(lmtest)

# Prueba de Durbin-Watson para verificar independencia de los residuos
dw_test <- dwtest(model_sig)
dw_test
```

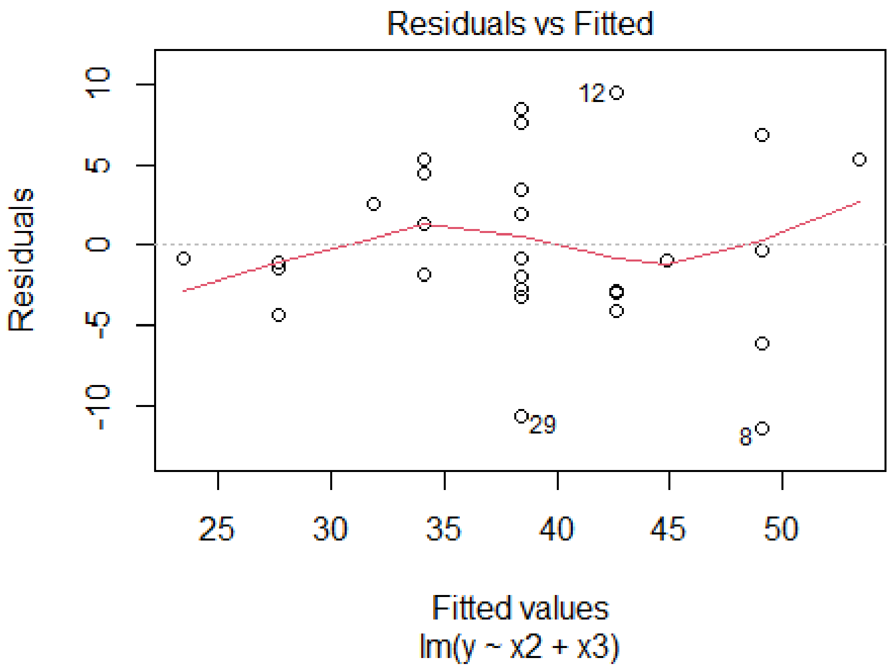
```
##
##  Durbin-Watson test
##
## data:  model_sig
## DW = 2.3511, p-value = 0.8267
## alternative hypothesis: true autocorrelation is greater than 0
```

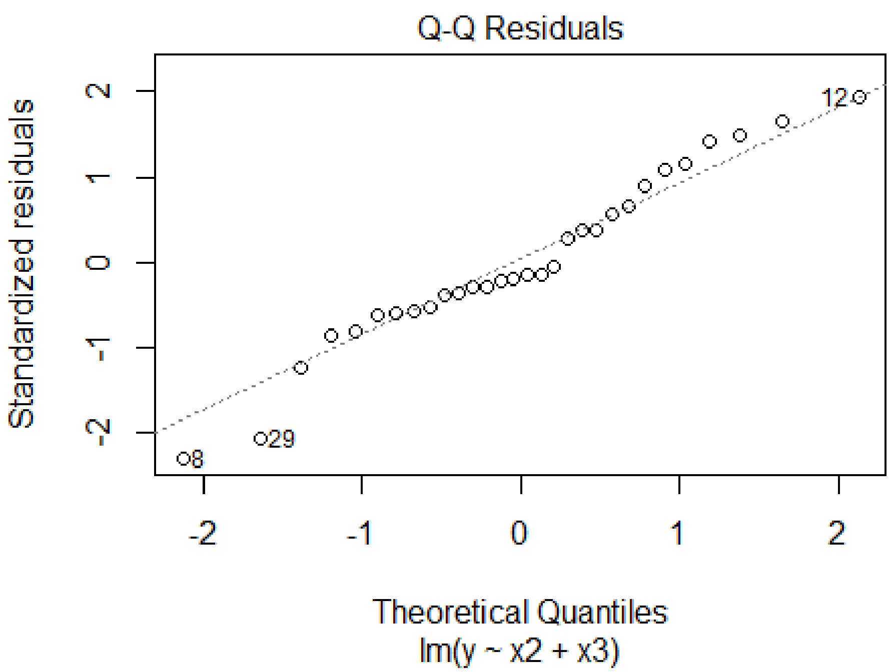
Observación * No se encontró evidencia de autocorrelación significativa en los residuos, lo que sugiere que los errores del modelo son independientes entre sí. Esto indica que el supuesto de independencia de los errores en el modelo de regresión se cumple.

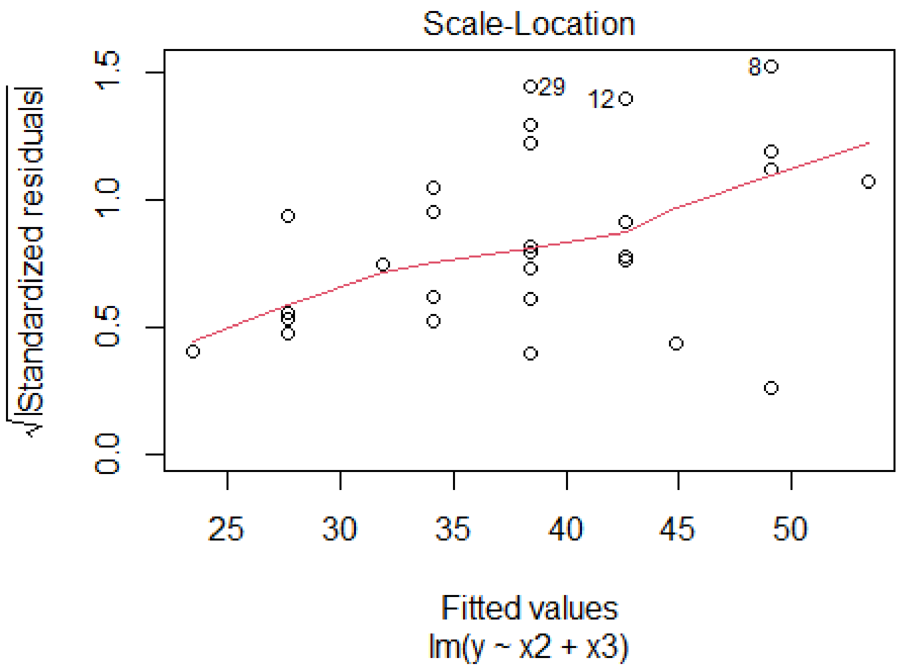
Resultados * El supuesto de que los residuos tienen una media cero se cumple. * El supuesto de normalidad de los residuos no se cumple. * El supuesto de linealidad se cumple. * El supuesto de homocedasticidad se cumple. * El supuesto de independencia de los residuos se cumple.

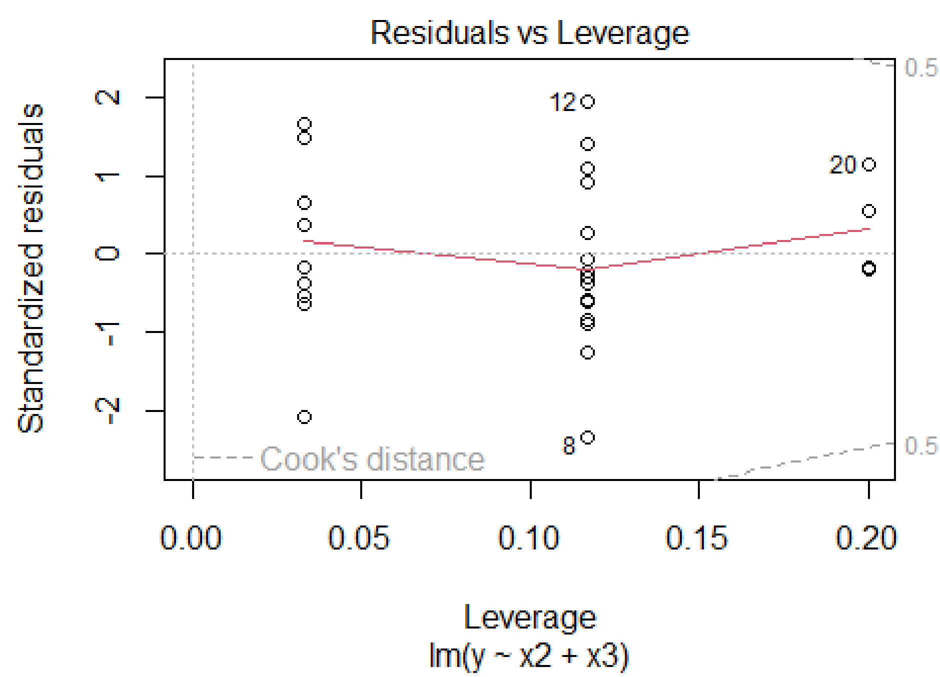
Usa plot(Modelo) para los gráficos y añade pruebas de hipótesis.

```r
# Generar los gráficos de diagnóstico del modelo
plot(model_sig)
```


Residuals vs Fitted
Residuals
Fitted values
lm(y ~ x2 + x3)
12
29
8


Q-Q Residuals
Standardized residuals
Theoretical Quantiles
lm(y ~ x2 + x3)
12
29
8





Interpretación

Gráfico 1: Residuals vs Fitted (Residuos vs Valores Ajustados): **Objetivo**: Evaluar la homocedasticidad y la linealidad del modelo. * Los puntos están distribuidos de manera bastante aleatoria alrededor de la línea horizontal (0), lo que indica que no hay un patrón claro que sugiera heterocedasticidad (es decir, no parece que la varianza de los residuos cambie sistemáticamente con los valores ajustados). Sin embargo, se observa una ligera curvatura en la línea roja (tendencia suave), lo que podría indicar una ligera violación de la linealidad.

Gráfico 2: Q-Q Plot de Residuos: **Objetivo**: Evaluar si los residuos siguen una distribución normal. * La mayoría de los puntos se alinean bien con la línea diagonal, lo que sugiere que los residuos siguen aproximadamente una distribución normal. Sin embargo, algunos puntos en los extremos (cuantiles más bajos y más altos) se desvían de la línea, lo que indica posibles desviaciones de la normalidad en los valores extremos, aunque estas no son graves.

Gráfico 3: Scale-Location (Gráfico de Escala-Ubicación): **Objetivo**: Evaluar la homocedasticidad (varianza constante de los residuos). * La línea roja muestra una ligera pendiente ascendente, lo que podría sugerir una ligera tendencia a una varianza creciente a medida que aumentan los valores ajustados. Aunque no es un patrón fuerte, indica una posible violación leve de la homocedasticidad.

Gráfico 4: Residuals vs Leverage (Residuos vs Apalancamiento): **Objetivo**: Detectar posibles puntos de influencia (observaciones que tienen un alto apalancamiento y podrían tener un efecto significativo en el modelo). * No se observan puntos con **Cook's Distance** mayor a 0.5, lo que indica que no hay observaciones que tengan una influencia significativa en el ajuste del modelo. Los puntos 8 y 12 podrían tener un apalancamiento (leverage) relativamente alto, pero no parecen ser lo suficientemente influyentes como para distorsionar el modelo.

No multicolinealidad de Xi

```r
# Calcular el VIF para detectar multicolinealidad
vif_values <- vif(model_sig)

# Imprimir los valores de VIF
print(vif_values)

## x2 x3
##  1  1

# Interpretar los resultados
if (any(vif_values > 10)) {
  cat("Advertencia: Se detectó multicolinealidad alta en algunas variables (VIF > 10).\n")
} else {
  cat("No se detectó multicolinealidad significativa (VIF < 10).\n")
}

## No se detectó multicolinealidad significativa (VIF < 10).
```

Observación Los valores de VIF menores a 10 indican que no hay evidencia de multicolinealidad preocupante en las variables del modelo. En este caso, los valores son exactamente 1 para todas las variables, lo que significa que no existe colinealidad entre las variables explicativas.

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

Contextualización del efecto de las variables predictoras:

El experimento tiene como objetivo evaluar el impacto de diferentes factores en la resistencia al corte. En este caso, las variables x2 y x3 tienen efectos significativos sobre la variable respuesta.

x2 representa un factor crucial en el experimento. Un aumento en esta variable (potencia) genera un aumento considerable en la resistencia, lo que significa que es fundamental controlar este parámetro para optimizar la resistencia al corte.

x3 como la temperatura, también contribuye al aumento de la resistencia, su impacto es menor que el de x2, lo que sugiere que esta variable también debe gestionarse, pero no es tan crítica como x2.

Conclusiones

- El modelo final es confiable, significativo, y explica una parte importante de la variabilidad en la resistencia al corte.
- Las variables x2 y x3 tienen efectos positivos sobre la variable respuesta, siendo x2 la variable más influyente en el resultado.
- No se encontraron problemas graves de multicolinealidad, normalidad o heterocedasticidad en el modelo, lo que respalda la validez de las inferencias realizadas a partir de los coeficientes estimados.
- En términos del problema, para optimizar la resistencia al corte, al realizar los experimentos debenenfocarse en controlar y aumentar x2 (potencia), sin descuidar x3 (temperatura), aunque su efecto es menor.

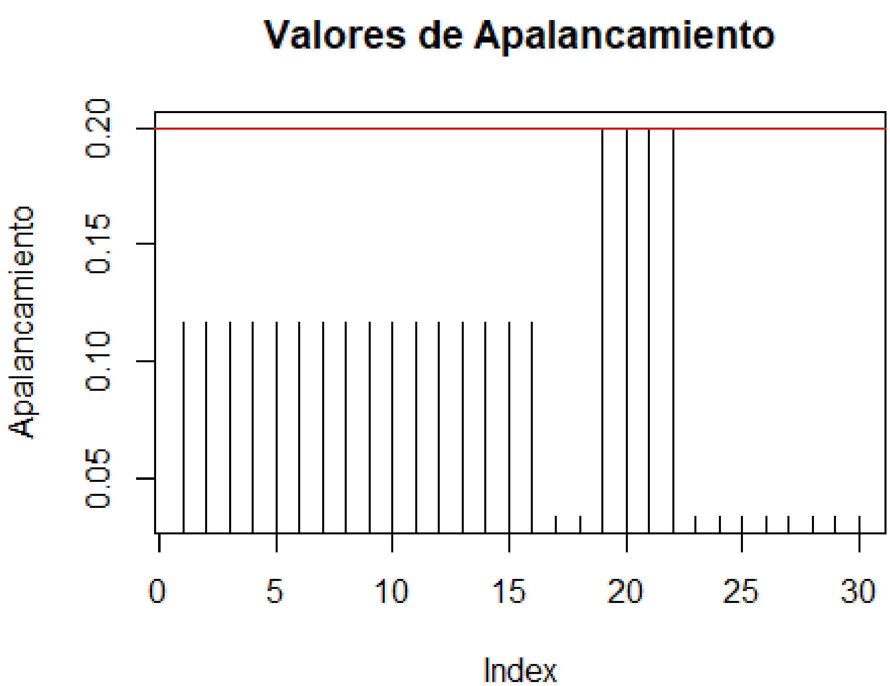
4. Haz el análisis de datos atípicos e incluyentes del mejor modelo encontrado

Detección de datos atípicos en x

```
# Cálculo de leverage (apalancamiento)
leverage <- hatvalues(model_sig)
leverage
```

```
##          1          2          3          4          5          6
7
## 0.11666667 0.11666667 0.11666667 0.11666667 0.11666667 0.11666667
0.11666667
##          8          9         10         11         12         13
14
## 0.11666667 0.11666667 0.11666667 0.11666667 0.11666667 0.11666667
0.11666667
##         15         16         17         18         19         20
21
## 0.11666667 0.11666667 0.03333333 0.03333333 0.20000000 0.20000000
0.20000000
##         22         23         24         25         26         27
28
## 0.20000000 0.03333333 0.03333333 0.03333333 0.03333333 0.03333333
0.03333333
##         29         30
## 0.03333333 0.03333333
```

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




```
# Identificar las observaciones con alto leverage (apalancamiento)
high_leverage_points <- which(leverage > 2 * mean(leverage))       # 2
```

```r
# Muestra las observaciones con leverage alto (índices)
cat("Observaciones con leverage alto:\n")

## Observaciones con leverage alto:

# Ver los valores de leverage para esos puntos:
print(leverage[high_leverage_points])

##  19  20
## 0.2 0.2
```

- Sabemos que la media de los valores de apalancamiento puede ser aproximada usando la fórmula:

$$\text{Media del apalancamiento} = \frac{p}{n}$$

donde:

p es el número de variables predictoras (incluyendo el intercepto).

n es el número de observaciones.

- En este caso, el modelo tiene 2 variables predictoras (potencia y temperatura) más el intercepto, lo que nos da p = 3. El número total de observaciones es 30, de acuerdo a los resultados de hatvalues.

Por lo tanto, la media de los valores de apalancamiento sería:

$$\text{Media del apalancamiento} = \frac{3}{30} = 0.1$$

Interpretación

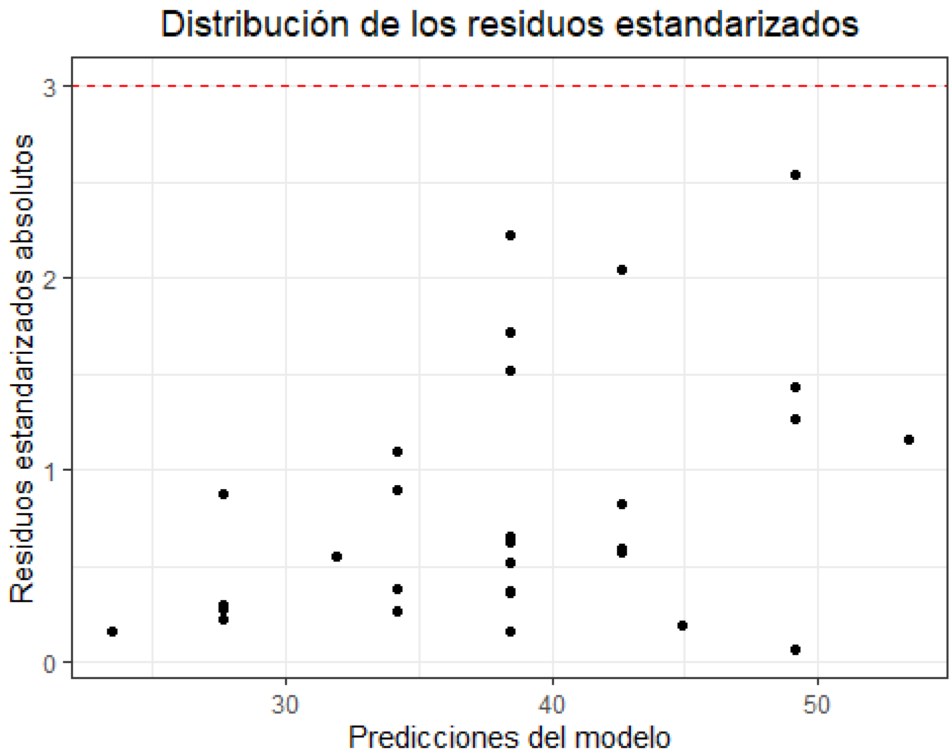
- La mayoría de las observaciones tienen un valor de apalancamiento de aproximadamente 0.1167. Esto indica que estas observaciones no son atípicas o influyentes en términos de las variables predictoras.

- Hay varias observaciones con un apalancamiento bajo, en torno a 0.0333. Estas observaciones tienen poca influencia en el modelo.

- Las observaciones 19, 20, 21, 22 tienen un valor de apalancamiento relativamente alto, de 0.2000. Esto indica que estas observaciones son más influyentes en el modelo. Aunque aún no superan el valor comúnmente considerado como indicativo de un apalancamiento extremadamente alto (que suele ser 2p/n), estos puntos son más atípicos en cuanto a sus valores de las variables predictoras y tienen un mayor impacto en la estimación de los coeficientes del modelo.

Detección de datos atípicos en y. (Estandarización extrema)

```r
library(ggplot2)

# Calcula los residuos estudentizados
residuos_estandarizados <- rstudent(model_sig)
residuos_estandarizados

##           1           2           3           4           5
6
## -0.29928810 -0.27913928 -0.57306020 -0.59350704  0.90139527
0.26638516
##           7           8           9          10          11
12
## -0.06483984 -2.53583165 -0.21876720 -0.87273741 -0.82051976
2.04358872
##          13          14          15          16          17
18
##  1.09233365 -0.37666314 -1.26507229  1.43227007 -0.61917389
1.71793108
##          19          20          21          22          23
24
## -0.15951102  1.15435457  0.54601898 -0.18765390 -0.52152469
0.65579682
##          25          26          27          28          29
30
## -0.36641522 -0.15469425  0.36383979  1.51868782 -2.21695209
0.36383979

# Predicciones del modelo
predicciones <- predict(model_sig)

# Genera el gráfico sin usar un DataFrame explícito
ggplot() +
  geom_hline(yintercept = 3, color = "red", linetype = "dashed") +  #
Línea que indica residuos > 3
  geom_point(aes(x = predicciones, y = abs(residuos_estandarizados),
                 color = ifelse(abs(residuos_estandarizados) > 3, 'red',
'black'))) +
  scale_color_identity() +
  labs(title = "Distribución de los residuos estandarizados",
       x = "Predicciones del modelo",
       y = "Residuos estandarizados absolutos") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
```




```
# Identifica las observaciones atípicas
atipicos <- which(abs(residuos_estandarizados) > 3)

# Muestra las observaciones atípicas
cat("Observaciones atípicas (residuos estandarizados > 3):\n")

## Observaciones atípicas (residuos estandarizados > 3):

print(atipicos)

## named integer(0)

# Ver los valores de los residuos atípicos:
print(residuos_estandarizados[atipicos])

## named numeric(0)
```

- Los valores de rstudent() se refieren a los residuos estudentizados, que son una versión estandarizada de los residuos.

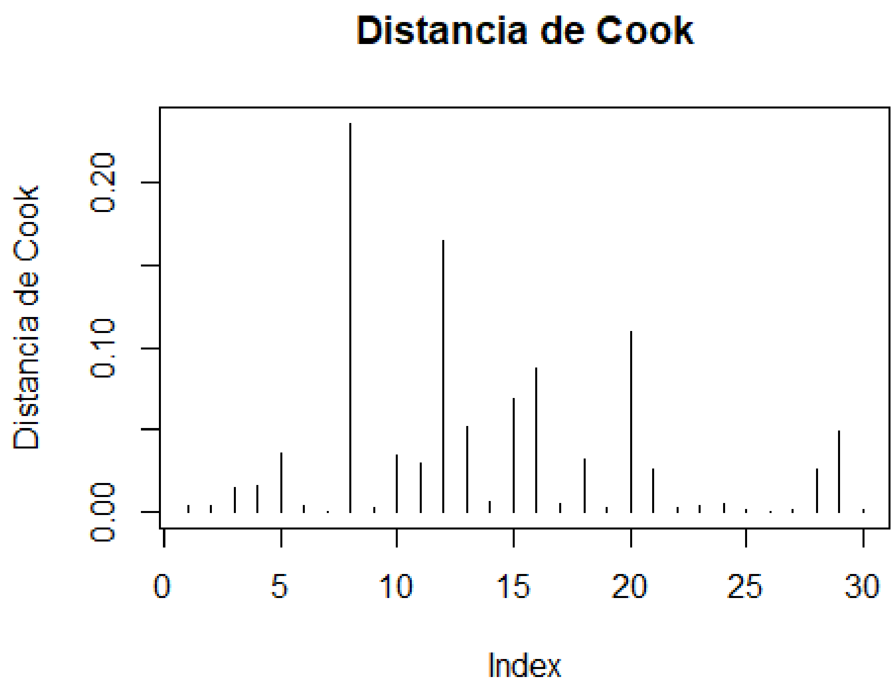
- Valores cercanos a 0 indican que la observación se ajusta bien al modelo, mientras que valores altos o muy bajos indican observaciones que no se ajustan bien.

- Como regla general, valores de residuos estudentizados mayores a 2 o menores a -2 pueden considerarse observaciones potencialmente atípicas en y (dependiendo del tamaño de la muestra y el contexto, a veces se utiliza un umbral de 3 o -3).

- Con tres el umbral de 3, no se observan datos atípicos.

Detección de datos influyentes. Distancia de cook.

```
# Cálculo de la distancia de Cook
cooksd <- cooks.distance(model_sig)
cooksd

##            1            2            3            4            5
6
## 0.0040810940 0.0035516786 0.0148265717 0.0158890789 0.0360211607
0.0032353979
##            7            8            9           10           11
12
## 0.0001921786 0.2356962355 0.0021840222 0.0338312969 0.0300031458
0.1645077394
##           13           14           15           16           17
18
## 0.0521573001 0.0064511061 0.0689254904 0.0869280642 0.0045096144
0.0316364764
##           19           20           21           22           23
24
## 0.0021997116 0.1096935439 0.0255078004 0.0030432424 0.0032129304
0.0050499237
##           25           26           27           28           29
30
## 0.0015943414 0.0002853777 0.0015721209 0.0252869645 0.0493389168
0.0015721209

# Graficar la distancia de Cook
plot(cooksd, type = "h", main = "Distancia de Cook", ylab = "Distancia de
Cook")
abline(h = 1, col = "red") # Límite comúnmente usado para identificar
puntos influyentes
```




```r
# Identificar las observaciones con distancia de Cook mayor a 1
puntos_influyentes <- which(cooksd > 1)

# Mostrar las observaciones influyentes
data[puntos_influyentes, ]
```

```
## # A tibble: 0 × 5
## # ℹ 5 variables: Fuerza <dbl>, Potencia <dbl>, Temperatura <dbl>,
Tiempo <dbl>,
## #   Resistencia <dbl>
```

```r
if (length(puntos_influyentes) > 0) {
  print(data[puntos_influyentes, ])
} else {
  cat("No hay observaciones influyentes con una distancia de Cook mayor a
1.\n")
}
```

```
## No hay observaciones influyentes con una distancia de Cook mayor a 1.
```

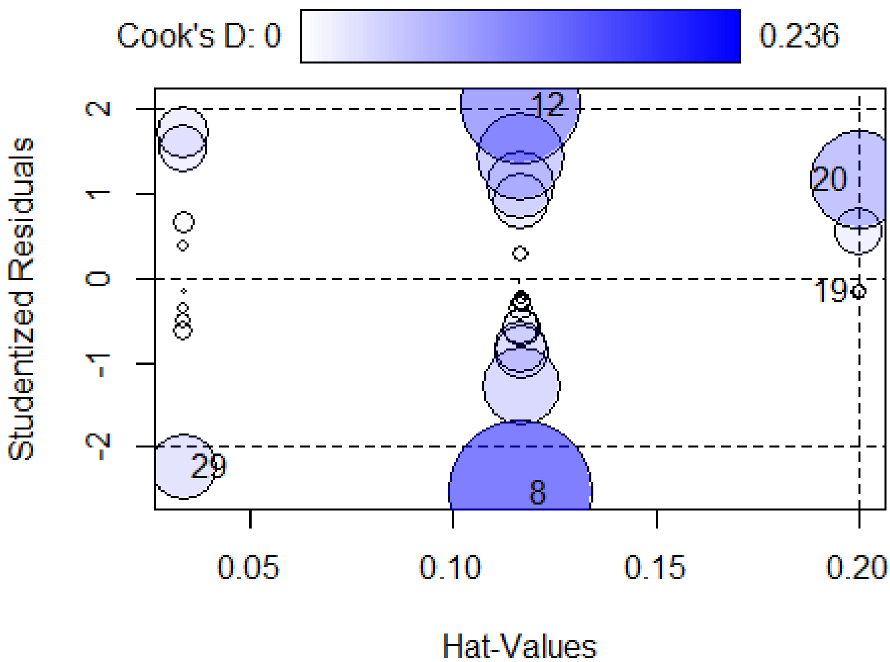
- La distancia de Cook mide el impacto que cada observación tiene en los coeficientes estimados del modelo. Si la distancia de Cook para una observación es mayor a un umbral (generalmente 1), se considera que esa observación tiene un alto potencial de influencia en el modelo, lo que podría significar que esa observación es un valor atípico que está afectando considerablemente el ajuste de los coeficientes.

- En este caso, ninguna observación supera el umbral de 1 en la distancia de Cook, lo que significa que no hay puntos de datos que estén influyendo desproporcionadamente en el ajuste del modelo.

```r
# Calcular medidas de influencia para cada observación en el modelo
ajustado
I = influence.measures(model_sig)

# Mostrar un resumen de las medidas de influencia para todas las
observaciones
# Esto incluye medidas como:
# - Distancia de Cook (cooks.distance)
# - Apalancamiento (hatvalues)
# - DFBETAs: Cambio en cada coeficiente de regresión si una observación
se elimina
# - DFFITS: Cambio en la predicción si una observación se elimina
summary(I)
```

```
## Potentially influential observations of
##   lm(formula = y ~ x2 + x3) :
##
##    dfb.1_ dfb.x2 dfb.x3 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
```

```r
library(car)

# Generar un gráfico de influencia para el modelo ajustado (model_sig)
influencePlot(model_sig)
```




```
##       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
```

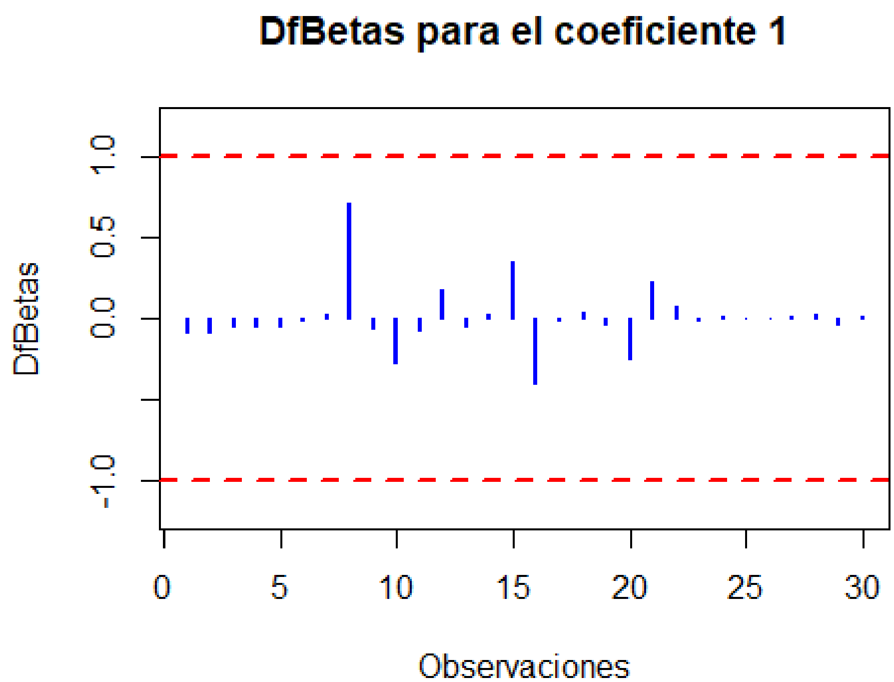
- Las observaciones 8, 19, 21 y 22 han sido marcadas como potencialmente influyentes en el modelo, basándose en las métricas DFBETAs, DFFITS, CovRatio, Distancia de Cook (cook.d), y valores de apalancamiento (hat).
- Las observaciones 8 y 12 son las más influyentes en términos de sus residuos, y tiene un impacto significativo en los coeficientes del modelo.
- Las observaciones 19, 20, 21 y 22 tienen apalancamiento alto, pero no necesariamente tienen un impacto considerable en los coeficientes, ya que sus distancias de Cook son bajas.

```
# Cálculo de los valores de DfBeta para cada coeficiente
dfbetas_values <- dfbetas(model_sig)
dfbetas_values
```

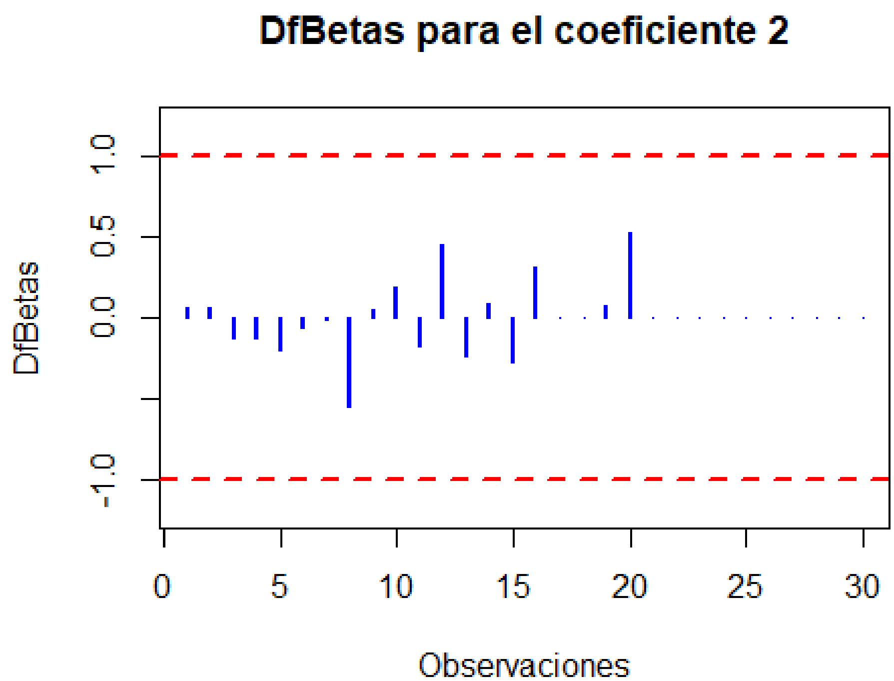
```
##     (Intercept)            x2            x3
## 1  -0.094659114  6.500122e-02  6.500122e-02
## 2  -0.088286427  6.062518e-02  6.062518e-02
## 3  -0.049908871 -1.244607e-01  1.244607e-01
## 4  -0.051689624 -1.289015e-01  1.289015e-01
```

```
## 5  -0.045449787 -1.957705e-01  1.957705e-01
## 6  -0.013431565 -5.785516e-02  5.785516e-02
## 7   0.018129912 -1.408231e-02 -1.408231e-02
## 8   0.709045573 -5.507474e-01 -5.507474e-01
## 9  -0.069191890  4.751320e-02  4.751320e-02
## 10 -0.276030188  1.895465e-01  1.895465e-01
## 11 -0.071460581 -1.782055e-01  1.782055e-01
## 12  0.177979915  4.438391e-01 -4.438391e-01
## 13 -0.055077205 -2.372397e-01  2.372397e-01
## 14  0.018991956  8.180601e-02 -8.180601e-02
## 15  0.353727704 -2.747561e-01 -2.747561e-01
## 16 -0.400477985  3.110692e-01  3.110692e-01
## 17 -0.010852267  5.680779e-17 -3.058881e-17
## 18  0.030110195 -1.576162e-16  8.487028e-17
## 19 -0.041488301  7.280648e-02  2.474947e-17
## 20 -0.255763326  5.268884e-01  3.044832e-16
## 21  0.220916541 -1.562791e-16 -2.492224e-01
## 22  0.068692986 -1.585856e-17 -8.565189e-02
## 23 -0.009140768  4.784870e-17 -2.576468e-17
## 24  0.011494157 -6.016786e-17  3.239808e-17
## 25 -0.006422163  3.361776e-17 -1.810187e-17
## 26 -0.002711328  1.419284e-17 -7.642300e-18
## 27  0.006377023 -3.338147e-17  1.797464e-17
## 28  0.026618056 -1.393361e-16  7.502715e-17
## 29 -0.038856541  2.034003e-16 -1.095232e-16
## 30  0.006377023 -3.338147e-17  1.797464e-17

# Número de coeficientes en el modelo (incluyendo el intercepto)
num_coef <- ncol(dfbetas_values)

# Iterar sobre cada coeficiente
for (i in 1:num_coef) {
  # Graficar los valores de DfBeta para cada coeficiente
  plot(dfbetas_values[, i], type = "h",
       main = paste("DfBetas para el coeficiente", i),
       ylab = "DfBetas", xlab = "Observaciones",
       lwd = 2, col = "blue", ylim = c(-1.2, 1.2))  # Ajustar los límites
del eje Y

  # Añadir líneas de referencia en -1 y 1
  abline(h = c(-1, 1), col = "red", lty = 2, lwd = 2)  # Límites comunes
para identificar puntos influyentes

  # Identificar las observaciones influyentes con DfBetas > 1 o < -1
  puntos_influyentes <- which(abs(dfbetas_values[, i]) > 1)

  # Mostrar las observaciones influyentes, si existen
  if (length(puntos_influyentes) > 0) {
    points(puntos_influyentes, dfbetas_values[puntos_influyentes, i],
           col = "red", pch = 19, cex = 1.5)  # Resaltar puntos
```

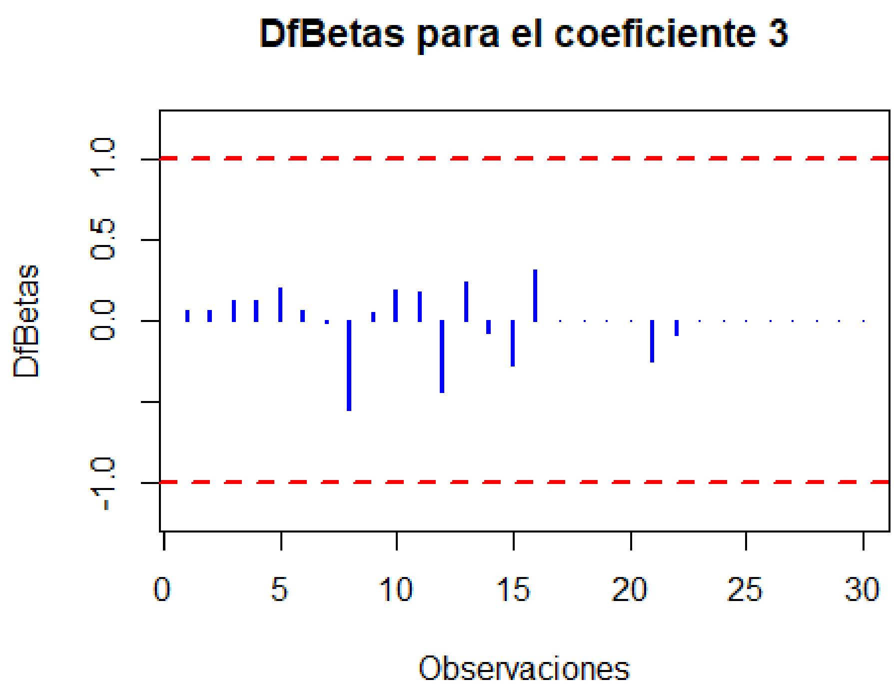
```
influyentes en rojo
    text(puntos_influyentes, dfbetas_values[puntos_influyentes, i],
         labels = puntos_influyentes, pos = 3, cex = 0.8, col = "red")  #
Etiquetar los puntos
  } else {
    print(paste("No hay observaciones influyentes para el coeficiente",
i))
  }
}
```




```
## [1] "No hay observaciones influyentes para el coeficiente 1"
```




```
## [1] "No hay observaciones influyentes para el coeficiente 2"
```




```
## [1] "No hay observaciones influyentes para el coeficiente 3"
```

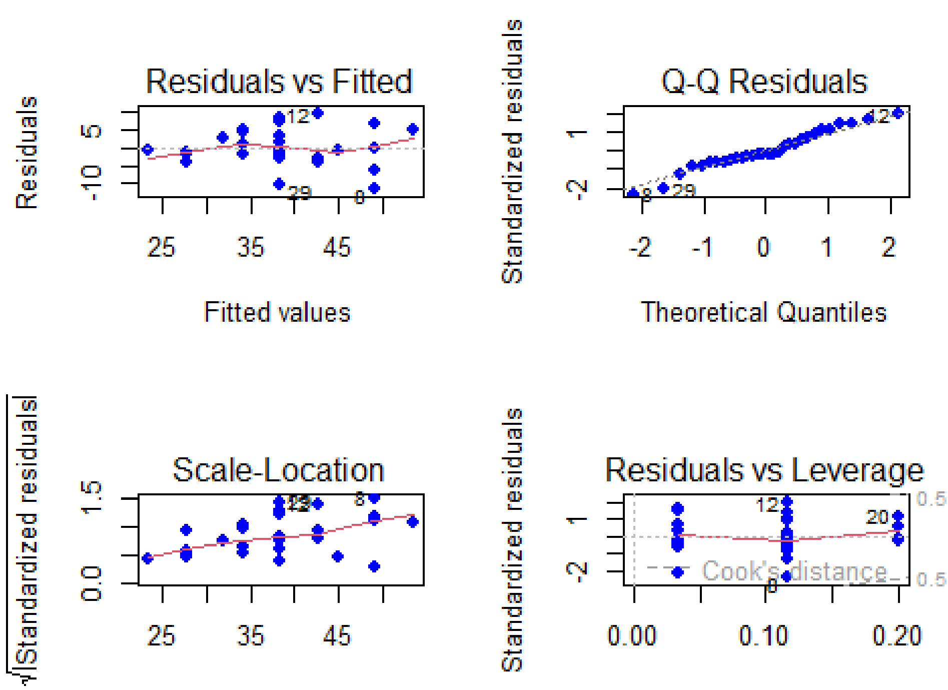
- DfBeta mide la influencia de una observación en un coeficiente específico. Valores cercanos a 0 significan que la observación no tiene un impacto significativo en el coeficiente cuando se elimina. Sin embargo, valores de DfBeta mayores a 1 o menores a -1 suelen indicar que esa observación tiene un impacto considerable sobre el coeficiente cuando se elimina.

Observaciones

- Observación 8 parece tener un impacto considerable en el modelo, ya que sus valores de DfBeta son relativamente altos para el intercepto, x2, y x3. No supera el umbral de 1.

- Observaciones 12 y 21 también muestran valores elevados de DfBeta en algunas de las variables, especialmente en los coeficientes de x2 y x3. Deberían ser consideradas potencialmente influyentes, aunque no muestran un impacto tan fuerte como la observación 8.

```r
# Configura la ventana gráfica para mostrar 4 gráficos en una cuadrícula de 2x2
par(mfrow = c(2, 2))

# Genera los gráficos de diagnóstico del modelo ajustado (model_sig)
plot(model_sig, col = 'blue', pch = 19)
```



Interpretación

- Las observaciones 8, 12 y 29 son puntos destacados en varios de los gráficos, lo que sugiere que podrían ser influyentes en el modelo. No obstante, no parecen ser extremadamente problemáticas según la distancia de Cook.

Conclusión Al analizar la influencia de las observaciones en el modelo de regresión ajustado, utilizando diferentes métricas como el apalancamiento, los residuos estudentizados, la distancia de Cook, y los DFBETAs observamos que ciertos resultados indican que hay algunas observaciones que son más influyentes o atípicas que otras, particularmente las observaciones 8, 12, 19, 21, y 22. Sin embargo, la influencia de estas observaciones no parece ser lo suficientemente significativa como para distorsionar el modelo en gran medida. Por lo tanto, no hay una fuerte evidencia que justifique la eliminación o el ajuste de estas observaciones en el modelo actual.