A3-Regresión Múltiple-Detección datos atípicos

Nallely Serna

2024-09-24

```r
# Cargar los datos
data <- read.csv("AlCorte.csv")

# Ajustar el modelo completo con todas las variables
model <- lm(Resistencia ~ ï..Fuerza + Potencia + Temperatura + Tiempo,
data = data)

# Realizar la selección de variables con step() usando AIC
modelo_refinado <- step(model, direction = "both", trace = TRUE)
```

```
## Start:  AIC=102.96
## Resistencia ~ ï..Fuerza + Potencia + Temperatura + Tiempo
##
##               Df Sum of Sq     RSS    AIC
## - ï..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
## + ï..Fuerza    1     26.88  665.12 102.96
## - Temperatura  1    252.20  944.20 109.47
## - Potencia     1   1341.01 2033.02 132.48
##
## Step:  AIC=101.84
## Resistencia ~ Potencia + Temperatura
##
##               Df Sum of Sq     RSS    AIC
## <none>                      732.04 101.84
## + Tiempo       1     40.04  692.00 102.15
## + ï..Fuerza    1     26.88  705.16 102.72
## - Temperatura  1    252.20  984.24 108.72
## - Potencia     1   1341.01 2073.06 131.07
```

```r
# Ver el resumen del modelo refinado
summary(modelo_refinado)
```

```
##
## 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)

modelo_refinado2 = step(modelo_nulo, scope = list(lower = modelo_nulo,
upper = model), 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
## + ï..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
## + ï..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
## + ï..Fuerza  1    26.882 705.16 102.72
```

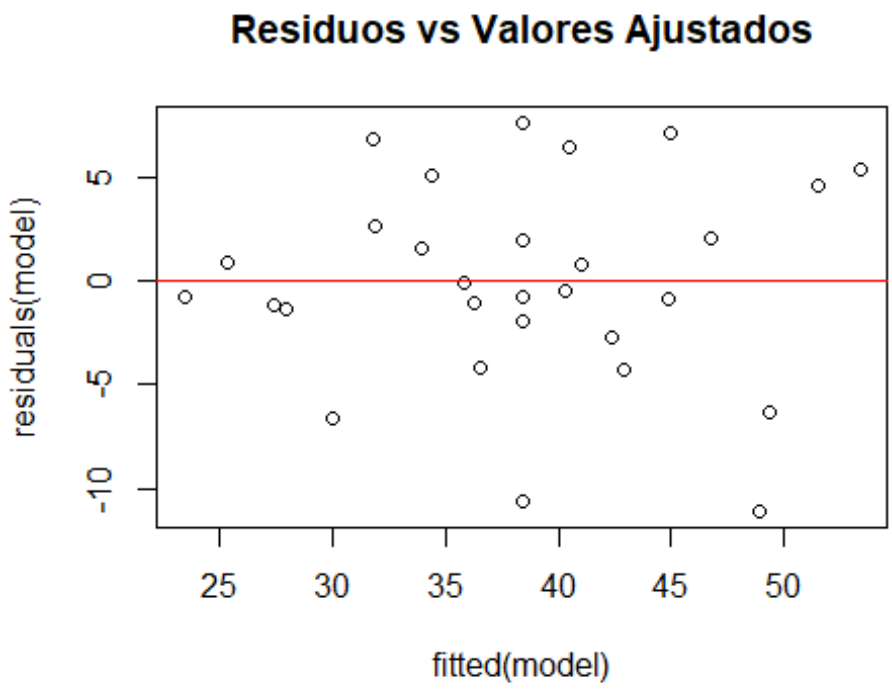
```
n = length(data$Resistencia)
modelo_refinado3 = step(model, direction = "both", k=log(n))
```

```
## Start:  AIC=109.97
## Resistencia ~ ï..Fuerza + Potencia + Temperatura + Tiempo
##
##               Df Sum of Sq     RSS    AIC
## - ï..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
## + ï..Fuerza    1     26.88  665.12 109.97
## - Temperatura  1    252.20  944.20 113.68
## - Potencia     1   1341.01 2033.02 136.69
##
## Step:  AIC=106.04
## Resistencia ~ Potencia + Temperatura
##
##               Df Sum of Sq     RSS    AIC
## <none>                      732.04 106.04
## + Tiempo       1     40.04  692.00 107.76
## + ï..Fuerza    1     26.88  705.16 108.32
## - Temperatura  1    252.20  984.24 111.52
## - Potencia     1   1341.01 2073.06 133.87
```

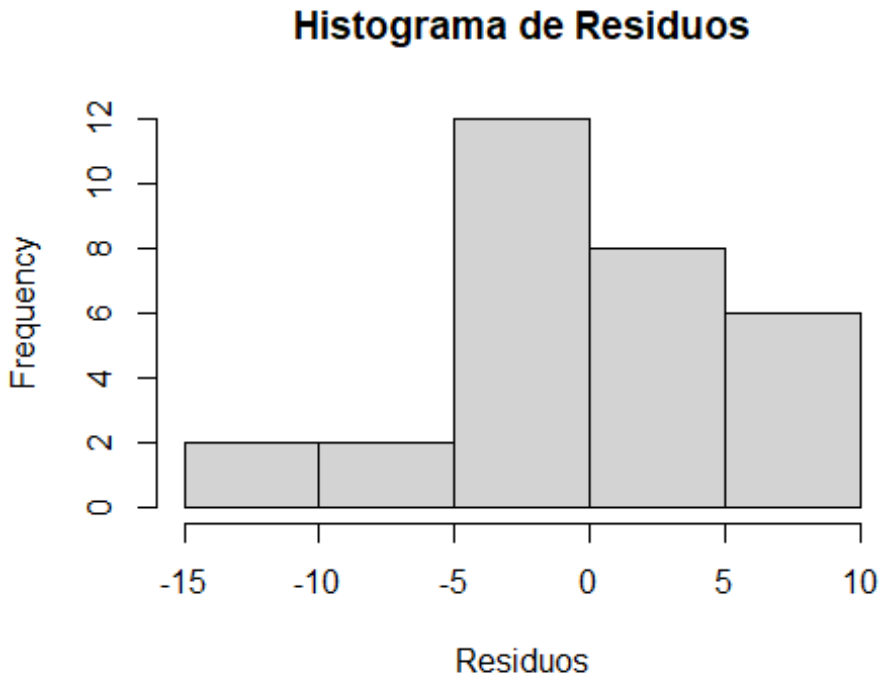
```
# Verificar la normalidad de los residuos
shapiro.test(residuals(model))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(model)
## W = 0.95583, p-value = 0.2415
```

```
# Gráfico de residuos vs valores ajustados (para verificar
homocedasticidad)
plot(fitted(model), residuals(model), main="Residuos vs Valores
Ajustados")
abline(h = 0, col = "red")
```




```r
# Histograma de los residuos
hist(residuals(model), main="Histograma de Residuos", xlab="Residuos")
```



```r
# Prueba de independencia de residuos (Durbin-Watson)
library(lmtest)
dwtest(model)

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

# Calcular el VIF
vif(model)

##   ï..Fuerza    Potencia Temperatura      Tiempo
##           1           1           1           1
```

Colinealidad (VIF): Los valores de VIF son todos 1, lo que indica que no hay problemas de multicolinealidad entre las variables.

Interpretación de los coeficientes: Aunque las variables Potencia y Temperatura son estadísticamente significativas (p-valor < 0.05), las variables Fuerza y Tiempo no lo son (p-valor > 0.05). Esto significa que estas últimas no tienen un impacto significativo en la resistencia al corte según tu modelo.

Normalidad de los residuos: El valor p del test de Shapiro-Wilk (0.2415) sugiere que no hay evidencia para rechazar la hipótesis de que los residuos siguen una distribución normal.

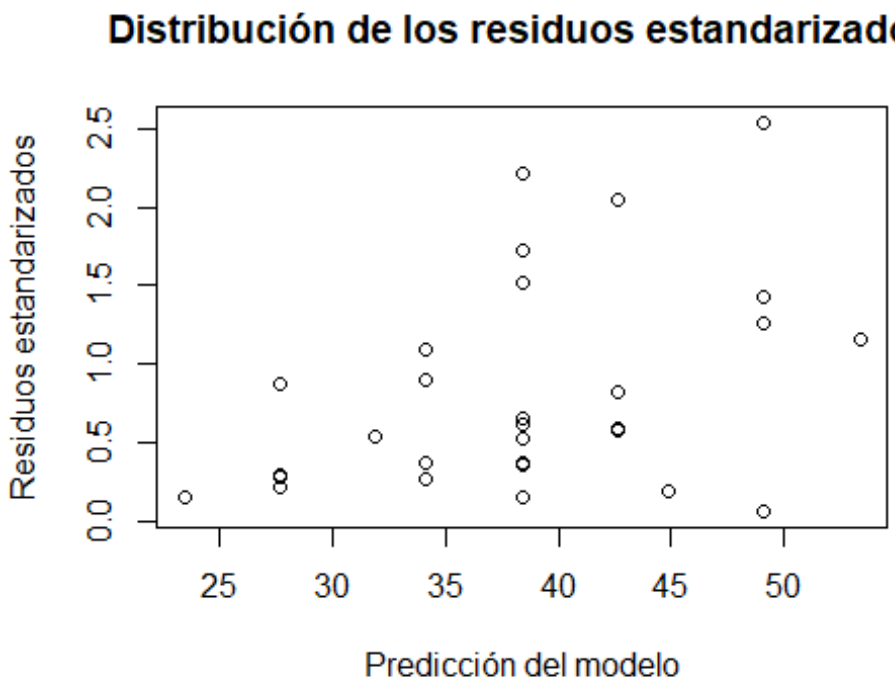
Independencia de los residuos: La prueba de Durbin-Watson muestra que no hay autocorrelación significativa en los residuos (p-valor = 0.7917), lo que es un buen resultado.

Homocedasticidad: Los puntos están distribuidos de manera uniforme alrededor de 0.

##A3-Regresión Múltiple-Detección datos atípicos

```r
# Calcular los residuos estandarizados
data$residuos_estandarizados <- rstudent(modelo_refinado)

# Gráfico de residuos estandarizados
plot(predict(modelo_refinado), abs(data$residuos_estandarizados),
     main = "Distribución de los residuos estandarizados",
     xlab = "Predicción del modelo", ylab = "Residuos estandarizados")
abline(h = 3, col = "red", lty = 2)  # Línea discontinua para residuos >
3
```

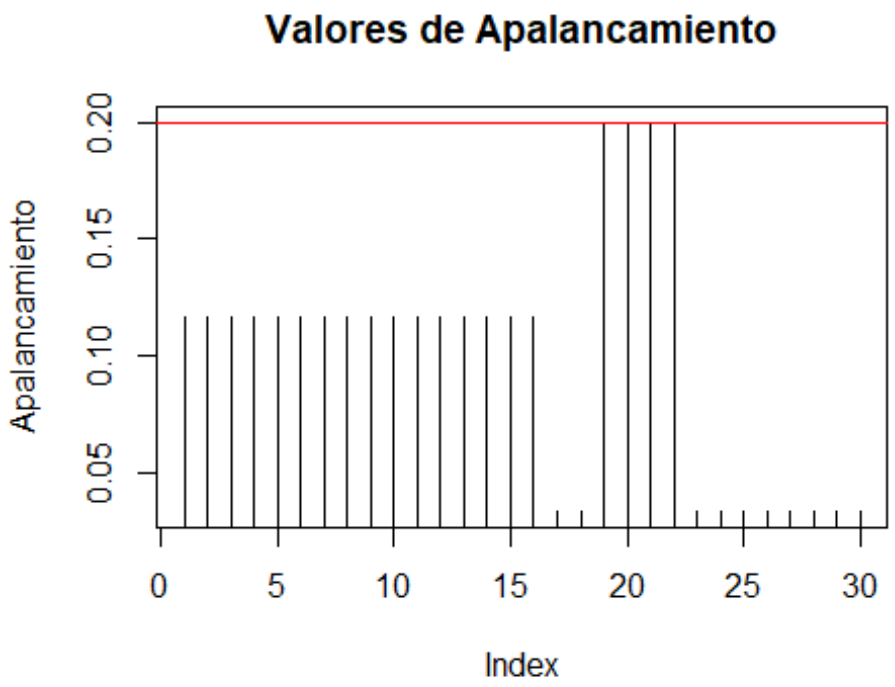



```
# Identificar los puntos atípicos
Atipicos <- which(abs(data$residuos_estandarizados) > 3)
data[Atipicos, ]  # Mostrar observaciones con residuos atípicos
```

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

```
# Cálculo del leverage
leverage <- hatvalues(modelo_refinado)

# Gráfico de leverage
plot(leverage, type = "h", main = "Valores de Apalancamiento", ylab =
"Apalancamiento")
abline(h = 2 * mean(leverage), col = "red")  # Límite común para leverage
```

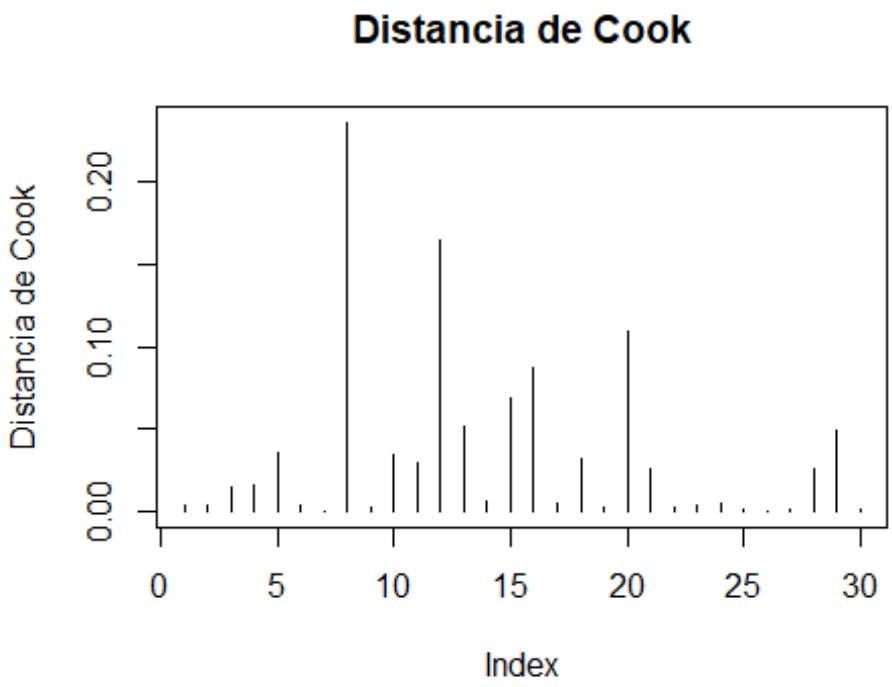



```r
# Identificar puntos con alto leverage
high_leverage_points <- which(leverage > 2 * mean(leverage))
data[high_leverage_points, ]  # Mostrar observaciones con alto leverage
```

```
##    ï..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
```

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

# Gráfico de la distancia de Cook
plot(cooksd, type = "h", main = "Distancia de Cook", ylab = "Distancia de
Cook")
abline(h = 1, col = "red")  # Límite común para la distancia de Cook
```

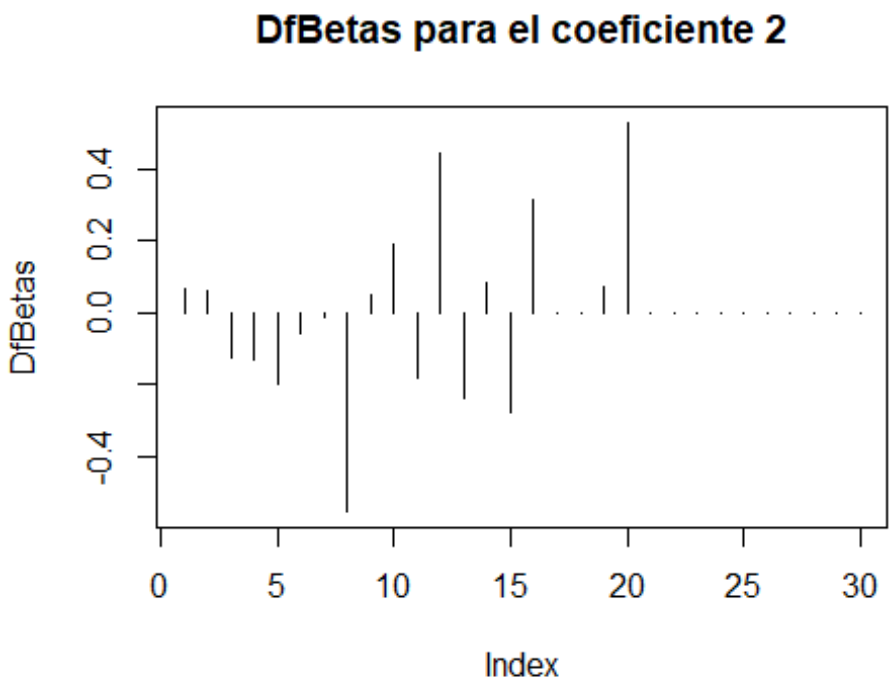



```r
# Identificar puntos influyentes según la distancia de Cook
puntos_influyentes <- which(cooksd > 1)
data[puntos_influyentes, ]  # Mostrar observaciones influyentes
```

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

```r
# Cálculo de DfBetas para cada coeficiente
dfbetas_values <- dfbetas(modelo_refinado)

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




```r
# Identificar puntos influyentes según DfBetas para el coeficiente 2
puntos_influyentes_dfbeta <- which(abs(dfbetas_values[, 2]) > 1)
data[puntos_influyentes_dfbeta, ]  # Mostrar observaciones influyentes
```

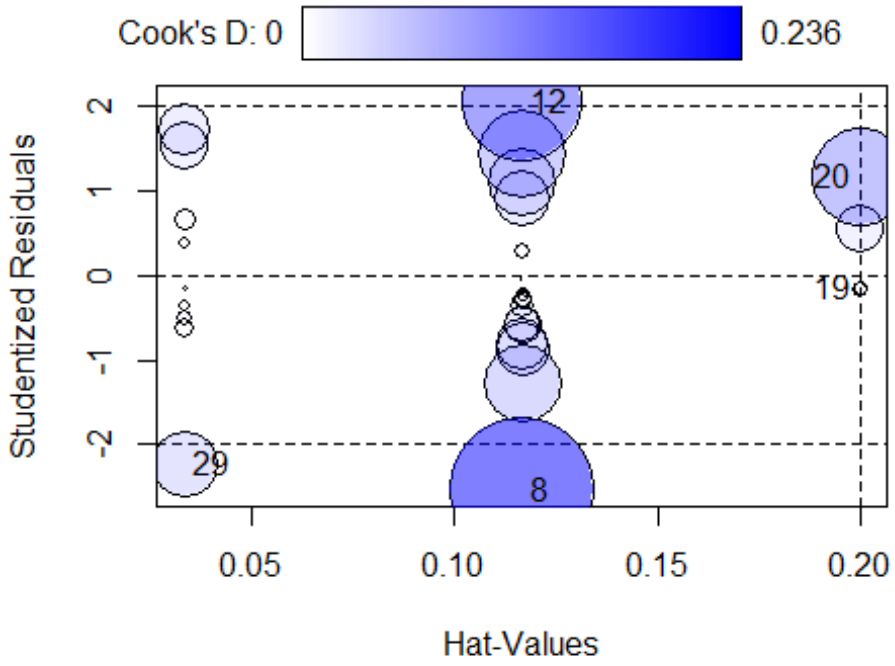
```
## [1] ï..Fuerza               Potencia                Temperatura
## [4] Tiempo                  Resistencia
residuos_estandarizados
## <0 rows> (or 0-length row.names)
```

```r
# Calcular las medidas de influencia
influencia <- influence.measures(modelo_refinado)

# Resumen de las observaciones con posible influencia
summary(influencia)
```

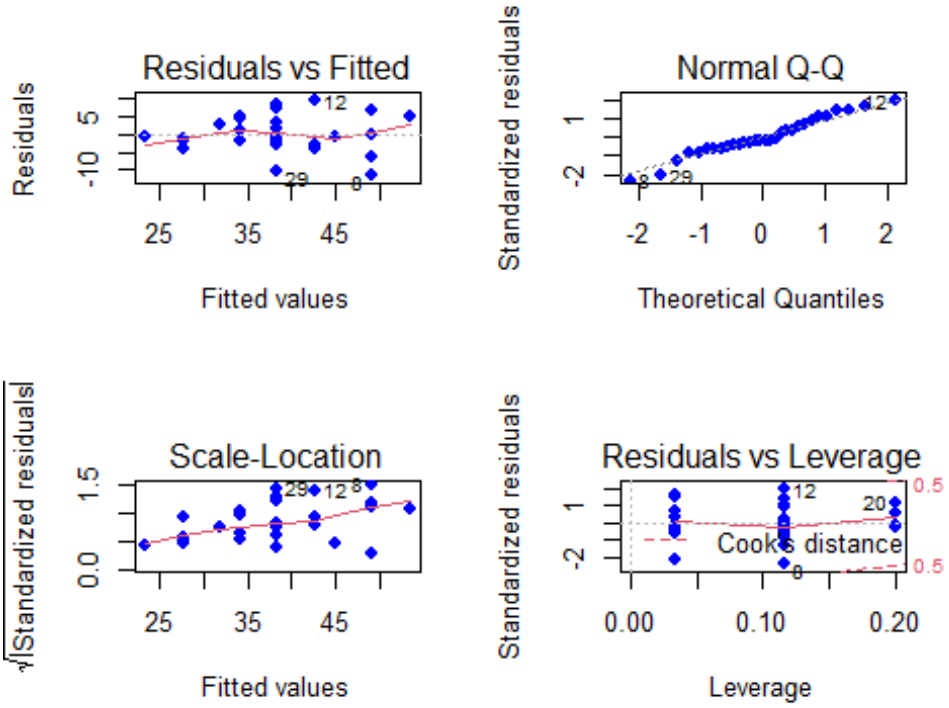
```
## Potentially influential observations of
##   lm(formula = Resistencia ~ Potencia + Temperatura, data = data) :
##
##    dfb.1_ dfb.Ptnc dfb.Tmpr dffit cov.r   cook.d hat
## 8   0.71  -0.55    -0.55    -0.92  0.65_*  0.24   0.12
## 19 -0.04   0.07     0.00    -0.08  1.40_*  0.00   0.20
## 21  0.22   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
```

```r
# Gráfico de influencia usando car::influencePlot()
influencePlot(modelo_refinado)
```




```
##       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

# Gráficos diagnósticos del modelo
par(mfrow = c(2, 2))
plot(modelo_refinado, col = 'blue', pch = 19)
```


Residuals vs Fitted
Residuals
Fitted values
Normal Q-Q
Standardized residuals
Theoretical Quantiles
Scale-Location
√|Standardized residuals|
Fitted values
Residuals vs Leverage
Standardized residuals
Cook's distance
Leverage