Regresión multiple detección datos atipicos

Jacobo Hirsch Rodriguez

2024-09-28

#Lectura del dataset vamos a leer el dataset de las variables

```
alcorte=read.csv("./ALCorte.csv") #leer el dataset

print(alcorte)
```

```
##    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
## 7      30       90          225     15        48.8
## 8      40       90          225     15        37.8
## 9      30       60          175     25        26.6
## 10     40       60          175     25        23.4
## 11     30       90          175     25        38.6
## 12     40       90          175     25        52.1
## 13     30       60          225     25        39.5
## 14     40       60          225     25        32.3
## 15     30       90          225     25        43.0
## 16     40       90          225     25        56.0
## 17     25       75          200     20        35.2
## 18     45       75          200     20        46.9
## 19     35       45          200     20        22.7
## 20     35      105          200     20        58.7
## 21     35       75          150     20        34.5
## 22     35       75          250     20        44.0
## 23     35       75          200     10        35.7
## 24     35       75          200     30        41.8
## 25     35       75          200     20        36.5
## 26     35       75          200     20        37.6
## 27     35       75          200     20        40.3
## 28     35       75          200     20        46.0
## 29     35       75          200     20        27.8
## 30     35       75          200     20        40.3
```

#Análisis descriptivo de las variables

```r
medias_alcorte <- colMeans(alcorte)
desviaciones_alcorte <- apply(alcorte, 2, sd)
minimos_alcorte <- apply(alcorte, 2, min)
maximos_alcorte <- apply(alcorte, 2, max)

medidas_alcorte <- data.frame(Medias = medias_alcorte, Desviaciones = desviaciones_alcorte, Minimos = mi
print(medidas_alcorte)
```

```
##                Medias Desviaciones Minimos Maximos
## Fuerza       35.00000     4.548588    25.0    45.0
## Potencia     75.00000    13.645765    45.0   105.0
## Temperatura 200.00000    22.742941   150.0   250.0
## Tiempo       20.00000     4.548588    10.0    30.0
## Resistencia  38.40667     8.954403    22.7    58.7
```

obtenemos la matriz de correlación de las variables:

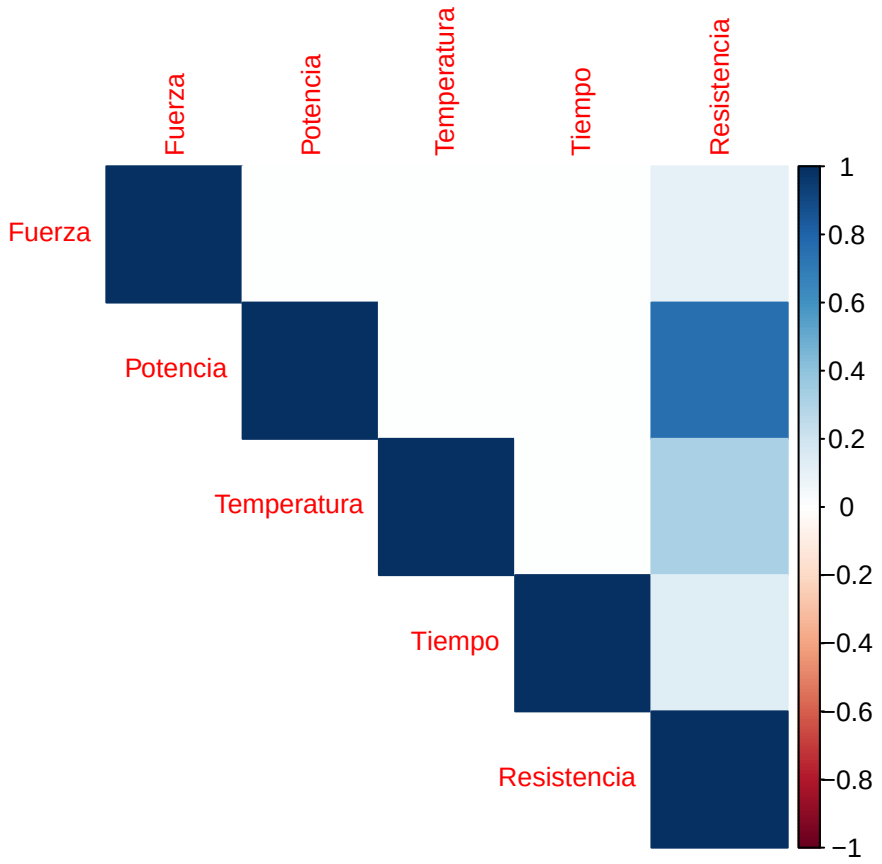
```r
correlacion_alcorte <- cor(alcorte)
```

vamos a visualizar con un mapa de calor las correlaciones entre las variables

```r
library(corrplot)
```

```
## corrplot 0.94 loaded
```

```r
# Visualizar la matriz de correlación con un heatmap
corrplot(correlacion_alcorte, method = "color", type = "upper", tl.cex = 0.8)
```



de acuerdo al mapa de calor, se observa que no existe mucha relación entre las variables explicativas, solo entre la variable dependiente y las explicativas, y ni aún así todas presentan una relación con esta, las que parecen ser que podrían explicar más la resistencia son la temperatura y la potencia, tendremos que comprobarlo buscando el mejor modelo, su significancia y podemos hacer una prueba VIF para buscar colinealidad entre las variables predictorias, aunque si tenemos razón, el mapa de calor no muestra una relación entre potencia y temperatura por lo que no hay indicios de colinalidad.

#2) Vamos a encontrar el mejor modelo de regresión

vamos a escribir el modelo inicial, que despues utilizaremos para encontrar el modelo optimo

```
# Modelo completo
modelo_alcorte <- lm(Resistencia ~ Fuerza + Potencia + Temperatura + Tiempo, data = alcorte)
summary(modelo_alcorte)
```

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

```
## Fuerza        0.21167    0.21057   1.005  0.32444
## Potencia      0.49833    0.07019   7.100 1.93e-07 ***
## Temperatura   0.12967    0.04211   3.079  0.00499 **
## Tiempo        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
```

Análisis del modelo inicial:

el valor p del estadistico de f nos indica que el modelo es estadisticamente significativo, por lo que al menos una de las variables tiene un efecto significativo. Despues de considerar la cantidad de variables (usando el R cuadrado ajustado) se explica un 66.8% de la variabilidad en la resistencia. Tambien se puede ver que las variables potencia y temperatura tienen mucha influencia en la resistencia al corte.

Para encontrar el mejor modelo, vamos a usar el método de selección hacia atrás (backward selection) basado en el AIC:

```
# Selección hacia atrás
modelo_optimo_alcorte <- step(modelo_alcorte, direction = "backward")
```

```
## Start:  AIC=102.96
## Resistencia ~ Fuerza + Potencia + Temperatura + Tiempo
##
##               Df Sum of Sq     RSS    AIC
## - Fuerza       1     26.88  692.00 102.15
## - Tiempo       1     40.04  705.16 102.72
## <none>                      665.12 102.96
## - Temperatura  1    252.20  917.32 110.61
## - Potencia     1   1341.01 2006.13 134.08
##
## Step:  AIC=102.15
## Resistencia ~ Potencia + Temperatura + Tiempo
##
##               Df Sum of Sq     RSS    AIC
## - Tiempo       1     40.04  732.04 101.84
## <none>                      692.00 102.15
## - Temperatura  1    252.20  944.20 109.47
## - Potencia     1   1341.01 2033.02 132.48
##
## Step:  AIC=101.84
## Resistencia ~ Potencia + Temperatura
##
##               Df Sum of Sq     RSS    AIC
## <none>                      732.04 101.84
## - Temperatura  1     252.2  984.24 108.72
## - Potencia     1    1341.0 2073.06 131.07
```

```
summary(modelo_optimo_alcorte)
```

```
##
```

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

Los resutlados coinciden con las predicciones iniciales, el mejor modelo es el que tiene las variables potencia y temperatura, el R ajustado no cambio pero el modelo es menos complejo y por ende es mejor, el valor p del estadístico aún muestra que el modelo es significativo que también es buena señal aunque era de esperarse.

Análisis del modelo encontrado

residuos del modelo optimo

```
residuos_alcorte <- residuals(modelo_optimo_alcorte)
```

##Normalidad de los residuos

hipotesis nula : los residuos siguen una distribución normal.

hipotesis alternativa : los residuos no siguen una distribución normal

regla de decisión : Si el valor p es menor a 0.05, se rechaza la hipótesis nula y se concluye que los residuos no siguen una distribución normal.

con interacción

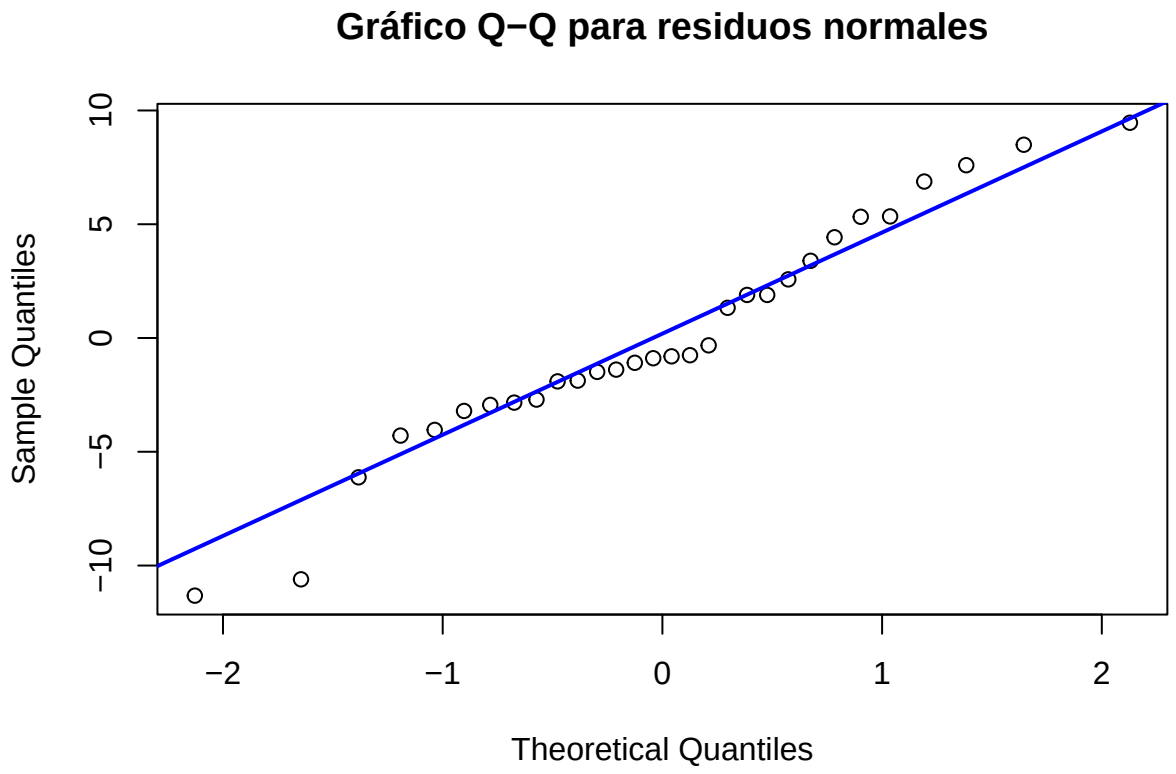
sacamos la prueba de shapiro-wilk

```
# Prueba de Shapiro-Wilk
shapiro.test(residuos_alcorte)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuos_alcorte
## W = 0.96588, p-value = 0.4333
```

de acuerdo a la regla de decisión, no rechazamos la hipotesis nula, parece ser que los residuos siguen una distribución normal, vamos a realizar un qqplot para reafirmar los resultados de la prueba

```
qqnorm(residuos_alcorte,
       main = "Gráfico Q-Q para residuos normales")
qqline(residuos_alcorte, col = "blue", lwd = 2)
```



de acuerdo al qqplot en general, los datos no siguen exactamente una distribución normal, especialmente en los extremos. Los puntos en el centro están cerca de la línea, lo que sugiere que la mayor parte de los datos se comporta de manera normal, pero en las partes más alejadas (los extremos), los puntos se desvían bastante. Esto podría significar que hay valores atípicos o que la distribución tiene colas más largas o más cortas de lo esperado. Aún así, los datos en su mayoría muestran una tendencia normal, sumado al resultado de la prueba de shapiro-wilk podría decirse que si pasa la prueba.

##Verificación de media cero

hipotesis nula : la media de los errores es igual a 0

hiporesis alrernativa : la media de los errores es diferente de 0

regla de decisión con prueba t : Si el valor p es menor que 0.05, se rechaza la hipótesis nula y se concluye que la media de los residuos es significativamente diferente de 0.

```
media_prueba_alcorte = mean(residuos_alcorte)
print(media_prueba_alcorte)
```

```
## [1] 2.442491e-16
```

realizamos la prueba t para el modelo optimo

```
# Realizar la prueba t para verificar si la media es 0
prueba_t_alcorte <- t.test(residuos_alcorte, mu = 0)

# Mostrar el resultado de la prueba
print(prueba_t_alcorte)
```

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

para ambos casos el valor p fue de 1, el valor más alto posible que indica que no hay evidencia para descartar la hipotesis nula

##Homocedasticidad

hipotesis nula: la varianza de los errores es constante (hay homocedasticidad)

hipotesis alternativa : la varianza de los errores no es constante (hay heterocedasticidad)

regla de decisión: si el valor p es menor o igual que 0.05 entonces rechazamos la hipotesis nula

para realizar las pruebas vamos a necesitar descargar un paquete que contiene la prueba que vamos a utilizar

```
# Cargar el paquete lmtest
library(lmtest)
```

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

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

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

vamos a realizar la prueba de white

```
# Realizar la prueba de White
prueba_white_alcorte <- bptest(modelo_optimo_alcorte, ~ fitted(modelo_optimo_alcorte) + I(fitted(modelo

# Mostrar los resultados de la prueba
print(prueba_white_alcorte)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  modelo_optimo_alcorte
## BP = 4.0002, df = 2, p-value = 0.1353
```

vamos a verificar la hipotesis nula

```
# Extraer el valor p de la prueba
p_value_white_alcorte <- prueba_white_alcorte$p.value

# Comparar el valor p con el umbral de significancia de 0.05
if (p_value_white_alcorte > 0.05) {
  print("No se rechaza la hipótesis nula: no hay evidencia de heterocedasticidad.")
} else {
  print("Se rechaza la hipótesis nula: hay evidencia de heterocedasticidad.")
}
```

```
## [1] "No se rechaza la hipótesis nula: no hay evidencia de heterocedasticidad."
```

no hay evidencia suficiente para rechazar la hipotesis nula, por lo que podemos decir que no presenta heterocedasticidad.

##independencia

hipotesis nula: los errores no estan correlacionados

hipotesis alternativa : los errores estan correlacionados

regla de decisión: si el valor p de la prueba es menor a 0.05 se rechaza la hipotesis nula

hacemos el modelo durbin watson para verificar independencia entre los residuos

```
# Realizar la prueba de Durbin-Watson
prueba_dw_alcorte <- dwtest(modelo_optimo_alcorte)

# Mostrar los resultados de la prueba
print(prueba_dw_alcorte)
```

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

verificamos que el valor p no sea menor a 0

```
# Extraer el valor p
p_value_dw_alcorte <- prueba_dw_alcorte$p.value

# Interpretar los resultados
if (p_value_dw_alcorte > 0.05) {
  print("No hay evidencia de correlación en los residuos (hipótesis nula no rechazada).")
} else {
  print("Hay evidencia de correlación en los residuos (hipótesis nula rechazada).")
}
```

```
## [1] "No hay evidencia de correlación en los residuos (hipótesis nula no rechazada)."
```

Parece ser que no existe evidencia de correlación en los residuos, hasta el momento una muy buena señal de que el modelo será de utilidad.

Para terminar de analizar los residuos, vamos a verificar la linealidad entre estos:

##linealidad

para verificar la linealidad vamos a utilizar una prueba RESET

La prueba RESET de Ramsey (Regression Equation Specification Error Test) es utilizada para detectar posibles errores de especificación en un modelo de regresión lineal. La prueba examina si hay variables omitidas o si la forma funcional del modelo es incorrecta.

hipotesis nula: no hay términos omitidos que indican linealidad

hipotesis alternativa: hay una especificación errónea en el modelo que indica no linealidad

vamos a hacer la prueba reset

```r
# Realizar la prueba RESET de Ramsey
prueba_reset_alcorte <- resettest(modelo_optimo_alcorte)

# Mostrar los resultados de la prueba
print(prueba_reset_alcorte)
```

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

extraemos el valor p para evalualr la regla de decisión

```r
# Extraer el valor p
p_value_reset_alcorte <- prueba_reset_alcorte$p.value

# Interpretar los resultados
if (p_value_reset_alcorte > 0.05) {
  print("No se rechaza la hipótesis nula: el modelo no parece tener errores de especificación.")
} else {
  print("Se rechaza la hipótesis nula: es probable que haya errores de especificación en el modelo.")
}
```

```
## [1] "No se rechaza la hipótesis nula: el modelo no parece tener errores de especificación."
```

Excelentes noticias para nuestro modelo! parece ser que las pruebas sobre los residuos han terminado, y obtuvimos buenos resultados pero aún falta verificar que las variables explicativas no esten relacionadas. para ellos vamos a utilizar el Factor de Inflación de la Varianza (VIF)

```r
library(car)
```

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

```r
# Cálculo del VIF
vif(modelo_optimo_alcorte)
```

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

como lo vimos desde el mapa de calor, los resultados de la medida VIF nos indican que nuestras variables predictoras no estan relacionadas.

#Conclusiones generales:

El modelo que incluye Potencia y Temperatura para predecir la Resistencia al corte es estadísticamente significativo, con un valor p global de 1.674e-07. Esto indica que al menos una de las variables predictoras tiene un impacto importante en la resistencia. El modelo explica alrededor del 68.52% de la variabilidad en la resistencia al corte, y el R^2 ajustado de 0.6619, que toma en cuenta el número de variables, sugiere que el modelo sigue siendo sólido. Ambas variables predictoras son significativas por separado, siendo la Potencia la que tiene el mayor efecto (coeficiente = 0.49833, p = 1.47e-07). Esto significa que a mayor potencia, mayor será la resistencia al corte. De manera similar, la Temperatura también es significativa (coeficiente = 0.12967, p = 0.00508), lo que implica que un aumento en la temperatura está asociado con un incremento en la resistencia, aunque con un efecto menor comparado con la potencia.

El modelo ha superado las pruebas de residuos, lo que sugiere que se cumplen los supuestos de homocedasticidad, normalidad e independencia, asegurando la validez del modelo. Además, el análisis de VIF muestra que no hay problemas de multicolinealidad entre las variables predictoras, reforzando la fiabilidad de los coeficientes estimados. Aunque el error estándar residual es de 5.207, lo que indica cierta dispersión en los valores predichos, este valor es aceptable dado el buen ajuste del modelo. En general, el modelo es adecuado para describir y predecir la resistencia al corte en función de la potencia y la temperatura, ofreciendo un equilibrio entre simplicidad y capacidad predictiva.

#Actividad de detección de datos atípicos

```r
# Cargar las librerías necesarias
library(dplyr)
```

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

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

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

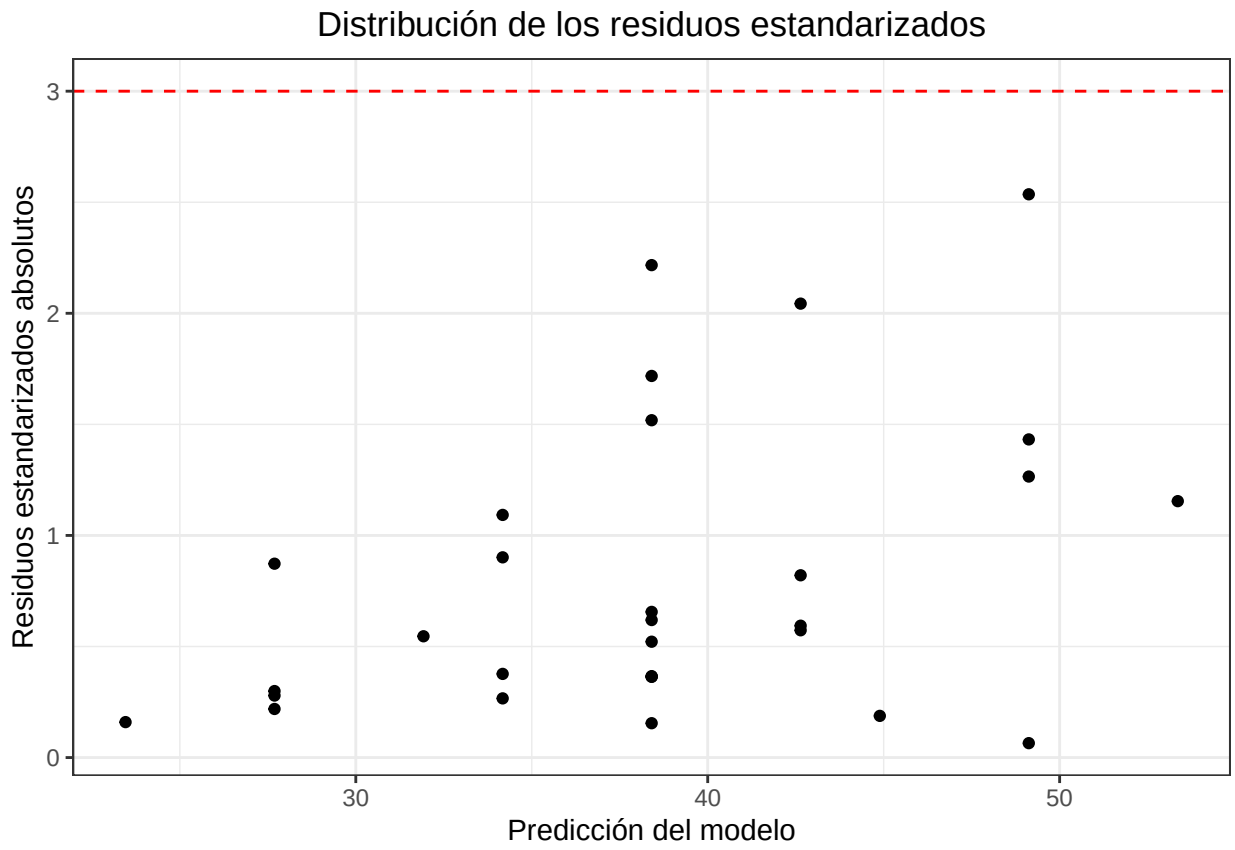
```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(ggplot2)

# Asumiendo que tienes un dataframe llamado "alcorte" y un modelo llamado "modelo_optimo_alcorte"
```

```r
# Calcular los residuos estandarizados y añadirlos al dataframe "alcorte"
alcorte$residuos_estandarizados <- rstudent(modelo_optimo_alcorte)

# Crear un gráfico de los residuos estandarizados vs las predicciones del modelo
ggplot(data = alcorte, aes(x = predict(modelo_optimo_alcorte), y = abs(residuos_estandarizados))) +
  geom_hline(yintercept = 3, color = "red", linetype = "dashed") + # Umbral de residuos > 3
  geom_point(aes(color = ifelse(abs(residuos_estandarizados) > 3, 'red', 'black'))) + # Residuos > 3 en
  scale_color_identity() + # Usar los colores definidos sin transformarlos
  labs(title = "Distribución de los residuos estandarizados", x = "Predicción del modelo", y = "Residuos
  theme_bw() + # Tema con fondo blanco
  theme(plot.title = element_text(hjust = 0.5)) # Centrar el título del gráfico
```




```r
# Identificar las observaciones con residuos estandarizados absolutos > 3
Atipicos <- which(abs(alcorte$residuos_estandarizados) > 3)

# Mostrar las observaciones atípicas
alcorte[Atipicos, ]
```

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

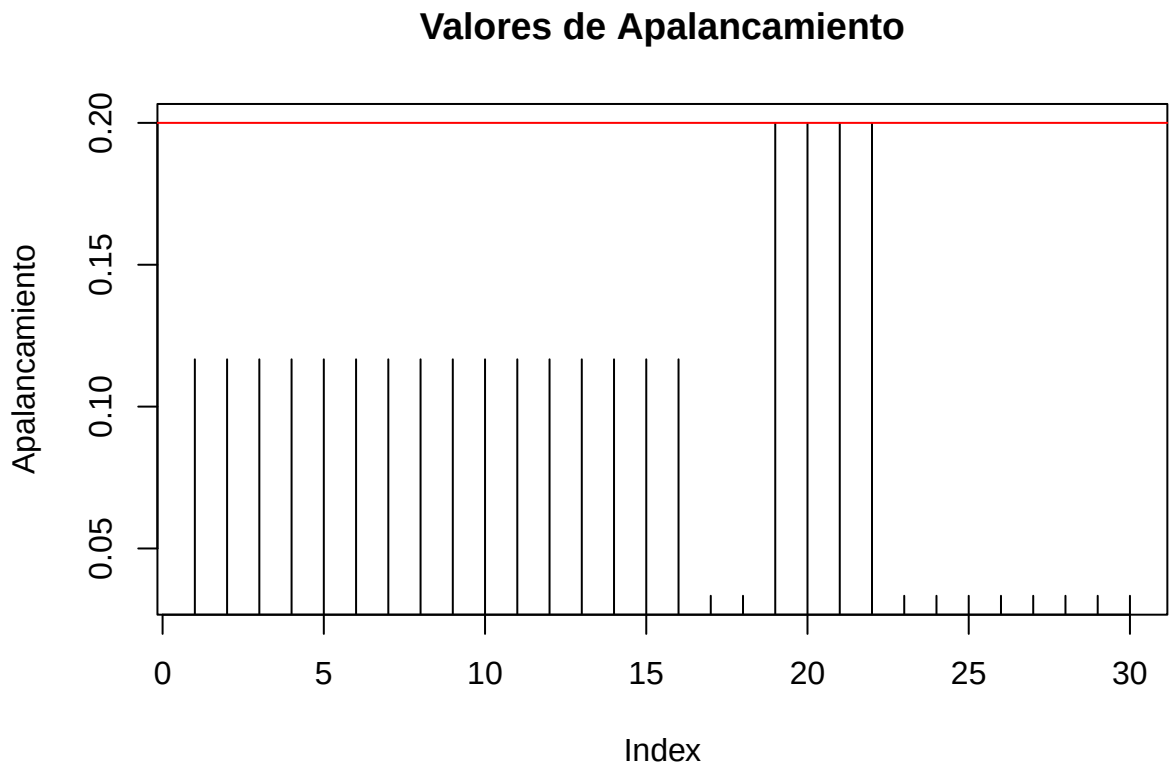
podemos observar que no se siguen ningún patrón en particular, los residuos parecen estar dispersos, aún más importante y el propósito de este gráfico es recalcar que ningún dato parece pasar el umbral de 3 desviaciones estándar. por lo que no hay evidencia de outliers significativos

#Leverage

```
# Cálculo del leverage para el modelo_optimo_alcorte
leverage = hatvalues(modelo_optimo_alcorte)

# Graficar los valores de leverage
plot(leverage, type="h", main="Valores de Apalancamiento", ylab="Apalancamiento")

# Añadir una línea roja en 2 veces el valor promedio del leverage
abline(h = 2 * mean(leverage), col="red")
```




```
# Identificar las observaciones con leverage alto (mayor a 2 veces el promedio)
high_leverage_points = which(leverage > 2 * mean(leverage))

# Mostrar las observaciones con alto leverage en el dataframe "alcorte"
alcorte[high_leverage_points, ]
```

```
##    Fuerza Potencia Temperatura Tiempo Resistencia residuos_estandarizados
## 19     35       45         200     20        22.7               -0.159511
## 20     35      105         200     20        58.7                1.154355
```

A pesar de que las observaciones 19 y 20 tienen un alto leverage, lo que indica que pueden influir significativamente en el ajuste del modelo debido a su ubicación en áreas extremas del espacio de predictores (específicamente en relación con la variable Potencia), sus residuos estandarizados no son alarmantes. Esto

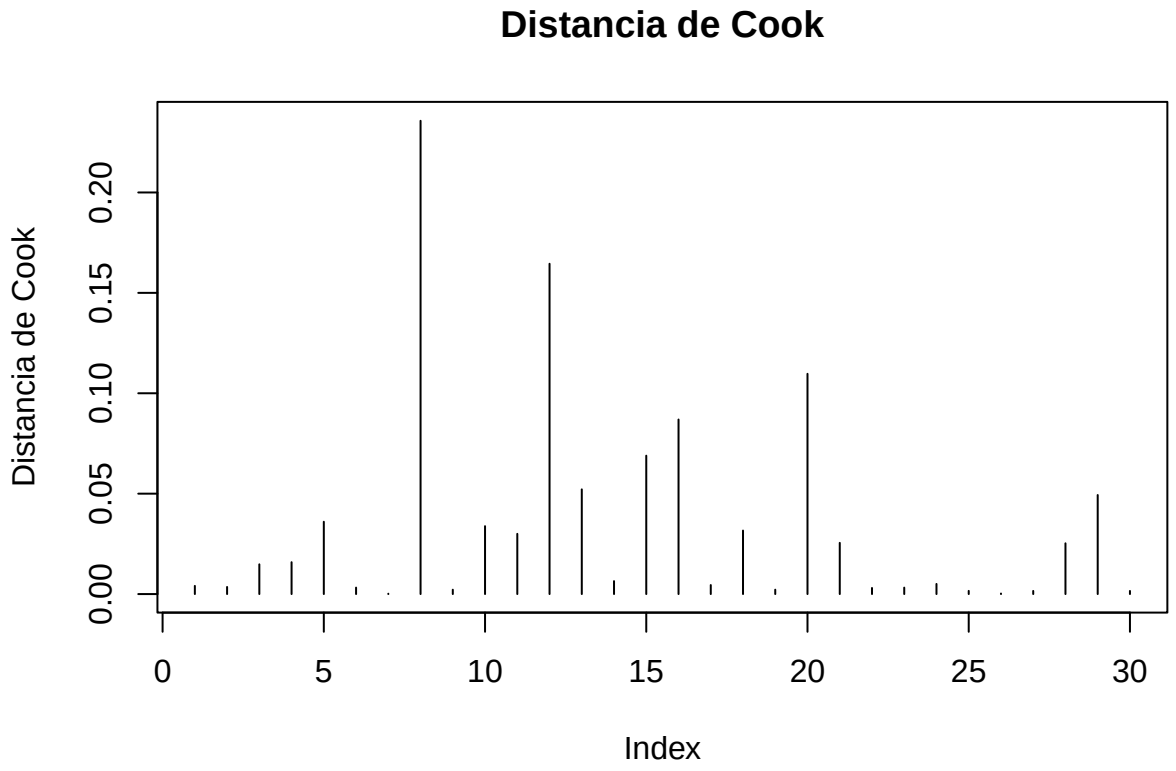
sugiere que, aunque son puntos influyentes, no están afectando negativamente la calidad del ajuste del modelo respecto a la variable de respuesta.

#Distancia de Cook

```
# Cálculo de la distancia de Cook para el modelo_optimo_alcorte
cooksdistance <- cooks.distance(modelo_optimo_alcorte)

# Graficar las distancias de Cook
plot(cooksdistance, type="h", main="Distancia de Cook", ylab="Distancia de Cook")

# Añadir una línea roja en 1 (umbral comúnmente usado)
abline(h = 1, col="red")
```




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

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

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

No se identifican observaciones influyentes en el modelo según la distancia de Cook, ya que todos los valores están por debajo de 0.20, muy lejos del umbral crítico de 1. Aunque la línea roja no es visible en la gráfica

actual, esto se debe a que está fuera del rango visual, y podría mostrarse ajustando manualmente los límites del eje Y.

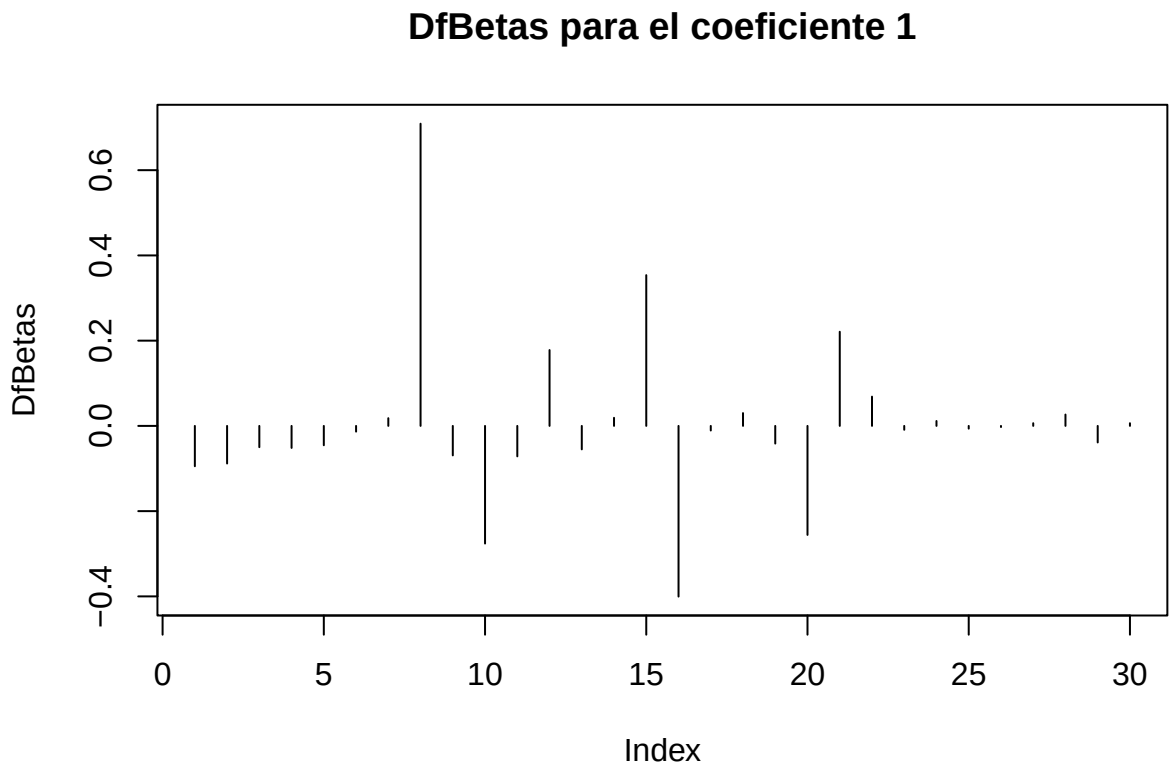
#DFbetas

obtenemos los dfbetas

```
# Calcular DFBETAS para el modelo_optimo_alcorte
dfbetas_values = dfbetas(modelo_optimo_alcorte)
```

calculos de los dfbetas para el primer coeficiente

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



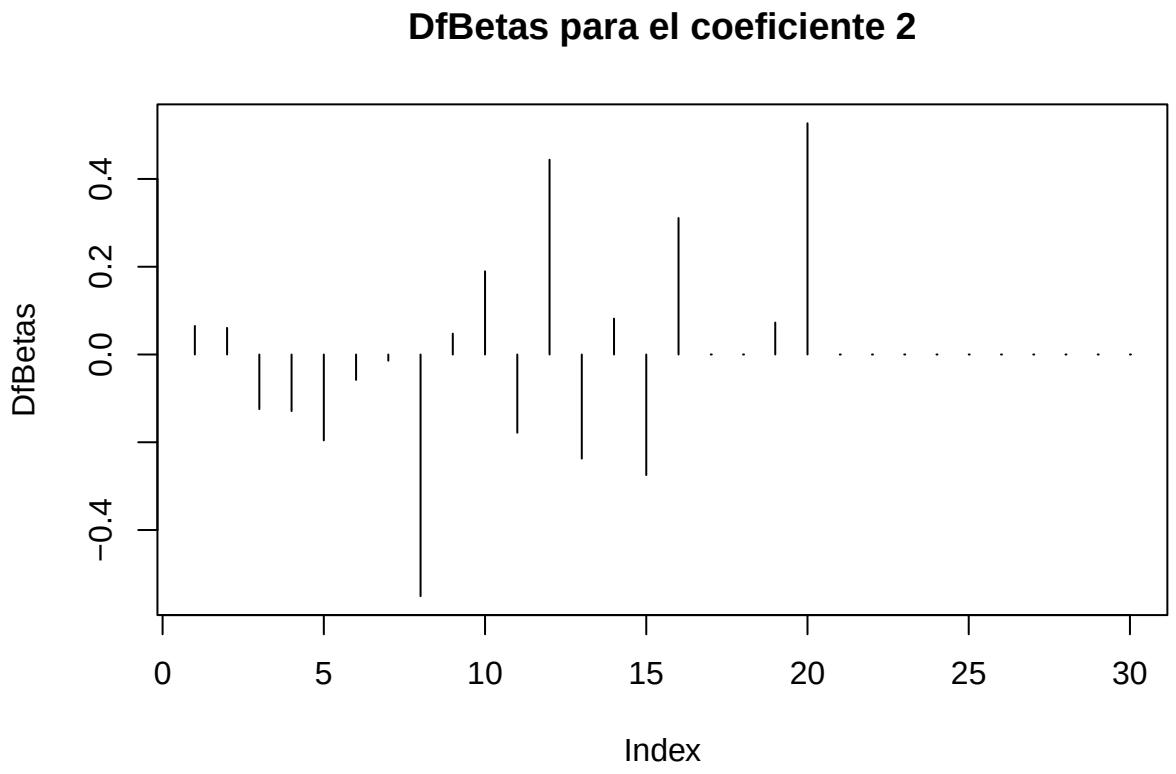

```
# Identificar las observaciones influyentes para el coeficiente 2
puntos_influyentes = which(abs(dfbetas_values[, 1]) > 1)

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

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

calculos de los dfbetas para el segundo coeficiente

```
# Graficar los 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")
```




```
# Identificar las observaciones influyentes para el coeficiente 2
puntos_influyentes = which(abs(dfbetas_values[, 2]) > 1)

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

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

No se identifican observaciones influyentes en ambos gráficos para los coeficientea , ya que todas las barras están lejos del umbral crítico de 1 o -1. Esto indica que ninguna observación está generando un cambio significativo en el valor del coeficiente al ser eliminada. Los valores bajos de DFBETAS sugieren que el modelo es estable en relación con las observaciones para este coeficiente.

#Conclusiones generales:

En general, las observaciones en el modelo no parecen tener una influencia preocupante ni un gran impacto en los coeficientes o el ajuste general. Aunque algunas tienen un leverage alto, no afectan negativamente el modelo. Los valores bajos de la distancia de Cook y DFBETAS confirman que el modelo se mantiene estable

en cuanto a las variables predictoras y los coeficientes. En resumen, el modelo es bastante robusto frente a la influencia de observaciones individuales.