Act3_Atipicos

Rodolfo Jesús Cruz Rebollar

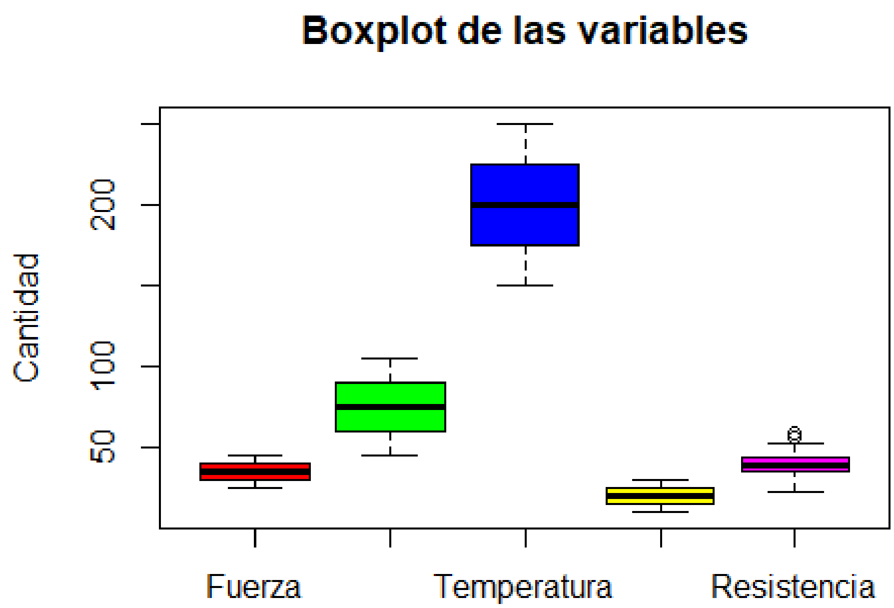
2024-09-26

```r
# Leer la base de datos

datos_corte = read.csv("Corte.csv")

head(datos_corte)

##   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
```

1. Análisis descriptivo de los datos

Diagramas de caja y bigote de las variables

```r
# Boxplot por cada variable

boxplot(datos_corte, main = "Boxplot de las variables",
        col = c("red", "green", "blue", "yellow", "magenta"), ylab = 
"Cantidad")
```



Cálculo de medidas estadísticas por variable

```
# Calcular estadísticos descriptivos básicos por cada variable

summary(datos_corte)

##       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
```

Al graficar el diagrama de caja y bigote, además de calcular ciertas medidas estadísticas básicas por cada variable, es posible observar que particularmente las variables fuerza, tiempo y resistencia presentan una caja mayormente pequeña al graficar sus respectivos boxplots, lo cual aunado a que las medidas estadísticas de dichas varables presentan poca variación, es posible afirmar que en términos generales, las variables de fuerza, tiempo y resistencia presentan una escasa variación entre sus datos, mientras que por el contrario, también se logra apreciar que las variables de potencia y temperatura son las que poseen la mayor cantidad de variación entre sus datos, esto debido a que la caja de sus boxplots correspondientes es más grande o larga que la de las variables mencionadas al principio, lo cual se confirma observando que las medidas estadísticas de potencia y temperatura poseen un mayor rango de variación que las de fuerza, tiempo y resistencia.

Cálculo de sesgo y curtosis por variable

```r
# Librería para calcular sesgo y curtosis

library(moments)
```

Sesgo y curtosis para la variable fuerza:

```r
# Calcular sesgo de la variable fuerza

cat("Sesgo de fuerza: ", skewness(datos_corte$Fuerza), "\n")

## Sesgo de fuerza:  0

# Calcular curtosis de la variable fuerza

cat("Curtosis de fuerza: ", kurtosis(datos_corte$Fuerza))

## Curtosis de fuerza:  2.5
```

Sesgo y curtosis para la variable potencia:

```r
# Calcular sesgo de la variable potencia

cat("Sesgo de potencia: ", skewness(datos_corte$Potencia), "\n")

## Sesgo de potencia:  0

# Calcular curtosis de la variable potencia

cat("Curtosis de potencia: ", kurtosis(datos_corte$Potencia))

## Curtosis de potencia:  2.5
```

Sesgo y curtosis para la variable temperatura:

```r
# Calcular sesgo de la variable temperatura

cat("Sesgo de temperatura: ", skewness(datos_corte$Temperatura), "\n")

## Sesgo de temperatura:  0

# Calcular curtosis de la variable temperatura

cat("Curtosis de temperatura: ", kurtosis(datos_corte$Temperatura))

## Curtosis de temperatura:  2.5
```

Sesgo y curtosis para la variable tiempo:

```r
# Calcular sesgo de la variable tiempo

cat("Sesgo de tiempo: ", skewness(datos_corte$Tiempo), "\n")
```

```
## Sesgo de tiempo:  0
# Calcular curtosis de la variable tiempo

cat("Curtosis de tiempo: ", kurtosis(datos_corte$Tiempo))
## Curtosis de tiempo:  2.5
```

Sesgo y curtosis para la variable resistencia:

```
# Calcular sesgo de la variable resistencia

cat("Sesgo de resistencia: ", skewness(datos_corte$Resistencia), "\n")
## Sesgo de resistencia:  0.2352167
# Calcular curtosis de la variable resistencia

cat("Curtosis de resistencia: ", kurtosis(datos_corte$Resistencia))
## Curtosis de resistencia:  2.838434
```

Por otra parte, es posible apreciar que al momento de calcular tanto el sesgo como la curtosis por cada variable, el sesgo y la curtosis poseen un valor de 0 y 2.5 respectivamente para el caso de todas las variables, a excepción de la resistencia, cuyo sesgo y curtosis son de 0.23 y 2.83 respectivamente, lo cual indica que dichas variables (todas excepto resistencia) poseen distribuciones de tipo leptocúrticas, ya que poseen un alto grado de curtosis, mientras que el sesgo es prácticamente nulo, además, de forma similar, en cuanto a la variable de resistencia, su sesgo de 0.23 indica que la distribución tiene una mayor concentración de datos a la izquierda del gráfico, mientras que en la región derecha del mismo es donde hay una minoría de los datos en cuestión, además al mismo tiempo, la curtosis de 2.83 señala que los datos de resistencia igualmente poseen un elevado nivel de curtosis, propiciando que la variable resistencia también siga una distribución de tipo leptocúrtica, motivo por el cual, en el caso de todas las variables, la alta curtosis es aparentemente el principal impedimiento para que los datos de dichas variables sigan una distribución normal.

2. Mejor modelo de regresión para la resistencia

```
# Crear modelo base de la resistencia contra todas las variables
predictoras

model_complete = lm(Resistencia ~., data = datos_corte)

summary(model_complete)
##
## Call:
## lm(formula = Resistencia ~ ., data = datos_corte)
##
```

```
## 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
```

Modelo con selección mixta de variables

```
# Obtención del mejor modelo mediante selección mixta de variables

modelo_mixto = step(model_complete, 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
```

Modelo con selección hacia adelante de variables

```
# Crear modelo nulo como punto de partida

null_model = lm(Resistencia ~ 1, data = datos_corte)

summary(null_model)

##
## Call:
## lm(formula = Resistencia ~ 1, data = datos_corte)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.7067  -3.7317   0.1933   4.2933  20.2933
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   38.407      1.635   23.49   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.954 on 29 degrees of freedom

# Realizar la selección de variables hacia adelante empleando como base
el modelo nulo

forward_model = step(null_model, scope = list(lower = null_model, upper =
model_complete),
                     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
```

Modelo con selección de variables hacia atrás

```r
# Generar mejor modelo mediante selección de variables hacia atrás
(backward)

backward_model = step(model_complete, direction = "backward")
```

```
## Start:  AIC=102.96
## Resistencia ~ Fuerza + Potencia + Temperatura + Tiempo
##
##               Df Sum of Sq     RSS    AIC
## - Fuerza       1     26.88  692.00 102.15
## - Tiempo       1     40.04  705.16 102.72
## <none>                      665.12 102.96
## - Temperatura  1    252.20  917.32 110.61
## - Potencia     1   1341.01 2006.13 134.08
##
## Step:  AIC=102.15
## Resistencia ~ Potencia + Temperatura + Tiempo
##
##               Df Sum of Sq     RSS    AIC
## - Tiempo       1     40.04  732.04 101.84
## <none>                      692.00 102.15
## - Temperatura  1    252.20  944.20 109.47
## - Potencia     1   1341.02 2033.02 132.48
##
## Step:  AIC=101.84
## Resistencia ~ Potencia + Temperatura
##
##               Df Sum of Sq     RSS    AIC
## <none>                      732.04 101.84
## - Temperatura  1     252.2  984.24 108.72
## - Potencia     1    1341.0 2073.06 131.07
```

En términos generales, es posible apreciar que durante el proceso iterativo de selección de variables mediante las 3 metodologías: mixto, hacia adelante, o hacia atrás, se comienza calculando el valor del indicador AIC para el modelo base del algoritmo (ya sea el modelo nulo, o el modelo que contempla todas las variables predictoras) y acto seguido, se procede a calcular el valor del AIC en caso de agregar o quitar cada una de las variables iniciales, por lo que se identifica aquella acción (agregar o quitar variable) que posea el AIC más bajo en cada iteración del algoritmo y a continuación, se procede a quitar o agregar la variable en cuestión al modelo actual para que en la próxima iteración del algoritmo, el AIC más bajo en la iteración previa será el AIC inicial en la iteración actual (sería el valor del AIC del modelo en caso de no agregar ni quitar ninguna variable del mismo), por lo cual, nuevamente se calcula el AIC para cada posible alternativa de agregación o eliminación de variables, se busca el menor AIC de todos y se realiza la agregación o eliminación de la variable correspondiente y así sucesivamente hasta que llegue un punto en el que ya no sea posible obtener un AIC más bajo que el mejor AIC hasta ese momento, por lo que llegado a dicho punto, el modelo que se tenga en ese momento, será el mejor modelo (aquel con el AIC mínimo).

Summary de los 3 modelos obtenidos (mixto, hacia adelante, hacia atrás)

Nota: Dado que los 3 modelos son en realidad el mismo, basta con imprimir únicamente el summary de uno de ellos para poder llevar a cabo el posterior proceso de validación de dichos modelos.

```r
# Imprimir summary del modelo mixto (combinación de forward y backward)

summary(modelo_mixto)

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

Hipótesis para la significancia global del modelo:

H_0: el modelo no es estadísticamente significativo.

H_1: el modelo sí es estadísticamente significativo.

Hipótesis para la significancia de cada β_i

H_0: $\beta_i = 0$ (el coeficiente β_i no es estadísticamente significativo)

$H_1: \beta_i \neq 0$ (el coeficiente β_i sí es estadísticamente significativo)

Análisis del modelo encontrado en base a su significancia

En términos generales, en el resumen del modelo calculado, es posible observar que en primer lugar, el modelo obtenido posee un p valor igual a 1.674e-07, lo cual al ser mucho menor que 0.05, se tiene suficiente evidencia estadística para rechazar H_0 de la hipótesis de significancia global del modelo, motivo por el cual, es posible concluir que el modelo encontrado sí es estadísticamente significativo.

Además de lo anterior, en cuanto a la significancia individual de los coeficientes del modelo, es posible apreciar que en el caso particular del coeficiente β_0 (intercepto) del modelo, el p valor de dicho coeficiente es igual a 0.02001, lo cual al ser menor que 0.05, se tiene evidencia estadística suficiente para rechazar H_0 de la hipótesis asociada a la significancia de los coeficientes β_i, por lo cual, es posible afirmar que el intercepto del modelo β_0 sí es estadísticamente significativo, de manera similar, en el caso del coeficiente β_1 asociado a la potencia, se puede apreciar que su p valor es de 1.47e-07, lo cual al ser menor que 0.05, se tiene evidencia estadística para rechazar H_0, motivo por el cual, es posible afirmar que el coeficiente β_1 sí es estadísticamente significativo y en consecuencia, la variable potencia también resulta ser estadísticamente significativa, además, en cuanto al coeficiente β_2 asociado a la variable temperatura, su p valor es igual a 0.00508, lo cual al ser considerablemente menor que 0.05, se tiene evidencia estadística para rechazar H_0, motivo por el cual, se puede afirmar que el coeficiente β_2 sí es estadísticamente significativo y por tanto, la variable temperatura asociada a él también es significativa en el contexto estadístico, por lo cual se concluye que todas las variables de modelo son estadísticamente significativas.

Adicionalmente, también cabe mencionar que el modelo encontrado tiene un coeficiente de determinación R^2 igual a 0.6619, lo cual significa que el modelo es capaz de explicar el 66.19% de la variabilidad total de los datos en cuestión, por lo cual el modelo sí consigue explicar la mayoría de la variabilidad presente en los datos analizados, por lo que se puede afirmar que las predicciones derivadas del mismo son mayormente confiables, aunque aún tienen un margen de error medianamente considerable, además de que también el modelo obtenido posee solamente 2 variables predictoras o independientes (temperatura y potencia), motivo por el cual, dado que dicho modelo es el mejor que se obtuvo por medio de los 3 métodos de selección de variables (aquel con el AIC más bajo), además de contar con una cantidad mayormente reducida de variables predictoras, debido a eso es posible afirmar que el modelo en cuestión resulta ser económico, ya que explica el mayor porcentaje posible de variabilidad de los datos, empleando el menor número posible de variables predictoras.

3. Análisis de la validez del modelo encontrado

Análisis de residuos

Media cero

Hipótesis:

$H_0: \mu_R = 0$ (la media de los residuos es cero)

$H_1: \mu_R \neq 0$ (la media de los residuos es diferente de 0)

```
# Verificar que la media de los residuos sea cero con un t test

t.test(modelo_mixto$residuals, mu = 0, conf.level = 0.95, alternative =
"two.sided")

##
##  One Sample t-test
##
## data:  modelo_mixto$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
```

Se aprecia que el p valor resultante del t test es igual a 1, lo cual al ser mayor que la significancia de 0.05, se tiene suficiente evidencia estadística para no rechazar H_0, motivo por el cual, eso indica que en efecto, la media de los residuos del modelo es igual a cero.

Normalidad

Hipótesis:

H_0: los residuos siguen una distribución normal.

H_1: los residuos no siguen una distribución normal.

```
# Librería para tests de normalidad

library(nortest)
```

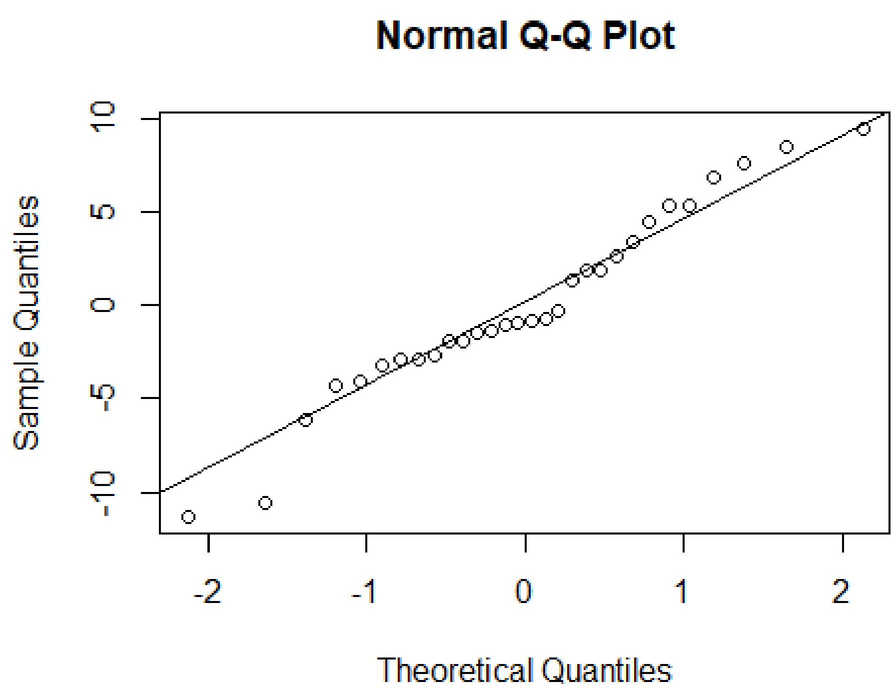
Prueba de normalidad de Anderson Darling

```
# Realizar test de normalidad de Anderson Darling de los residuos del
modelo

ad.test(modelo_mixto$residuals)
```

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

Q-plot de los residuos del modelo

```r
# Graficar el QQ-plot para los residuos del modelo

qqnorm(modelo_mixto$residuals) # graficar percentiles de residuos

qqline(modelo_mixto$residuals) # graficar la recta de normalidad ideal
```



En el qq-plot de los residuos del modelo es posible observar que los percentiles graficados se ubican en su mayoría sobre la recta de normalidad ideal, no obstante, hay algunos otros percetiles que se alejan o desvían de dicha recta principal del gráfico, lo cual indica que hay presencia de un cierto grado de sesgo en los residuos del modelo, ya que los percetiles que se alejan de la recta de normalidad se ubican sobre todo en los extremos de dicha recta, sin embargo, los percentiles que se alejan de la normalidad son pocos en comparación con aquellos que sí se ubican sobre la recta ideal de normalidad, motivo por el cual, los residuos del modelo cumplen aparentemente en su mayoría con el supuesto de normalidad.

Además de lo anterior, también se realizó un test de normalidad de Anderson Darling para los residuos del modelo, en el cual, el p valor resultante es de 0.3204, el cual resulta ser considerablemente mayor a 0.05, por lo cual, se tiene evidencia estadística suficiente para no rechazar H_0 de la hipótesis de normalidad, por lo tanto, se concluye que los residuos del mejor modelo encontrado sí siguen una distribución normal.

Homocedasticidad

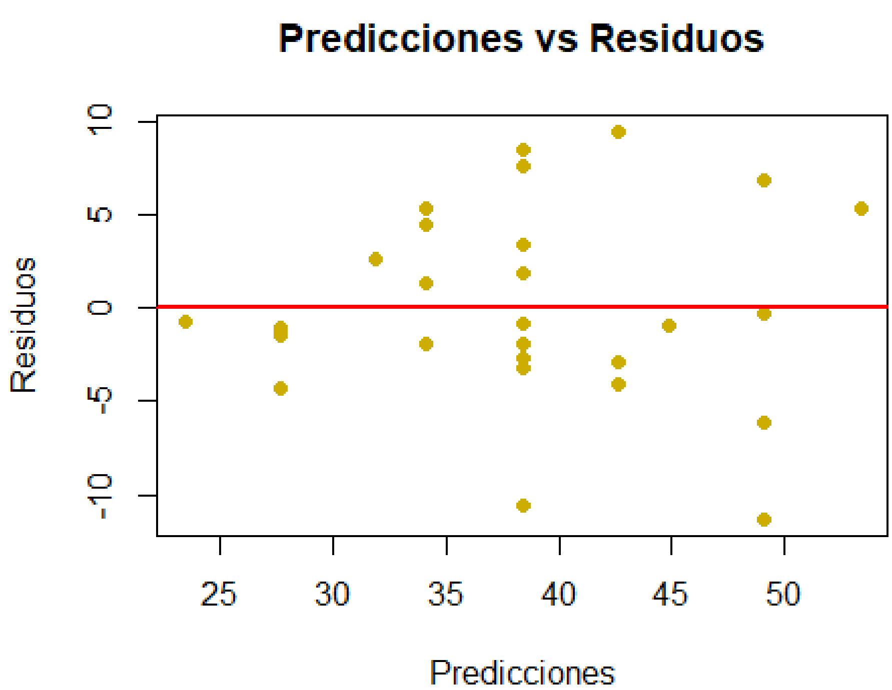
Hipótesis:

H_0: los residuos tienen varianza constante (sí hay homocedasticidad).

H_1: los residuos no tienen varianza constante (no hay homocedasticidad).

Gráfico de predicciones del modelo vs residuos

```
# Graficar las predicciones del modelo contra sus residuos

plot(modelo_mixto$fitted.values, modelo_mixto$residuals,
     col = "gold3", xlab = "Predicciones", ylab = "Residuos",
     main = "Predicciones vs Residuos", pch = 19)

# Graficar línea recta sobre la media cero de los residuos

abline(h = 0, col = "red", lwd = 2)
```



Test de homocedasticidad de Breusch Pagan

```r
# Librería para tests aplicables a modelos de regresión lineal

library(lmtest)

## Loading required package: zoo

##
## Attaching package: 'zoo'

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

# Aplicar test de homocedasticidad de Breusch Pagan a los residuos del
modelo

bptest(modelo_mixto)

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

En el gráfico de predicciones del modelo vs residuos del mismo, es posible apreciar que los residuos del modelo se encuentran distribuidos de una forma aproximadamente equitativa en ambas regiones del gráfico delimitadas por la recta de la media cero, esto sin seguir alguna tendencia o patrón en particular en cuanto a la dispersión de los residuos a lo largo del gráfico, por lo cual, es posible afirmar que la dispersión de los residuos del modelo resulta ser mayormente equitativa en todo el gráfico, además de que también dicha dispersión se atribuye mayormente al azar. Adicionalmente, en cuanto al test de homocedasticidad de Breusch Pagan realizado, es posible apreciar que el p valor del test es igual a 0.135, lo cual al ser mayor que 0.05, se tiene evidencia estadística suficiente para no rechazar H_0 de la hipótesis, motivo por el cual, se puede afirmar que en base al gráfico de predicciones vs residuos y al test de Breusch Pagan, los residuos del modelo obtenido sí tienen varianza constante, es decir, que sí presentan homocedasticidad.

Independencia

Hipótesis:

H_0: los residuos no presentan autocorrelación (sí son independientes).

H_1: los residuos sí presentan autocorrelación (no son independientes).

Nota: se recurrirá a la gráfica previa de predicciones vs residuos para analizar si los residuos del modelo son independientes.

Test de Durbin Watson para independencia

```r
# Realizar test de Durbin Watson para comprobar que los residuos sean independientes

dwtest(modelo_mixto)
```

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

En primera instancia, de acuerdo con el gráfico de predicciones del modelo vs los residuos del mismo, se observa en dicho gráfico que los residuos del modelo se encuentran dispersos de forma que éstos mismos no exhiben ninguna tendencia o patrón en particular, lo cual sugiere que dichos residuos no están aparentemente autocorrelacionados, es decir que son independientes entre ellos, mientras que al mismo tiempo, de acuerdo con el test de independencia de Durbin Watson, se puede apreciar que el p valor resultante de la prueba es de 0.8267, lo cual al ser mucho mayor que 0.05, indica que existe suficiente evidencia estadística para no rechazar H_0, por lo cual, se concluye que los residuos del modelo no presentan autocrrelación, es decir que sí son independientes, confirmando a su vez de esa manera lo que sugiere el gráfico de predicciones vs residuos del modelo (los residuos son independientes ya que no se observa que sigan alguna tendencia o patrón en particular).

Linealidad

Hipótesis:

H_0: No hay términos omitidos que indican linealidad.

H_1: Existe una especificación errónea en el modelo que indica no linealidad.

Test de RESET de Ramsey para verificar linealidad

```r
# Aplicar prueba de RESET de Ramsey para verificar que los residuos se comporten de forma
# lineal

resettest(modelo_mixto)
```

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

En el test de RESET de Ramsey se puede apreciar que el p valor resultante del test es igual a 0.4647, lo cual al ser considerablemente superior a 0.05, se cuenta con suficiente evidencia estadística para no rechazar H_0, motivo por el cual, se puede concluir que no hay términos omitidos en el modelo que indiquen linealidad, en otras palabras, es posible afirmar por tanto, que los residuos del modelo presentan un comportamiento de carácter lineal.

No multicolinealidad de X_i

Matriz de correlación entre predictores del modelo (potencia, temperatura)

```
# Calcular la matriz de correlación entre los predictores del mejor
modelo obtenido

cor(datos_corte)[2:3, 2:3]

##             Potencia Temperatura
## Potencia           1           0
## Temperatura        0           1
```

VIF (factor de inflación de varianza) por variable predictora

```
# Importar librería car para emplear el comando VIF() para el factor de
inflación de
# varianza

library(car)

## Loading required package: carData

# Calcular el VIF por cada variable predictora del modelo encontrado

vifs = vif(modelo_mixto)

# Crear un dataframe con el valor del VIF por predictor

VIF = data.frame(vifs, row.names = c("Potencia", "Temperatura"))

# Nombre de la columna del dataframe

colnames(VIF) = c("VIF")

# Mostrar VIF por variable predictora

VIF

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

En términos generales, en la matriz de correlación entre ambas variables predictoras del mejor modelo encontrado, se aprecia que el grado de correlación entre ambas variables predictoras (potencia y temperatura) es prácticamente nulo (igual a 0), por lo que eso indica que dichas variables predictoras no presentan correlación alguna entre ellas, por lo cual tampoco presentan colinealidad, lo cual se confirma al calcular el VIF (factor de inflación de la varianza) para cada variable predictora, dado que el VIF para cada variable predictora es igual a 1, lo cual al ser un VIF considerablemente inferior a 10, es un VIF mayormente bajo, por lo cual, es posible determinar que en efecto, no hay presencia de colinealidad entre ambas variables predictoras del mejor modelo obtenido, tal como sugiere la matriz de correlación realizada tomando en cuenta únicamente las variables predictoras presentes en el modelo.

4. Conclusiones sobre modelo final e interpretación del efecto de variables predictoras en la variable respuesta

En conclusión, el modelo final encontrado cumple con todos los supuestos de validez de un modelo de regresión múltiple, por lo que las predicciones derivadas del mismo serán mayormente confiables, es decir que dichas predicciones del modelo ilustrarán de la mejor manera posible la relación real existente de la resistencia al corte con la potencia y la temperatura, además de que también el modelo final obtenido es capaz de explicar la mayoría de la variabilidad de los datos originales utilizando la menor cantidad posible de variables predictoras (sólo 2 variables de las 4 variables predictoras iniciales), por lo cual, el modelo en general resulta ser económico, además de altamente preciso y confiable para explicar la resistencia en función de la temperatura y la potencia.

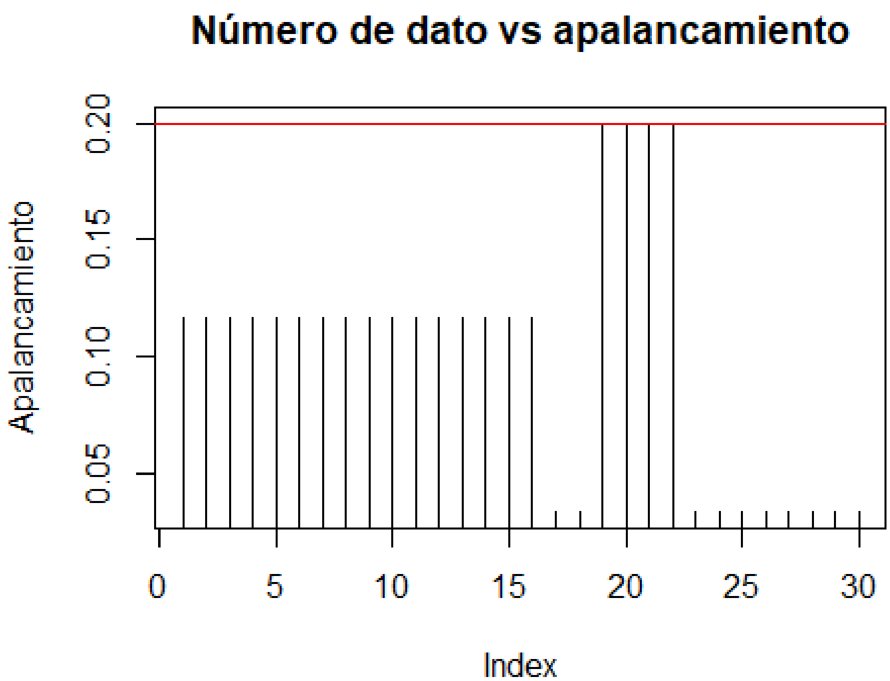
Además de lo anterior, en cuanto al efecto de las variables predictoras en la variable respuesta, es posible concluir que en el contexto del problema, conforme incrementa la potencia, la resistencia al corte también tiende a aumentar, mientras que de la misma forma, al momento de incrementar el valor de la temperatura, los valores de resistencia al corte igualmente tienden en su mayoría a aumentar, motivo por el cual, es posible concluir que existe una tendencia mayormente creciente entre la resistencia al corte y la potencia y la temperatura, motivo por el cual, en caso de que una de las variables predictoras aumente su valor pero al mismo tiempo la otra adopte valores cada vez más pequeños, los valores de la resistencia al corte tenderán a reflejar una ligera disminución, lo cual también ocurrirá en el caso de que ambas variables predictoras tengan valores cada vez más pequeños, por lo cual, hay una relación directamente proporcional entre resistencia al corte y las variables predictoras del modelo, es decir que como ya se mencionó previamente, si ambas variables predictoras aumentan sus valores, la resistencia también será mayor, en cambio, si ambas variables predictoras o una de ellas adopta valores cada vez menores, entonces se producirá el efecto opuesto en la resistencia, es decir, ésta misma disminuirá.

Análisis de datos atípicos e influyentes del mejor modelo

Detección de datos atípicos

Distancia de Leverage

```
# Calcular grados leverage de cada dato implicado en el mejor modelo
encontrado

grados_leverage = hatvalues(modelo_mixto)

# Grados leverage de los primeros datos

head(grados_leverage)

##         1         2         3         4         5         6
## 0.1166667 0.1166667 0.1166667 0.1166667 0.1166667 0.1166667

# Graficar los datos que se tienen contra los valores de apalancamiento
correspondientes
# a cada uno de ellos

plot(grados_leverage, type = "h",
     main = "Número de dato vs apalancamiento", ylab = "Apalancamiento")

# Línea recta horizontal sobre el valor de la cota mínima para considerar
un dato como
# atípico

abline(h = 2*mean(grados_leverage), col = "red2")
```



En el gráfico de apalancamiento previo, se puede observar que se graficaron los 30 datos que se tienen en total (sobre el eje horizontal) contra sus correspondientes valores de apalancamiento (sobre el eje vertical), además de una recta horizontal que pasa justo sobre el valor de la cota mínima para considerar un dato como atípico mediante el criterio de grados leverage (0.2), por lo cual al observar que particularmente los datos número 19, 20, 21 y 22 logran alcanzar la recta horizontal de color rojo, es posible afirmar que los 4 datos antes mencionados son datos atípicos, ya que además dichos datos poseen barras verticales que efectivamente alcanzan al valor de la cota mínima calculada para ser considerados como atípicos, motivo por el cual, existen en total 4 datos atípicos en la base de datos original, mismos que son los datos número 19, 20, 21 y 22.

Además de lo anterior, los datos atípicos sobre el eje x serán aquellos que tengan una cantidad de grados leverage igual o superior a la siguiente cota mínima, donde h_{ii} se refiere a la cantidad de grados leverage por dato, k es la cantidad de parámetros β del modelo a estimar y n es el total de datos que se tienen:

$$hii > 2\frac{\sum h_{ii}}{n}$$

```
# Calcular la cota mínima para identificar datos atípicos en x por grados leverage

cota_min_leverage = 2 * mean(grados_leverage)
```

```r
cat("Cota mínima para datos atípicos en x mediante grados leverage: ",
    cota_min_leverage)

## Cota mínima para datos atípicos en x mediante grados leverage:  0.2

# Agregar los grados leverage a la base de datos inicial

datos_corte$leverage = grados_leverage

# Buscar y contar aquellos datos cuyos grados leverage sean superiores o
iguales a la cota # mínima previamente definida

outliers_x = which(datos_corte$leverage > cota_min_leverage)

cat("Cantidad de datos atípicos por leverage: ", length(outliers_x))

## Cantidad de datos atípicos por leverage:  2

# Mostrar datos atípicos sobre el eje x por el criterio de grados
leverage

datos_corte[outliers_x, ]

##    Fuerza Potencia Temperatura Tiempo Resistencia leverage
## 19     35       45         200     20        22.7      0.2
## 20     35      105         200     20        58.7      0.2
```

Además de lo anterior, es posible evidenciar que al momento de mostrar los datos atípicos, solamente se logran visualizar 2 registros de los 4 identificados inicialmente en el gráfico de apalancamiento anterior, esto probablemente debido a que pese al hecho de que los 2 registros faltantes (datos número 21 y 22) también tienen 0.2 grados leverage al igual que los 2 registros que sí se muestran, los 2 registros faltantes en realidad es posible que tengan grados leverage muy ligeramente por debajo de 0.2 (cota mínima), por ejemplo 0.1999, por lo cual al no alcanzar en si el valor de 0.2, no se mostrarán como atípicos, no obstante, dado que sus grados leverage se encuentran muy cerca de 0.2, sí es posible considerarlos como datos atípicos, por lo cual, realmente los datos atípicos encontrados sobre el eje x son los datos número 19, 20, 21 y 22.

Estandarización extrema de los residuos

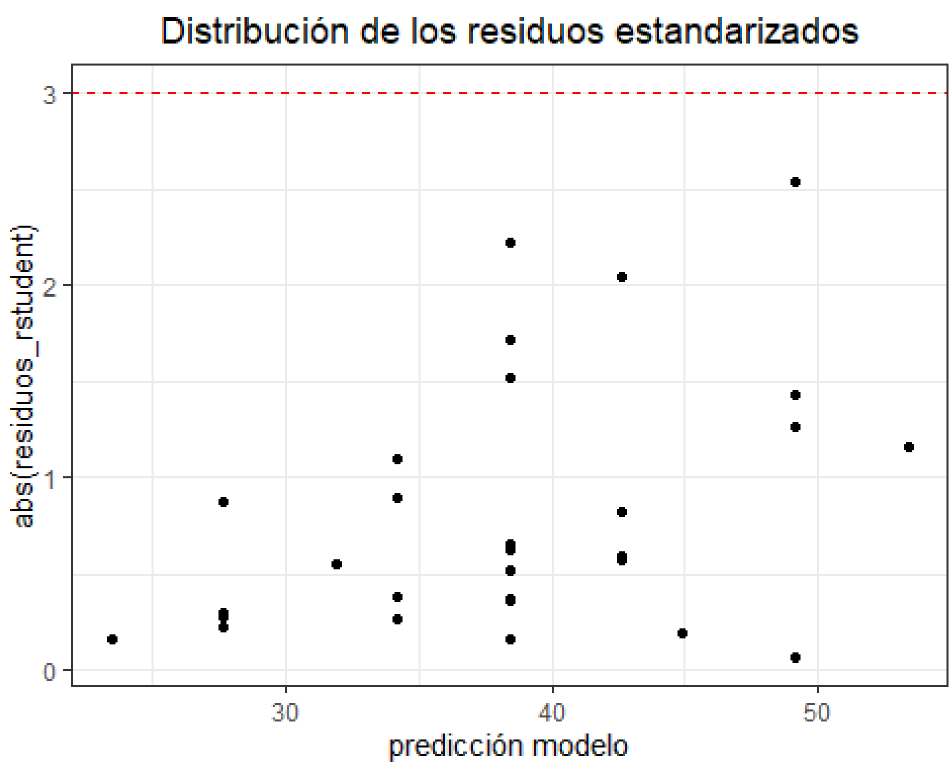
```r
# Aplicar estandarización extrema a cada uno de los errores del modelo
(residuos)

residuos_rstudent = rstudent(modelo_mixto)

# Agregar el cálculo de estandarización extrema a la base de datos
original

datos_corte$t_extreme = residuos_rstudent
```

```r
# Los datos atípicos sobre el eje y acorde al criterio de estandarización extrema
# serán aquellos cuya estandarización extrema supere las 3 desviaciones estándar

outliers_y = which(abs(datos_corte$t_extreme) > 3) # contar cantidad de outliers

cat("Cantidad de datos atípicos por estandarización extrema: ",
length(outliers_y), "\n")

## Cantidad de datos atípicos por estandarización extrema:  0

# Mostrar datos atípicos sobre el eje y, determinados mediante estandarización extrema

datos_corte[outliers_y, ]

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

Al desplegar los registros con los datos atípicos sobre el eje y, se puede notar que no se muestra ningún registro como resultado, lo cual se debe a que los residuos estandarizados mediante estandarización extrema correspondientes a cada uno de los datos originales, son todos inferiores a 3, lo cual a su vez significa que todos los datos presentes en la base de datos original, se ubican en realidad a menos de 3 desviaciones estándar dentro de la distribución de los datos, motivo por el cual, se puede afirmar que prácticamente ningún dato se considera como atípico en base al criterio de estandarización extrema.

```r
# Importar librería dplyr

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
# Importar librería ggplot

library(ggplot2)

# Nota: los datos en rojo corresponden a observaciones cuyo residuo
estandarizado
# con estandarización extrema, es superior a 3

# Realizar gráfica auxiliar de las predicciones del modelo vs sus
residuos
# estandarizados

ggplot(data = datos_corte, aes(x = predict(modelo_mixto),
                               y = abs(residuos_rstudent))) +
geom_hline(yintercept = 3, color = "red", linetype = "dashed") +
geom_point(aes(color = ifelse(abs(residuos_rstudent) > 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))
```



Además de lo anterior, en el gráfico anterior de las predicciones del modelo contra los valores absolutos de los residuos estandarizados, es posible observar que los puntos de color negro a lo largo del gráfico que corresponden a los residuos estandarizados de los datos iniciales, se ubican todos por debajo de la línea punteada en color rojo ubicada a su vez sobre el valor 3 (cantidad máxima de desviaciones estándar para considerar un dato como atípico), por lo cual, en base al gráfico se afirma que no existen datos atípicos al emplear la estandarización extrema, con lo cual se respalda el hecho de no haber obtenido como resultado ningún registro al momento de mostrar anteriormente los datos atípicos determinados mediante el criterio de estandarización extrema.

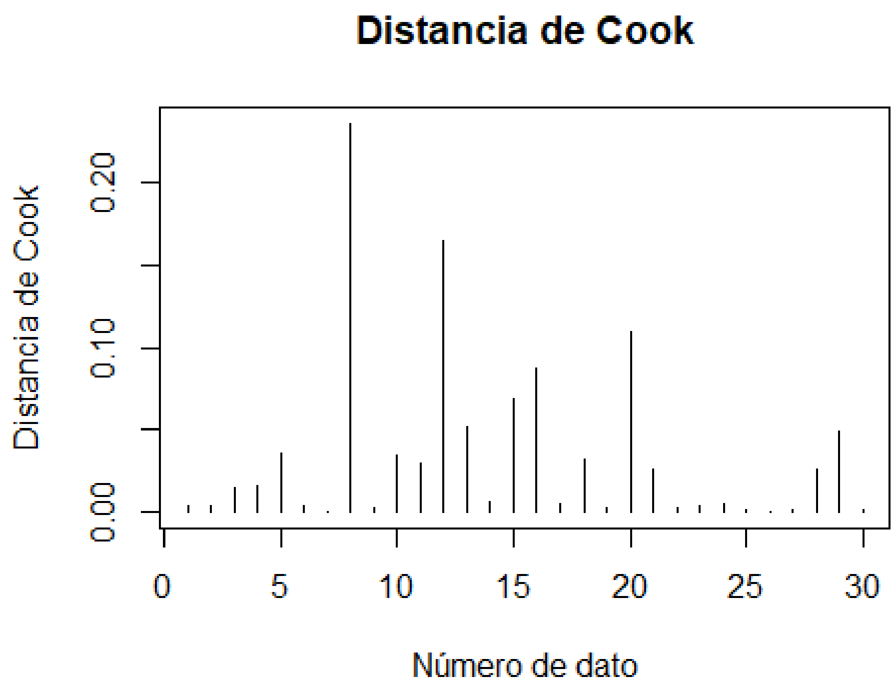
Detección de datos influyentes

Distancia de Cook

```
# Calcular la distancia de Cook para cada uno de los datos

cook = cooks.distance(modelo_mixto)

# Mostrar la distancia de Cook solamente para los primeros datos

head(cook)
```

```
##           1           2           3           4           5 
6 
## 0.004081094 0.003551679 0.014826572 0.015889079 0.036021161 
0.003235398
```

```
# Graficar la distancia de Cook correspondiente a cada uno de los datos

plot(cook, type = "h", main = "Distancia de Cook",
     xlab = "Número de dato", ylab = "Distancia de Cook")

# Graficar una recta en color magenta sobre el valor 1
# (límite mínimo para considerar un dato como influyente)

abline(h = 1, col = "magenta3")
```



En el gráfico previo se graficaron los datos originales con su respectivo número, contra su distancia de Cook correspondiente, además de que las líneas verticales en color negro ubicadas sobre cada dato indican la distancia de Cook que corresponde a dicho dato, por lo que tomando esto en cuenta, es posible observar que ninguna línea vertical en color negro sobrepasa una distancia de Cook de 1, motivo por el cual, eso es indicativo de que no hay presencia de datos influyentes que afecten las estimaciones realizadas por el modelo, motivo por el cual, no hay datos presentes que ocasionen que la recta del modelo de regresión lineal se incline más hacia una región particular del gráfico buscando ajustarse a dicho dato, por lo cual, solamente existen ciertos datos atípicos en la base de datos original, pero éstos mismos no son influyentes.

```
# Identificar cuáles datos son influyentes

influyentes = which(cook > 1)

# Filtrar y mostrar los datos influyentes identificados

datos_corte[influyentes, ]

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

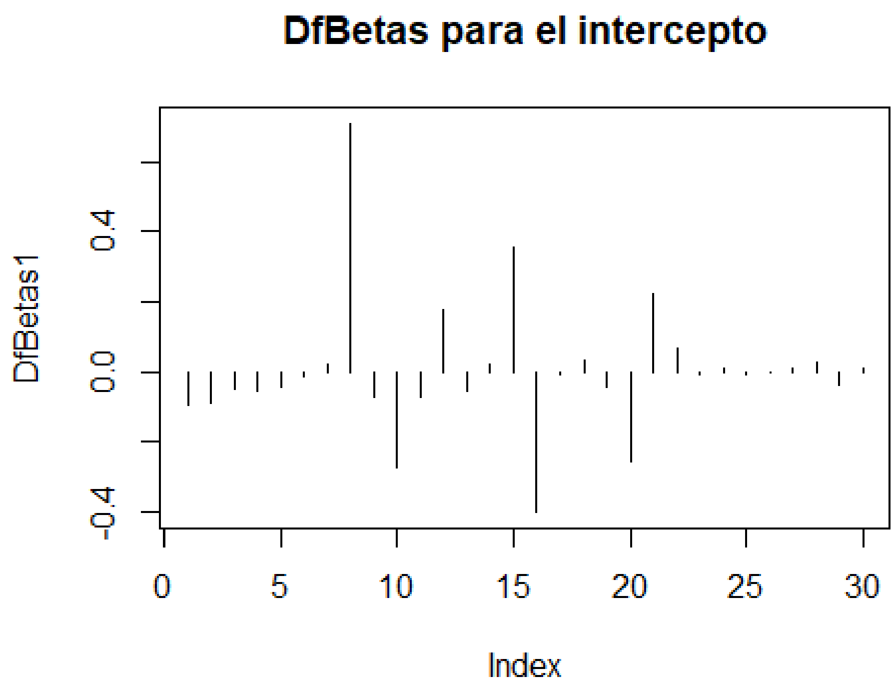
En el resultado anterior se aprecia que no se muestra ningún registro, esto debido a que para todos los datos originales, su distancia de Cook correspondiente resulta ser menor que 1, por lo cual, ningún datos satisface la condición de tener una distancia de Cook superior a 1, por tal motivo, ninguno de los datos se considera como influyente de acuerdo al criterio de la distancia de Cook.

DfBetas

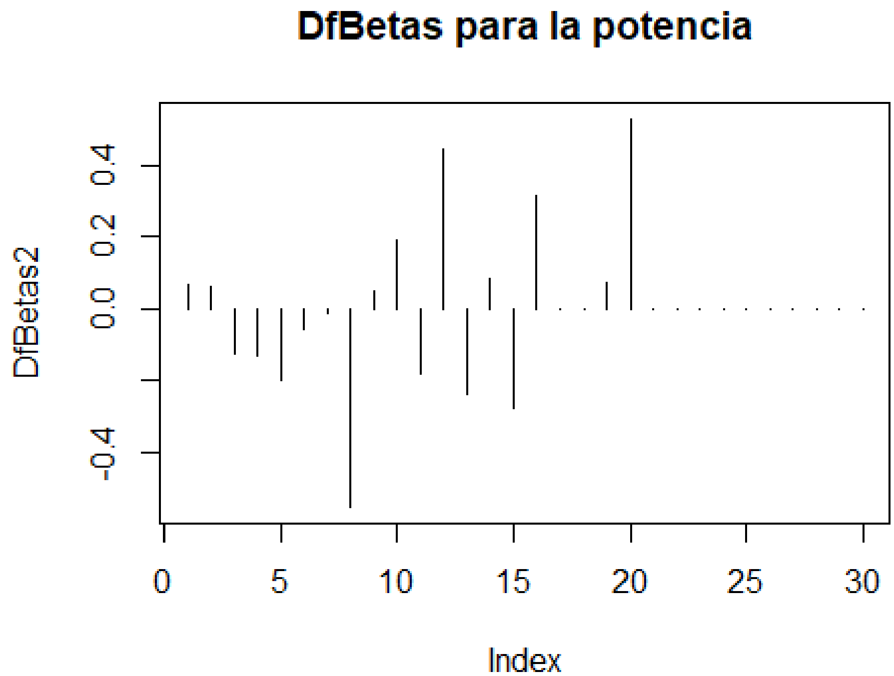
```
# Calcular el valor de dfbeta para todos los datos para cada coeficiente
beta del modelo

df_betas = dfbetas(modelo_mixto)

# Mostrar los dfbetas para los primeros datos, asociados a cada uno de
los parámetros beta
# del modelo de regresión

head(df_betas)
```

```
##   (Intercept)    Potencia Temperatura
## 1 -0.09465911  0.06500122  0.06500122
## 2 -0.08828643  0.06062518  0.06062518
## 3 -0.04990887 -0.12446072  0.12446072
## 4 -0.05168962 -0.12890149  0.12890149
## 5 -0.04544979 -0.19577054  0.19577054
## 6 -0.01343156 -0.05785516  0.05785516
```

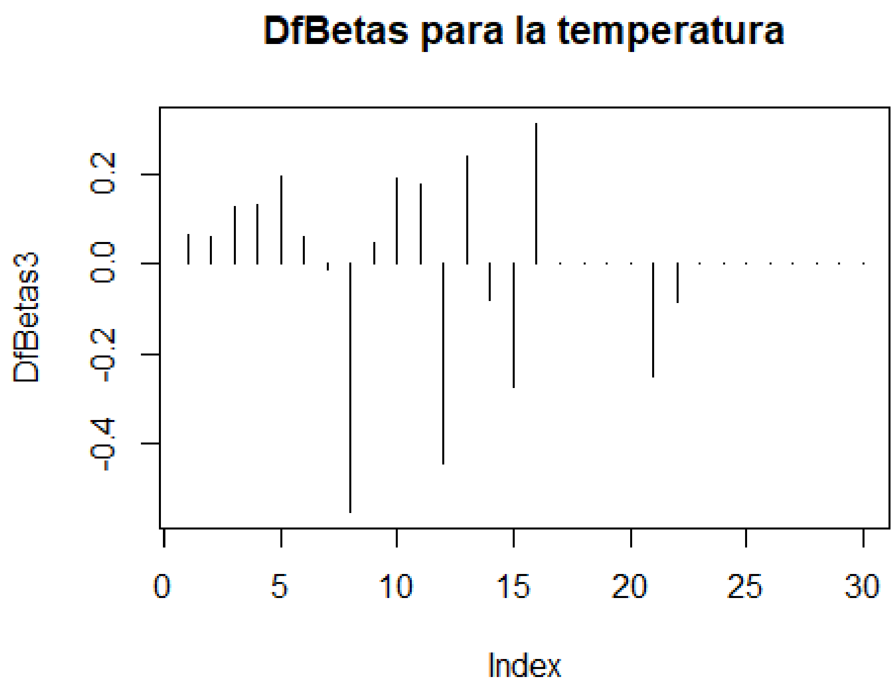
```
# Gráfico auxiliar, para la variable 1 (intercepto del modelo)

plot(df_betas[, 1], type="h", main="DfBetas para el intercepto", ylab =
"DfBetas1")

abline(h = c(-1, 1), col="red")
```




```
# Gráfico auxiliar de los datos vs sus dfbetas para la variable potencia
(coeficiente 2)

plot(df_betas[, 2], type="h", main="DfBetas para la potencia", ylab =
"DfBetas2")

abline(h = c(-1, 1), col="blue")
```




```
# Gráfico auxiliar de datos vs sus dfbetas para la variable temperatura (coeficiente 3)

plot(df_betas[, 3], type="h", main="DfBetas para la temperatura", ylab = "DfBetas3")

abline(h = c(-1, 1), col="green")
```




```
# Identificar los datos influyentes presentes en los dfbetas del
intercepto

dfbetas_influyentes1 = which(abs(df_betas[, 1]) > 1)

cat("Datos influyentes en dfbetas del intercepto: ",
dfbetas_influyentes1, "\n")

## Datos influyentes en dfbetas del intercepto:

# Identificar datos influyentes en los dfbetas de la variable 2
(potencia)

dfbetas_influyentes2 = which(abs(df_betas[, 2]) > 1)

cat("Datos influyentes en dfbetas de la potencia: ",
dfbetas_influyentes2, "\n")

## Datos influyentes en dfbetas de la potencia:

# Identificar datos influyentes en los dfbetas de la variable 3
(temperatura)

dfbetas_influyentes3 = which(abs(df_betas[, 3]) > 1)

cat("Datos influyentes en dfbetas de la temperatura: ",
dfbetas_influyentes3)
```

```
## Datos influyentes en dfbetas de la temperatura:
```

En el resultado anterior se aprecia que no se identificaron datos influyentes, esto debido a que básicamente, el valor absoluto de los dfbetas son todos inferiores a 1, motivo por el cual ningún dfbeta calculado satisface el criterio para considerar que cada uno de los coeficientes β del modelo de regresión sean influyentes en el modelo, lo cual significa que para el caso de todos los coeficientes del modelo, el cambio en dichos coeficientes no será significativo en caso de eliminar cualquiera de las observaciones en base a las que se construyó el modelo, por lo que debido a ese motivo, no hay factores influyentes en los coeficientes del modelo.

Resumen de las medidas de influencia de los datos

Otra forma más rápida de calcular las medidas de influencia de los datos anteriormente obtenidas es por medio de la función **influence.measures(modelo)**, misma que arroja como resultado, 3 métricas para evaluar si existe presencia de datos influyentes: distancia de Cook, DfBetas y distancia de leverage.

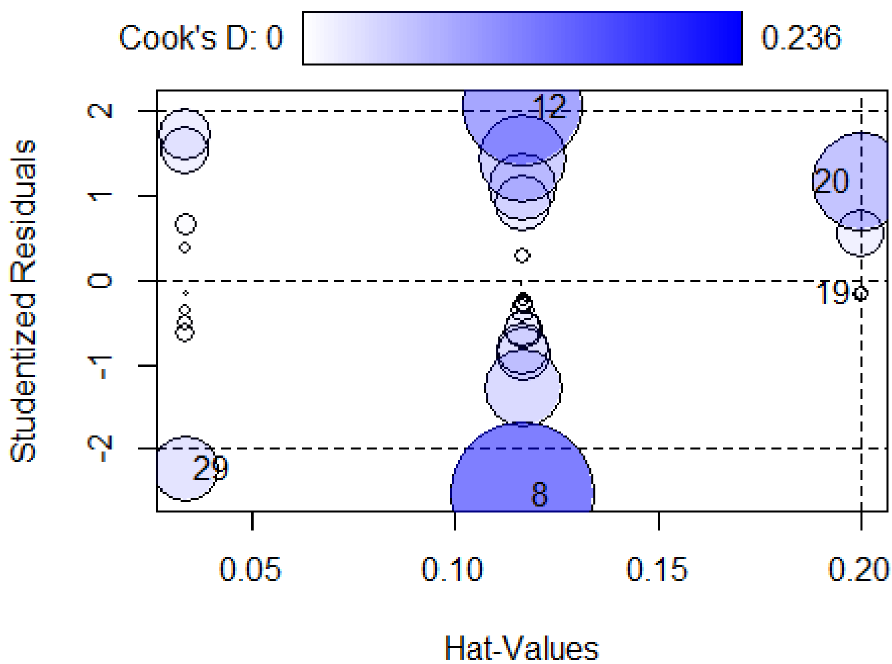
```
# Calcular con la función influence.measures() la distancia de Cook y de
Leverage, además
# de los DfBetas

medidas_influencia = influence.measures(modelo_mixto)

# Mostrar resumen de los datos que son candidatos a ser datos influyentes
en el modelo

summary(medidas_influencia)

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

En la tabla previa se muestra que las observaciones candidatas a ser influyentes en el modelo de regresión son la número 8, 19, 21 y 22, por lo que se observa que se determinó que las observaciones 19, 21 y 22 son candidatas a ser influyentes, sin embargo, dado que todas las observaciones sugeridas por la función influence.measures() poseen una distancia de Cook menor a 1, no se pueden considerar como observaciones influyentes en el modelo de regresión lineal.

Gráfico de influencia (influencePlot)

```
# Librería car para usar la función influencePlot
```

```r
library(car)

# Gráfica de los residuos del modelo estandarizados con estandarización extrema,
# grados leverage y distancia de Cook para los datos, además de mostrar los registros
# influyentes

influencePlot(modelo_mixto)
```

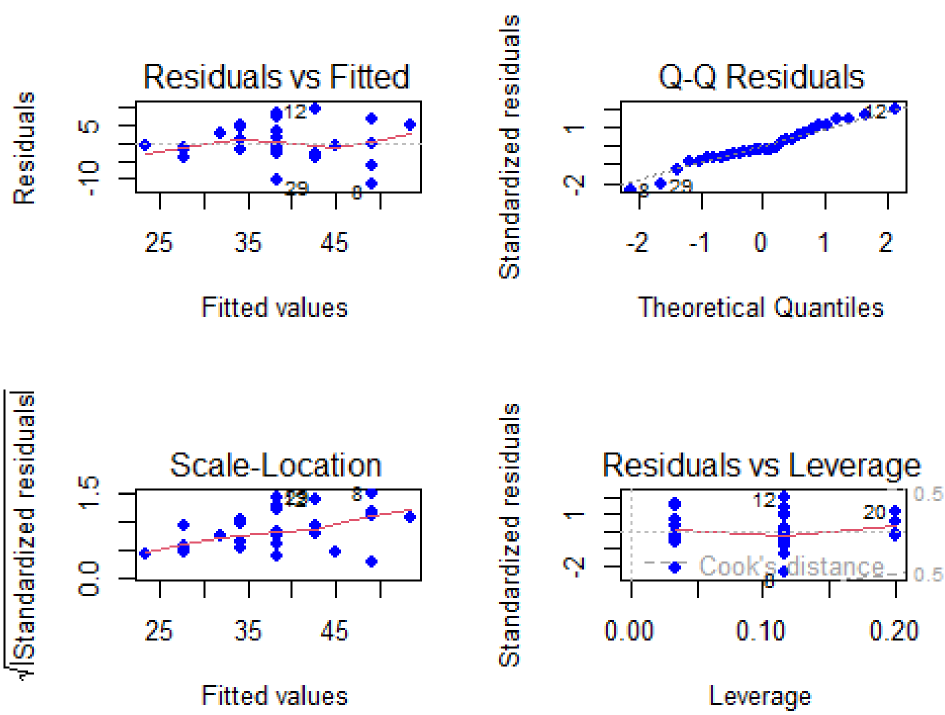



```
##        StudRes        Hat       CookD
## 8   -2.535832 0.11666667 0.235696235
## 12   2.043589 0.11666667 0.164507739
## 19  -0.159511 0.20000000 0.002199712
## 20   1.154355 0.20000000 0.109693544
## 29  -2.216952 0.03333333 0.049338917
```

En el gráfico de influencia (influencePlot) se puede apreciar que se graficaron los grados leverage de los datos (hat values) contra los residuos con estandarización extrema, además de que los círculos graficados poseen una determinada tonalidad de azul, por lo que entre más fuerte sea dicha tonalidad, significa que la observación señalada con número en el círculo tiene fuerte grado de influencia en el modelo, mientras que si por el contrario, la tonalidad de azul es clara, significará que el grado de influencia de la observación es bajo, por lo que se evidencia que la observación 8 tiene el mayor grado de influencia de todas las observaciones, aunque dicho grado de influencia no es suficiente para considerar formalmente a dicha observación como influyente.

Plot del modelo

```r
# Realizar gráficas de los diferentes atributos del modelo de regresión: residuos vs
# valores ajustados, QQ-plot de residuos, residuos estandarizados vs valores ajustados y
# residuos estandarizados vs distancia de Cook y de Leverage

par(mfrow = c(2, 2))

plot(modelo_mixto, col = "blue", pch = 19)
```



En los gráficos anteriores, se aprecia que en el primer gráfico correspondiente a las predicciones del modelo vs los residuos del mismo, las observaciones 8, 12 y 29 se encuentran mayormente alejadas de la zona principal de puntos, lo cual significa que son candidatas a ser datos atípicos en el modelo de regresión, lo cual ya se comprobó anteriormente que no es así, sino que las observaciones que realmente son datos atípicos en el modelo son la número 19, 20, 21 y 22. Por otro lado, en el gráfico de QQ-residuals se aprecia que los puntos en el gráfico se ubican en su mayoría sobre la recta central de color rojo que señala normalidad de los residuos, mientras que las observaciones 8 y 29 son aquellas que se ubican en su mayoría alejadas de la recta central de normalidad, indicando que las observaciones 8 y 29 representan en general los únicos alejamientos de la normalidad entre todos residuos graficados.

Por otra parte, en la gráfica de predicciones del modelo vs residuos estandarizados, es posible observar que entre las observaciones que más se alejan de la recta principal en color rojo son las observaciones número 8, 12 y 29, lo cual es un indicativo de que los residuos estandarizados mediante el método de estandarización extrema correspondientes a dichas observaciones, son los que tuvieron los mayores valores entre todos los residuos estandarizados, lo cual significa que las observaciones 8, 12 y 29 son las que se encuentran alejadas a una mayor cantidad de desviaciones estándar, respecto a la media de las observaciones, además, también cabe mencionar que en el último gráfico de la distancia de Leverage y Cook vs los residuos también estandarizados mediante estandarización extrema, se puede notar que aquellos residuos con mayor distancia de Leverage y Cook son los correspondientes a las observaciones número 12 y 20, mientras que el residuo con la menor distancia de Leverage y Cook corresponde a la observación número 8, lo cual indica que los mayores candidatos a ser datos atípicos y/o influyentes en el modelo son las observaciones 12 y 20, de las cuales, la observación número 20 sí que resultó ser realmente atípica pero no influyente, reforzando así el descubrimiento realizado al encontrar aquellos datos que son atípicos sobre el eje x por medio de los grados leverage, donde se determinó precisamente que la observación número 20 de la base de datos original sí es un dato atípico pero sin una influencia significativa en las estimaciones del modelo de regresión lineal.

En conclusión, el mejor modelo encontrado para la problemática inicial, tiene como datos atípicos a las observaciones número 19, 20, 21 y 22, además de que no tiene datos influyentes.