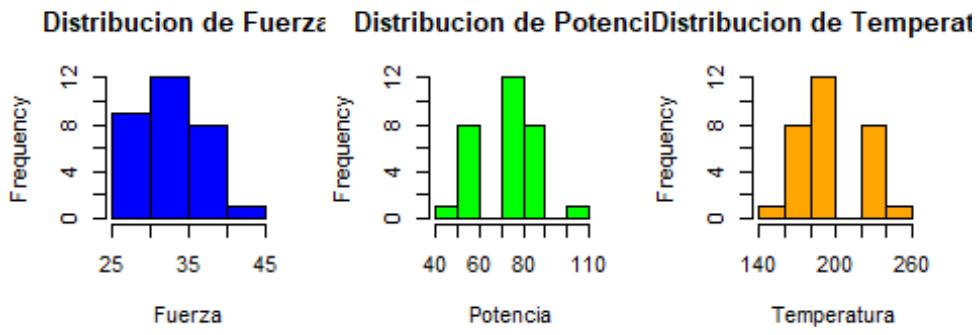
A2_A01571214_Lautaro_Coteja

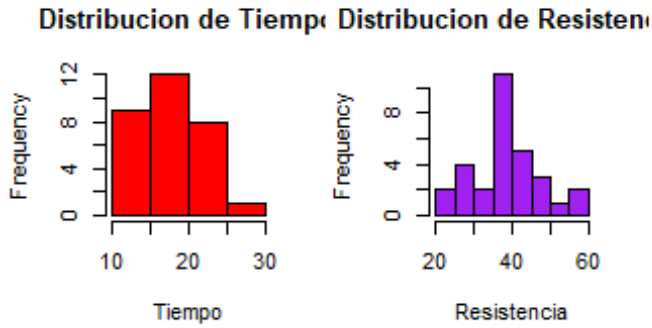
A01571214 - Lautaro Coteja

2024-09-26

R Markdown

```
# Cargar los datos
data = read.csv("C:/Users/lauta/Downloads/AlCorte.csv")
head(data)

##   Fuerza Potencia Temperatura Tiempo Resistencia
## 1     30       60         175     15        26.2
## 2     40       60         175     15        26.3
## 3     30       90         175     15        39.8
## 4     40       90         175     15        39.7
## 5     30       60         225     15        38.6
## 6     40       60         225     15        35.5

# Estadisticas descriptivas
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

# Generar histogramas
par(mfrow=c(2,3)) # Dividir la pantalla en una matriz de 2x3

hist(data$Fuerza, main="Distribucion de Fuerza", col="blue", xlab="Fuerza")
hist(data$Potencia, main="Distribucion de Potencia", col="green",
xlab="Potencia")
hist(data$Temperatura, main="Distribucion de Temperatura", col="orange",
xlab="Temperatura")
hist(data$Tiempo, main="Distribucion de Tiempo", col="red", xlab="Tiempo")
hist(data$Resistencia, main="Distribucion de Resistencia", col="purple",
xlab="Resistencia")

par(mfrow=c(1,1)) # Volver a configuracion normal
```






```r
# Ajustar el modelo de regresion multiple
library(stats)
X = data[c('Fuerza', 'Potencia', 'Temperatura', 'Tiempo')]
y = data$Resistencia

# Agregar la constante al modelo
X = cbind(1, X) # agregar intercepto
colnames(X)[1] = "Intercepto"

# Modelo de regresion
model = lm(Resistencia ~ Fuerza + Potencia + Temperatura + Tiempo, data=data)

# Resumen del modelo
summary(model)
```

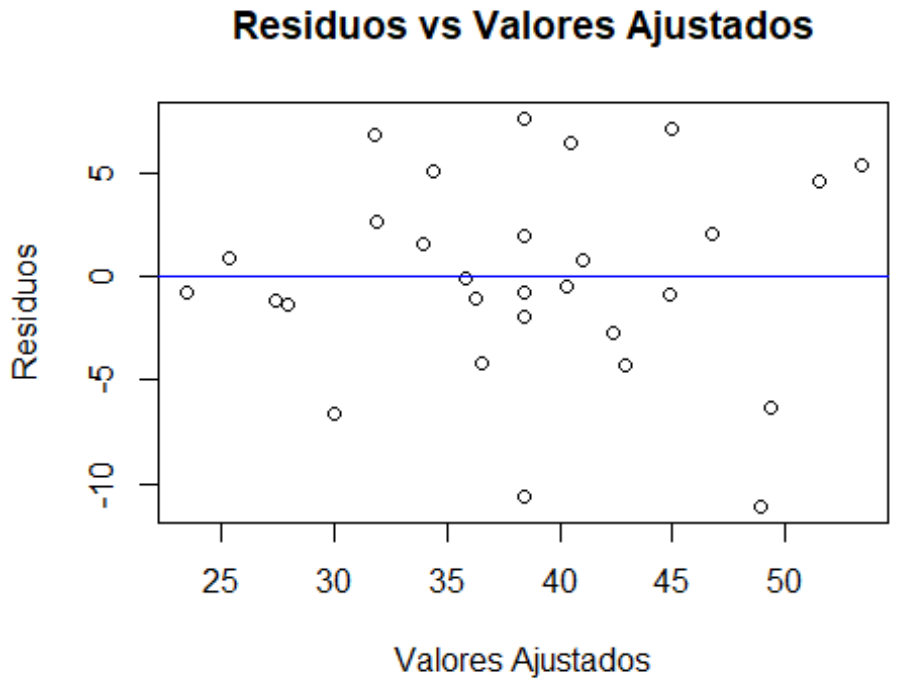
```
##
## Call:
## lm(formula = Resistencia ~ Fuerza + Potencia + Temperatura +
##     Tiempo, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.0900  -1.7608  -0.3067   2.4392   7.5933
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (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
```

Interpretacion de Resultados del Modelo

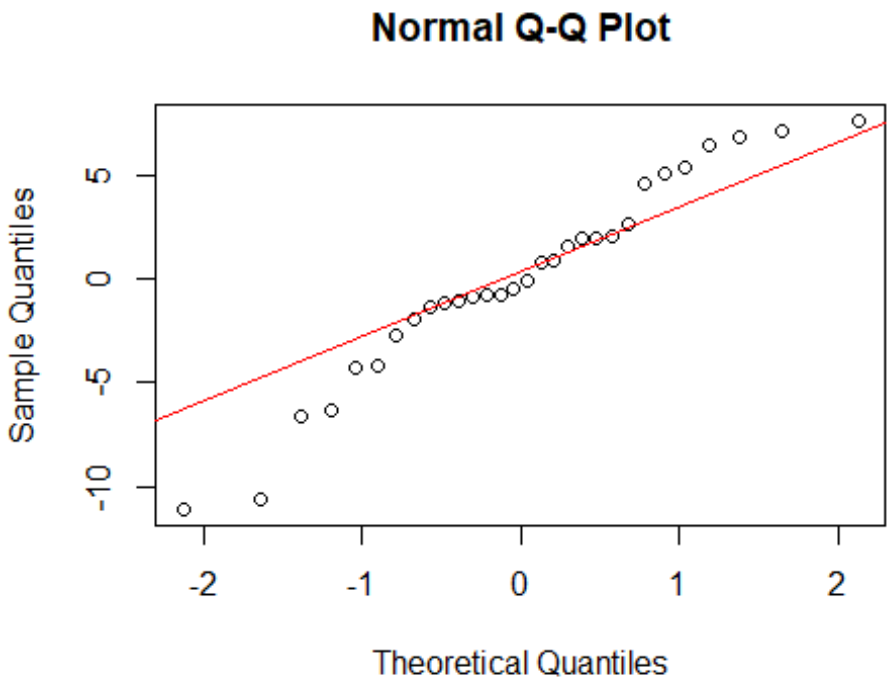
La R cuadrada es de 0.714 por lo que el modelo explica el 71.4% de la variabilidad en la resistencia, en la prueba F se ve que el modelo es globalmente significativo, potencia es muy significativa por lo que tiene efecto importante en la resistencia, al igual que la temperatura, fuerza y tiempo no contribuyen significativamente a explicar la resistencia.

```r
# Graficar residuos vs valores ajustados
plot(model$fitted.values, model$residuals, main="Residuos vs Valores
Ajustados", xlab="Valores Ajustados", ylab="Residuos")
abline(h=0, col="blue")
```




```r
# Grafico Q-Q para verificar la normalidad de los residuos
qqnorm(model$residuals)
```

```r
qqline(model$residuals, col = "red")

# Prueba de Durbin-Watson para autocorrelacion de los residuos
library(lmtest)

## Cargando paquete requerido: zoo

##
## Adjuntando el paquete: 'zoo'

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




```r
dwtest(model)

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

Interpretacion de Residuos

En los residuos vs valores ajustados vemos que los residuos parecen dispersos de manera aleatoria alrededor de 0, por lo que no hay problemas de heterocedasticidad, en la QQPlot los residuos siguen razonablemente bien la linea de normalidad, por lo que los residuos pueden considerarse normales, y en la prueba de Durbin-Watson el valor de 2.26 indica que no hay autocorrelacion significativa de los errores, osea que hay independencia.

```
# Cargar la libreria car para calcular el VIF
library(car)

## Cargando paquete requerido: carData

# Calcular el VIF para cada predictor
vif(model)

##      Fuerza    Potencia Temperatura      Tiempo
##           1           1           1           1
```

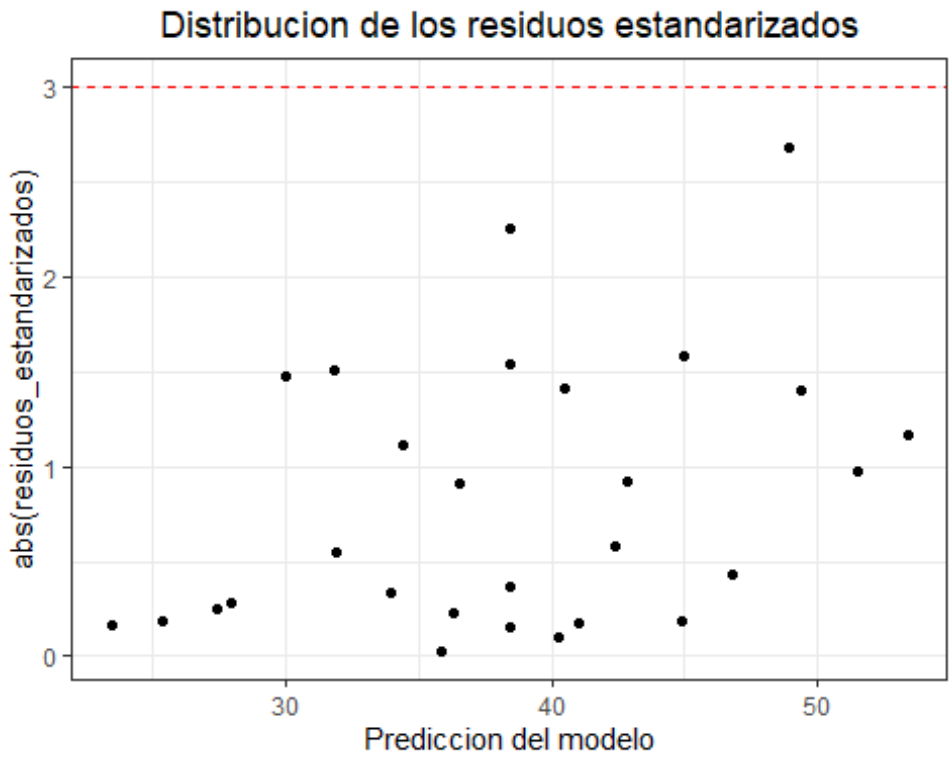
Conclusion

El analisis realizado nos indica que Potencia y Temperatura son los factores mas relevantes para explicar la Resistencia. El modelo es significativo en general y los residuos cumplen con los supuestos de normalidad, homocedasticidad, e independencia.

A3 - Regresion Multiple - Deteccion de Datos Atipicos

```
# Librerias
library(dplyr)

##
## Adjuntando el paquete: '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

library(ggplot2)

# Calcular residuos estandarizados
data$residuos_estandarizados = rstudent(model)

# Graficar residuos estandarizados
```

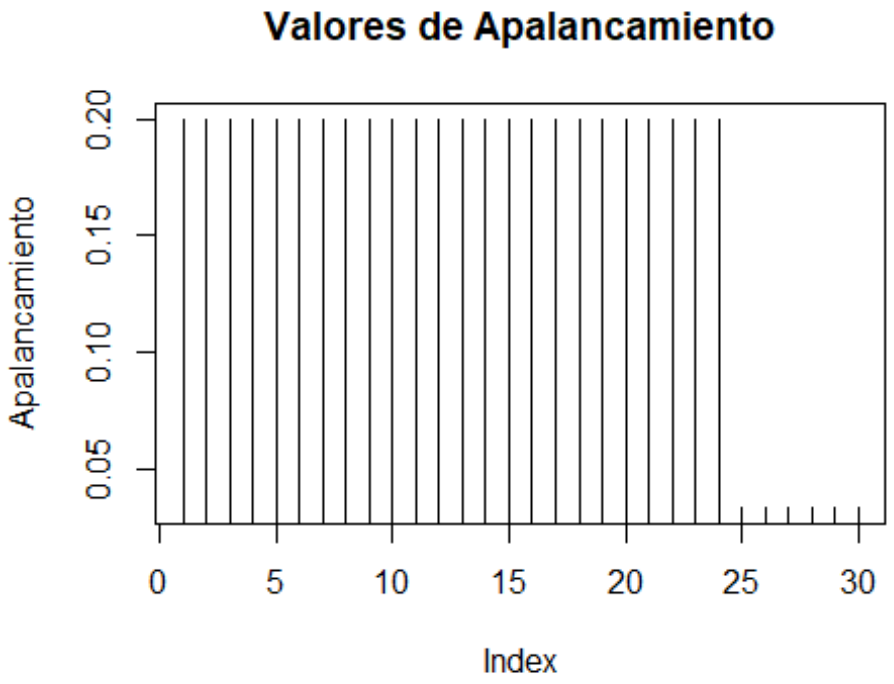
```r
ggplot(data = data, aes(x = predict(model), y =
abs(residuos_estandarizados))) +
  geom_hline(yintercept = 3, color = "red", linetype = "dashed") +
  geom_point(aes(color = ifelse(abs(residuos_estandarizados) > 3, 'red',
'black'))) +
  scale_color_identity() +
  labs(title = "Distribucion de los residuos estandarizados", x = "Prediccion
del modelo") +
  theme_bw() + theme(plot.title = element_text(hjust = 0.5))
```




```r
# Identificar los datos atipicos
atipicos = which(abs(data$residuos_estandarizados) > 3)
data[atipicos, ]
```

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

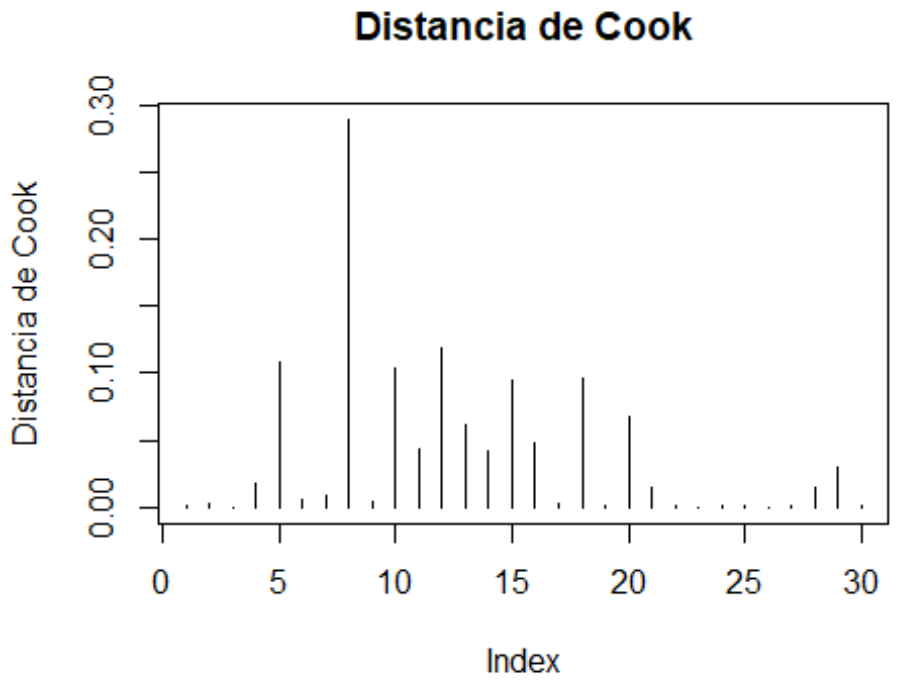
```r
# Calcular leverage
leverage = hatvalues(model)

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




```r
# Identificar datos con alto leverage
high_leverage_points = which(leverage > 2*mean(leverage))
data[high_leverage_points, ]
```

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

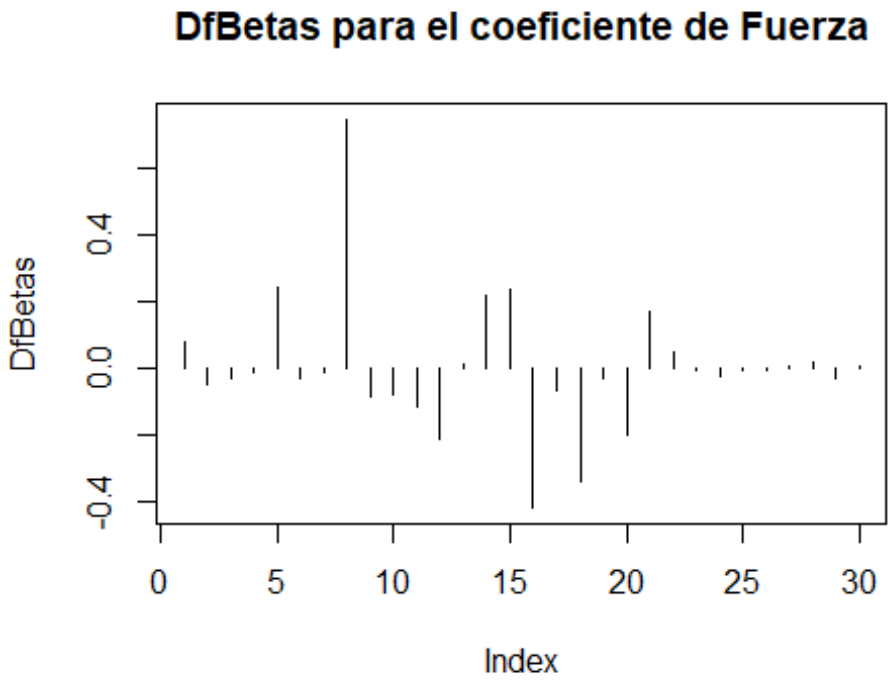
```r
# Calcular distancia de Cook
cooksd = cooks.distance(model)

# Graficar distancia de Cook
plot(cooksd, type="h", main="Distancia de Cook", ylab="Distancia de Cook")
abline(h = 1, col="red")
```



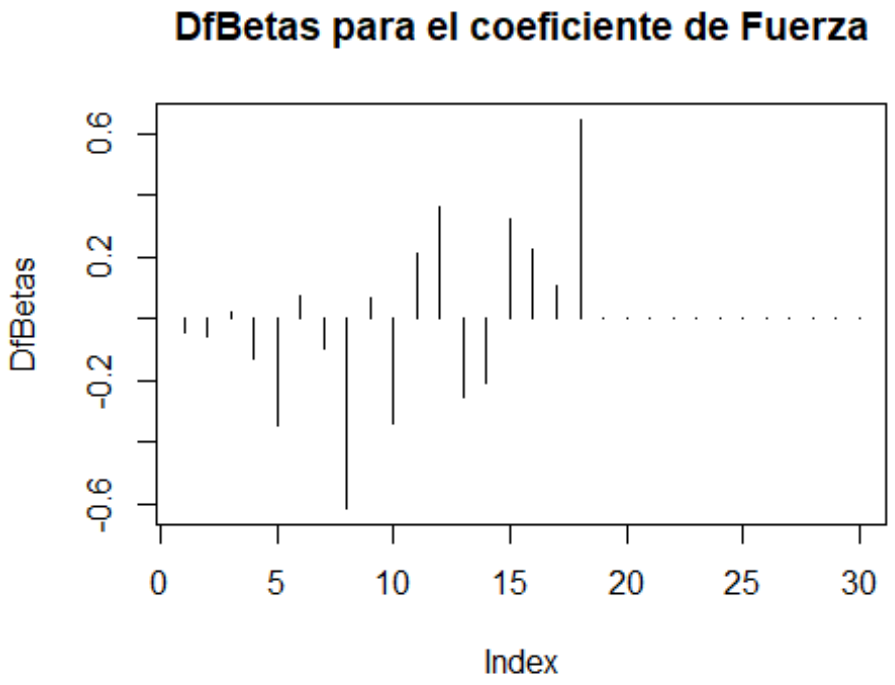

```r
# Identificar puntos influyentes
puntos_influyentes = which(cooksd > 1)
data[puntos_influyentes, ]
```

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

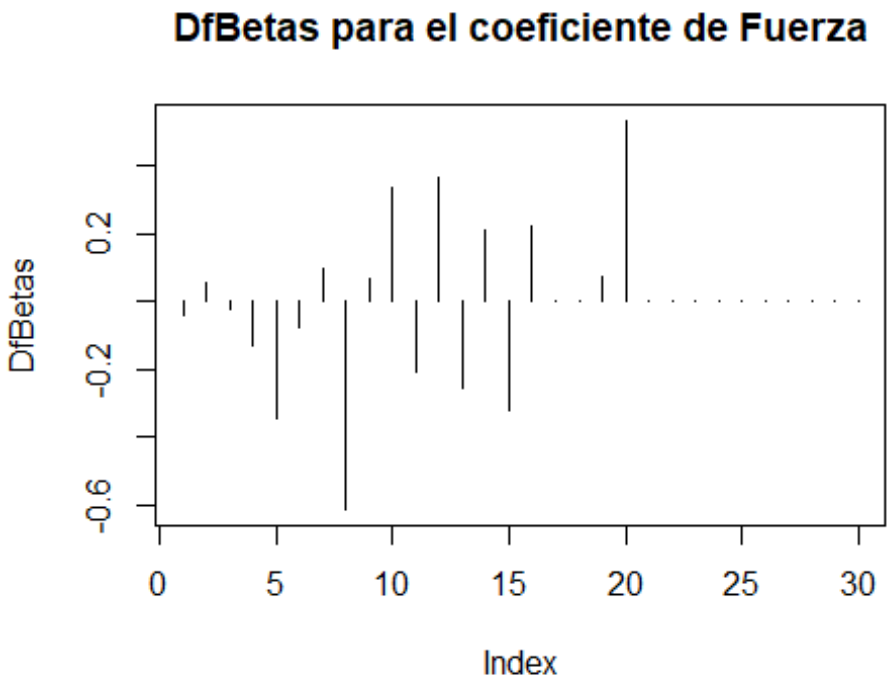
```r
# Calcular DfBetas
dfbetas_values = dfbetas(model)

# Graficar DfBetas para un coeficiente específico (por ejemplo, Fuerza)
plot(dfbetas_values[, 1], type="h", main="DfBetas para el coeficiente de
Fuerza", ylab="DfBetas")
abline(h = c(-1, 1), col="red")
```




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



```r
plot(dfbetas_values[, 3], type="h", main="DfBetas para el coeficiente de
Fuerza", ylab="DfBetas")
abline(h = c(-1, 1), col="red")
```




```r
# Identificar puntos influyentes en el coeficiente de Fuerza
puntos_influyentes = which(abs(dfbetas_values[, 2]) > 1)
data[puntos_influyentes, ]
```

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

Conclusion

Ausencia de Datos Atípicos o Influyentes La falta de datos atípicos o influyentes en este análisis sugiere que el conjunto de datos es relativamente homogéneo y bien comportado. Esto aumenta la confianza en la estabilidad del modelo, ya que no hay observaciones que distorsionen los resultados.

Significado de las Variables La Potencia y la Temperatura son variables que tienen un impacto significativo en la predicción de la Resistencia. Esto es coherente con expectativas prácticas, ya que ambas variables están directamente relacionadas con el comportamiento de los materiales en un proceso de corte. Por el contrario, Fuerza y Tiempo no parecen ser tan relevantes en este contexto particular.

Validez del Modelo La ausencia de autocorrelación (confirmada por la prueba de Durbin-Watson) y la normalidad de los residuos (verificada con el gráfico Q-Q) sugieren que el modelo cumple con los supuestos de regresión lineal múltiple. Esto refuerza la validez de los resultados obtenidos.

En resumen, el modelo ajustado es robusto y confiable, sin datos atípicos o influyentes que comprometan su interpretación. Las variables Potencia y Temperatura son las más importantes para predecir la Resistencia en este conjunto de datos, mientras que la Fuerza y el Tiempo no contribuyen de manera significativa.