

Actividad 14

Saúl Francisco Vázquez del Río

2024-09-17

```
M = read.csv("C:\\Users\\saulv\\OneDrive\\Escritorio\\Septimo
semestre\\AlCorte.csv")
Fuerza = M$Fuerza
Potencia = M$Potencia
Temperatura = M$Temperatura
Tiempo = M$Tiempo
Resistencia = M$Resistencia
head(M)
```

```
##   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
```

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

```
summary(M)
```

```
##      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
```

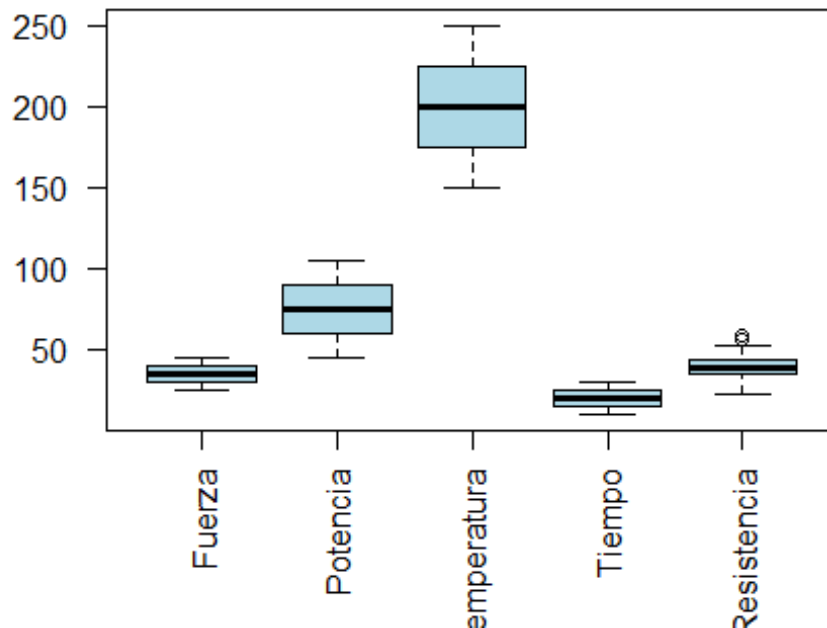
```
sapply(M, sd, na.rm = TRUE)
```

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

Crear los boxplots de todas las columnas numéricas

```
boxplot(M,
  main = "Boxplots de cada variable",
  col = "lightblue",
  las = 2) # las = 2 para rotar las etiquetas del eje X
```

Boxplots de cada variable



Encuentra el mejor modelo de regresión que explique la variable Resistencia. Analiza el modelo basándote en:

Significancia del modelo: Economía de las variables Significación global (Prueba para el modelo) Significación individual (Prueba para cada β_i) Variación explicada por el modelo

```
Modelo1 = lm(Resistencia~., data =M)
summary(Modelo1)
```

```
##
## Call:
## lm(formula = Resistencia ~ ., data = M)
##
## 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

Pasos = step(Modelo1, 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

summary(Pasos)

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

Modelo6 = lm(Resistencia ~ Potencia + Temperatura, data = M)
summary(Modelo6)

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

El mejor resultado

```
modelo_0=lm(Resistencia~1,data=M)
Pasos2 = step(modelo_0, scope = list(lower = modelo_0, upper=Modelo1),
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

summary(Pasos2)

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

Pasos3 = step(Modelo1, direccion = "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

summary(Pasos3)

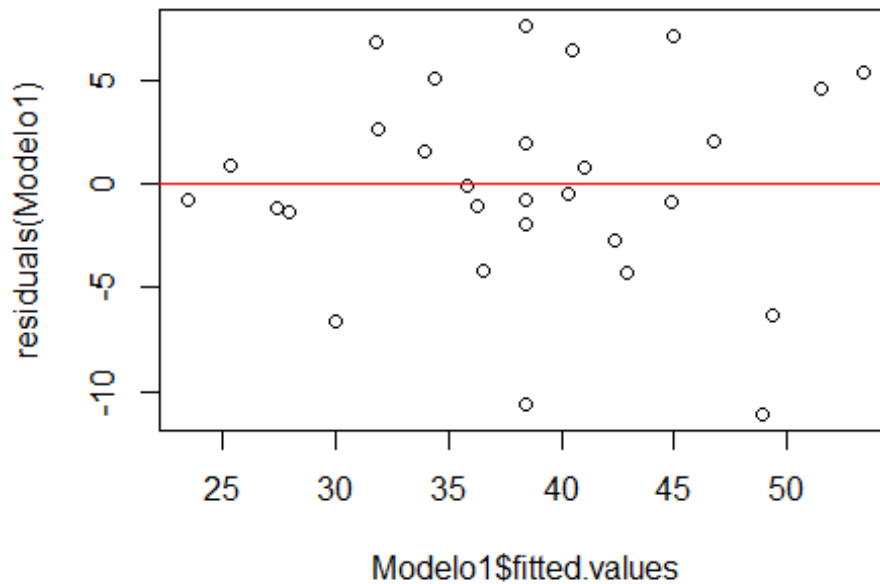
##
## Call:
## lm(formula = Resistencia ~ Potencia + Temperatura, data = M)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.3233  -2.8067  -0.8483   3.1892   9.4600
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -24.90167   10.07207  -2.472  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
```

Analiza la validez del modelo encontrado:

Análisis de residuos (homocedasticidad, independencia, etc) No multicolinealidad de X_i

```
n = length(Resistencia)
plot(Modelo1$fitted.values, residuals(Modelo1), main = "Residuos vs
Valores Ajustados")
abline(h = 0, col = "red")
```

Residuos vs Valores Ajustados



```
library(lmtest)

## Cargando paquete requerido: zoo

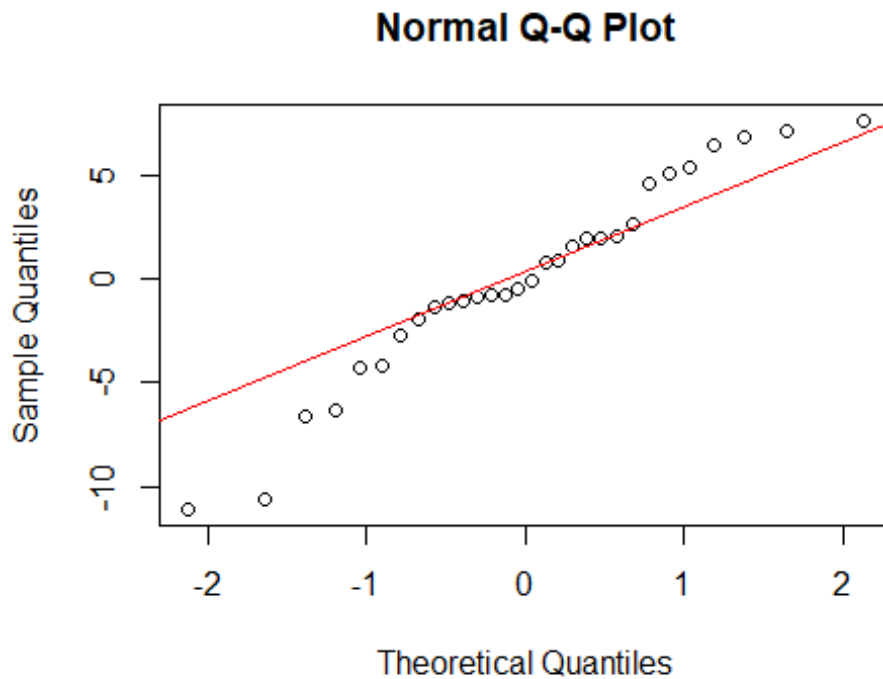
##
## Adjuntando el paquete: 'zoo'

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

bptest(Modelo1)

##
## studentized Breusch-Pagan test
##
## data:  Modelo1
## BP = 4.2293, df = 4, p-value = 0.3759

qqnorm(residuals(Modelo1))
qqline(residuals(Modelo1), col = "red")
```



```
dwtest(Modelo1)

##
## Durbin-Watson test
##
## data: Modelo1
## DW = 2.2611, p-value = 0.7917
## alternative hypothesis: true autocorrelation is greater than 0

library(car)

## Cargando paquete requerido: carData

vif(Modelo1)

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

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

Las variables de temperatura y potencia afectan a la resistencia, esto es correcto ya que un aumento de potencia en un objeto aumenta su resistencia haciendo que estas dos variables sean las que tengan más importancia, además que el modelo_0 sea el mejor ya que tiene la fórmula que la resistencia sea la afectada mediante la temperatura y potencia. Obteniendo los mejores resultados gracias a esta fórmula.

A3

```
library(ggplot2)

library(dplyr)

##
## Adjuntando el paquete: 'dplyr'

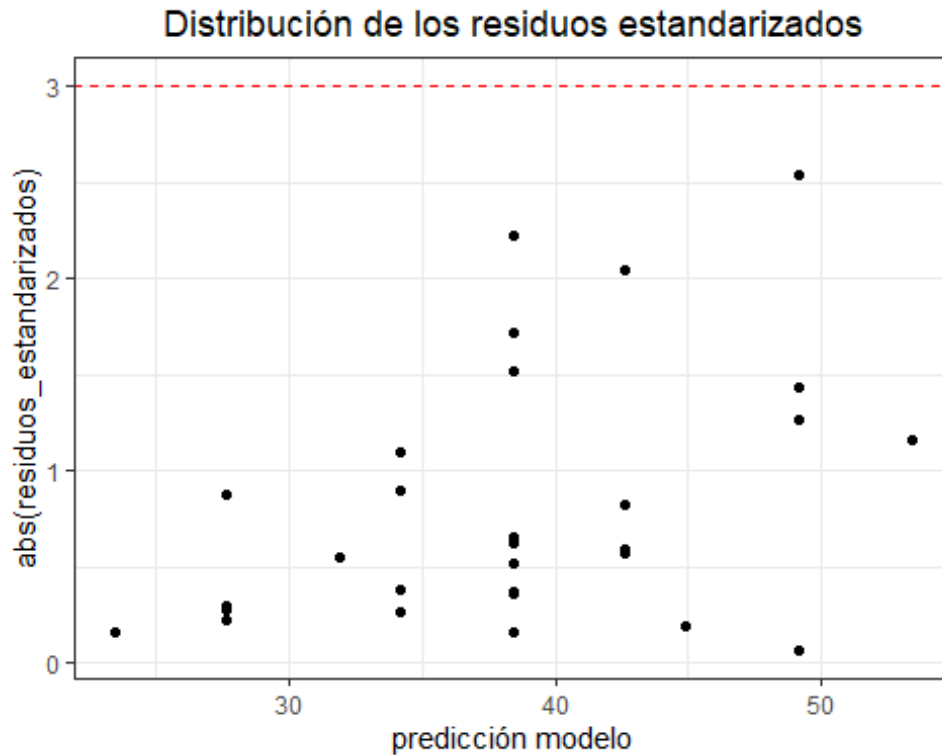
## The following object is masked from 'package:car':
##
##      recode

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

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

M$residuos_estandarizados <- rstudent(Pasos3)

#Introduce una columna en Datos con Los residuos estandarizados de Los n
datos
#Gráfico auxiliar:
ggplot(data = M, aes(x = predict(Pasos3), y =
abs(residuos_estandarizados))) +
geom_hline(yintercept = 3, color = "red", linetype = "dashed") +
# se identifican en rojo observaciones con residuos estandarizados
absolutos > 3
geom_point(aes(color = ifelse(abs(residuos_estandarizados) > 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))
```



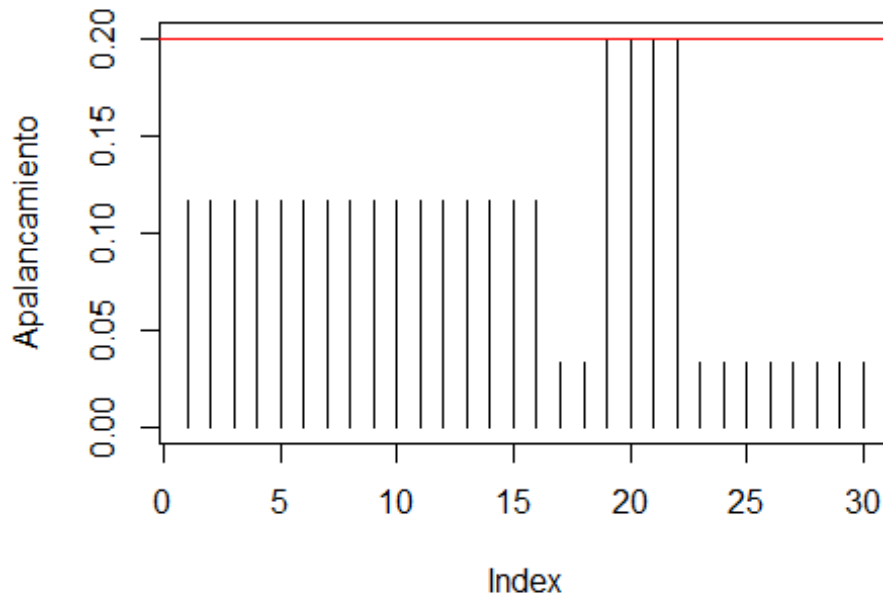
```
#Cuenta e identifica cuántos datos atípicos hay:
Atipicos = which(abs(M$residuos_estandarizados)>3)
#Muestra las observaciones con altos residuos estandarizados
M[Atipicos, ]

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

Distancia de leverage (diagonal de la matriz sombrero)

```
leverage = hatvalues(Pasos3)
#Calcula el leverage de los n datos
#Gráfico auxiliar:
plot(leverage, type="h", main="Valores de Apalancamiento",
      ylab="Apalancamiento", ylim=c(0, 2*mean(leverage)) )
abline(h = 2*mean(leverage), col="red") # Límite comúnmente usado
```

Valores de Apalancamiento



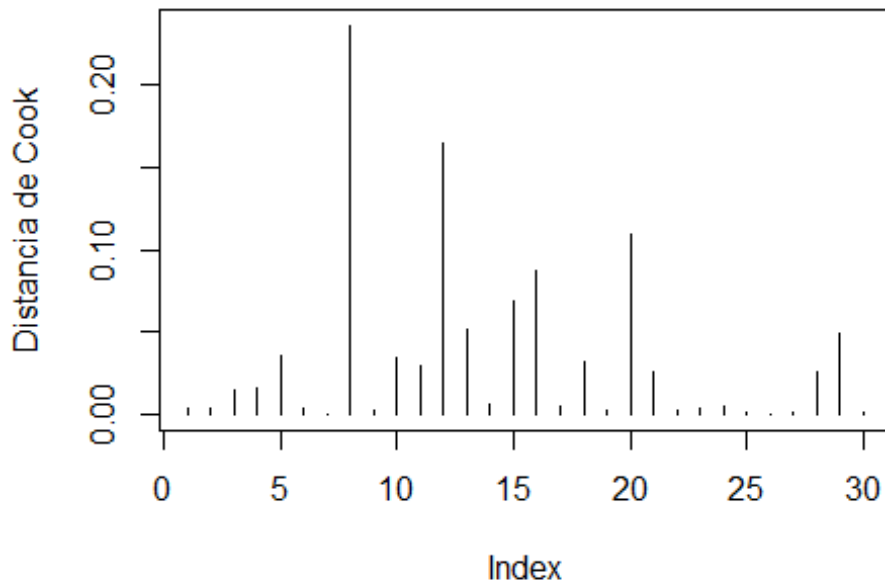
```
#Cuenta e identifica cuántos datos atípicos hay:
high_leverage_points = which(leverage > 2*mean(leverage))
#Muestra las observaciones con alto Leverage
M[high_leverage_points, ]

##      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
```

Distancia de Cook

```
cooks_d <- cooks.distance(Pasos3)
#Calcula la distancia de Cook de Los n datos
#Gráfico auxiliar:
plot(cooks_d, type="h", main="Distancia de Cook", ylab="Distancia de Cook")
abline(h = 1, col="red") # Límite comúnmente usado
```

Distancia de Cook



#Cuenta e identifica cuántos datos atípicos hay:

```
puntos_influyentes = which(cooksd > 1)
```

#Muestra las observaciones influyentes

```
M[puntos_influyentes, ]
```

```
## [1] Fuerza
```

```
Potencia
```

```
Temperatura
```

```
## [4] Tiempo
```

```
Resistencia
```

```
residuos_estandarizados
```

```
## <0 rows> (o 0- extensión row.names)
```

DfBetas

```
dfbetas_values = dfbetas(Pasos3)
```

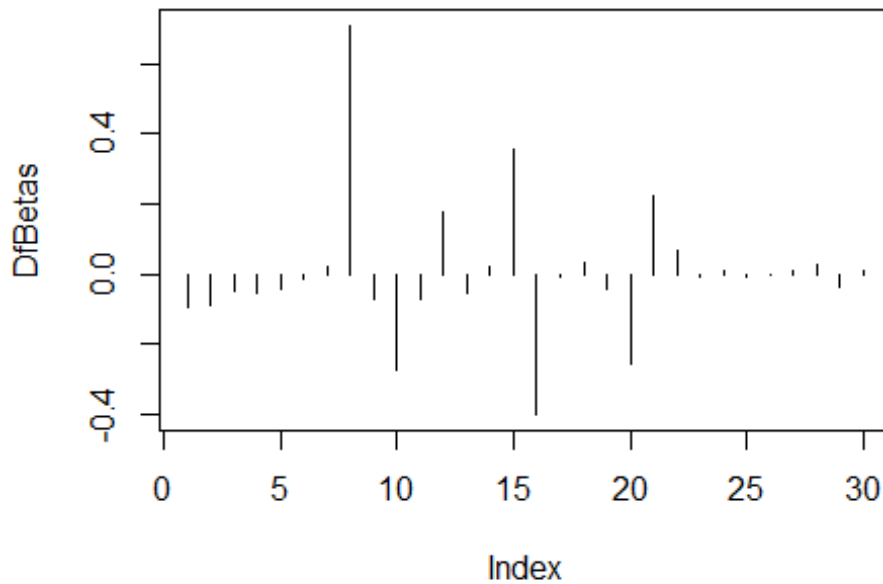
#Calcula la DfBeta de los n datos para cada β_j

#Gráfico auxiliar, para la variable 1:

```
plot(dfbetas_values[, 1], type="h", main="DfBetas para el coeficiente 1",  
      ylab="DfBetas" )
```

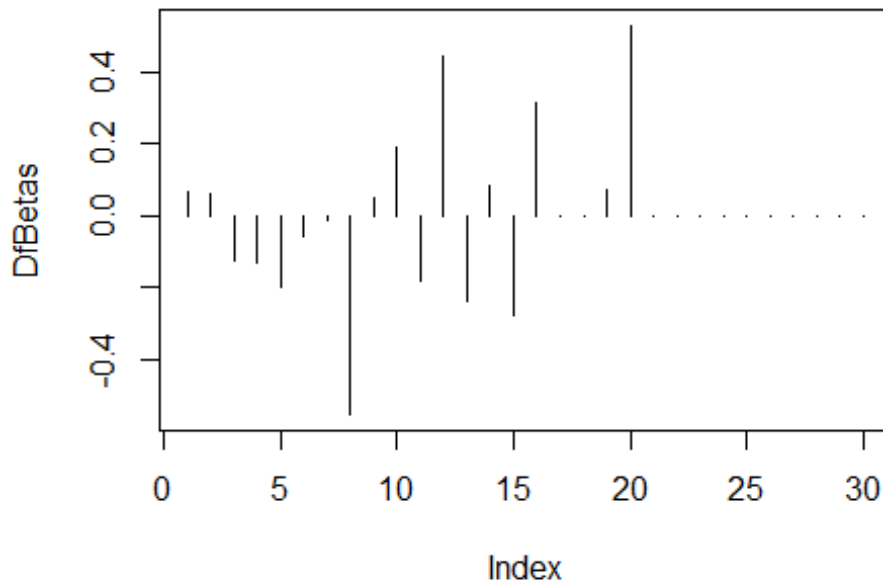
```
abline(h = c(-1, 1), col="red") # Límites comunes
```

DfBetas para el coeficiente 1



```
#Cuenta e identifica cuántos datos atípicos hay:  
puntos_influyentes = which(abs(dfbetas_values[, 1]) > 1)  
  
#Gráfico auxiliar, para la variable 2:  
plot(dfbetas_values[, 2], type="h", main="DfBetas para el coeficiente 2"  
 , ylab="DfBetas" )  
abline(h = c(-1, 1), col="red") # Límites comunes
```

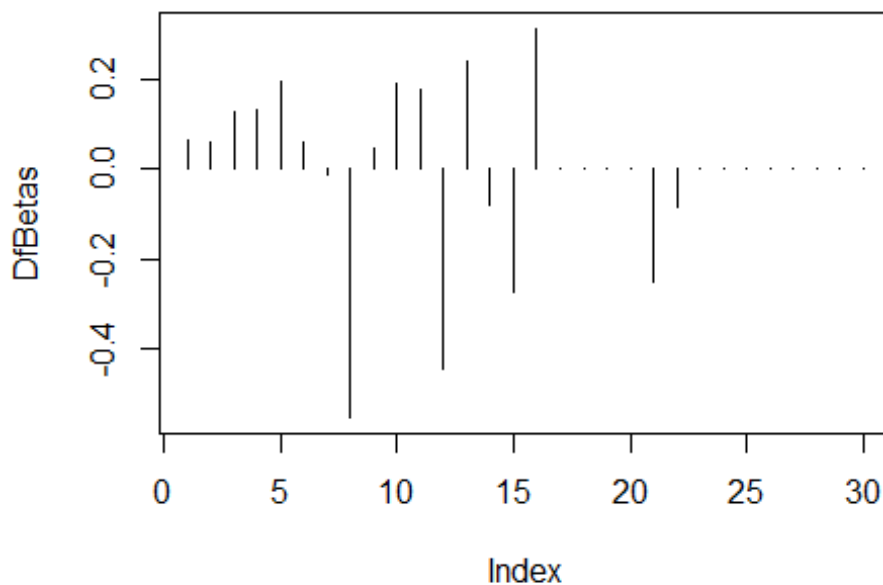
DfBetas para el coeficiente 2



```
#Cuenta e identifica cuántos datos atípicos hay:
puntos_influyentes = which(abs(dfbetas_values[, 2]) > 1)

#Gráfico auxiliar, para la variable 3:
plot(dfbetas_values[, 3], type="h", main="DfBetas para el coeficiente 3",
     , ylab="DfBetas" )
abline(h = c(-1, 1), col="red") # Límites comunes
```

DfBetas para el coeficiente 3



```
#Cuenta e identifica cuántos datos atípicos hay:  
puntos_influyentes = which(abs(dfbetas_values[, 3]) > 1)
```

Influence.measures

```
influencia = influence.measures(Pasos3)
```

```
#Calcula las medidas de los n datos
```

```
#Resumen de datos influyentes:
```

```
summary(influencia)
```

```
## Potentially influential observations of
```

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

```
##
```

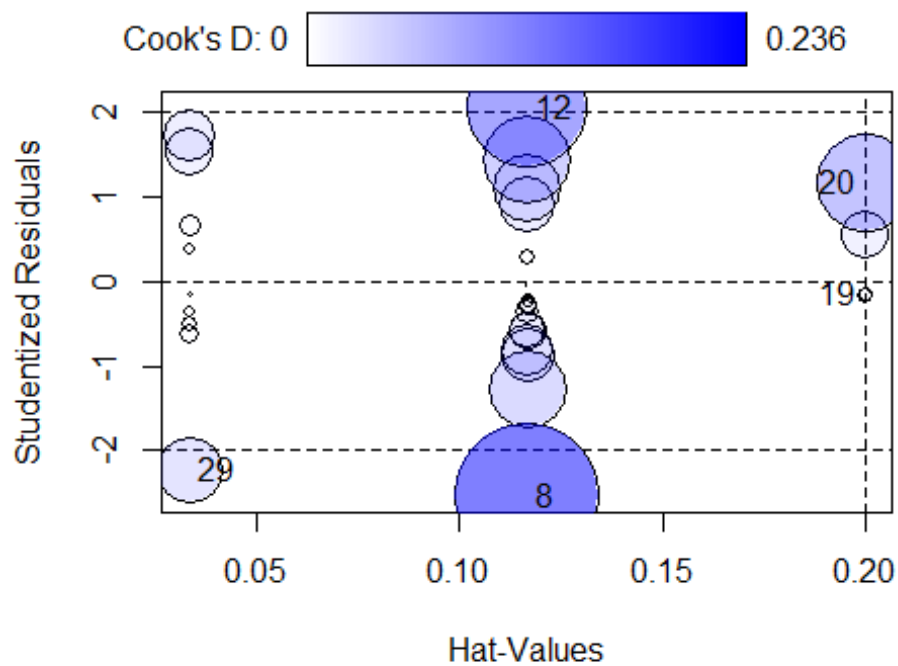
##	dfb.1_	dfb.Ptnc	dfb.Tmpr	dffit	cov.r	cook.d	hat
## 8	0.71	-0.55	-0.55	-0.92	0.65_*	0.24	0.12
## 19	-0.04	0.07	0.00	-0.08	1.40_*	0.00	0.20
## 21	0.22	0.00	-0.25	0.27	1.35_*	0.03	0.20
## 22	0.07	0.00	-0.09	-0.09	1.39_*	0.00	0.20

```
# Detecta los datos con posible influencia
```

influencePlot

```
library(car)
```

```
influencePlot(Pasos3)
```

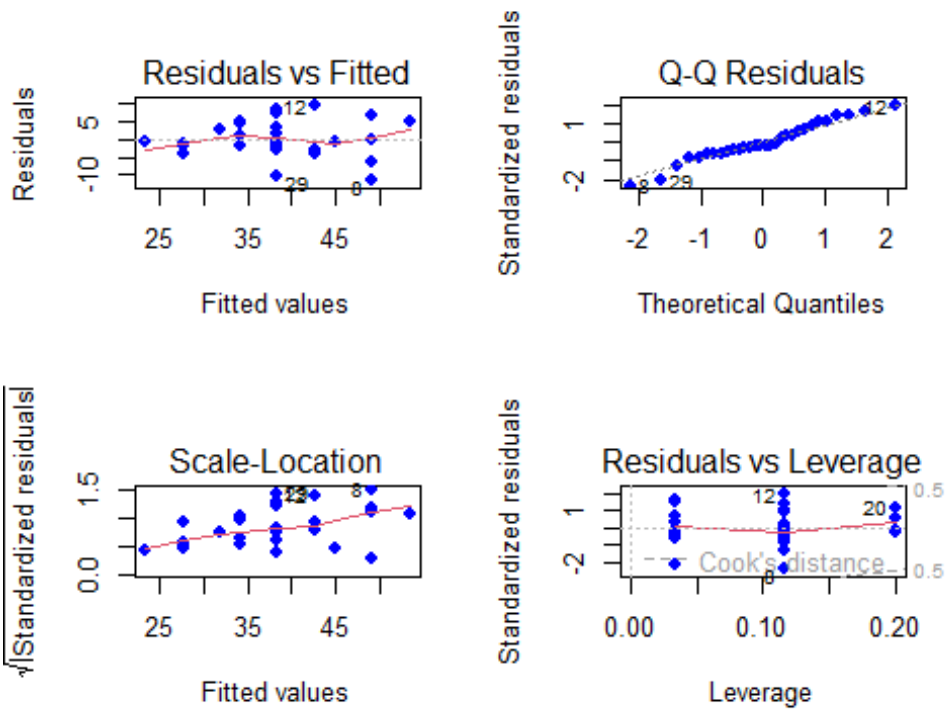


```
##      StudRes      Hat      CookD
## 8  -2.535832 0.1166667 0.235696235
## 12  2.043589 0.1166667 0.164507739
## 19 -0.159511 0.2000000 0.002199712
## 20  1.154355 0.2000000 0.109693544
## 29 -2.216952 0.0333333 0.049338917
```

```
# grafica los residuos con estandarización
#extrema, el leverage y la distancia de cook
#Muestra las observaciones influyentes
```

Plot del modelo

```
par(mfrow=c(2, 2))
plot(Pasos3, col="blue", pch=19)
```

Conclusion

Mediante las graficas mostradas se puede observar que el modelo se adapta correctamente pero en cuanto a la grafica de residuos esta se podria mejorar, ya que no se aprecia correctamente la relacion de las variables y se podria hacer que se capturen mejor las variables pra tener una grafica más completa, pero como se planteo estos mismo podrian ser datos atipicos.