

A3_A01742161

Rogelio Lizárraga

2024-09-24

1. Análisis descriptivo de los datos: medidas principales y gráficos

```
data = read.csv('AlCorte.csv')
library(car)
```

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

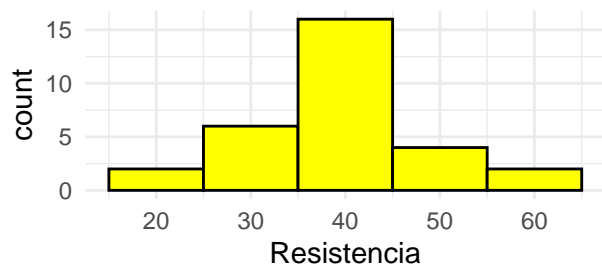
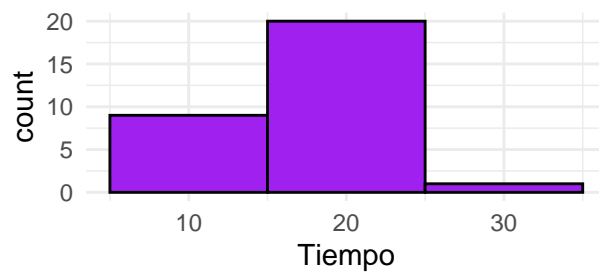
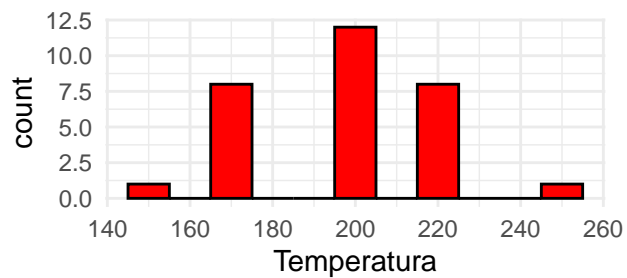
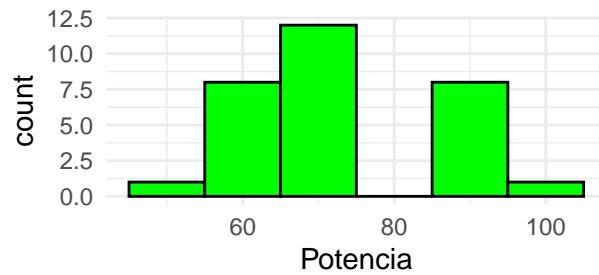
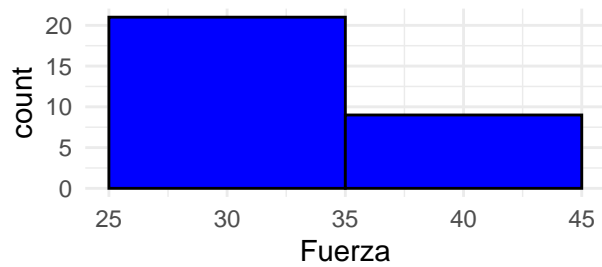
```
library(ggplot2)
library(gridExtra)
head(data)
```

```
##   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
```

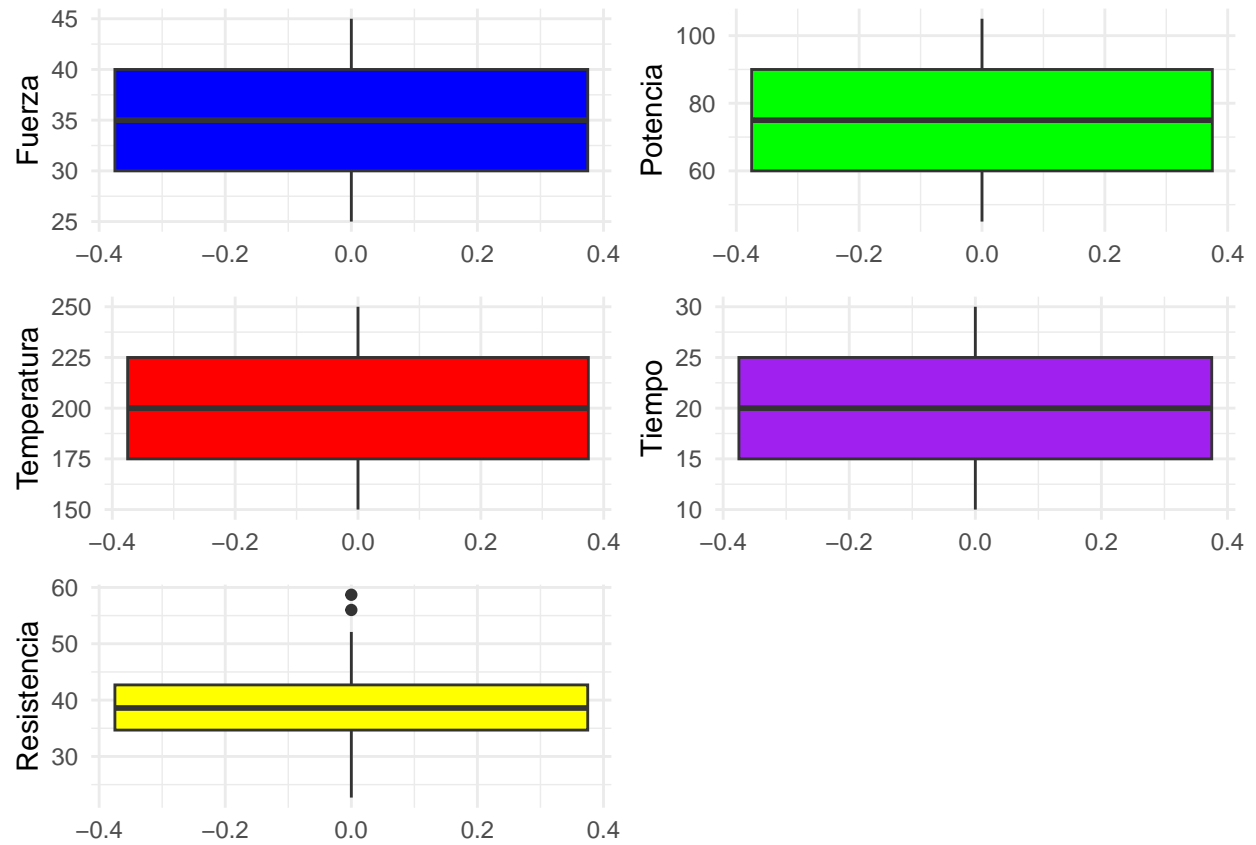
```
summary(data)
```

```
##      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
```

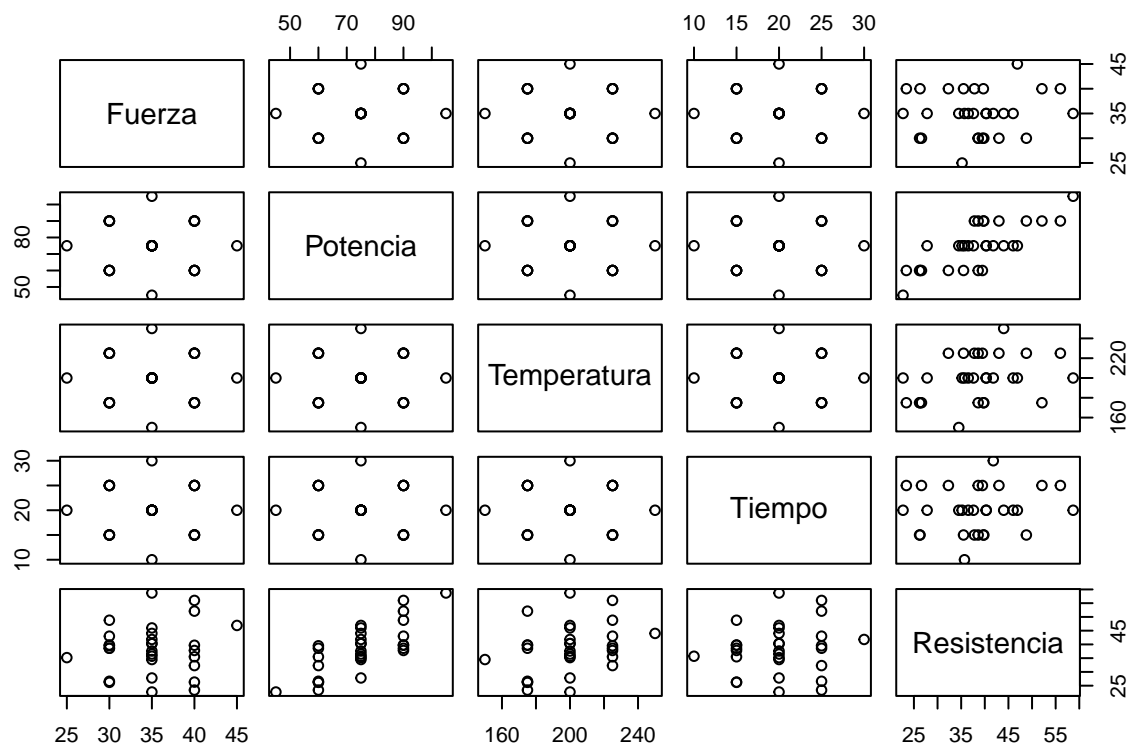
```
grid.arrange(
  ggplot(data, aes(x = Fuerza)) + geom_histogram(binwidth = 10, fill = "blue", color = "black") + theme_minimal(),
  ggplot(data, aes(x = Potencia)) + geom_histogram(binwidth = 10, fill = "green", color = "black") + theme_minimal(),
  ggplot(data, aes(x = Temperatura)) + geom_histogram(binwidth = 10, fill = "red", color = "black") + theme_minimal(),
  ggplot(data, aes(x = Tiempo)) + geom_histogram(binwidth = 10, fill = "purple", color = "black") + theme_minimal(),
  ggplot(data, aes(x = Resistencia)) + geom_histogram(binwidth = 10, fill = "yellow", color = "black") + theme_minimal(),
  ncol = 2
)
```



```
grid.arrange(
  ggplot(data, aes(y = Fuerza)) + geom_boxplot(fill = "blue") + theme_minimal(),
  ggplot(data, aes(y = Potencia)) + geom_boxplot(fill = "green") + theme_minimal(),
  ggplot(data, aes(y = Temperatura)) + geom_boxplot(fill = "red") + theme_minimal(),
  ggplot(data, aes(y = Tiempo)) + geom_boxplot(fill = "purple") + theme_minimal(),
  ggplot(data, aes(y = Resistencia)) + geom_boxplot(fill = "yellow") + theme_minimal(),
  ncol = 2
)
```



```
par(mfrow = c(1, 1))  
pairs(data)
```



```
library(e1071)
cat('sesgo fuerza:', skewness(data$Fuerza))

## sesgo fuerza: 0

cat('\nsesgo potencia:', skewness(data$Potencia))

##
## sesgo potencia: 0

cat('\nsesgo temperatura:', skewness(data$Temperatura))

##
## sesgo temperatura: 0

cat('\nsesgo tiempo:', skewness(data$Tiempo))

##
## sesgo tiempo: 0

cat('\nsesgo resistencia:', skewness(data$Resistencia))
```

```
##
## sesgo resistencia: 0.2235544
```

```
cat('\ncurtosis fuerza:', kurtosis(data$Fuerza))
```

```
##
## curtosis fuerza: -0.6638889
```

```
cat ('\ncurtosis potencia:', kurtosis(data$Potencia))
```

```
##
## curtosis potencia: -0.6638889
```

```
cat ('\ncurtosis temperatura:', kurtosis(data$Temperatura))
```

```
##
## curtosis temperatura: -0.6638889
```

```
cat ('\ncurtosis tiempo:', kurtosis(data$Tiempo))
```

```
##
## curtosis tiempo: -0.6638889
```

```
cat ('\ncurtosis resistencia:', kurtosis(data$Resistencia), '\n\n')
```

```
##
## curtosis resistencia: -0.3476408
```

```
cor(data, use = 'complete.obs')
```

```
##          Fuerza  Potencia Temperatura   Tiempo Resistencia
## Fuerza      1.000000 0.000000  0.000000 0.000000  0.1075208
## Potencia    0.000000 1.000000  0.000000 0.000000  0.7594185
## Temperatura 0.000000 0.000000  1.000000 0.000000  0.3293353
## Tiempo      0.000000 0.000000  0.000000 1.000000  0.1312262
## Resistencia 0.1075208 0.7594185  0.3293353 0.1312262  1.0000000
```

Como podemos observar, la fuerza está normalmente distribuida, sin sesgo y con una dispersión de datos simétrica. La potencia tiene una distribución simétrica con pocas observaciones extremas, indicando un comportamiento que es razonablemente cercano a lo normal. La temperatura sigue un comportamiento similar, con simetría y menos valores extremos, lo que indica que se comporta de manera bastante estable. El tiempo tiene un comportamiento simétrico y no está influido por valores extremos.

Por otro lado, la resistencia tiene una ligera inclinación hacia la derecha, pero su sesgo no es significativo. Los valores extremos son menos frecuentes, pero no tanto como en las otras variables.

Observando los gráficos, la fuerza tiene una dispersión moderada y sin outliers significativos. La potencia tiene una simetría sin valores atípicos importantes. La temperatura tiene una alta dispersión, pero sin outliers. El tiempo no cuenta con outliers. Finalmente, La resistencia presenta un sesgo ligera a la derecha, con algunos outliers, con un sesgo no significativo.

Finalmente, observando las gráficas de dispersión concluimos que no parece haber una relación clara entre las variables contra resistencia, excepto en potencia, pues es visible la relación directa entre las variables. Esto también lo podemos observar en la matriz de correlación, pues la mayor correlación está entre la potencia y la resistencia, con un 0.759, lo cual es bastante alto, y una correlación moderada del 0.3293 para la temperatura. Para fuerza y tiempo es menor a 0.14, por lo que parece que ser que estas variables no serán útiles para el modelo

```
length(data$Fuerza)
```

```
## [1] 30
```

2. Modelo de regresión lineal múltiple

$H_0 : \beta_1$ No es significativo $H_1 : \beta_1$ Es significativo

```
modelo_sin_vars = lm(Resistencia ~ 1, data = data)
modelo <- lm(Resistencia ~ Fuerza + Potencia + Temperatura + Tiempo, data=data)
summary(modelo)
```

```
##
## 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
```

```
vif(modelo)
```

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

Como podemos observar el valor p del modelo global < 0.05 , por lo que la hipótesis inicial se rechaza y el modelo es estadísticamente significativo. Las β_{α_i} que sí rechazan (valor $p < 0.05$) y por lo tanto tienen significancia son: β_{α_0} (intercepto), β_{α_2} (potencia) y β_{α_3} (temperatura) Las β_{α_i} que no rechazan (valor $p > 0.05$) y por lo tanto NO tienen significancia son: β_{α_1} (fuerza), β_{α_4} (tiempo)

Tenemos un coeficiente de determinación de R^2 de 0.6682, por lo que el modelo explica el 66.82% de la variabilidad de la resistencia.

De igual manera el VIF para cada variable es = 1, por lo que no hay multicolinealidad entre las variables.

Ahora, escogeremos el modelo basándonos en el criterio de información de Akaike (AIC)

3 analisis modelo

Forward

```
modelo_forward_aic <- step(modelo_sin_vars, scope = list(lower = modelo_sin_vars, upper = modelo), direc
```

```
## 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(modelo_forward_aic)
```

```
##
## Call:
## lm(formula = Resistencia ~ Potencia + Temperatura, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.3233  -2.8067  -0.8483   3.1892   9.4600
##
## Coefficients:
```

```
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept) -24.90167   10.07207  -2.472  0.02001 *
## Potencia     0.49833    0.07086   7.033 1.47e-07 ***
## Temperatura  0.12967    0.04251   3.050 0.00508 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.207 on 27 degrees of freedom
## Multiple R-squared:  0.6852, Adjusted R-squared:  0.6619
## F-statistic: 29.38 on 2 and 27 DF,  p-value: 1.674e-07

modelo_forward_bic <- step(modelo_sin_vars, scope = list(lower = modelo_sin_vars, upper = modelo), direc

## Start:  AIC=133.91
## Resistencia ~ 1
##
##           Df Sum of Sq    RSS    AIC
## + Potencia     1   1341.01  984.24 111.52
## + Temperatura  1    252.20 2073.06 133.87
## <none>                        2325.26 133.91
## + Tiempo        1     40.04 2285.22 136.79
## + Fuerza        1     26.88 2298.38 136.97
##
## Step:  AIC=111.52
## Resistencia ~ Potencia
##
##           Df Sum of Sq    RSS    AIC
## + Temperatura  1    252.202 732.04 106.04
## <none>                        984.24 111.52
## + Tiempo        1     40.042 944.20 113.68
## + Fuerza        1     26.882 957.36 114.09
##
## Step:  AIC=106.04
## Resistencia ~ Potencia + Temperatura
##
##           Df Sum of Sq    RSS    AIC
## <none>                        732.04 106.04
## + Tiempo  1     40.042 692.00 107.76
## + Fuerza  1     26.882 705.16 108.32

summary(modelo_forward_bic)

##
## Call:
## lm(formula = Resistencia ~ Potencia + Temperatura, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.3233  -2.8067  -0.8483   3.1892   9.4600
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept) -24.90167   10.07207  -2.472  0.02001 *
```



```
## Potencia      0.49833    0.07086    7.033 1.47e-07 ***
## Temperatura  0.12967    0.04251    3.050 0.00508 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.207 on 27 degrees of freedom
## Multiple R-squared:  0.6852, Adjusted R-squared:  0.6619
## F-statistic: 29.38 on 2 and 27 DF,  p-value: 1.674e-07
```

Como podemos observar el valor p del nuevo modelo global < 0.05 , por lo que la hipótesis inicial se rechaza y el modelo es estadísticamente significativo. OBServamos que las β_{α_i} sí echazan (valor $p < 0.05$) la hipótesis inicial y por lo tanto tienen significancia para el modelo creado, que contiene las variables de potencia (β_{α_1}) y temperatura (β_{α_2}) para predecir la resistencia.

Tenemos un coeficiente de determinación de R^2 de 0.6619, por lo que el modelo explica el 66.19% de la variabilidad de la resistencia.

Finalmente, tenemos un AIC de 101.84, por lo que el modelo tiene un buen ajuste, sin ser excesivamente complejo. BIC penaliza más fuertemente los modelos con más parámetros, por lo que el BIC de 106.04 es bajo, balanceando la complejidad y el ajuste, penalizando más la complejidad en el modelo que AIC.

Backward

```
modelo_mixto_aic <- step(modelo, 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
```

```
summary(modelo_mixto_aic)
```

```
##
## Call:
## lm(formula = Resistencia ~ Potencia + Temperatura, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.3233  -2.8067  -0.8483   3.1892   9.4600
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -24.90167    10.07207  -2.472  0.02001 *
## Potencia      0.49833     0.07086   7.033 1.47e-07 ***
## Temperatura   0.12967     0.04251   3.050 0.00508 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.207 on 27 degrees of freedom
## Multiple R-squared:  0.6852, Adjusted R-squared:  0.6619
## F-statistic: 29.38 on 2 and 27 DF,  p-value: 1.674e-07
```

```
modelo_mixto_bic <- step(modelo, direction = "backward", k = log(length(data$Fuerza)))
```

```
## Start:  AIC=109.97
## Resistencia ~ Fuerza + Potencia + Temperatura + Tiempo
##
##              Df Sum of Sq    RSS    AIC
## - Fuerza      1     26.88  692.00 107.76
## - Tiempo      1     40.04  705.16 108.32
## <none>                    665.12 109.97
## - Temperatura 1     252.20  917.32 116.21
## - Potencia    1    1341.01 2006.13 139.69
##
## Step:  AIC=107.76
## Resistencia ~ Potencia + Temperatura + Tiempo
##
##              Df Sum of Sq    RSS    AIC
## - Tiempo      1     40.04  732.04 106.04
## <none>                    692.00 107.76
## - Temperatura 1     252.20  944.20 113.68
## - Potencia    1    1341.02 2033.02 136.69
##
## Step:  AIC=106.04
## Resistencia ~ Potencia + Temperatura
##
##              Df Sum of Sq    RSS    AIC
## <none>                    732.04 106.04
## - Temperatura 1     252.2   984.24 111.52
## - Potencia    1    1341.0 2073.06 133.87
```

```
summary(modelo_mixto_bic)
```

```
##
## Call:
## lm(formula = Resistencia ~ Potencia + Temperatura, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.3233  -2.8067  -0.8483   3.1892   9.4600
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -24.90167    10.07207  -2.472  0.02001 *
## Potencia      0.49833     0.07086   7.033 1.47e-07 ***
## Temperatura   0.12967     0.04251   3.050 0.00508 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.207 on 27 degrees of freedom
## Multiple R-squared:  0.6852, Adjusted R-squared:  0.6619
## F-statistic: 29.38 on 2 and 27 DF,  p-value: 1.674e-07
```

Como podemos observar el valor p del nuevo modelo global < 0.05 , por lo que la hipótesis inicial se rechaza y el modelo es estadísticamente significativo. OBServamos que las β_{α_i} sí echazan (valor $p < 0.05$) la hipótesis inicial y por lo tanto tienen significancia para el modelo creado, que contiene las variables de potencia (β_{α_1}) y temperatura (β_{α_2}) para predecir la resistencia.

Tenemos un coeficiente de determinación de R^2 de 0.6619, por lo que el modelo explica el 66.19% de la variabilidad de la resistencia.

Finalmente, tenemos un AIC de 101.84, por lo que el modelo tiene un buen ajuste, sin ser excesivamente complejo. BIC penaliza más fuertemente los modelos con más parámetros, por lo que el BIC de 106.04 es bajo, balanceando la complejidad y el ajuste, penalizando más la complejidad en el modelo que AIC.

Mixto

```
modelo_mixto_aic <- step(modelo, direction = "both")
```

```
## 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(modelo_mixto_aic)
```

```
##
## Call:
## lm(formula = Resistencia ~ Potencia + Temperatura, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.3233  -2.8067  -0.8483   3.1892   9.4600
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -24.90167    10.07207  -2.472  0.02001 *
## Potencia      0.49833     0.07086   7.033 1.47e-07 ***
## Temperatura   0.12967     0.04251   3.050  0.00508 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.207 on 27 degrees of freedom
## Multiple R-squared:  0.6852, Adjusted R-squared:  0.6619
## F-statistic: 29.38 on 2 and 27 DF,  p-value: 1.674e-07
```

```
modelo_mixto_bic <- step(modelo, direction = "both", k = log(length(data$Fuerza)))
```

```
## Start:  AIC=109.97
## Resistencia ~ Fuerza + Potencia + Temperatura + Tiempo
##
##           Df Sum of Sq      RSS      AIC
## - Fuerza      1      26.88   692.00  107.76
## - Tiempo      1      40.04   705.16  108.32
## <none>                665.12  109.97
## - Temperatura  1     252.20   917.32  116.21
## - Potencia     1    1341.01  2006.13  139.69
##
## Step:  AIC=107.76
## Resistencia ~ Potencia + Temperatura + Tiempo
```

```
##
##           Df Sum of Sq      RSS      AIC
## - Tiempo      1      40.04   732.04 106.04
## <none>                692.00 107.76
## + Fuerza       1      26.88   665.12 109.97
## - Temperatura  1     252.20   944.20 113.68
## - Potencia     1    1341.02  2033.02 136.69
##
## Step: AIC=106.04
## Resistencia ~ Potencia + Temperatura
##
##           Df Sum of Sq      RSS      AIC
## <none>                732.04 106.04
## + Tiempo      1      40.04   692.00 107.76
## + Fuerza       1      26.88   705.16 108.32
## - Temperatura  1     252.20   984.24 111.52
## - Potencia     1    1341.01  2073.06 133.87
```

```
summary(modelo_mixto_bic)
```

```
##
## Call:
## lm(formula = Resistencia ~ Potencia + Temperatura, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.3233  -2.8067  -0.8483   3.1892   9.4600
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -24.90167    10.07207  -2.472  0.02001 *
## Potencia      0.49833     0.07086   7.033 1.47e-07 ***
## Temperatura   0.12967     0.04251   3.050  0.00508 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.207 on 27 degrees of freedom
## Multiple R-squared:  0.6852, Adjusted R-squared:  0.6619
## F-statistic: 29.38 on 2 and 27 DF,  p-value: 1.674e-07
```

Como podemos observar el valor p del nuevo modelo global < 0.05 , por lo que la hipótesis inicial se rechaza y el modelo es estadísticamente significativo. Observamos que las β_{i1} sí echazan (valor $p < 0.05$) la hipótesis inicial y por lo tanto tienen significancia para el modelo creado, que contiene las variables de potencia (β_{11}) y temperatura (β_{21}) para predecir la resistencia.

Tenemos un coeficiente de determinación de R^2 de 0.6619, por lo que el modelo explica el 66.19% de la variabilidad de la resistencia.

Finalmente, tenemos un AIC de 101.84, por lo que el modelo tiene un buen ajuste, sin ser excesivamente complejo. BIC penaliza más fuertemente los modelos con más parámetros, por lo que el BIC de 106.04 es bajo, balanceando la complejidad y el ajuste, penalizando más la complejidad en el modelo que AIC. Este modelo minimiza AIC y BIC, por lo que utilizaremos el modelo de Potencia y Temperatura.

3. Conclusión modelo utilizado

Como todas las direcciones utilizadas llegaron al mismo modelo, en nuestro caso no importó la dirección utilizada. Para nuestro caso, utilizaremos el modelo que utiliza las variables de Potencia y Temperatura predecir la Resistencia, al obtener el AIC y BIC más bajo.

```
modelo = modelo_mixto_aic
```

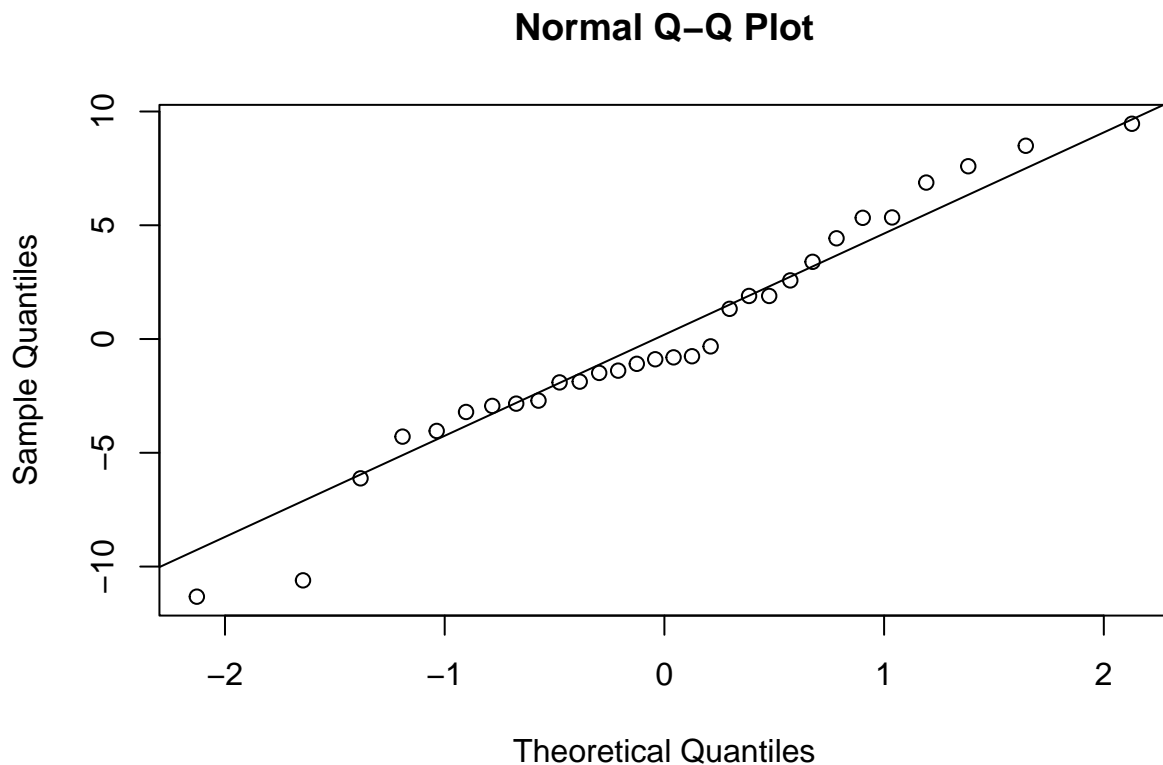
Normalidad

H_0 : Los residuos siguen una distribución normal. H_1 : Los residuos NO siguen una distribución normal.

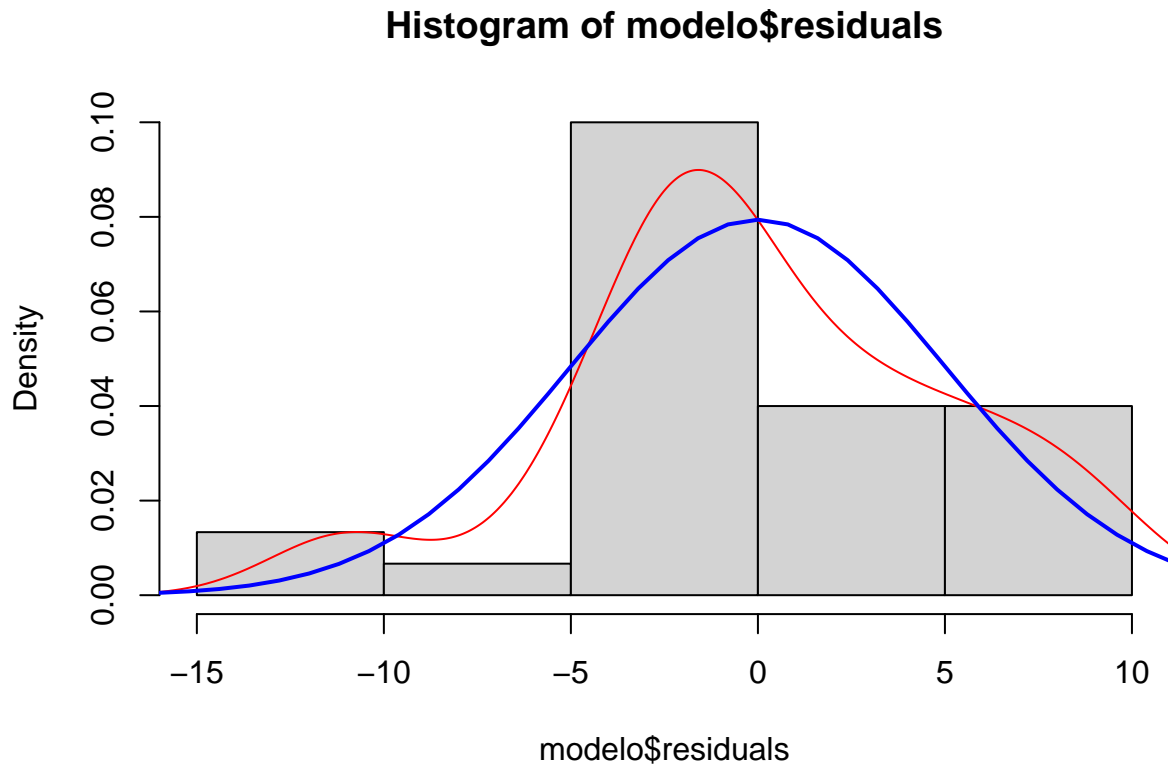
```
library(nortest)
shapiro.test(residuals(modelo))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(modelo)
## W = 0.96588, p-value = 0.4333
```

```
qqnorm(modelo$residuals)
qqline(modelo$residuals)
```



```
hist(modelo$residuals,freq=FALSE)
lines(density(modelo$residuals),col="red")
curve(dnorm(x,mean=mean(modelo$residuals),sd=sd(modelo$residuals)), from=-
40, to=40, add=TRUE, col="blue",lwd=2)
```



Como podemos observar, el valor $p > 0.05$, por lo que H_0 NO se rechaza y los residuos siguen una distribución normal.

Verificación de media cero

$H_0: \mu_e = 0$ $H_1: \mu_e \neq 0$

```
t.test(modelo$residuals)
```

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

Como tenemos un valor $p \approx 1$, H_0 no se rechaza, por lo que los residuos tienen media cero.

Homocedasticidad

H_0 : La varianza de los errores es constante (Hay homocedasticidad). H_1 : La varianza de los errores NO es constante (Hay heterocedasticidad).

```
library(lmtest)
```

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

```
##
```

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

```
## The following objects are masked from 'package:base':
```

```
##
```

```
##      as.Date, as.Date.numeric
```

```
bptest(modelo)
```

```
##
```

```
## studentized Breusch-Pagan test
```

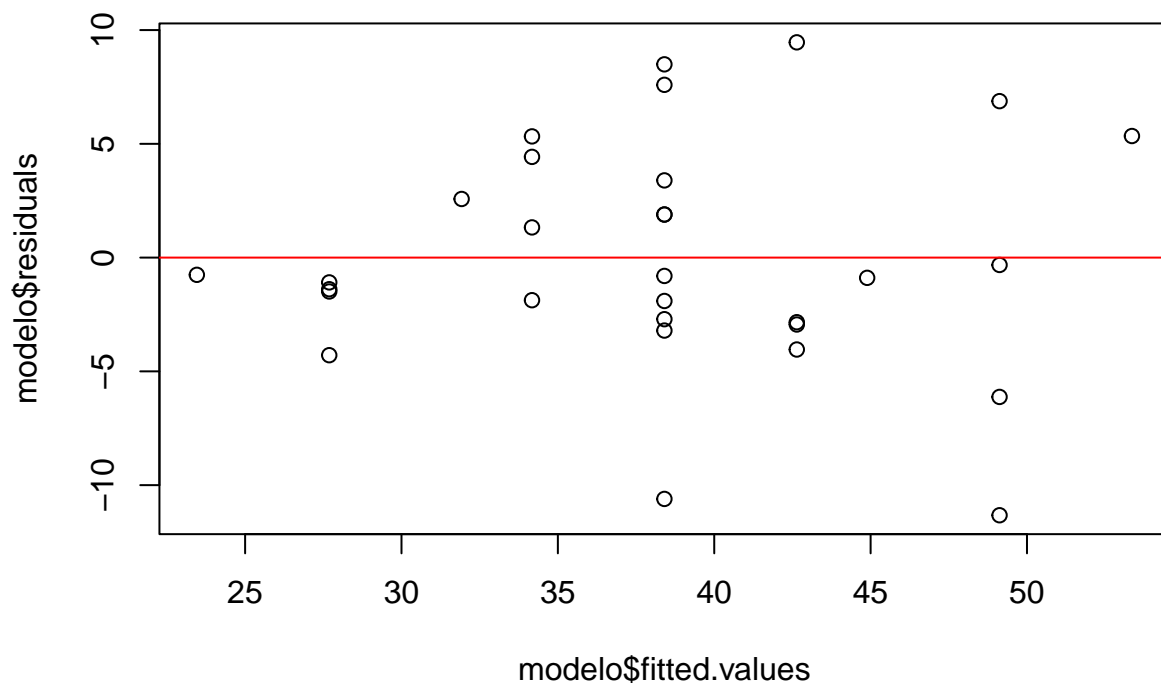
```
##
```

```
## data:  modelo
```

```
## BP = 4.0043, df = 2, p-value = 0.135
```

```
plot(modelo$fitted.values,modelo$residuals)
```

```
abline(h=0, col= 'red')
```

Como el valor $p > 0.05$, No rechaza H_0 , por lo que la varianza de los errores es constante (hay homocedasticidad). Además, esto se puede observar en el gráfico, pues la varianza fluctúa dentro de un rango de variabilidad sin importar el punto del gráfico.

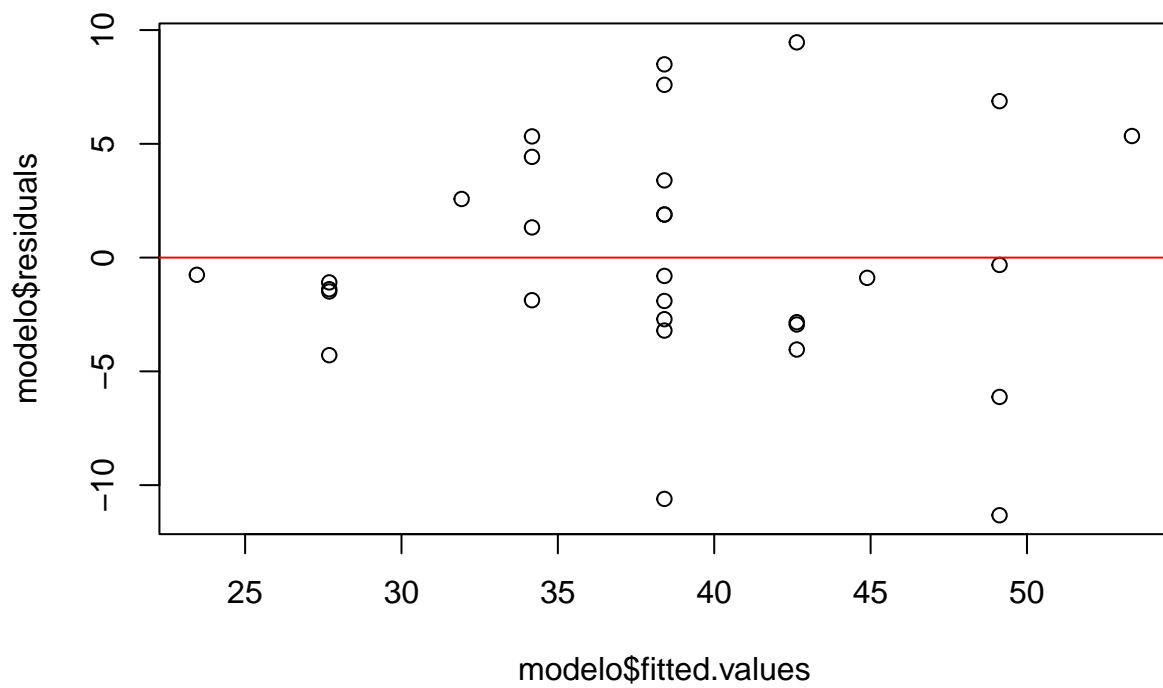
Independencia

H_0 : La autocorrelación de los residuos es 0 (hay independencia). H_1 : La autocorrelación de los residuos $\neq 0$ (no hay independencia).

```
dwtest(modelo)
```

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

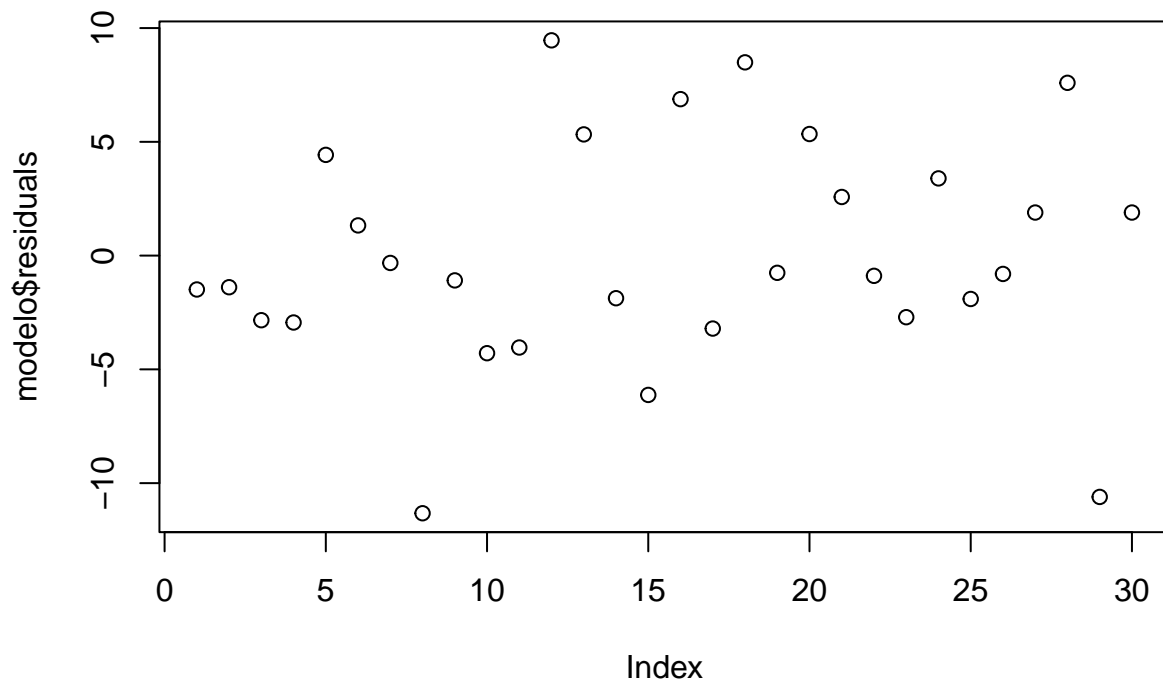
```
plot(modelo$fitted.values,modelo$residuals)
abline(h=0, col= 'red')
```



Como el valor $p > 0.05$, no se rechaza H_0 , por lo que sí hay independencia en los residuos. Además, no se observa un patrón en la gráfica, o una dependencia entre los residuos.

Linealidad

```
plot(modelo$residuals)
```



Observamos que los residuos se distribuyen aleatoriamente en torno a cero, por lo que la relación lineal es adecuada.

Multicolinealidad

```
library(car)
vif(modelo)
```

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

Como todas las variables tienen un VIF de 1, potencia y temperatura son independientes entre sí. Es decir, no hay redundancia o relación lineal entre las variables predictoras, por lo que no hay multicolinealidad.

Conclusión modelo

```
summary(modelo)
```

```
##
## Call:
```

```
## lm(formula = Resistencia ~ Potencia + Temperatura, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.3233  -2.8067  -0.8483   3.1892   9.4600
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -24.90167    10.07207  -2.472  0.02001 *
## Potencia      0.49833     0.07086   7.033 1.47e-07 ***
## Temperatura   0.12967     0.04251   3.050  0.00508 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.207 on 27 degrees of freedom
## Multiple R-squared:  0.6852, Adjusted R-squared:  0.6619
## F-statistic: 29.38 on 2 and 27 DF,  p-value: 1.674e-07
```

Después de realizar los análisis anteriores, podemos concluir que el mejor modelo para explicar la variabilidad de la resistencia es aquel que utiliza solo la potencia y la temperatura, de acuerdo con el criterio de información de Akaike. Además, esto se observa en las variables, pues todas son estadísticamente significativas. La ecuación del modelo es:

$$Resistencia = -24.9 + 0.5Potencia + 0.13Temperatura$$

, por lo que cada unidad que se mueve en potencia y temperatura, la resistencia aumenta 0.5 y 0.13 unidades, respectivamente. Finalmente, observamos que el VIF de la potencia y la temperatura fue de 1, por lo que no hay multicolinealidad, y el modelo cumple con todos los supuestos y tiene un coeficiente de determinación del 0.6619 (66.19% de la variabilidad explicada de la resistencia), por lo que podemos concluir que este modelo es adecuado.

4. Haz el análisis de datos atípicos e influyentes del mejor modelo encontrado

Datos atípicos

Estandarización extrema de los residuos

```
Datos = data
library(dplyr)

##
## Attaching package: 'dplyr'

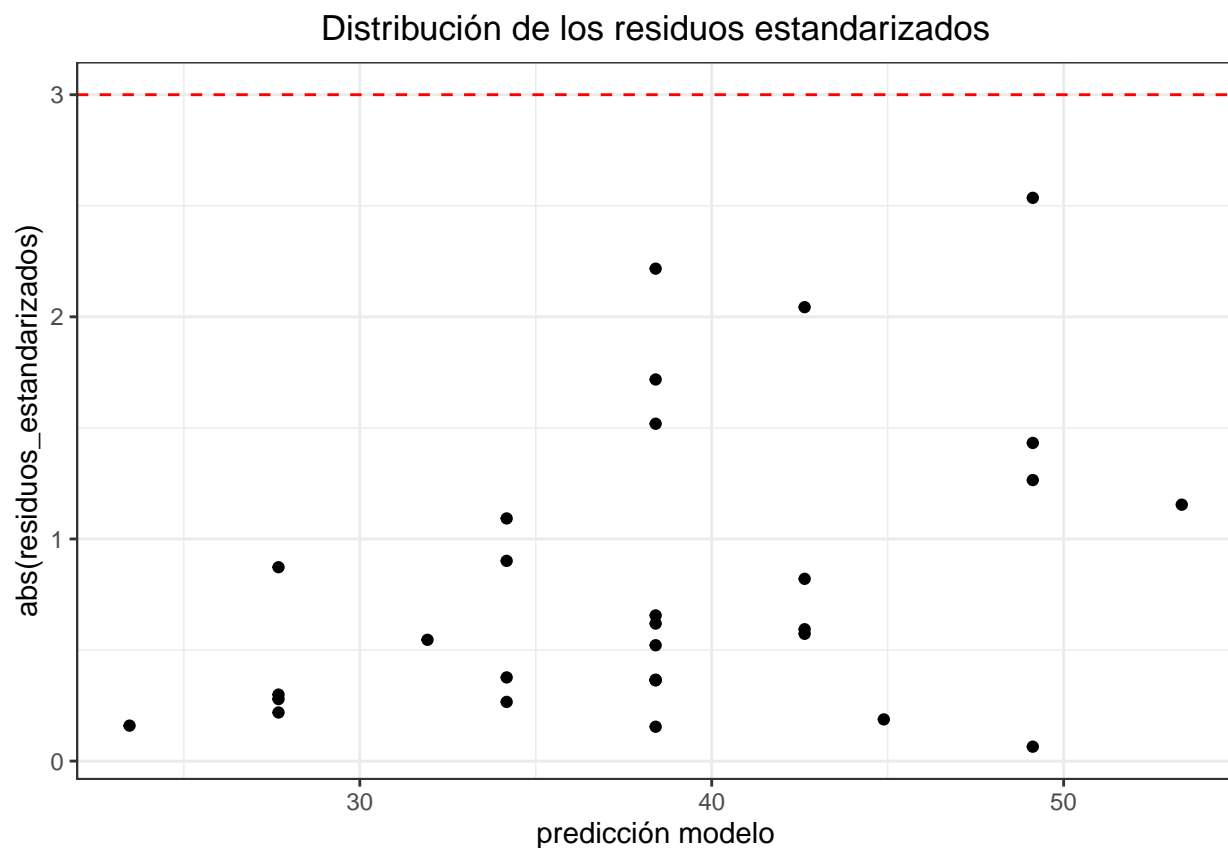
## The following object is masked from 'package:gridExtra':
##
##      combine

## 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
```

```
Datos$residuos_estandarizados <- rstudent(modelo)
ggplot(data = Datos, aes(x = predict(modelo), 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 = "Distribución de los residuos estandarizados", x = "predicción modelo") +
  theme_bw() + theme(plot.title = element_text(hjust = 0.5))
```



```
Atipicos = which(abs(Datos$residuos_estandarizados)>3)
Datos[Atipicos,]
```

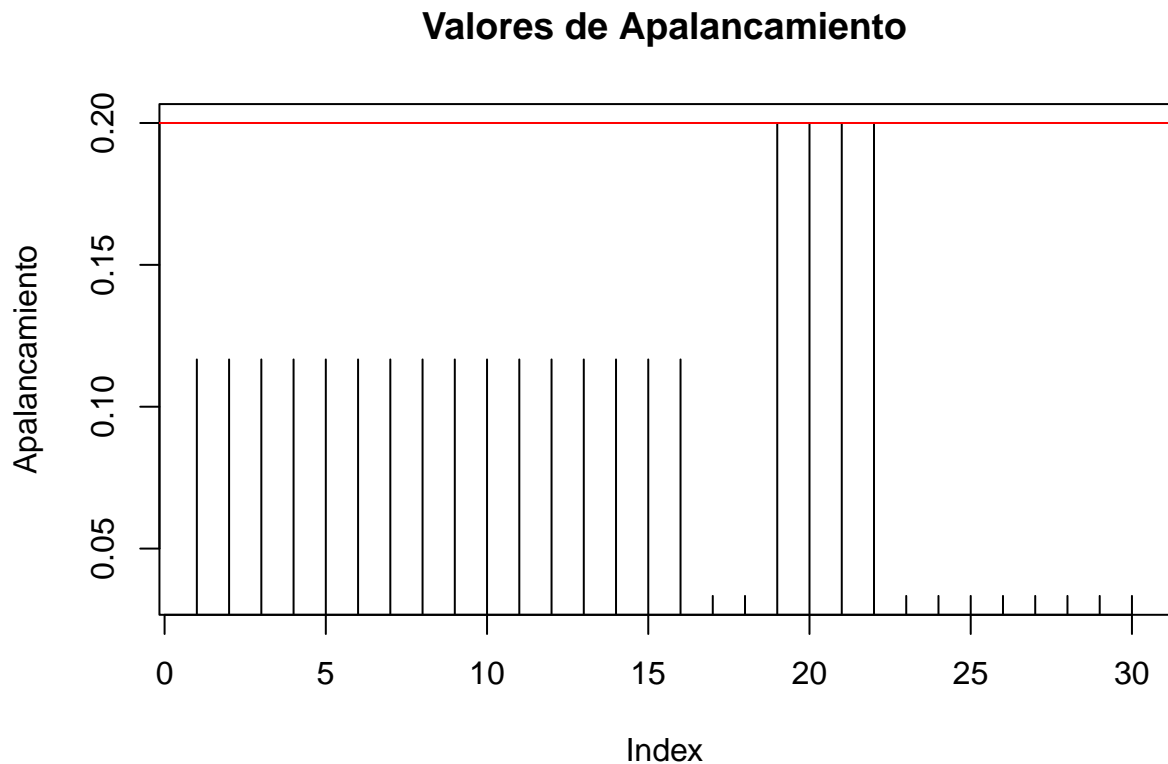
```
## [1] Fuerza          Potencia          Temperatura
## [4] Tiempo          Resistencia       residuos_estandarizados
## <0 rows> (or 0-length row.names)
```

En nuestros resultados, observamos que no hay datos atípicos, de acuerdo al análisis de la estandarización extrema de los residuos (nuestro criterio es que sea mayor a 3σ). Esto también se encuentra en la gráfica, pues no hay valores que sobrepasen la frontera.

Distancia de Leverage

```
leverage = hatvalues(modelo)

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



```
high_leverage_points = which(leverage > 2*mean(leverage))
Datos[high_leverage_points, ]
```

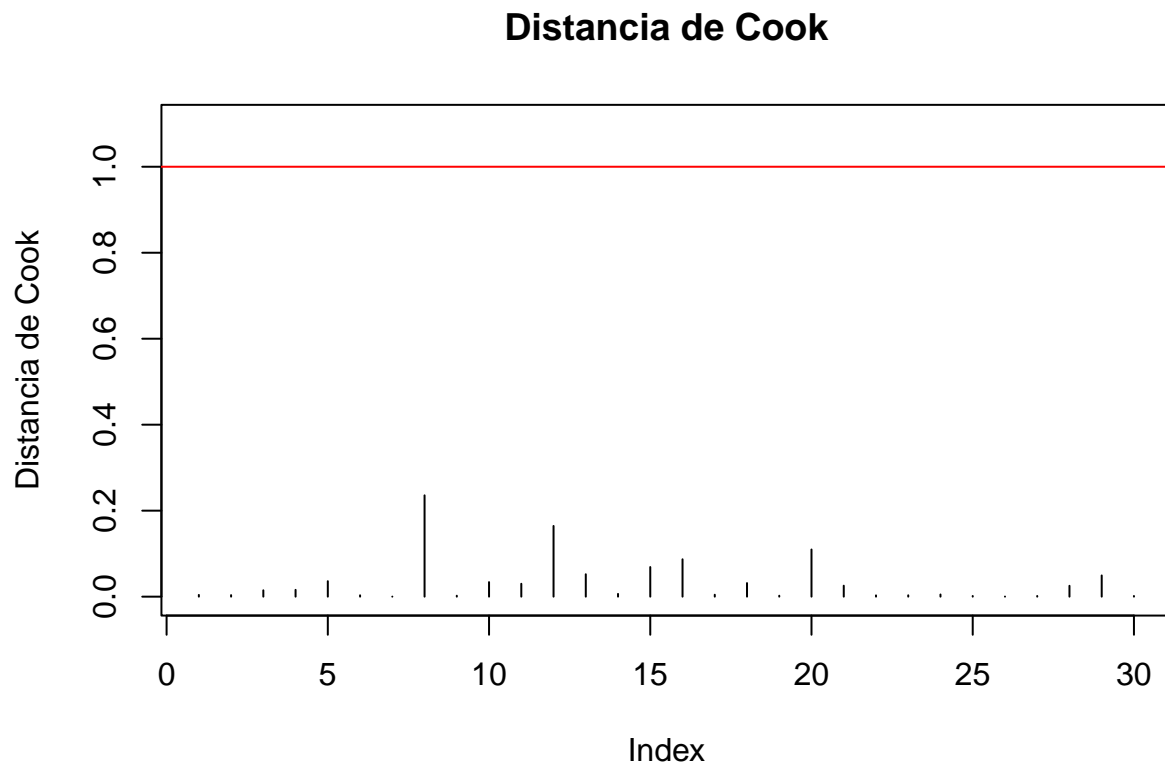
```
##      Fuerza Potencia Temperatura Tiempo Resistencia residuos_estandarizados
## 19      35         45          200     20         22.7          -0.159511
## 20      35        105          200     20         58.7           1.154355
```

De acuerdo con el criterio de datos atípicos, por medio de los valores de apalancamiento, encontramos dos valores que sobrepasan el doble de la media de las distancias de leverage, los cuales son el 19º y 20º dato. Observamos también en la gráfica que el 21 y 22 estuvieron cercanos al límite, pero no sobrepasaron la frontera establecida.

Datos influyentes

Distancia de Cook

```
cooks_d <- cooks.distance(modelo)
#Calcula la distancia de Cook de los n datos
plot(cooks_d, type="h", main="Distancia de Cook", ylab="Distancia de Cook", ylim=c(0, 1.1))
abline(h = 1, col="red") # Límite comúnmente usado
```



```
puntos_influyentes = which(cooks_d > 1)
Datos[puntos_influyentes, ]
```

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

De acuerdo con el criterio de distancias de cook para datos influyentes, no hay ningún dato que sea influyente, pues las distancias de cook están lejos de la frontera de una unidad. Esto también se puede observar en la gráfica, donde las distancias de cook son muy bajas, en comparación de la frontera.

DfBetas

```
dfbetas_values = dfbetas(modelo)
num_coefs = ncol(dfbetas_values)
par(mfrow = c(ceiling(num_coefs / 2), 2))
for (i in 1:num_coefs) {
  plot(dfbetas_values[, i], type = "h", main = paste("DfBetas para el coeficiente", i), ylab = "DfBetas")
  abline(h = c(-1, 1), col = "red")
  puntos_influyentes = which(abs(dfbetas_values[, i]) > 1)
  print(paste("Puntos influyentes para el coeficiente", i, ":"))
  if(length(puntos_influyentes) == 0){print('No hay puntos influyentes')}
  else{print(puntos_influyentes)}
}
```

```
## [1] "Puntos influyentes para el coeficiente 1 :"
```

```
## [1] "No hay puntos influyentes"
```

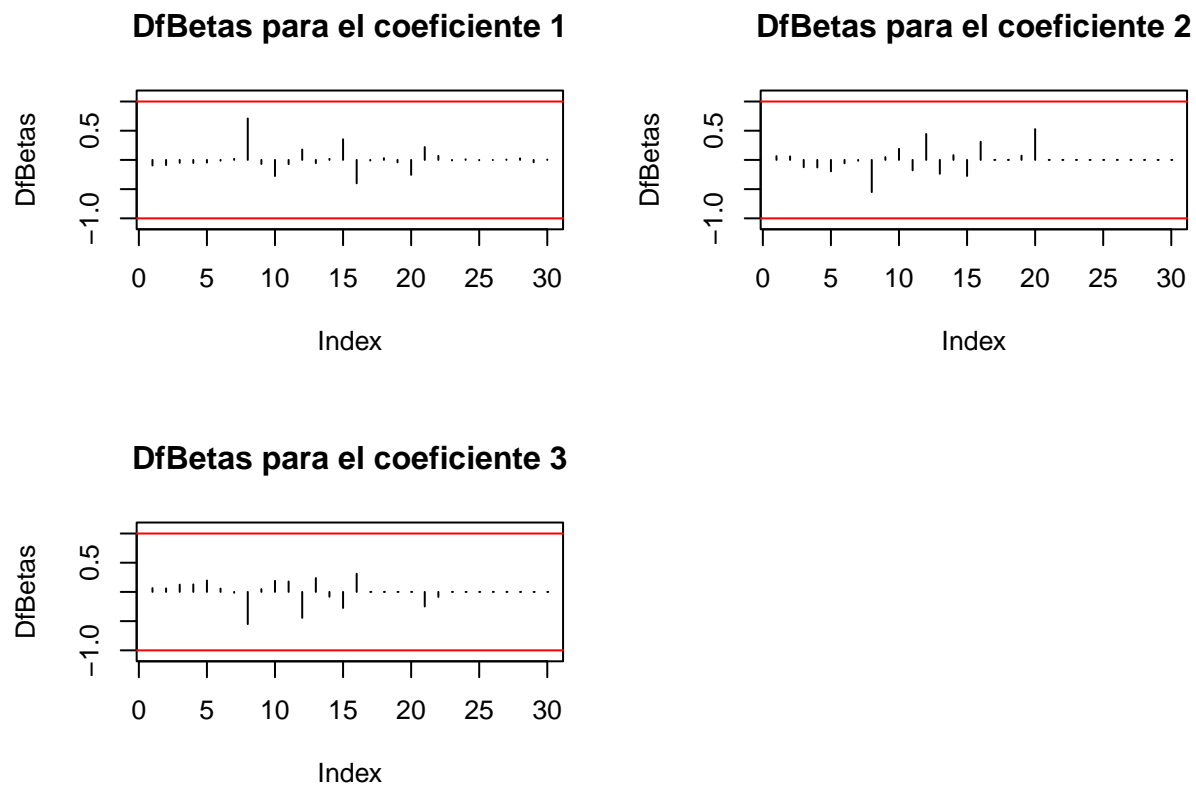
```
## [1] "Puntos influyentes para el coeficiente 2 :"
```

```
## [1] "No hay puntos influyentes"
```

```
## [1] "Puntos influyentes para el coeficiente 3 :"
```

```
## [1] "No hay puntos influyentes"
```

```
par(mfrow = c(1, 1))
```



De acuerdo con el criterio de datos influyentes utilizando dfbetas, observamos que no hay ningún dato influyente, pues ninguno sobrepasa las fronteras establecidas (-1 y 1). Lo mismo se observa en la gráfica, pues los valores quedan muy alejados de las fronteras.

Plot del modelo

```
par(mfrow=c(2, 2))  
plot(modelo, col='blue', pch=19)
```

