Problemas propuestos de Regresión Lineal Múltiple

Francisco Javier Mercader Martínez

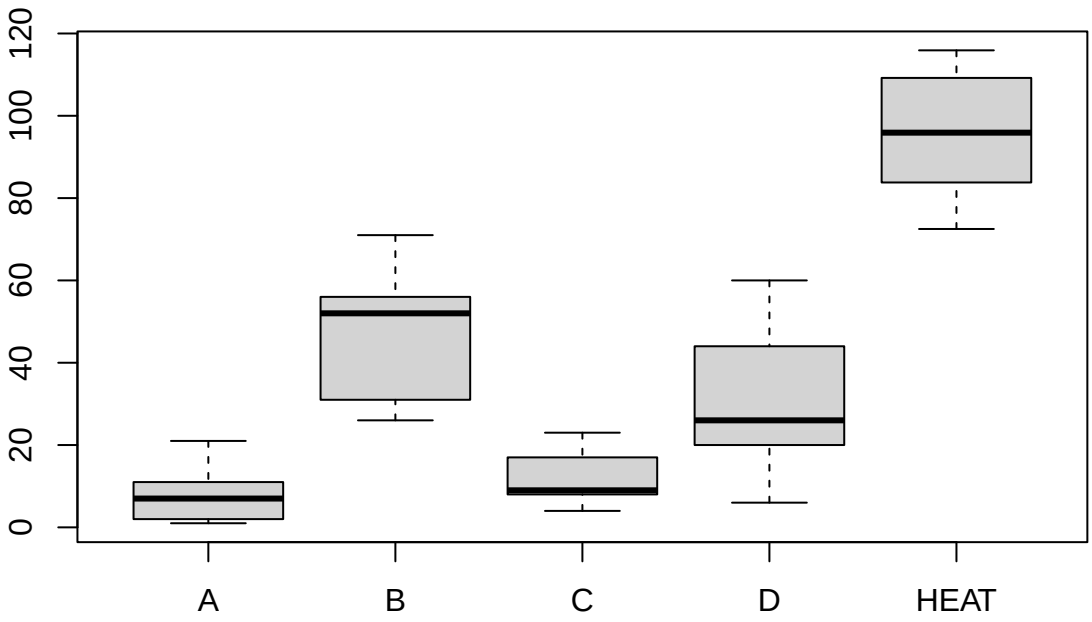
Problema 1

En el fichero **cemento_RLM.xlsx**, contiene los datos correspondientes a la presencia (en %) de cuatro componentes químicos en un tipo de cemento, así como el calor emitido (en calorías por gramo de cemento) durante el proceso de endurecimiento. Se desea proponer un modelo que permita predecir el calor emitido en función de los componentes químicos presentes del cemento.

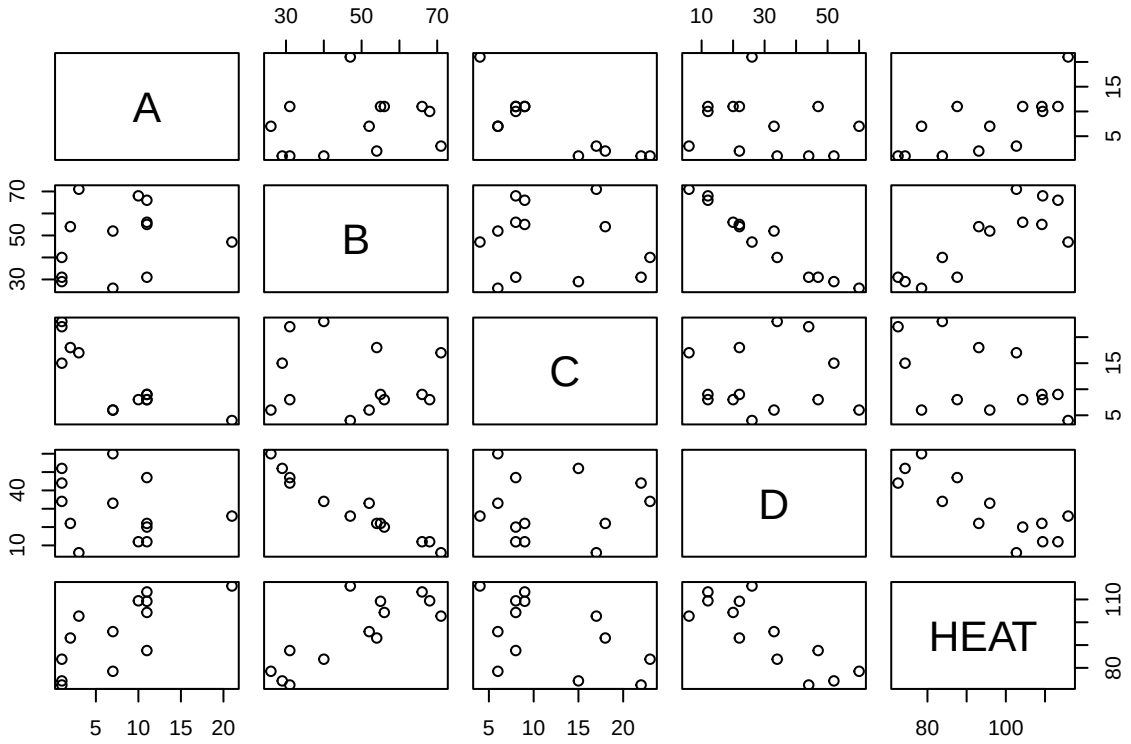
```
library(readxl)
cemento <- read_excel("../data/cemento_RLM.xlsx")
print.data.frame(cemento)
```

```
##     A  B  C  D  HEAT
## 1   7 26  6 60  78.5
## 2   1 29 15 52  74.3
## 3  11 56  8 20 104.3
## 4  11 31  8 47  87.6
## 5   7 52  6 33  95.9
## 6  11 55  9 22 109.2
## 7   3 71 17  6 102.7
## 8   1 31 22 44  72.5
## 9   2 54 18 22  93.1
## 10 21 47  4 26 115.9
## 11  1 40 23 34  83.8
## 12 11 66  9 12 113.3
## 13 10 68  8 12 109.4
```

```
boxplot(cemento)
```




```
plot(cemento)
```

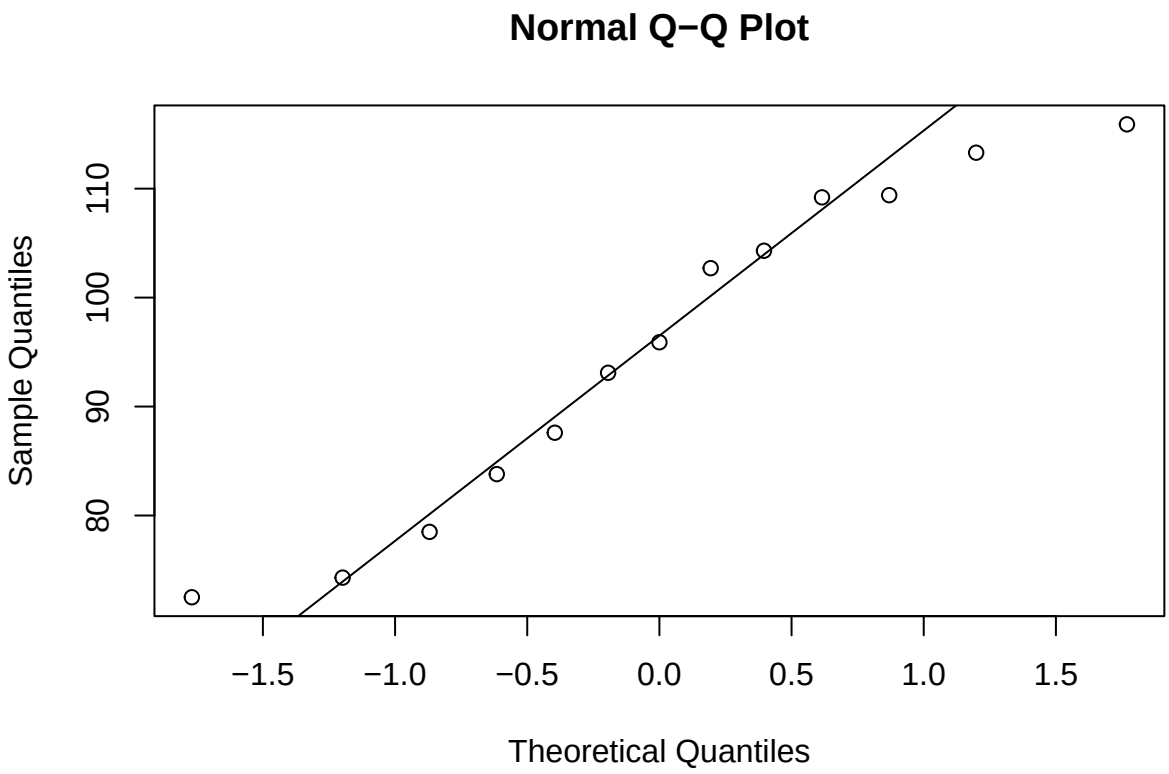


1) Realiza un análisis descriptivo previo de las variables del problema y comenta los resultados más relevantes. ¿Podemos suponer que nuestra variable respuesta es Normal?

```
shapiro.test(cemento$HEAT)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  cemento$HEAT
## W = 0.93, p-value = 0.4
```

```
qqnorm(cemento$HEAT)
qqline(cemento$HEAT)
```



2) Calcula la matriz de correlaciones de las cinco variables. ¿Qué información proporciona esta matriz? ¿Qué regresores del modelo presentan una más estrecha relación lineal entre sí? ¿Cuál es la primera variable que debería entrar en el modelo?

```
cor(cemento)
```

```
##            A       B        C        D    HEAT
## A     1.0000  0.2286 -0.82413 -0.24545  0.7307
## B     0.2286  1.0000 -0.13924 -0.97295  0.8163
## C    -0.8241 -0.1392  1.00000  0.02954 -0.5347
## D    -0.2454 -0.9730  0.02954  1.00000 -0.8213
## HEAT  0.7307  0.8163 -0.53467 -0.82131  1.0000
```

La matriz de correlaciones nos proporciona información sobre la relación lineal entre las variables.

- Las variables B y D tienen la correlación más fuerte entre sí (-0.9729550), lo que indica una fuerte relación lineal negativa.
- La variable Calor (la variable de respuesta) tiene la correlación más fuerte con la variable B (0.8162526), seguida de la variable A (0.7307175).

3) Realiza la selección del modelo mediante regresión por pasos, hacia delante y hacia atrás. Indica el orden de entrada y salida de las variables para cada uno de los métodos. Comenta los resultados obtenidos.

```
modelo_cte <- lm(HEAT ~ 1, data = cemento)


# Ajustar el modelo de regresión lineal completo
```

```r
modelo_completo <- lm(HEAT ~ ., data = cemento)

# Selección de modelo hacia adelante
modelo_forward <- step(modelo_cte, direction = "forward", scope =
 ↪  formula(modelo_completo))
```

```
## Start:  AIC=71.44
## HEAT ~ 1
##
##        Df Sum of Sq  RSS  AIC
## + D     1      1832  884 58.9
## + B     1      1809  906 59.2
## + A     1      1450 1266 63.5
## + C     1       776 1939 69.1
## <none>              2716 71.4
##
## Step:  AIC=58.85
## HEAT ~ D
##
##        Df Sum of Sq RSS  AIC
## + A     1       809  75 28.7
## + C     1       708 176 39.9
## <none>              884 58.9
## + B     1        15 869 60.6
##
## Step:  AIC=28.74
## HEAT ~ D + A
##
##        Df Sum of Sq  RSS  AIC
## + B     1      26.8 48.0 25.0
## + C     1      23.9 50.8 25.7
## <none>              74.8 28.7
##
## Step:  AIC=24.97
## HEAT ~ D + A + B
##
##        Df Sum of Sq  RSS  AIC
## <none>              48.0 25.0
## + C     1     0.109 47.9 26.9
```

```r
# Selección de modelo hacia atrás
modelo_backward <- step(modelo_completo, direction = "backward")
```

```
## Start:  AIC=26.94
## HEAT ~ A + B + C + D
##
##        Df Sum of Sq  RSS  AIC
## - C     1      0.11 48.0 25.0
## - D     1      0.25 48.1 25.0
## - B     1      2.97 50.8 25.7
## <none>              47.9 26.9
## - A     1     25.95 73.8 30.6
##
## Step:  AIC=24.97
```

```
## HEAT ~ A + B + D
##
##        Df Sum of Sq RSS  AIC
## <none>               48 25.0
## - D     1        10  58 25.4
## - B     1        27  75 28.7
## - A     1       821 869 60.6
```

```
# Regresión por pasos
modelo_stepwise <- step(modelo_cte, direction = "both", scope = formula(modelo_completo))
```

```
## Start:  AIC=71.44
## HEAT ~ 1
##
##        Df Sum of Sq  RSS  AIC
## + D     1      1832  884 58.9
## + B     1      1809  906 59.2
## + A     1      1450 1266 63.5
## + C     1       776 1939 69.1
## <none>              2716 71.4
##
## Step:  AIC=58.85
## HEAT ~ D
##
##        Df Sum of Sq  RSS  AIC
## + A     1       809   75 28.7
## + C     1       708  176 39.9
## <none>               884 58.9
## + B     1        15  869 60.6
## - D     1      1832 2716 71.4
##
## Step:  AIC=28.74
## HEAT ~ D + A
##
##        Df Sum of Sq  RSS  AIC
## + B     1        27   48 25.0
## + C     1        24   51 25.7
## <none>               75 28.7
## - A     1       809  884 58.9
## - D     1      1191 1266 63.5
##
## Step:  AIC=24.97
## HEAT ~ D + A + B
##
##        Df Sum of Sq RSS  AIC
## <none>               48 25.0
## - D     1        10  58 25.4
## + C     1         0  48 26.9
## - B     1        27  75 28.7
## - A     1       821 869 60.6
```

```
modelo_forward$coefficients
```

```
## (Intercept)           D           A           B
##     71.6483     -0.2365      1.4519      0.4161
```

```
modelo_backward$coefficients
```

```
## (Intercept)           A           B           D
##     71.6483      1.4519      0.4161     -0.2365
```

```
modelo_stepwise$coefficients
```

```
## (Intercept)           D           A           B
##     71.6483     -0.2365      1.4519      0.4161
```

4) Estudia si hay colinealidad entre los regresores de los modelos resultantes en el apartado anterior y en caso afirmativo explica cuál es tu decisión para solventarlo.

```
# Comprobar la colinealidad
library("rms")
round(vif(modelo_stepwise), digits = 4)
```

```
##      D      A      B
## 18.940  1.066 18.780
```

```
round(vif(modelo_forward), digits = 4)
```

```
##      D      A      B
## 18.940  1.066 18.780
```

```
round(vif(modelo_backward), digits = 4)
```

```
##      A      B      D
##  1.066 18.780 18.940
```

En el `modelo_completo` y el `modelo_forward`, todos los regresores tienen un VIF muy alto, lo que indica una fuerte colinealidad. Para solucionar esto, podrías considerar eliminar uno o más de los regresores, o combinarlos de alguna manera si tiene sentido en el contexto de tus datos.

En el `modelo_backward`, los regresores A tienen un VIF bajo, lo que indica que no hay colinealidad. Sin embargo, B y D tienen un VIF mayor a 5, lo que sugiere alguna colinealidad.

5) ¿Propondrías un único modelo o varios? ¿Cuál o cuáles y por qué?

El `modelo_backward` muestra una colinealidad moderada entre las variables B y D, pero la variable A no muestra colinealidad. Por lo tanto, este modelo puede ser más adecuado para la predicción.

6) Determina el (los) modelo(s) ajustado(s) y los intervalos de confianza al 95% para los parámetros de regresión.

```
# Modelo ajustado
confint(modelo_backward, level=0.95)
```

```
##                2.5 %   97.5 %
## (Intercept) 39.65599 103.6406
## A            1.18727   1.7166
## B           -0.00377   0.8360
## D           -0.62854   0.1555
```

7) Para el modelo que contempla sólo los regresores A y D, estudia si se verifican las hipótesis del modelo de regresión múltiple, comentando los procesos utilizados. Estudia si hay colinealidad entre los regresores y si aparecen observaciones influyentes, comentando los procesos utilizados. En caso de que se presente alguno de estos problemas, explica cuál es tu decisión para solventarlo.

```r
# Comprobar la colinealidad
modelo_ajustado <- lm(HEAT ~ A + D, data = cemento)
round(vif(modelo_ajustado), digits = 4)
```

```
##     A     D
## 1.064 1.064
```

Los valores de `vif` para los regresores A y D son ambos 1.064105. En este caso, los valores de `vif` son muy bajos, lo que indica que no hay colinealidad entre los regresores A y D. Por lo tanto, no es necesario tomar ninguna medida para tratar la colinealidad en este modelo.

8) Obtén una estimación puntual del calor emitido por el cemento sabiendo que A=15, B=39, C=4.5 y D=40. Determina también un intervalo de confianza para el calor emitido en ese caso, así como un intervalo de predicción. ¿Podemos concluir que el calor emitido por el cemento superará las 95 cal/gr? ¿Y en promedio?

```r
predict(modelo_ajustado, newdata = data.frame(A = 15, D = 40), interval = "confidence",
 ↪  level = 0.95)
```

```
##     fit   lwr   upr
## 1 100.1 96.87 103.4
```

```r
predict(modelo_ajustado, newdata = data.frame(A = 15, D = 40), interval = "prediction",
 ↪  level = 0.95)
```

```
##     fit   lwr   upr
## 1 100.1 93.23 107.1
```

El intervalo de confianza al 95% para el calor emitido por el cemento es (96.87177, 103.4055) cal/gr. El intervalo de predicción al 95% para el calor emitido por el cemento es (93.22567, 107.0515) cal/gr.

Dado que la estimación puntual del calor emitido por el cemento es de 100.1386 cal/gr, que está por encima de 95 cal/gr, podemos concluir que, en promedio, es probable que el calor emitido por el cemento supere las 95 cal/gr.

9) Responde a la cuestión anterior sabiendo que A=45 y D=40.

```r
predict(modelo_ajustado, newdata = data.frame(A = 45, D = 40), interval = "confidence",
 ↪  level = 0.95)
```

```
##     fit   lwr   upr
## 1 143.3 131.3 155.3
```

```r
predict(modelo_ajustado, newdata = data.frame(A = 45, D = 40), interval = "prediction",
 ↪  level = 0.95)
```

```
##     fit   lwr   upr
## 1 143.3 129.9 156.8
```

Dado que la estimación puntual del calor emitido por el cemento es de 143.3374 cal/gr, que está muy por encima de 95 cal/gr, podemos concluir que, en promedio, es más que probable que el calor emitido por el cemento supere las 95 cal/gr.

Problema 2

En el fichero **motor.dat** se encuentran los datos correspondientes a 200 ensayos, donde se midieron las siguientes variables: VRP (velocidad de rotación primaria), VRS (velocidad de rotación secundaria), Presion (presión), Temp_Esc (temperatura de escape), Temp_Amb (temperatura ambiente a la hora de efectuar la

prueba), LN_RFC (logaritmo neperiano de la rapidez de flujo de combustible) y Empuje (empuje del motor). Se desea proponer un modelo que permita predecir el "Empuje del motor" en función del resto de variables, analizando si serían necesarias todas o no.

```
motor <- read.table("../data/motor.dat", header = TRUE)
```

1) Indica la variable respuesta y los regresores del problema. Las variables del problema, ¿presentan datos atípicos? NO eliminies ningún dato. ¿Podemos suponer que nuestra variable respuesta es Normal? En caso negativo, justificar si la transformación logarítmica sería adecuada y realizarla.

```
summary(motor) # Los valores de la columna EMPUJE son caracteres, por lo que necesitamos
 ↪  convertirlos a numéricos
```

```
##       VRP            VRS           PRESION       TEMP_ESC       TEMP_AMB
##  Min.   :1403   Min.   :17008   Min.   :130   Min.   :1500   Min.   : 85.0
##  1st Qu.:1586   1st Qu.:17822   1st Qu.:154   1st Qu.:1558   1st Qu.: 89.0
##  Median :1802   Median :19140   Median :174   Median :1630   Median : 94.0
##  Mean   :1835   Mean   :19028   Mean   :175   Mean   :1622   Mean   : 93.9
##  3rd Qu.:2094   3rd Qu.:20107   3rd Qu.:197   3rd Qu.:1678   3rd Qu.: 99.0
##  Max.   :2300   Max.   :20950   Max.   :220   Max.   :1749   Max.   :102.0
##     LN_RFC             EMPUJE
##  Length:200         Length:200
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
```

```
motor$EMPUJE <- gsub(",", ".", motor$EMPUJE) # Reemplazar las comas por puntos para que
 ↪  el programa los reconozca

motor$EMPUJE <- as.numeric(motor$EMPUJE) # Convertir la columna EMPUJE a numérica

summary(motor) # Verificar que la columna EMPUJE ahora es numérica
```

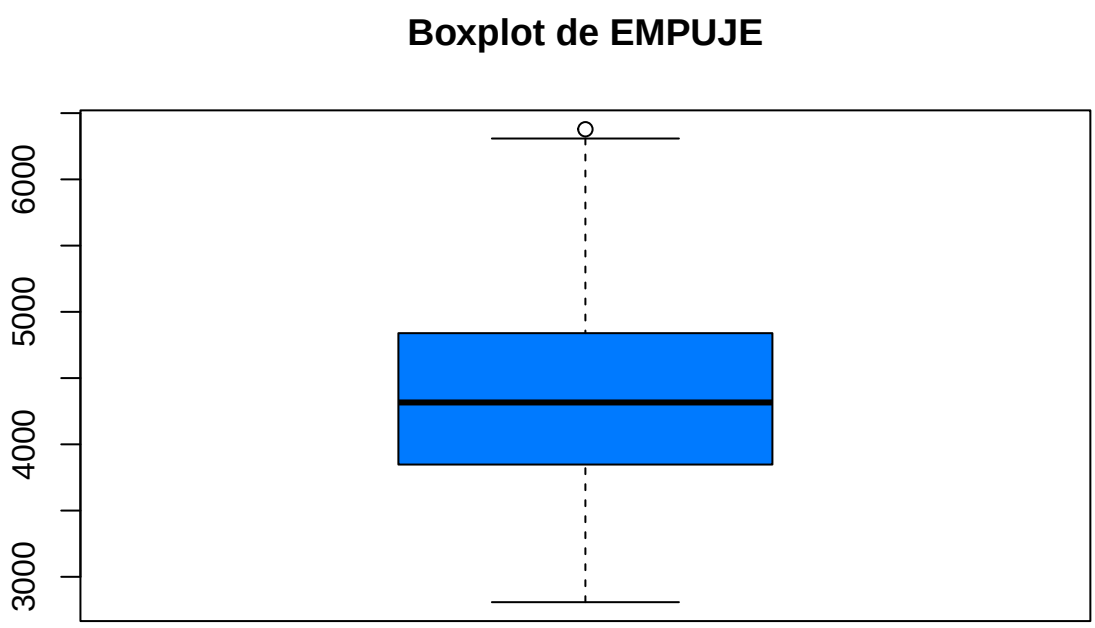
```
##       VRP            VRS           PRESION       TEMP_ESC       TEMP_AMB
##  Min.   :1403   Min.   :17008   Min.   :130   Min.   :1500   Min.   : 85.0
##  1st Qu.:1586   1st Qu.:17822   1st Qu.:154   1st Qu.:1558   1st Qu.: 89.0
##  Median :1802   Median :19140   Median :174   Median :1630   Median : 94.0
##  Mean   :1835   Mean   :19028   Mean   :175   Mean   :1622   Mean   : 93.9
##  3rd Qu.:2094   3rd Qu.:20107   3rd Qu.:197   3rd Qu.:1678   3rd Qu.: 99.0
##  Max.   :2300   Max.   :20950   Max.   :220   Max.   :1749   Max.   :102.0
##     LN_RFC              EMPUJE
##  Length:200         Min.   :2809
##  Class :character   1st Qu.:3849
##  Mode  :character   Median :4316
##                     Mean   :4360
##                     3rd Qu.:4839
##                     Max.   :6378
```

```
shapiro.test(motor$EMPUJE)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  motor$EMPUJE
```

```
## W = 0.98, p-value = 0.003
```

```
boxplot(motor$EMPUJE, main = "Boxplot de EMPUJE", col = "#007AFF", border = "black")
```

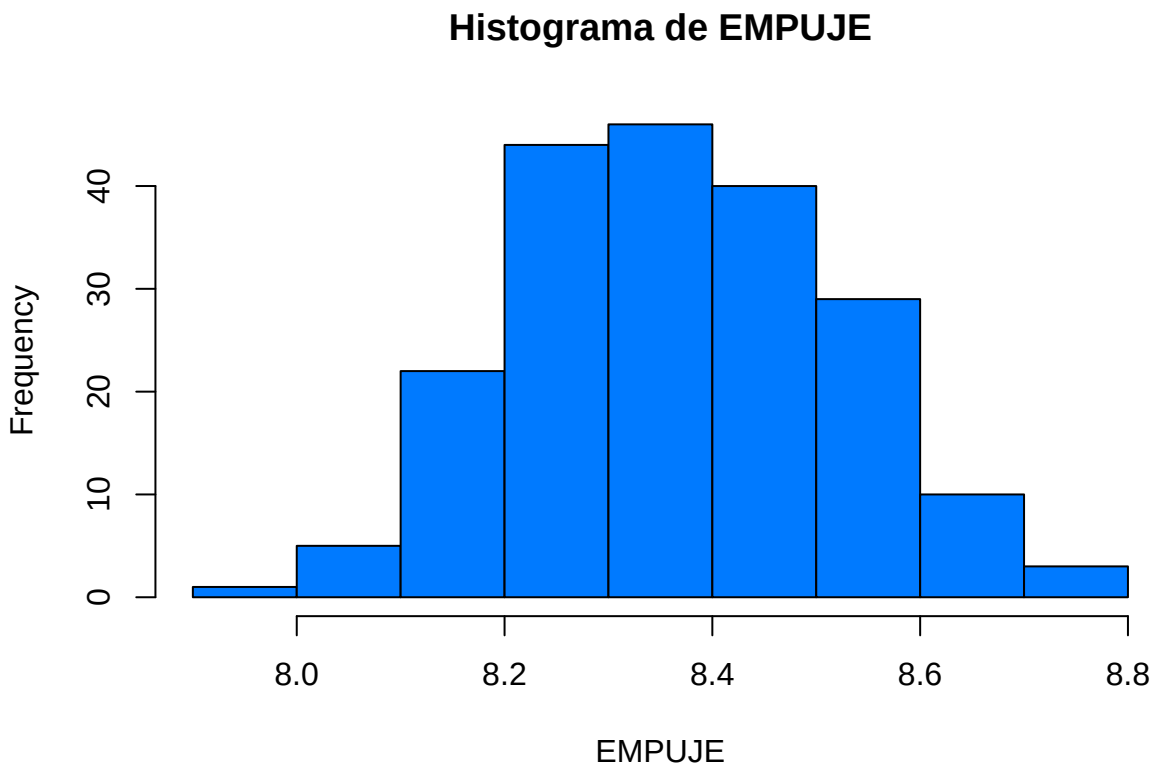


El `p-value` es 0.003448391, es menor que 0.05. Por lo tanto, rechazaríamos la hipótesis nula y concluiríamos que los datos no están normalmente distribuidos.

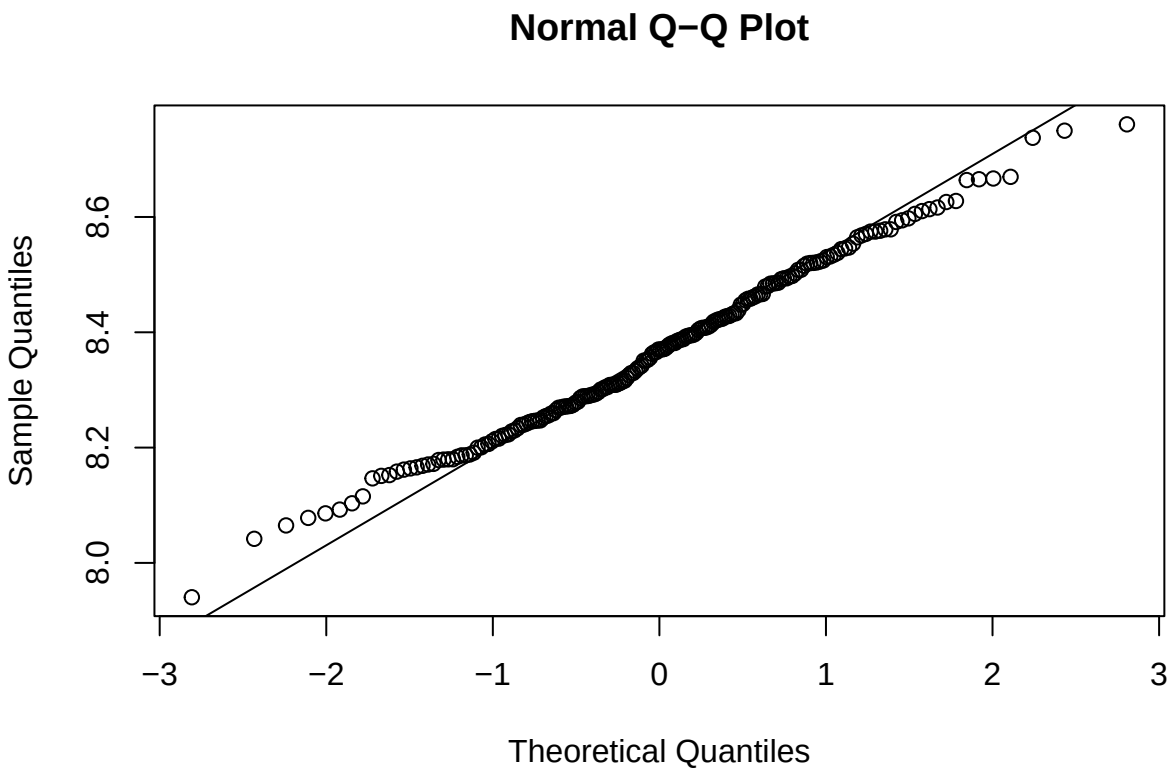
```
motor$EMPUJE <- log(motor$EMPUJE)
shapiro.test(motor$EMPUJE)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  motor$EMPUJE
## W = 0.99, p-value = 0.4
```

```
hist(motor$EMPUJE, main = "Histograma de EMPUJE", col = "#007AFF", border = "black", xlab
 ↪  = "EMPUJE")
```




```
qqnorm(motor$EMPUJE)
qqline(motor$EMPUJE)
```



Utilizando la tranformación logarítmica los valores de la variable respuesta se han normalo y hemos obtenido `p-value` con un valor mayor que 0.05, concluyendo que ahora los datos están normalmente distribuidos.

2) Calcula la matriz de correlaciones de las variables del problema. ¿Existen regresores altamente correlados dos a dos? ¿Cuál es la primera variable que debería entrar en el modelo? (indica el coeficiente de correlación en cada caso e interprétalo).

```
# Para poder hacer la matriz de correlaciones primero habría que convertir en valores
# numéricos la columma LN_RFC ya que nos encontramos el mismo problema que con la
# columma EMPUJE
motor$LN_RFC <- gsub(',', '.', motor$LN_RFC)
motor$LN_RFC <- as.numeric(motor$LN_RFC)
cor(motor)
```

```
##                VRP       VRS  PRESION TEMP_ESC TEMP_AMB    LN_RFC   EMPUJE
## VRP        1.00000 -0.017427  0.05410  0.02255 -0.01405 -0.012772  0.02237
## VRS       -0.01743  1.000000  0.04551  0.02927 -0.11999 -0.004835  0.06421
## PRESION    0.05410  0.045514  1.00000 -0.09177 -0.15822 -0.105858  0.84393
## TEMP_ESC   0.02255  0.029271 -0.09177  1.00000 -0.07088 -0.061965  0.14832
## TEMP_AMB  -0.01405 -0.119992 -0.15822 -0.07088  1.00000  0.216248 -0.15570
## LN_RFC    -0.01277 -0.004835 -0.10586 -0.06197  0.21625  1.000000 -0.24113
## EMPUJE     0.02237  0.064206  0.84393  0.14832 -0.15570 -0.241135  1.00000
```

Mirando tu matriz de correlaciones, la correlación más alta es entre las variables `PRESION` y `EMPUJE`, con un coeficiente de correlación de 0.84392883530. Esto indica una fuerte correlación positiva entre estas dos variables, lo que significa que cuando `PRESION` aumenta, `EMPUJE` también tiende a aumentar, y viceversa.

Por lo tanto, `PRESION` sería la primera variable que debería entrar en el modelo, ya que es la que tiene la

correlación más alta con la variable de respuesta.

3) Realiza la selección del modelo mediante regresión por pasos, hacia delante y hacia atrás. Para cada uno de los tres métodos, indica el modelo teórico resultante y estudia si existe multicolinealidad.

```
motor_cte <- lm(EMPUJE ~ 1, data = motor)
motor_completo <- lm(EMPUJE ~ ., data = motor)
motor_forward <- step(motor_cte, direction = "forward", scope = formula(motor_completo))
```

```
## Start:  AIC=-752.8
## EMPUJE ~ 1
##
##            Df Sum of Sq  RSS   AIC
## + PRESION   1      3.27 1.32 -1000
## + LN_RFC    1      0.27 4.33  -763
## + TEMP_AMB  1      0.11 4.48  -756
## + TEMP_ESC  1      0.10 4.49  -755
## <none>                  4.59  -753
## + VRS       1      0.02 4.57  -752
## + VRP       1      0.00 4.59  -751
##
## Step:  AIC=-999.9
## EMPUJE ~ PRESION
##
##            Df Sum of Sq  RSS   AIC
## + TEMP_ESC  1    0.2361 1.09 -1037
## + LN_RFC    1    0.1070 1.22 -1015
## <none>                  1.32 -1000
## + VRS       1    0.0031 1.32  -998
## + VRP       1    0.0025 1.32  -998
## + TEMP_AMB  1    0.0023 1.32  -998
##
## Step:  AIC=-1037
## EMPUJE ~ PRESION + TEMP_ESC
##
##            Df Sum of Sq  RSS   AIC
## + LN_RFC    1    0.0857 1.00 -1052
## <none>                  1.09 -1037
## + VRP       1    0.0040 1.08 -1036
## + VRS       1    0.0015 1.08 -1035
## + TEMP_AMB  1    0.0000 1.09 -1035
##
## Step:  AIC=-1052
## EMPUJE ~ PRESION + TEMP_ESC + LN_RFC
##
##            Df Sum of Sq   RSS   AIC
## <none>                  1.000 -1052
## + VRP       1   0.00422 0.996 -1051
## + TEMP_AMB  1   0.00282 0.997 -1050
## + VRS       1   0.00158 0.998 -1050
```

```
motor_backward <- step(motor_completo, direction = "backward")
```

```
## Start:  AIC=-1047
## EMPUJE ~ VRP + VRS + PRESION + TEMP_ESC + TEMP_AMB + LN_RFC
```

```
##
##            Df Sum of Sq  RSS   AIC
## - VRS       1      0.00 0.99 -1049
## - TEMP_AMB  1      0.00 0.99 -1049
## - VRP       1      0.00 1.00 -1049
## <none>                  0.99 -1047
## - LN_RFC    1      0.09 1.08 -1032
## - TEMP_ESC  1      0.22 1.21 -1010
## - PRESION   1      3.20 4.19  -761
##
## Step:  AIC=-1049
## EMPUJE ~ VRP + PRESION + TEMP_ESC + TEMP_AMB + LN_RFC
##
##            Df Sum of Sq  RSS   AIC
## - TEMP_AMB  1      0.00 1.00 -1051
## - VRP       1      0.00 1.00 -1050
## <none>                  0.99 -1049
## - LN_RFC    1      0.09 1.08 -1034
## - TEMP_ESC  1      0.22 1.21 -1011
## - PRESION   1      3.21 4.20  -763
##
## Step:  AIC=-1050
## EMPUJE ~ VRP + PRESION + TEMP_ESC + LN_RFC
##
##            Df Sum of Sq  RSS   AIC
## - VRP       1      0.00 1.00 -1052
## <none>                  1.00 -1051
## - LN_RFC    1      0.09 1.08 -1036
## - TEMP_ESC  1      0.22 1.21 -1013
## - PRESION   1      3.25 4.24  -763
##
## Step:  AIC=-1052
## EMPUJE ~ PRESION + TEMP_ESC + LN_RFC
##
##            Df Sum of Sq  RSS   AIC
## <none>                  1.00 -1052
## - LN_RFC    1      0.09 1.09 -1037
## - TEMP_ESC  1      0.21 1.21 -1015
## - PRESION   1      3.24 4.24  -765
```

```
motor_stepwise <- step(motor_cte, direction = "both", scope = formula(motor_completo))
```

```
## Start:  AIC=-752.8
## EMPUJE ~ 1
##
##            Df Sum of Sq  RSS   AIC
## + PRESION   1      3.27 1.32 -1000
## + LN_RFC    1      0.27 4.33  -763
## + TEMP_AMB  1      0.11 4.48  -756
## + TEMP_ESC  1      0.10 4.49  -755
## <none>                  4.59  -753
## + VRS       1      0.02 4.57  -752
## + VRP       1      0.00 4.59  -751
##
```

```
## Step:  AIC=-999.9
## EMPUJE ~ PRESION
##
##            Df Sum of Sq  RSS   AIC
## + TEMP_ESC  1      0.24 1.09 -1037
## + LN_RFC    1      0.11 1.21 -1015
## <none>                  1.32 -1000
## + VRS       1      0.00 1.32  -998
## + VRP       1      0.00 1.32  -998
## + TEMP_AMB  1      0.00 1.32  -998
## - PRESION   1      3.27 4.59  -753
##
## Step:  AIC=-1037
## EMPUJE ~ PRESION + TEMP_ESC
##
##            Df Sum of Sq  RSS   AIC
## + LN_RFC    1      0.09 1.00 -1052
## <none>                  1.09 -1037
## + VRP       1      0.00 1.08 -1036
## + VRS       1      0.00 1.08 -1035
## + TEMP_AMB  1      0.00 1.09 -1035
## - TEMP_ESC  1      0.24 1.32 -1000
## - PRESION   1      3.41 4.49  -755
##
## Step:  AIC=-1052
## EMPUJE ~ PRESION + TEMP_ESC + LN_RFC
##
##            Df Sum of Sq  RSS   AIC
## <none>                  1.00 -1052
## + VRP       1      0.00 1.00 -1051
## + TEMP_AMB  1      0.00 1.00 -1050
## + VRS       1      0.00 1.00 -1050
## - LN_RFC    1      0.09 1.09 -1037
## - TEMP_ESC  1      0.21 1.21 -1015
## - PRESION   1      3.24 4.24  -765
```

```
motor_forward$coefficients
```

```
## (Intercept)     PRESION    TEMP_ESC      LN_RFC
##  24.9610061   0.0049578   0.0004724  -1.7704779
```

```
motor_backward$coefficients
```

```
## (Intercept)     PRESION    TEMP_ESC      LN_RFC
##  24.9610061   0.0049578   0.0004724  -1.7704779
```

```
motor_stepwise$coefficients
```

```
## (Intercept)     PRESION    TEMP_ESC      LN_RFC
##  24.9610061   0.0049578   0.0004724  -1.7704779
```

```
# Comprobar la colinealidad
vif(motor_forward)
```

```
##  PRESION TEMP_ESC   LN_RFC
##    1.021    1.014    1.017
```

```r
vif(motor_backward)
```

```
##  PRESION TEMP_ESC   LN_RFC
##    1.021    1.014    1.017
```

```r
vif(motor_stepwise)
```

```
##  PRESION TEMP_ESC   LN_RFC
##    1.021    1.014    1.017
```

Todos los valores de `VIF` son muy bajos por lo que suponemos que no hay colinealidad.

4) ¿Qué modelo(s) de regresión propondrías y por qué? Indica el modelo ajustado que explica el "empuje del motor" y comenta la bondad del ajuste.

Aunque los valores son los mismo, voy a utilizar el `modelo_backward`.

```r
round(summary(modelo_backward)$r.squared, 4)
```

```
## [1] 0.9823
```

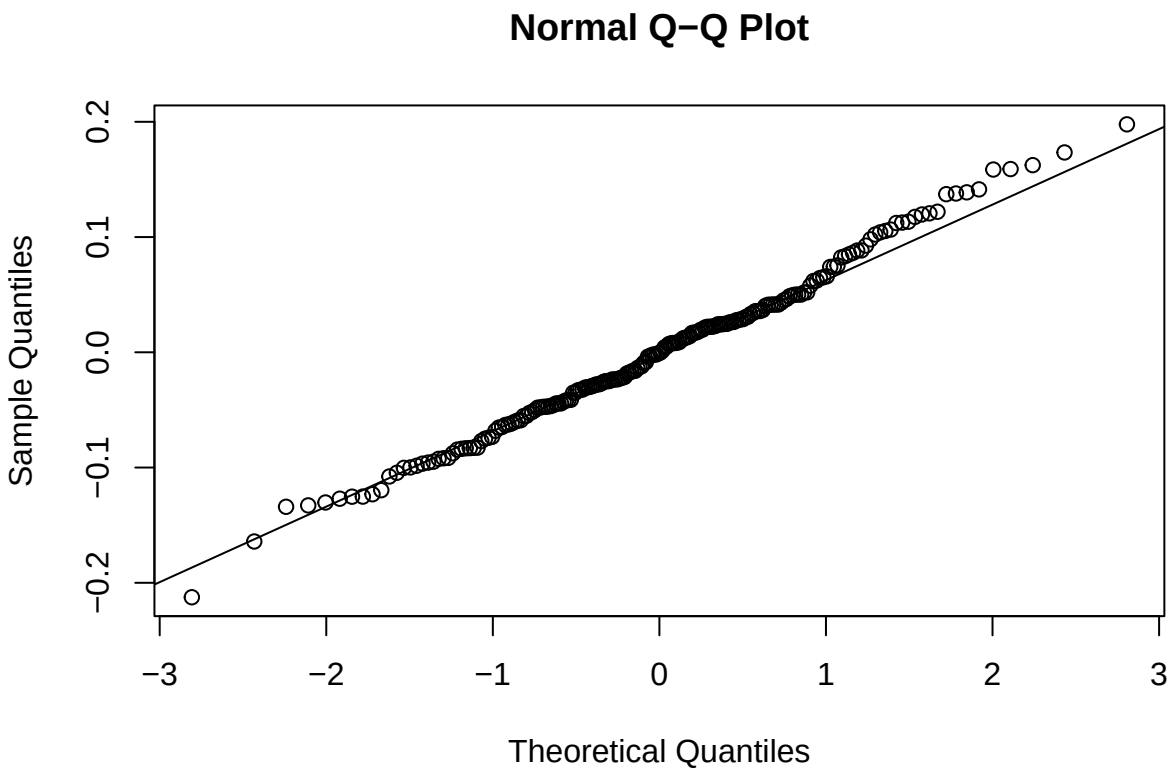
Así podemos comprobar que la bondad del ajuste es muy alta.

5) Para el modelo propuesto, estudia si se verifican las hipótesis del modelo de regresión múltiple y si existen observaciones influyentes. Comenta los procesos utilizados.

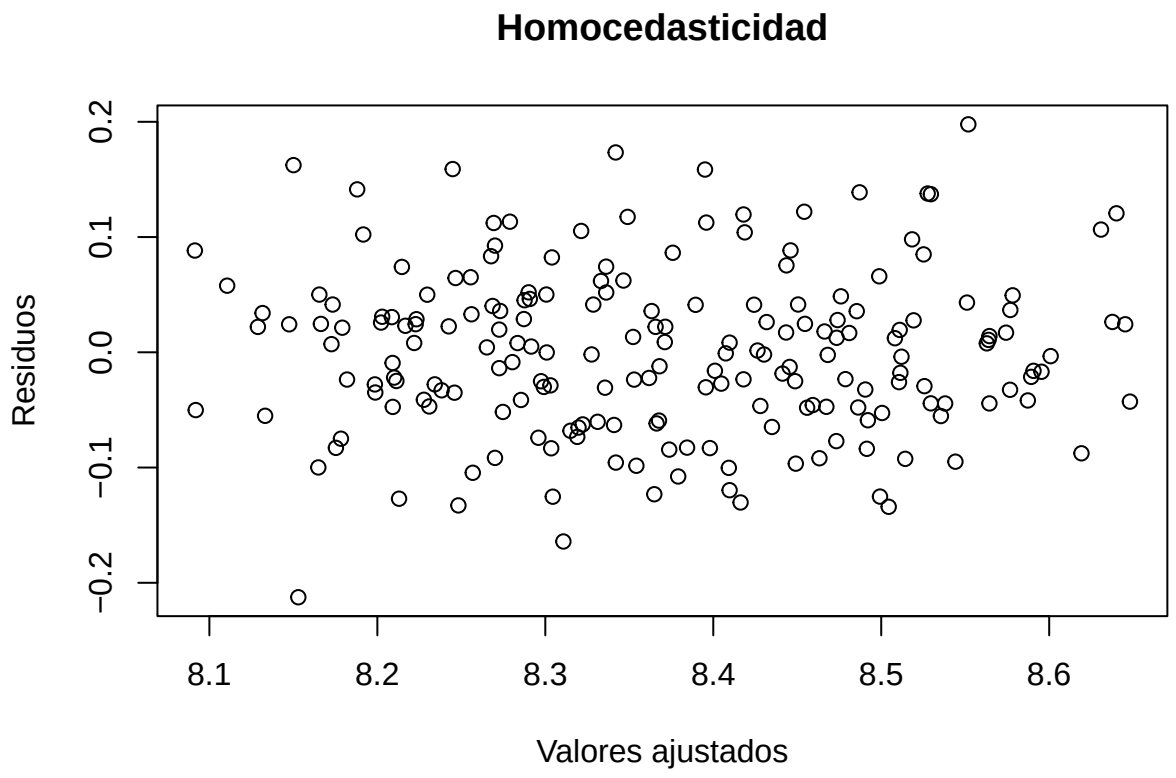
```r
# Normalidad de los residuos
shapiro.test(motor_backward$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  motor_backward$residuals
## W = 0.99, p-value = 0.6
```

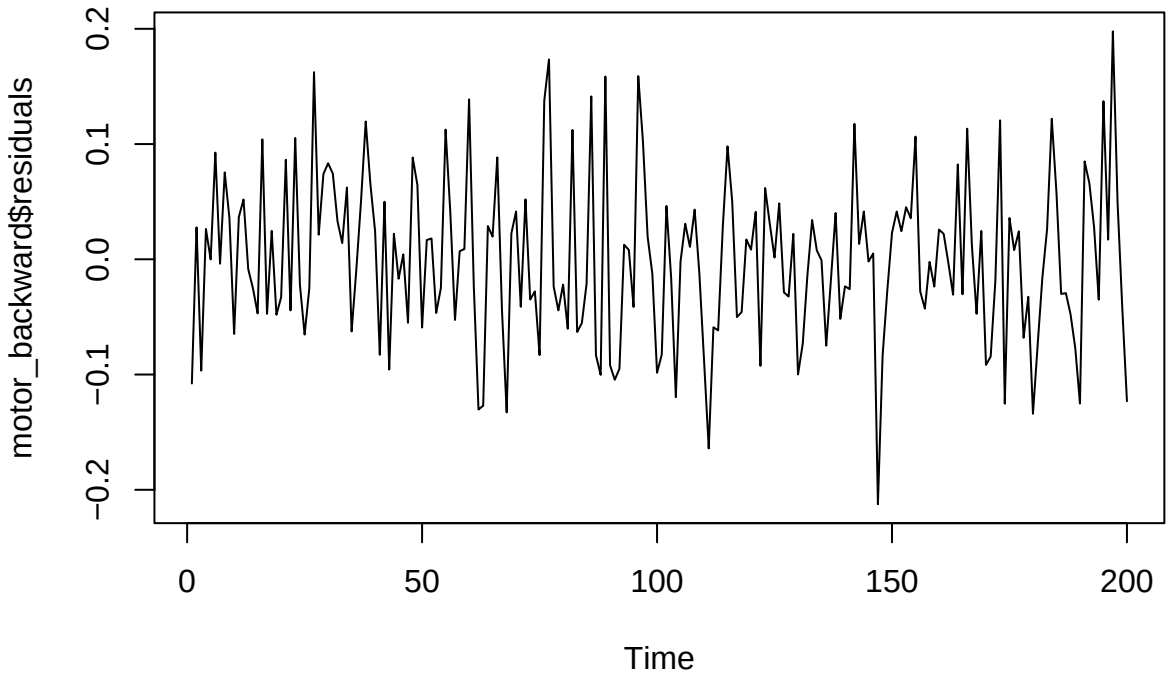
```r
qqnorm(motor_backward$residuals)
qqline(motor_backward$residuals)
```




```r
# Homocedasticidad
plot(motor_backward$fitted.values, motor_backward$residuals, main = "Homocedasticidad",
 ↪  xlab = "Valores ajustados", ylab = "Residuos")
```




```
# Hipótesis de independencia
ts.plot(motor_backward$residuals)
```




```r
library("lmtest")
```

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

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

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

```
##
## Attaching package: 'lmtest'
```

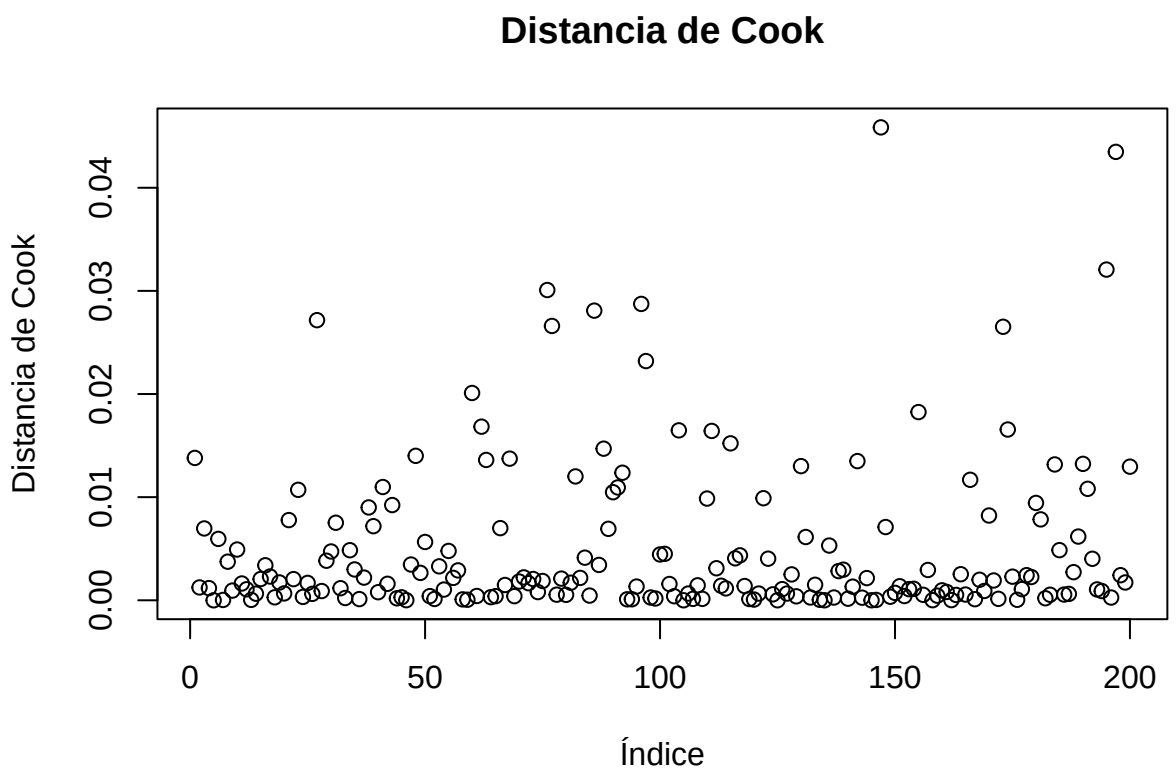
```
## The following object is masked from 'package:rms':
##
##     lrtest
```

```r
dwtest(motor_backward, alternative = "two.sided")
```

```
##
##  Durbin-Watson test
##
## data:  motor_backward
## DW = 1.8, p-value = 0.2
## alternative hypothesis: true autocorrelation is not 0
```

```r
# Observaciones influyentes
cook <- cooks.distance(motor_backward)
```

```r
plot(cook, main = "Distancia de Cook", xlab = "Índice", ylab = "Distancia de Cook")
```



6) Porporciona una estimación puntual del "empuje del motor" para un ensayo de las siguientes características:

VRP= 2000, VRS=19000, LN_RFC= 10.3089, Presion = 180, Temp_Esc = 1700 y Temp_Amb= 95.

Determinar también un intervalo predicción individual para el "empuje" en ese caso, así como un intervalo de confianza para el "empuje" promedio. ¿Podemos concluir que el "empuje del motor" será superior a 4000? ¿Y en promedio para los ensayos de esas características?

```r
predict(motor_backward, newdata = data.frame(VRP = 2000, VRS = 19000, LN_RFC = 10.3089,
 ↪  PRESION = 180, TEMP_ESC = 1700, TEMP_AMB = 95), interval = "prediction")
```

```
##     fit   lwr   upr
## 1 8.405 8.263 8.547
```

```r
predict(motor_backward, newdata = data.frame(VRP = 2000, VRS = 19000, LN_RFC = 10.3089,
 ↪  PRESION = 180, TEMP_ESC = 1700, TEMP_AMB = 95), interval = "confidence")
```

```
##     fit   lwr   upr
## 1 8.405 8.385 8.425
```

Como se puede apreciar ninguno de los datos obtenidos nos da un valor superior a 4000, por lo que podemos concluir que el "empuje del motor" no será superior a 4000.

Problema 3

Con los datos del Problema 1 (fichero **cemento_RLM.xlsx**), responder a las siguientes cuestiones:

1) Obtener la ecuación del modelo ajustado por mínimos cuadrados usando todos los predictores. Realizar el ajuste de tres formas diferentes:

   a) Primero con la función `lm()` de R.

```
lm(HEAT ~ ., data = cemento)
```

```
##
## Call:
## lm(formula = HEAT ~ ., data = cemento)
##
## Coefficients:
## (Intercept)            A            B            C            D
##      62.405        1.551        0.510        0.102       -0.144
```

   b) Después usando la inversa de $(t(M) * M)$, siendo M la matriz de diseño y $t(M)$ su traspuesta.

```
M <- model.matrix(HEAT ~ ., data = cemento)
t_M <- t(M)
solve(t_M %*% M) %*% t_M %*% cemento$HEAT
```

```
##                [,1]
## (Intercept) 62.4054
## A            1.5511
## B            0.5102
## C            0.1019
## D           -0.1441
```

   c) Por último. usando el método GD (Gradiente descendente). En este caso, debes probar con diferentes valores del número de iteraciones, learning rate y valores iniciales.

```
library("optimx")
optimx(par = rep(0, ncol(M)), fn = function(beta) sum((cemento$HEAT - M %*% beta)^2),
 ↪  method = "BFGS")
```

```
##         p1    p2     p3     p4      p5 value fevals gevals niter convcode kkt1
## BFGS 62.44 1.551 0.5098 0.1016 -0.1444 47.86     41     12    NA        0 TRUE
##       kkt2 xtime
## BFGS FALSE     0
```

¿Se obtienen los mismos resultados?

Si se obtienen los mismos resultados.

2) Repetir el apartado anterior, pero usando sólo los predictores A y D.

```
lm(HEAT ~ A + D, data = cemento)
```

```
##
## Call:
## lm(formula = HEAT ~ A + D, data = cemento)
##
## Coefficients:
## (Intercept)            A            D
##     103.097        1.440       -0.614
```

```r
M <- model.matrix(HEAT ~ A + D, data = cemento)
t_M <- t(M)
solve(t_M %*% M) %*% t_M %*% cemento$HEAT
```

```
##                 [,1]
## (Intercept) 103.097
## A             1.440
## D            -0.614
```