Hoja 4 (d): Modelos lineales con R

Estadística Computacional I. Grado en Estadística

Departamento de Estadística e Investigación Operativa. Universidad de Sevilla

Ejercicio 1

ANOVA de un factor.

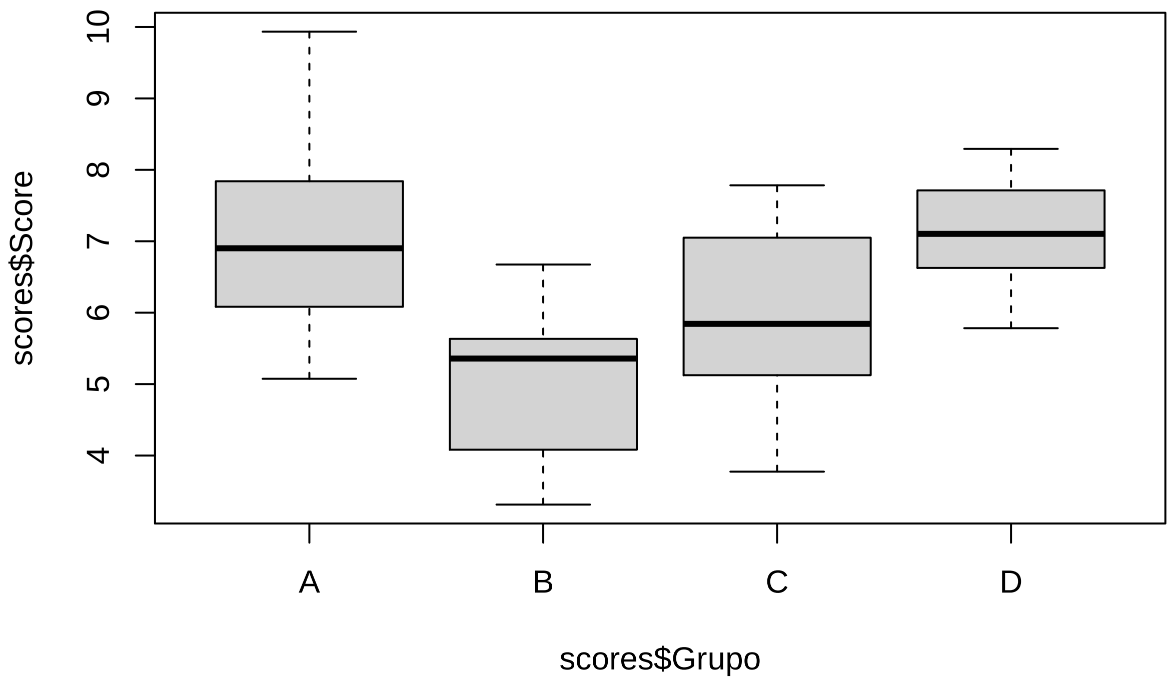
El fichero de datos "scores.txt" contiene la puntuación obtenida en una prueba de nivel de inglés para 40 alumnos. Se han considerado 4 academias, de cada una de las cuales han sido seleccionados aleatoriamente 10 alumnos. Se trata de estudiar si existen diferencias significativas entre las puntuaciones medias dependiendo de la academia.

Solución

```
scores = read.table(file = "scores.txt", header = TRUE)
head(scores)
```

```
##      Score Grupo
## 1 5.786680     B
## 2 5.304995     B
## 3 4.059006     B
## 4 5.547848     B
## 5 4.865416     B
## 6 3.313361     B
```

```
boxplot(scores$Score ~ scores$Grupo)
```



Las medias para cada academia

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
scores %>%
  group_by(Grupo) %>%
  summarise(
    Media = mean(Score)
  )
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 4 x 2
##   Grupo Media
##   <chr> <dbl>
## 1 A      6.97
## 2 B      5.07
## 3 C      5.95
## 4 D      7.10
```

Estudiamos la Normalidad

```
shapiro.test(scores %>% filter(Grupo=="A") %>% select(Score) %>% pull())
```

```
##
##  Shapiro-Wilk normality test
```

```
##
## data:  scores %>% filter(Grupo == "A") %>% select(Score) %>% pull()
## W = 0.94232, p-value = 0.5791
```

```
shapiro.test(scores %>% filter(Grupo=="B") %>% select(Score) %>% pull())
```

```
##
##  Shapiro-Wilk normality test
##
## data:  scores %>% filter(Grupo == "B") %>% select(Score) %>% pull()
## W = 0.9529, p-value = 0.7028
```

```
shapiro.test(scores %>% filter(Grupo=="C") %>% select(Score) %>% pull())
```

```
##
##  Shapiro-Wilk normality test
##
## data:  scores %>% filter(Grupo == "C") %>% select(Score) %>% pull()
## W = 0.97104, p-value = 0.9003
```

```
shapiro.test(scores %>% filter(Grupo=="D") %>% select(Score) %>% pull())
```

```
##
##  Shapiro-Wilk normality test
##
## data:  scores %>% filter(Grupo == "D") %>% select(Score) %>% pull()
## W = 0.98141, p-value = 0.9723
```

```
scores %>%
  group_by(Grupo) %>%
  summarise(
    pvalor.shapiro = shapiro.test(Score)$p.value
  )
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 4 x 2
##   Grupo pvalor.shapiro
##   <chr>          <dbl>
## 1 A              0.579
## 2 B              0.703
## 3 C              0.900
## 4 D              0.972
```

```
library(purrr)
scores %>%
  split( .$Grupo) %>%
  #map(~ shapiro.test(.x$Score))
  map_dbl(~ shapiro.test(.x$Score)$p.value)
```

```
##         A         B         C         D
## 0.5791077 0.7028269 0.9003301 0.9722507
```

Aceptamos la Normalidad en las 4 academias.

Estudiamos la homocedasticidad o igualdad de varianzas

```
library(car)
```

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

```
##
## Attaching package: 'car'

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

## The following object is masked from 'package:purrr':
##
##     some
```

```
leveneTest(scores$Score ~ scores$Grupo, center = "mean" )
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.

## Levene's Test for Homogeneity of Variance (center = "mean")
##       Df F value Pr(>F)
## group  3  0.7837 0.5109
##       36
```

El p-valor es $0.5109 > 0.05$, aceptamos la igualdad de varianzas.

Se cumplen las hipótesis para poder aplicar los test paramétricos ANOVA.

```
anova1factor = aov(Score ~ Grupo, data = scores)
anova1factor
```

```
## Call:
##    aov(formula = Score ~ Grupo, data = scores)
##
## Terms:
##                    Grupo Residuals
## Sum of Squares  27.23780  45.79052
## Deg. of Freedom        3        36
##
## Residual standard error: 1.127812
## Estimated effects may be unbalanced
```

```
summary(anova1factor)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## Grupo        3  27.24   9.079   7.138 0.000698 ***
## Residuals   36  45.79   1.272
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Se rechaza la igualdad de medias.

```
cbind(Coef = coef(anova1factor), confint(anova1factor))
```

```
##                  Coef      2.5 %     97.5 %
## (Intercept)  6.971713  6.2484030  7.6950234
## GrupoB      -1.903958 -2.9268735 -0.8810433
## GrupoC      -1.021658 -2.0445735  0.0012568
## GrupoD       0.125106 -0.8978091  1.1480211
```

Para realizar comparaciones múltiples, vamos a emplear los métodos de Tukey y Duncan.

Método de Tukey

```r
TukeyHSD(x = anova1factor, conf.level = 0.95)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = Score ~ Grupo, data = scores)
##
## $Grupo
##           diff        lwr        upr     p adj
## B-A -1.9039584 -3.2623485 -0.5455683 0.0031072
## C-A -1.0216583 -2.3800484  0.3367318 0.1976491
## D-A  0.1251060 -1.2332841  1.4834961 0.9945430
## C-B  0.8823001 -0.4760901  2.2406902 0.3140345
## D-B  2.0290644  0.6706743  3.3874545 0.0015414
## D-C  1.1467643 -0.2116258  2.5051545 0.1233214
```

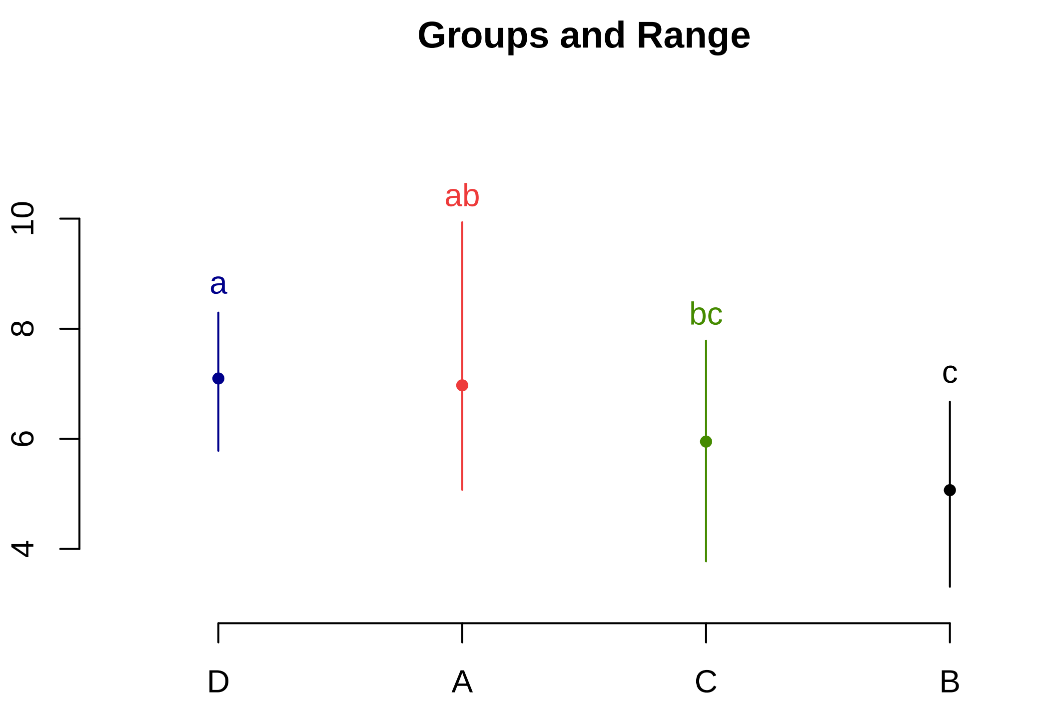
Diferencias significativas: B-A y D-B (p valores menores que 0.05).

Método de Duncan:

```r
library(agricolae)
resD = duncan.test(anova1factor, trt = "Grupo", console = TRUE)
```

```
##
## Study: anova1factor ~ "Grupo"
##
## Duncan's new multiple range test
## for Score
##
## Mean Square Error:  1.271959
##
## Grupo,  means
##
##      Score       std  r      Min      Max
## A 6.971713 1.4097619 10 5.074712 9.934203
## B 5.067755 0.9961449 10 3.313361 6.674236
## C 5.950055 1.2419218 10 3.774473 7.783607
## D 7.096819 0.7521519 10 5.783111 8.293345
##
## Alpha: 0.05 ; DF Error: 36
##
## Critical Range
##        2        3        4
## 1.022915 1.075364 1.109571
##
## Means with the same letter are not significantly different.
##
##      Score groups
## D 7.096819      a
## A 6.971713     ab
## C 5.950055     bc
## B 5.067755      c
```

```r
plot(resD)
```



$$H_0 : \mu_1 = \mu_2 = \mu_3 = \mu_4$$

Si hubiéramos usado la estadística no paramétrica, sería a través de

```
kruskal.test(scores$Score ~ scores$Grupo)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  scores$Score by scores$Grupo
## Kruskal-Wallis chi-squared = 14.549, df = 3, p-value = 0.002245
```

Ejercicio 2

ANOVA de dos factores.

A fin de investigar el efecto del fármaco Rhitalin sobre los niños hiperactivos se tomó una muestra de 4 niños para cada uno de los cruces de los dos siguientes factores: Tipo de niño (normal e hiperactivo) y medicamento administrado (Placebo y Rhitalin). Para cada niño se midió un índice de actividad.

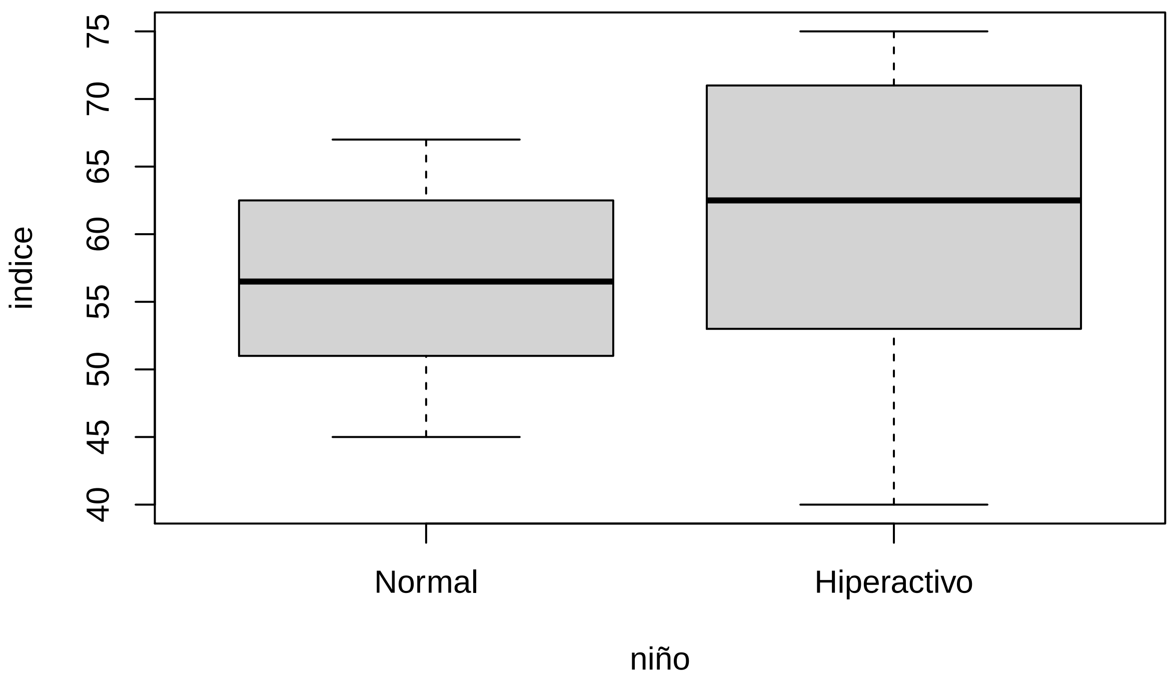
```
indice<-c(50,45,55,52,67,60,58,65,70,72,68,
          75,51,57,40,55)
niño<- gl(2,8)
levels(niño)<- c("Normal","Hiperactivo")
tratamiento<- gl(2,4,16)
levels(tratamiento)<- c("Placebo","Rhitalin")
datos2 = data.frame(niño,tratamiento,indice)
datos2
```

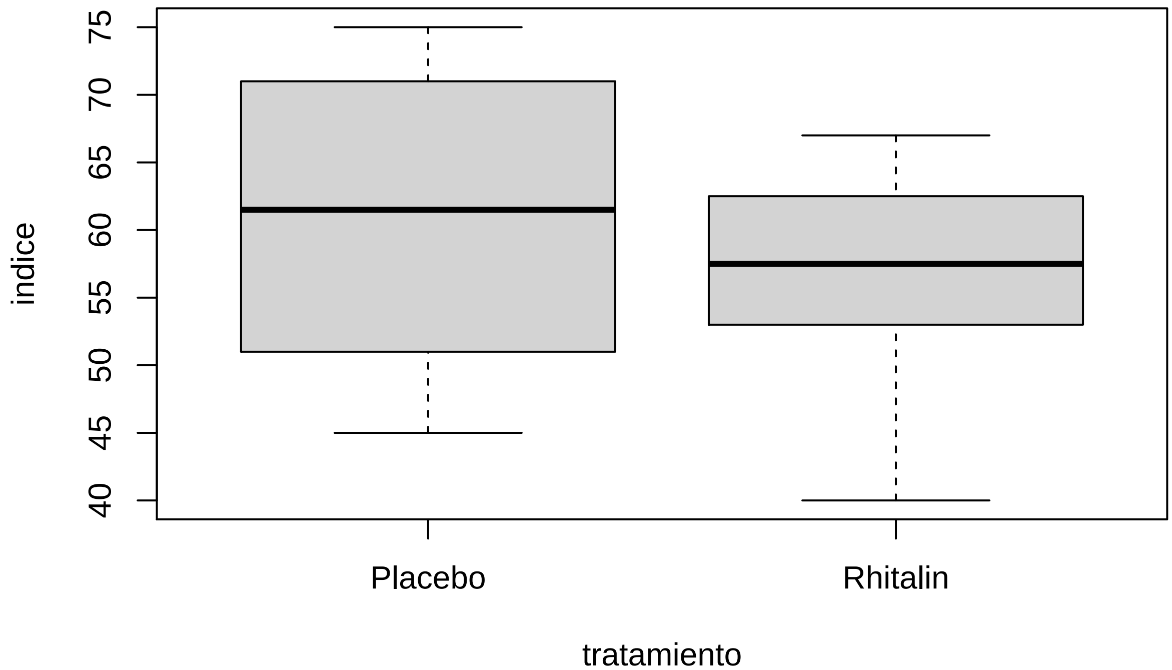
```
##          niño tratamiento indice
## 1      Normal     Placebo     50
## 2      Normal     Placebo     45
## 3      Normal     Placebo     55
```

```
## 4        Normal     Placebo     52
## 5        Normal    Rhitalin     67
## 6        Normal    Rhitalin     60
## 7        Normal    Rhitalin     58
## 8        Normal    Rhitalin     65
## 9  Hiperactivo     Placebo     70
## 10 Hiperactivo     Placebo     72
## 11 Hiperactivo     Placebo     68
## 12 Hiperactivo     Placebo     75
## 13 Hiperactivo    Rhitalin     51
## 14 Hiperactivo    Rhitalin     57
## 15 Hiperactivo    Rhitalin     40
## 16 Hiperactivo    Rhitalin     55
```

Solución

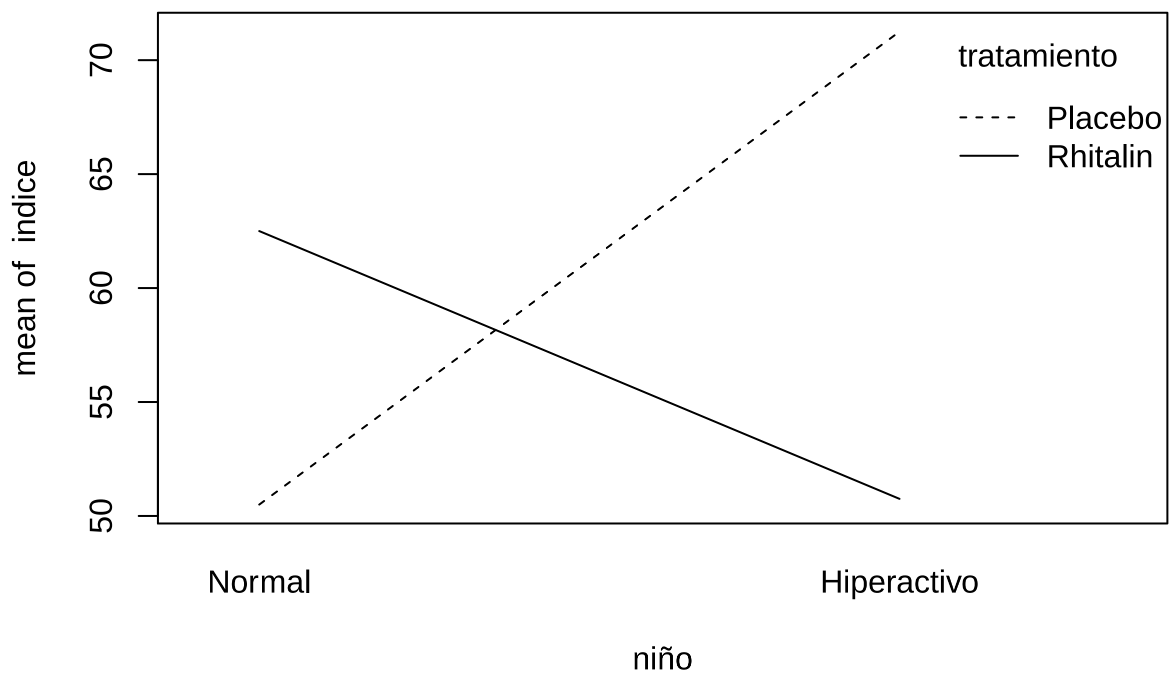
```
plot(indice ~ niño + tratamiento)
```



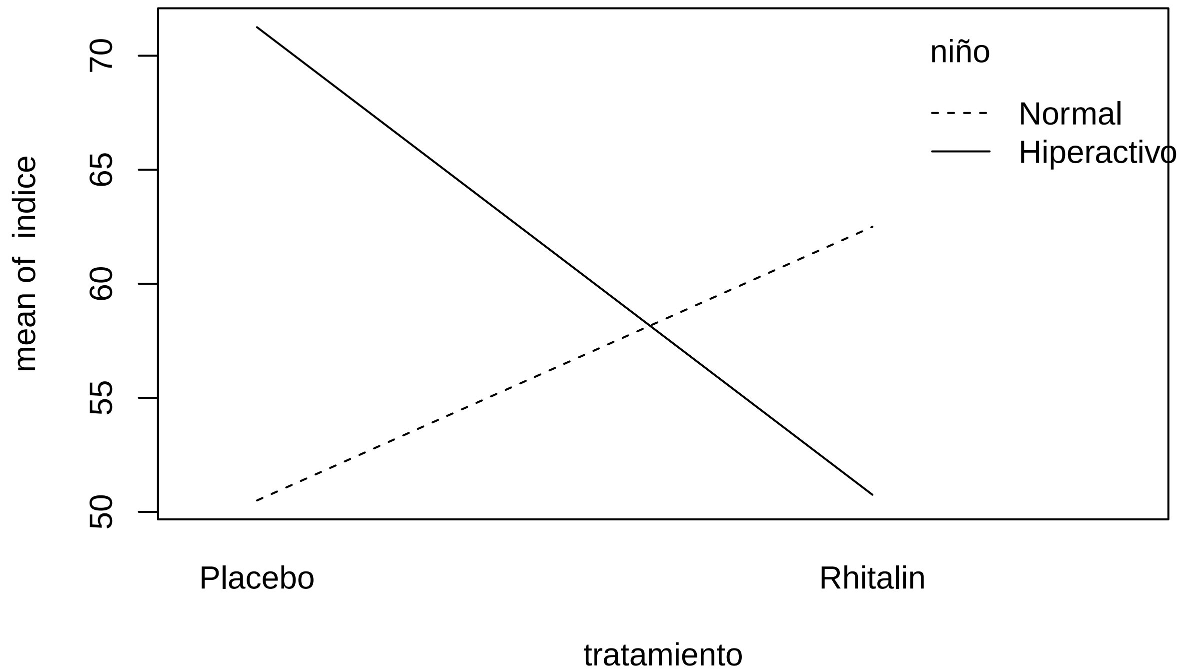


Interacciones

```
interaction.plot(niño, tratamiento, indice)
```




```
interaction.plot(tratamiento, niño, indice)
```



Anova de 2 factores

```
modelo2f = lm(indice ~ niño * tratamiento)
anova(modelo2f)
```

```
## Analysis of Variance Table
##
## Response: indice
##                   Df  Sum Sq Mean Sq F value    Pr(>F)
## niño               1   81.00   81.00  3.1817   0.09976 .
## tratamiento        1   72.25   72.25  2.8380   0.11787
## niño:tratamiento   1 1056.25 1056.25 41.4894 3.202e-05 ***
## Residuals         12  305.50   25.46
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(modelo2f)
```

```
##
## Call:
## lm(formula = indice ~ niño * tratamiento)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.750  -2.688   0.500   3.875   6.250
##
## Coefficients:
##                                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)                           50.500      2.523  20.017 1.38e-10 ***
## niñoHiperactivo                       20.750      3.568   5.816 8.27e-05 ***
## tratamientoRhitalin                   12.000      3.568   3.363  0.00564 **
## niñoHiperactivo:tratamientoRhitalin  -32.500      5.046  -6.441 3.20e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 5.046 on 12 degrees of freedom
## Multiple R-squared:  0.7983, Adjusted R-squared:  0.7479
## F-statistic: 15.84 on 3 and 12 DF,  p-value: 0.0001793
```

Estudiar las hipótesis:

```
shapiro.test(modelo2f$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  modelo2f$residuals
## W = 0.94799, p-value = 0.4586
```

Ejercicio 3

Regresión Lineal Simple.

```
x <- c(18,23,25,35,65,54,34,56,72,
       19,23,42,18,39,37)
#x=Edad
y <-c(202,186,187,180,156,169,174,172,
      153,199,193,174,
      198,183,178)
#y=Máximo de "frecuencia cardíaca"
```

Nube puntos y superponer la recta de mínimos cuadrados.

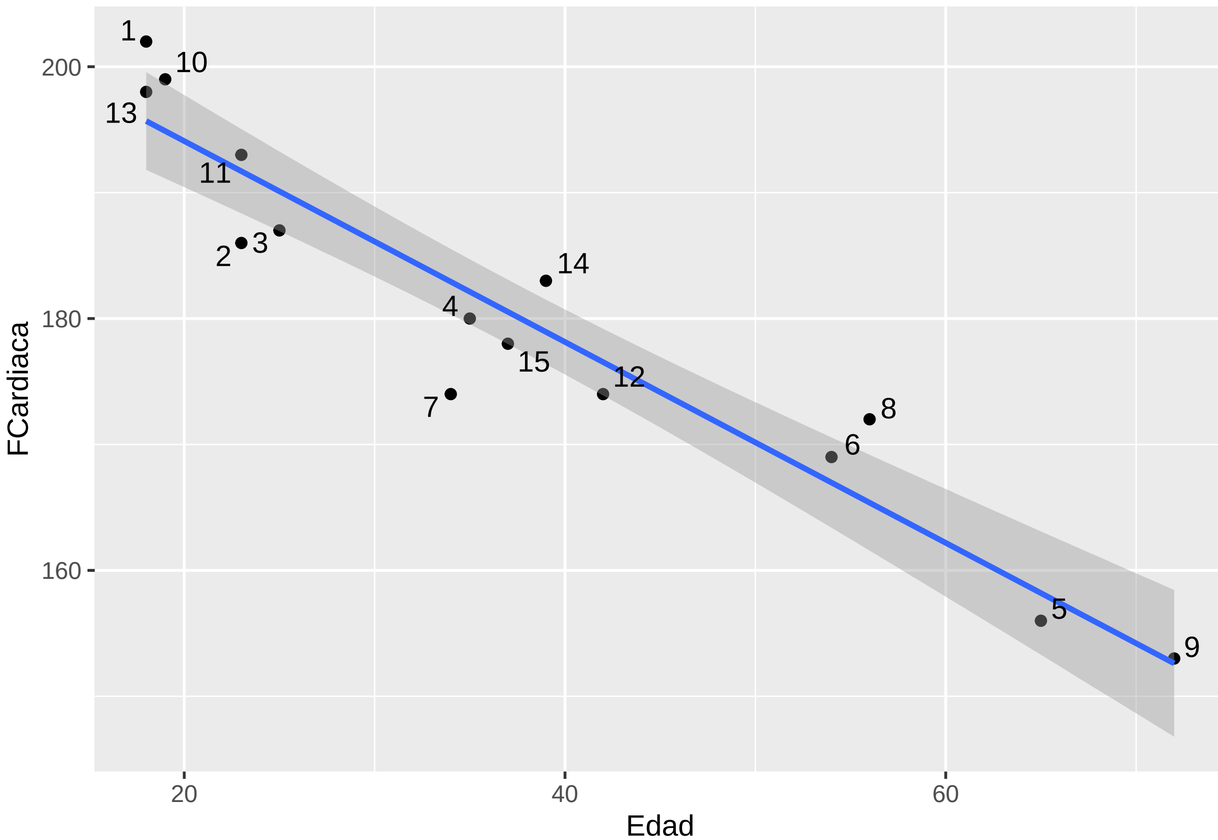
Solución

```
datos3 = tibble(Edad = x, FCardiaca = y)
head(datos3,10)
```

```
## # A tibble: 10 x 2
##     Edad FCardiaca
##    <dbl>     <dbl>
##  1    18       202
##  2    23       186
##  3    25       187
##  4    35       180
##  5    65       156
##  6    54       169
##  7    34       174
##  8    56       172
##  9    72       153
## 10    19       199
```

```
library(ggrepel)
datos3 %>%
  ggplot(aes(x = Edad, y = FCardiaca, label = row.names(.))) +
  geom_point() +
  geom_smooth(method = "lm") +
  geom_text_repel()
```

```
## `geom_smooth()` using formula 'y ~ x'
```

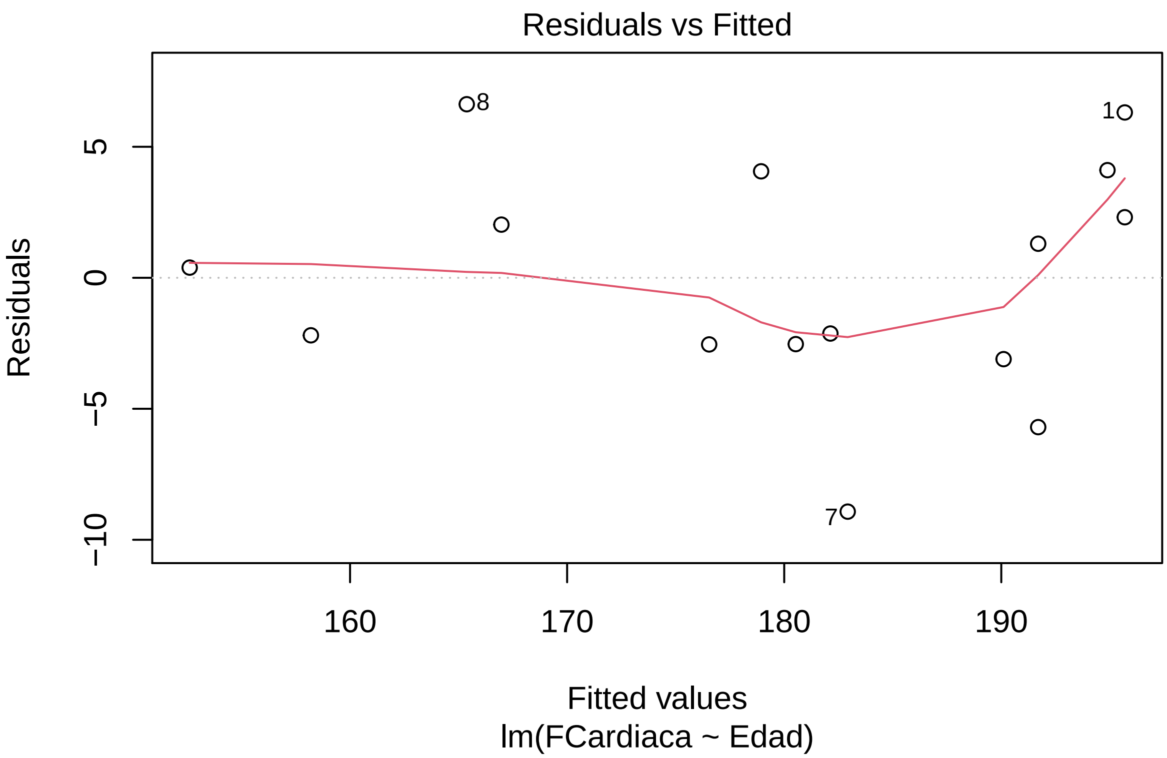


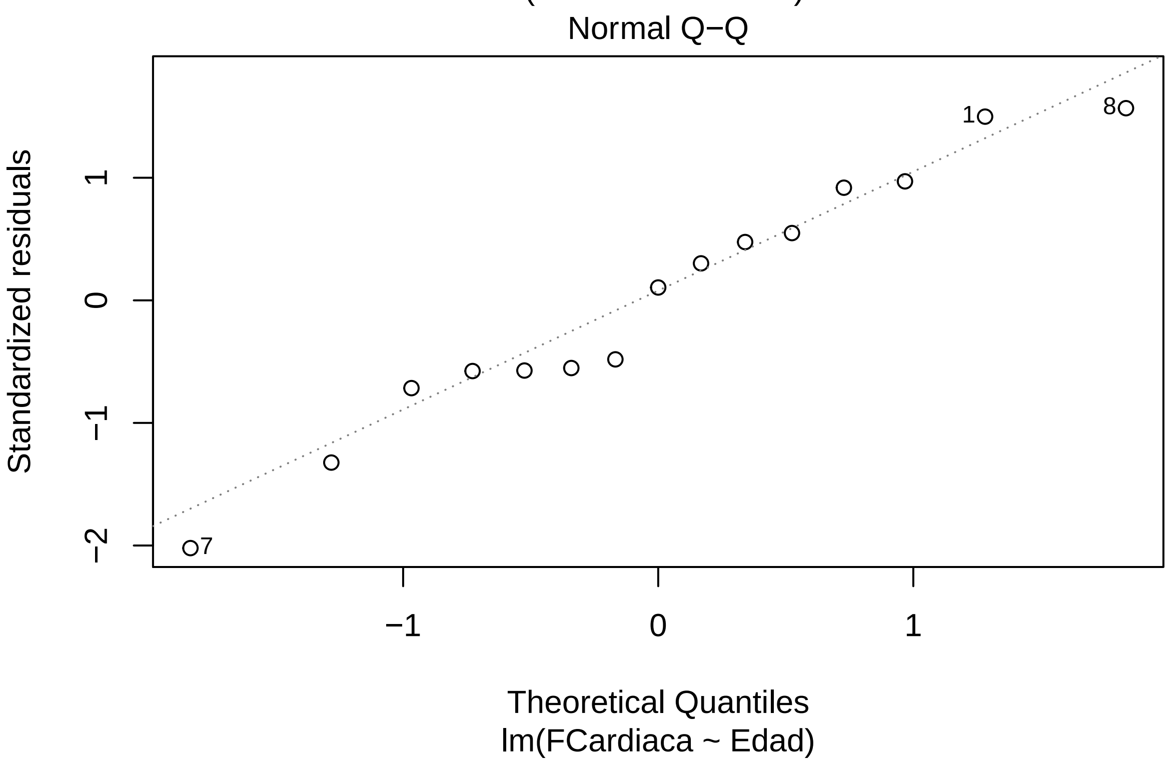

```
modelo = lm(FCardiaca ~ Edad, data = datos3)
summary(modelo)
```

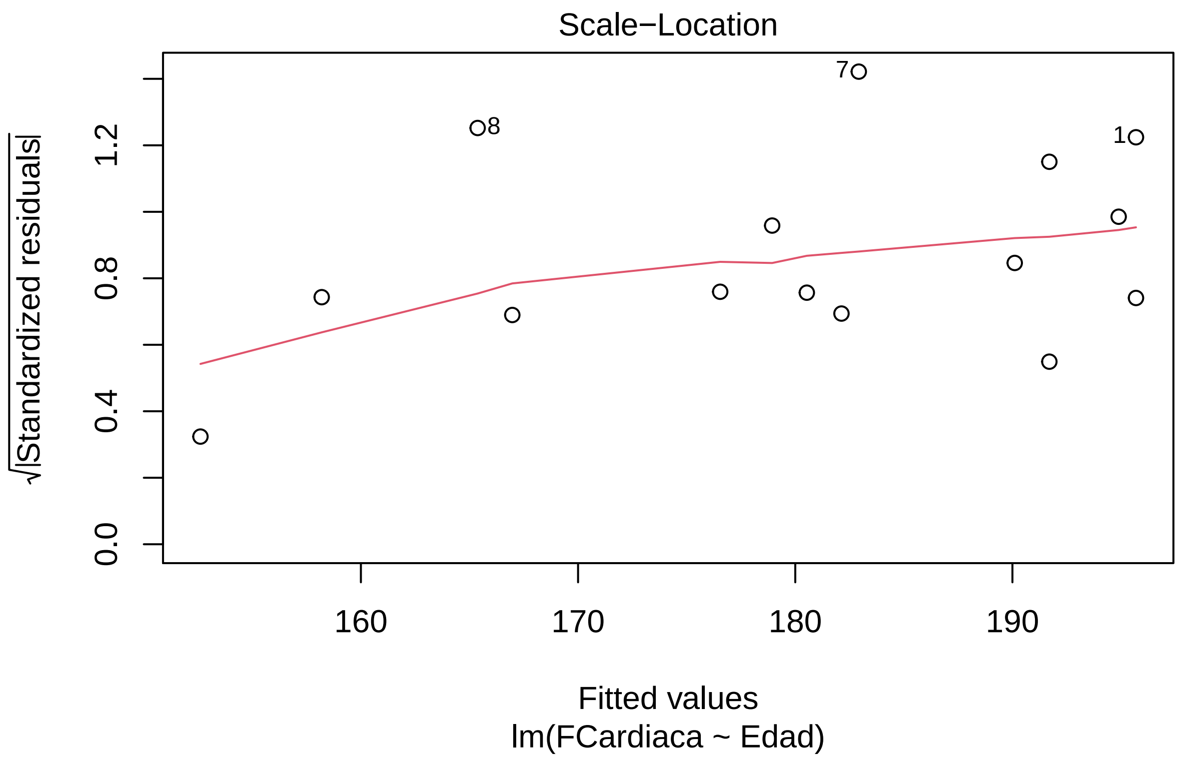
```
##
## Call:
## lm(formula = FCardiaca ~ Edad, data = datos3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.9258 -2.5383  0.3879  3.1867  6.6242
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 210.04846    2.86694   73.27  < 2e-16 ***
## Edad         -0.79773    0.06996  -11.40 3.85e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.578 on 13 degrees of freedom
## Multiple R-squared:  0.9091, Adjusted R-squared:  0.9021
## F-statistic:   130 on 1 and 13 DF,  p-value: 3.848e-08
```

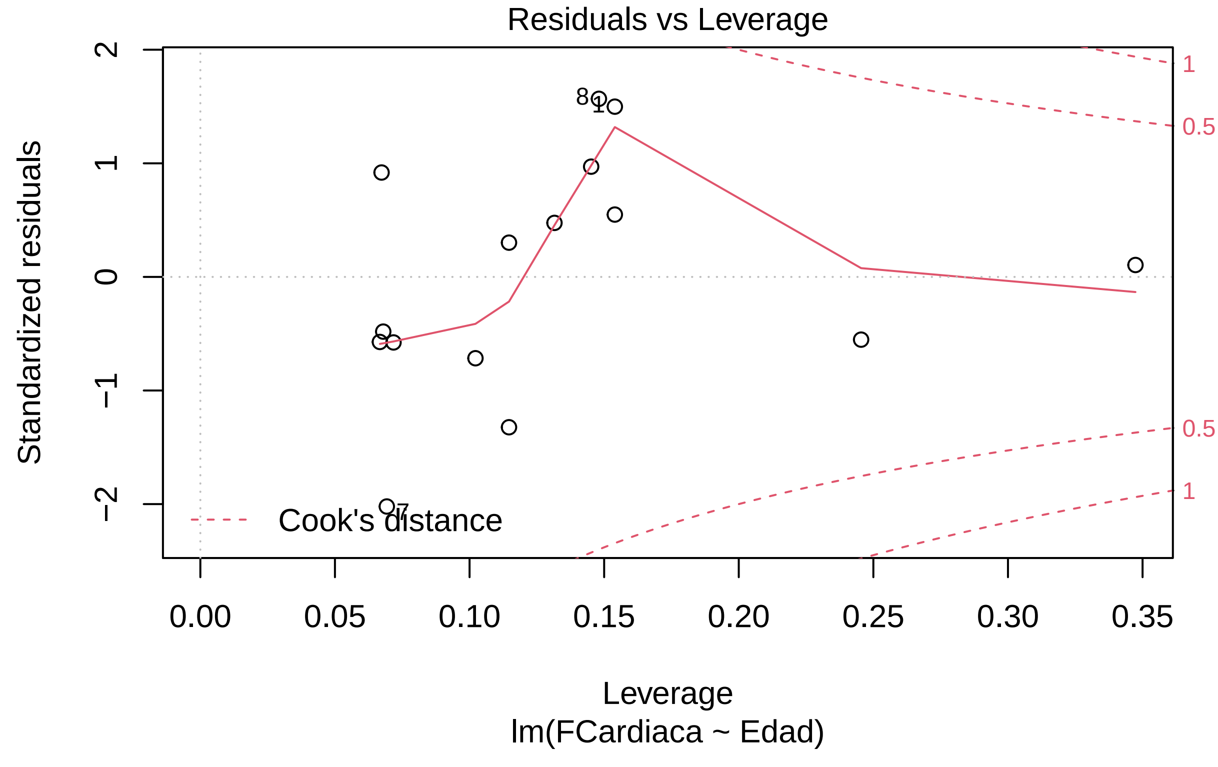
Se puede estudiar el cumplimiento de las hipótesis del modelo de regresión lineal con ayuda

```
plot(modelo)
```


Residuals vs Fitted
Residuals
5
0
−5
−10
160
170
180
190
Fitted values
lm(FCardiaca ~ Edad)
8
1
7


Normal Q−Q
Standardized residuals
1
0
−1
−2
−1
0
1
Theoretical Quantiles
lm(FCardiaca ~ Edad)
1
8
7





Este último gráfico estudia la existencia de posibles outliers, p

Ejercicio 4

Regresión Lineal Múltiple.

```r
library(ISLR) #para acceder a Hitters
data(Hitters)
# ?Hitters
```

Solución

```r
summary(Hitters)
```

```
##      AtBat            Hits         HmRun            Runs       
##  Min.   : 16.0   Min.   :  1   Min.   : 0.00   Min.   :  0.00  
##  1st Qu.:255.2   1st Qu.: 64   1st Qu.: 4.00   1st Qu.: 30.25  
##  Median :379.5   Median : 96   Median : 8.00   Median : 48.00  
##  Mean   :380.9   Mean   :101   Mean   :10.77   Mean   : 50.91  
##  3rd Qu.:512.0   3rd Qu.:137   3rd Qu.:16.00   3rd Qu.: 69.00  
##  Max.   :687.0   Max.   :238   Max.   :40.00   Max.   :130.00  
##                                                                
##       RBI             Walks            Years            CAtBat       
##  Min.   :  0.00   Min.   :  0.00   Min.   : 1.000   Min.   :   19.0  
##  1st Qu.: 28.00   1st Qu.: 22.00   1st Qu.: 4.000   1st Qu.:  816.8  
##  Median : 44.00   Median : 35.00   Median : 6.000   Median : 1928.0  
##  Mean   : 48.03   Mean   : 38.74   Mean   : 7.444   Mean   : 2648.7  
##  3rd Qu.: 64.75   3rd Qu.: 53.00   3rd Qu.:11.000   3rd Qu.: 3924.2  
##  Max.   :121.00   Max.   :105.00   Max.   :24.000   Max.   :14053.0  
##                                                                      
##      CHits            CHmRun           CRuns             CRBI        
##  Min.   :   4.0   Min.   :  0.00   Min.   :   1.0   Min.   :   0.00  
##  1st Qu.: 209.0   1st Qu.: 14.00   1st Qu.: 100.2   1st Qu.:  88.75  
##  Median : 508.0   Median : 37.50   Median : 247.0   Median : 220.50  
##  Mean   : 717.6   Mean   : 69.49   Mean   : 358.8   Mean   : 330.12  
##  3rd Qu.:1059.2   3rd Qu.: 90.00   3rd Qu.: 526.2   3rd Qu.: 426.25  
##  Max.   :4256.0   Max.   :548.00   Max.   :2165.0   Max.   :1659.00  
##                                                                      
##      CWalks          League  Division    PutOuts          Assists     
##  Min.   :   0.00   A:175   E:157    Min.   :   0.0   Min.   :  0.0  
##  1st Qu.:  67.25   N:147   W:165    1st Qu.: 109.2   1st Qu.:  7.0  
##  Median : 170.50                    Median : 212.0   Median : 39.5  
##  Mean   : 260.24                    Mean   : 288.9   Mean   :106.9  
##  3rd Qu.: 339.25                    3rd Qu.: 325.0   3rd Qu.:166.0  
##  Max.   :1566.00                    Max.   :1378.0   Max.   :492.0  
##                                                                     
##      Errors          Salary       NewLeague
##  Min.   : 0.00   Min.   :  67.5   A:176    
##  1st Qu.: 3.00   1st Qu.: 190.0   N:146    
##  Median : 6.00   Median : 425.0            
##  Mean   : 8.04   Mean   : 535.9            
##  3rd Qu.:11.00   3rd Qu.: 750.0            
##  Max.   :32.00   Max.   :2460.0            
##                  NA's   :59
```

```r
dim(Hitters)
```

```
## [1] 322  20
```

```r
str(Hitters)
```

```
## 'data.frame':    322 obs. of  20 variables:
##  $ AtBat    : int  293 315 479 496 321 594 185 298 323 401 ...
##  $ Hits     : int  66 81 130 141 87 169 37 73 81 92 ...
##  $ HmRun    : int  1 7 18 20 10 4 1 0 6 17 ...
##  $ Runs     : int  30 24 66 65 39 74 23 24 26 49 ...
##  $ RBI      : int  29 38 72 78 42 51 8 24 32 66 ...
##  $ Walks    : int  14 39 76 37 30 35 21 7 8 65 ...
##  $ Years    : int  1 14 3 11 2 11 2 3 2 13 ...
##  $ CAtBat   : int  293 3449 1624 5628 396 4408 214 509 341 5206 ...
##  $ CHits    : int  66 835 457 1575 101 1133 42 108 86 1332 ...
##  $ CHmRun   : int  1 69 63 225 12 19 1 0 6 253 ...
##  $ CRuns    : int  30 321 224 828 48 501 30 41 32 784 ...
##  $ CRBI     : int  29 414 266 838 46 336 9 37 34 890 ...
##  $ CWalks   : int  14 375 263 354 33 194 24 12 8 866 ...
##  $ League   : Factor w/ 2 levels "A","N": 1 2 1 2 2 1 2 1 2 1 ...
##  $ Division : Factor w/ 2 levels "E","W": 1 2 2 1 1 2 1 2 2 1 ...
##  $ PutOuts  : int  446 632 880 200 805 282 76 121 143 0 ...
##  $ Assists  : int  33 43 82 11 40 421 127 283 290 0 ...
##  $ Errors   : int  20 10 14 3 4 25 7 9 19 0 ...
##  $ Salary   : num  NA 475 480 500 91.5 750 70 100 75 1100 ...
##  $ NewLeague: Factor w/ 2 levels "A","N": 1 2 1 2 2 1 1 1 2 1 ...
```

Valores NA en el dataset:

```r
length(which(is.na(Hitters)))
```

```
## [1] 59
```

```r
sapply(Hitters,function(x) sum(is.na(x)))
```

```
##     AtBat      Hits     HmRun      Runs       RBI     Walks     Years    CAtBat
##         0         0         0         0         0         0         0         0
##     CHits    CHmRun     CRuns      CRBI    CWalks    League  Division   PutOuts
##         0         0         0         0         0         0         0         0
##   Assists    Errors    Salary NewLeague
##         0         0        59         0
```

Valores NA únicamente en la variable “Salary”.

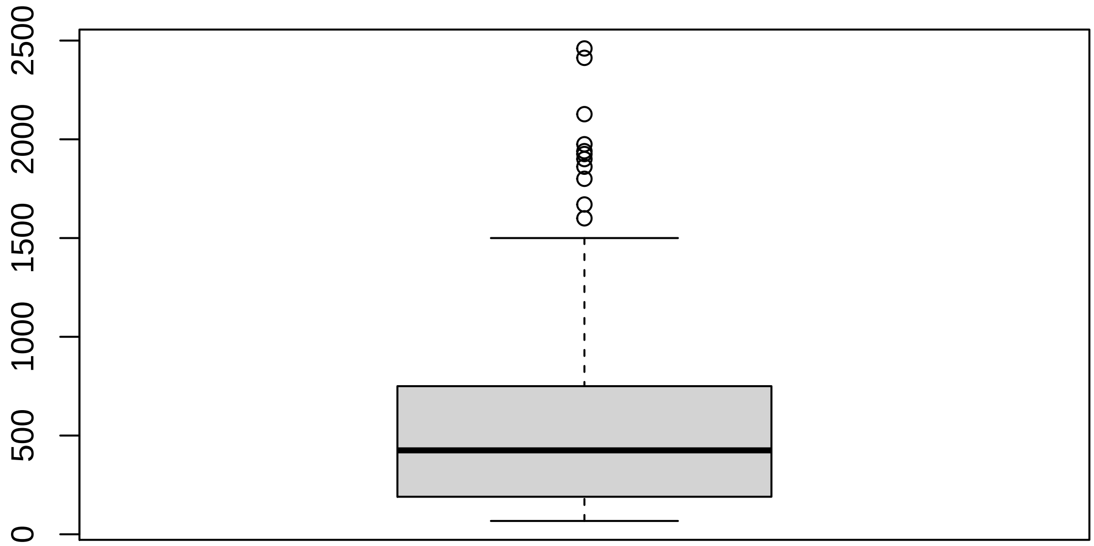
Vamos a trabajar sin las filas que tienen valores NA:

```r
Hitters2 = na.omit(Hitters)
dim(Hitters2)
```

```
## [1] 263  20
```

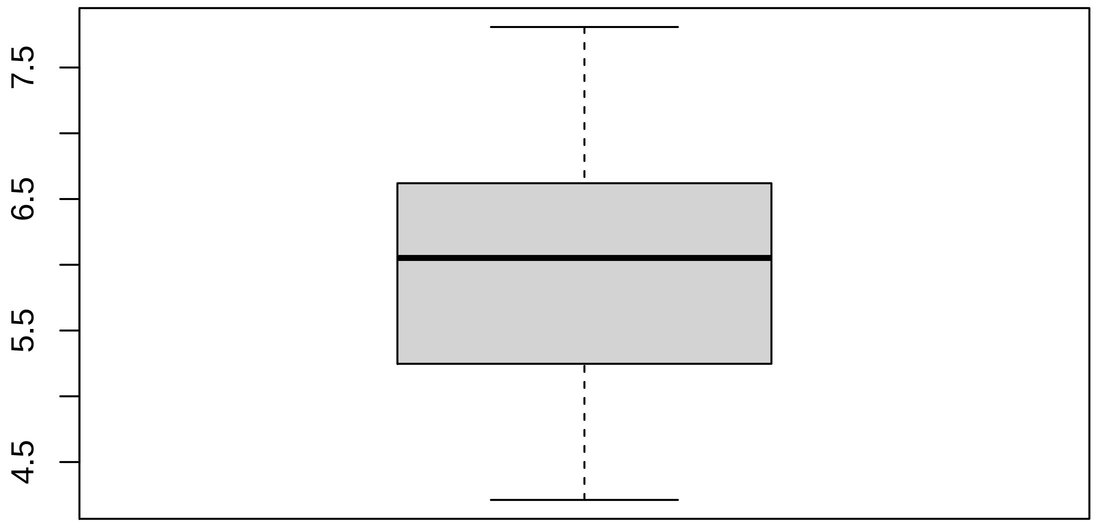
Vamos a intentar realizar un análisis de regresión lineal múltiple sobre la variable “Salary”.

```r
boxplot(Hitters2$Salary)
```



Mejorar la variable "Salary" por considerar una transformación de ella:

```
boxplot(log(Hitters2$Salary))
```



Decidimos trabajar con la variable transformada: `log(Salary)`.

El modelo de regresión lineal múltiple que vamos a considerar es:

```
modeloRLM = lm(data = Hitters2, formula = log(Salary) ~ . )
summary(modeloRLM)
```

```
##
## Call:
## lm(formula = log(Salary) ~ ., data = Hitters2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.22870 -0.45350  0.09424  0.40474  2.77223
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.618e+00  1.765e-01  26.171  < 2e-16 ***
```

```
## AtBat       -2.984e-03  1.232e-03  -2.421  0.01620 *
## Hits         1.308e-02  4.622e-03   2.831  0.00503 **
## HmRun        1.179e-02  1.205e-02   0.978  0.32889
## Runs        -1.419e-03  5.794e-03  -0.245  0.80670
## RBI         -1.675e-03  5.056e-03  -0.331  0.74063
## Walks        1.096e-02  3.554e-03   3.082  0.00229 **
## Years        5.696e-02  2.413e-02   2.361  0.01902 *
## CAtBat       1.283e-04  2.629e-04   0.488  0.62596
## CHits       -4.414e-04  1.311e-03  -0.337  0.73670
## CHmRun      -7.809e-05  3.144e-03  -0.025  0.98020
## CRuns        1.513e-03  1.459e-03   1.037  0.30072
## CRBI         1.312e-04  1.346e-03   0.097  0.92246
## CWalks      -1.466e-03  6.377e-04  -2.298  0.02239 *
## LeagueN      2.825e-01  1.541e-01   1.833  0.06797 .
## DivisionW   -1.656e-01  7.847e-02  -2.111  0.03580 *
## PutOuts      3.389e-04  1.505e-04   2.251  0.02526 *
## Assists      6.214e-04  4.300e-04   1.445  0.14970
## Errors      -1.197e-02  8.537e-03  -1.402  0.16225
## NewLeagueN  -1.742e-01  1.536e-01  -1.134  0.25788
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6135 on 243 degrees of freedom
## Multiple R-squared:  0.5586, Adjusted R-squared:  0.524
## F-statistic: 16.18 on 19 and 243 DF,  p-value: < 2.2e-16
```

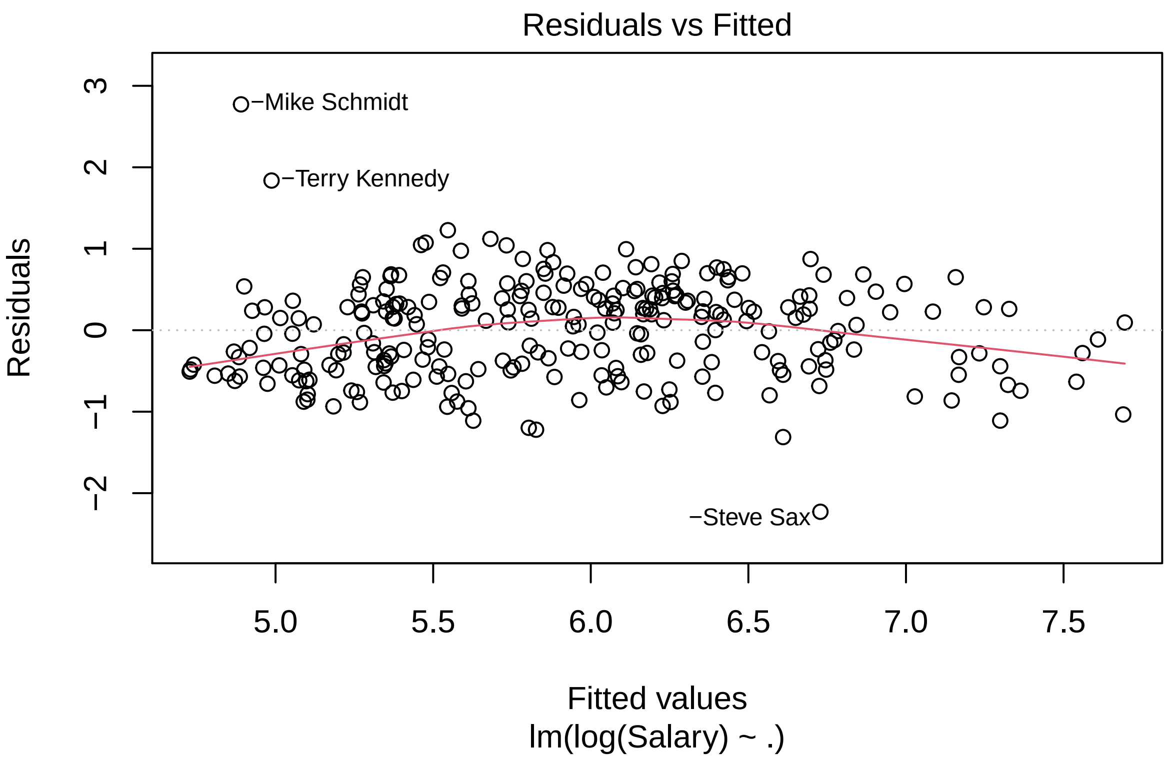
```
modeloRLM2 = lm(data = Hitters2, formula = Salary ~ . )
summary(modeloRLM2)
```

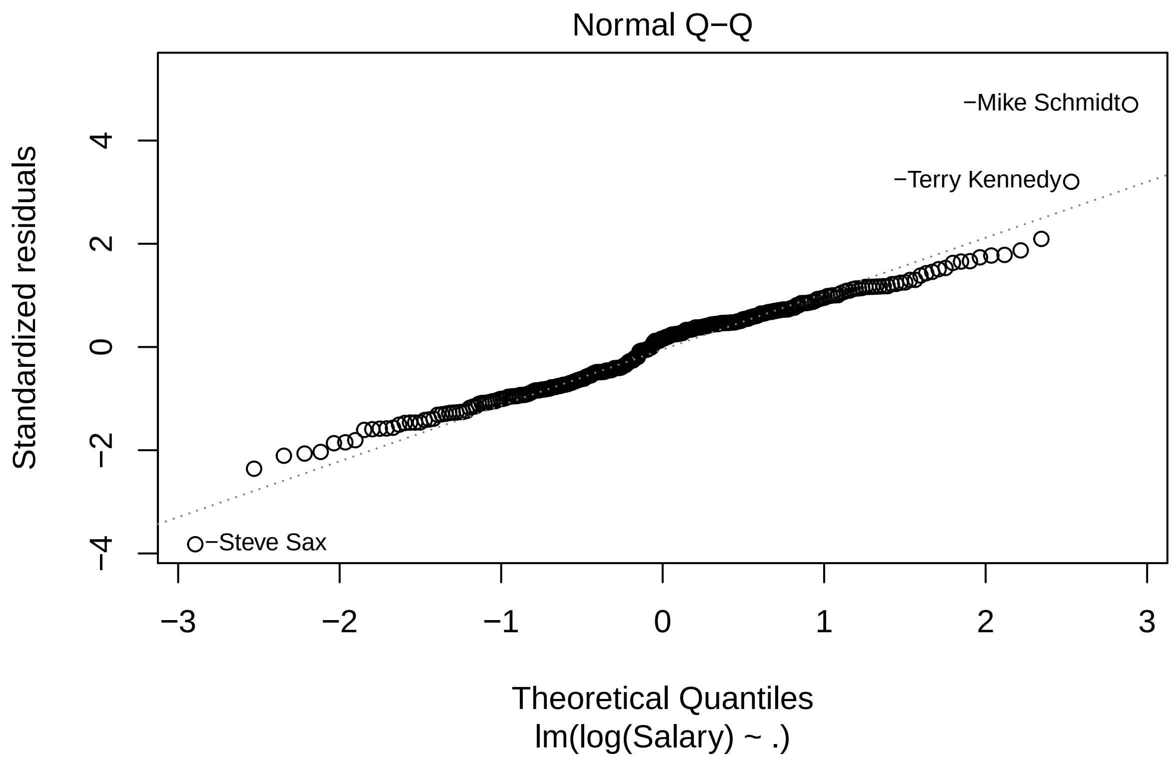
```
##
## Call:
## lm(formula = Salary ~ ., data = Hitters2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -907.62 -178.35  -31.11  139.09 1877.04
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  163.10359   90.77854   1.797 0.073622 .
## AtBat         -1.97987    0.63398  -3.123 0.002008 **
## Hits           7.50077    2.37753   3.155 0.001808 **
## HmRun          4.33088    6.20145   0.698 0.485616
## Runs          -2.37621    2.98076  -0.797 0.426122
## RBI           -1.04496    2.60088  -0.402 0.688204
## Walks          6.23129    1.82850   3.408 0.000766 ***
## Years         -3.48905   12.41219  -0.281 0.778874
## CAtBat        -0.17134    0.13524  -1.267 0.206380
## CHits          0.13399    0.67455   0.199 0.842713
## CHmRun        -0.17286    1.61724  -0.107 0.914967
## CRuns          1.45430    0.75046   1.938 0.053795 .
## CRBI           0.80771    0.69262   1.166 0.244691
## CWalks        -0.81157    0.32808  -2.474 0.014057 *
## LeagueN       62.59942   79.26140   0.790 0.430424
## DivisionW   -116.84925   40.36695  -2.895 0.004141 **
```

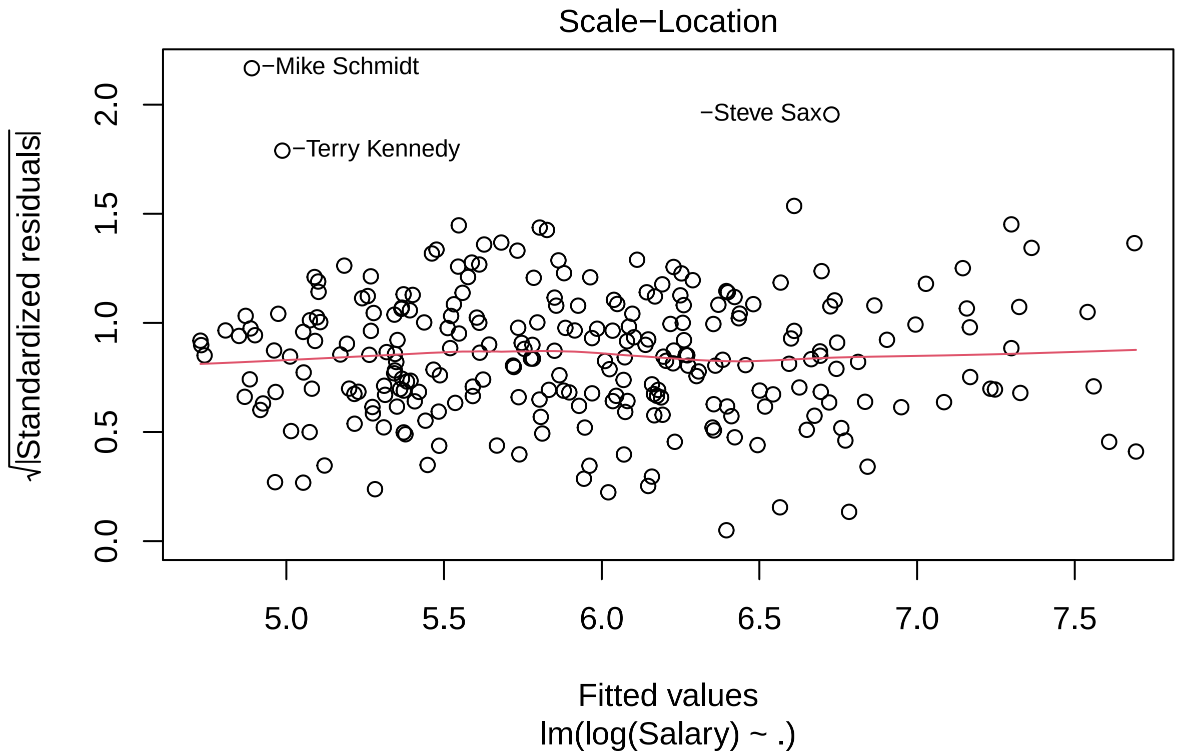
```
## PutOuts         0.28189    0.07744   3.640 0.000333 ***
## Assists         0.37107    0.22120   1.678 0.094723 .
## Errors         -3.36076    4.39163  -0.765 0.444857
## NewLeagueN    -24.76233   79.00263  -0.313 0.754218
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 315.6 on 243 degrees of freedom
## Multiple R-squared:  0.5461, Adjusted R-squared:  0.5106
## F-statistic: 15.39 on 19 and 243 DF,  p-value: < 2.2e-16
```

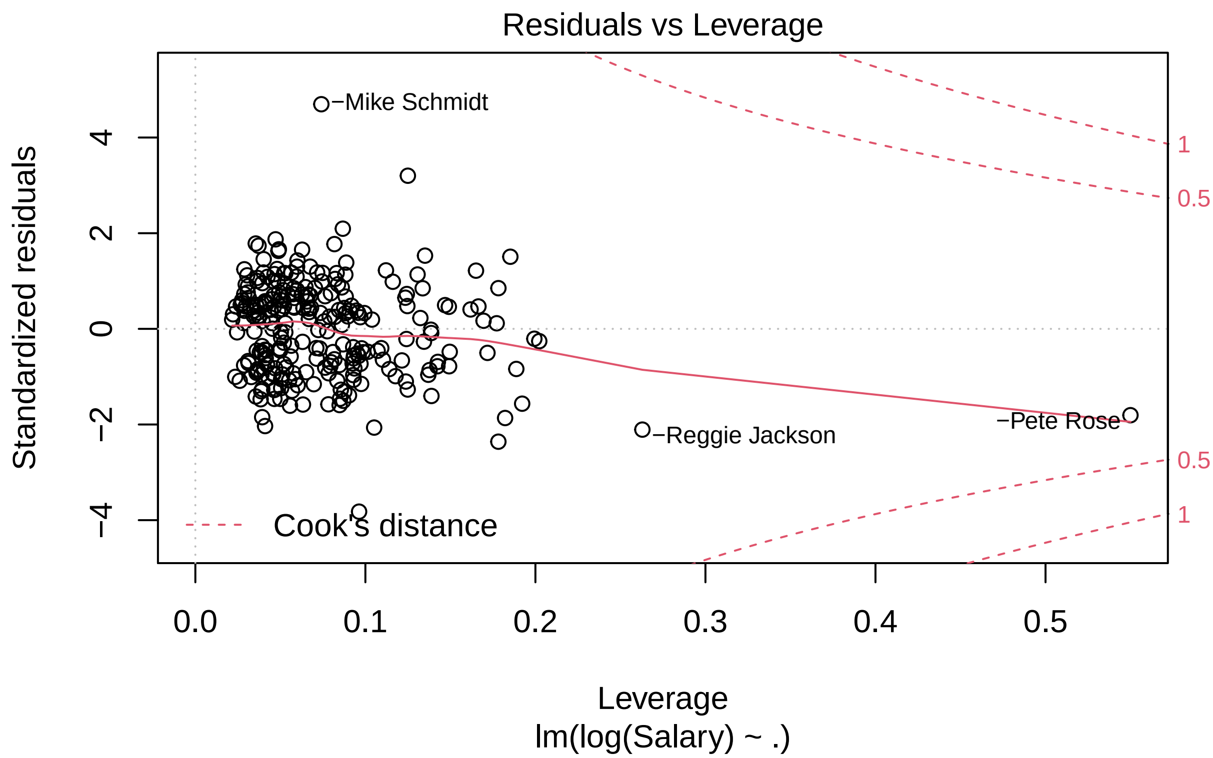
Para estudiar la validez de las hipótesis:

```
plot(modeloRLM)
```




Normal Q−Q
Standardized residuals
−Mike Schmidt
−Terry Kennedy
−Steve Sax
−4
−2
0
2
4
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(log(Salary) ~ .)


Scale−Location
√|Standardized residuals|
−Mike Schmidt
−Steve Sax
−Terry Kennedy
0.0
0.5
1.0
1.5
2.0
5.0
5.5
6.0
6.5
7.0
7.5
Fitted values
lm(log(Salary) ~ .)



Estudiamos en los datos los valores reseñables ("outliers"):

```
cuales = c("-Mike Schmidt", "-Terry Kennedy", "-Steve Sax")
Hitters2_red = Hitters2[rownames(Hitters2) %in% cuales,]
```

Mejora la Regresión sin esos individuos:

```
Hitters3 = Hitters2[!(rownames(Hitters2) %in% cuales),]
dim(Hitters3)
```

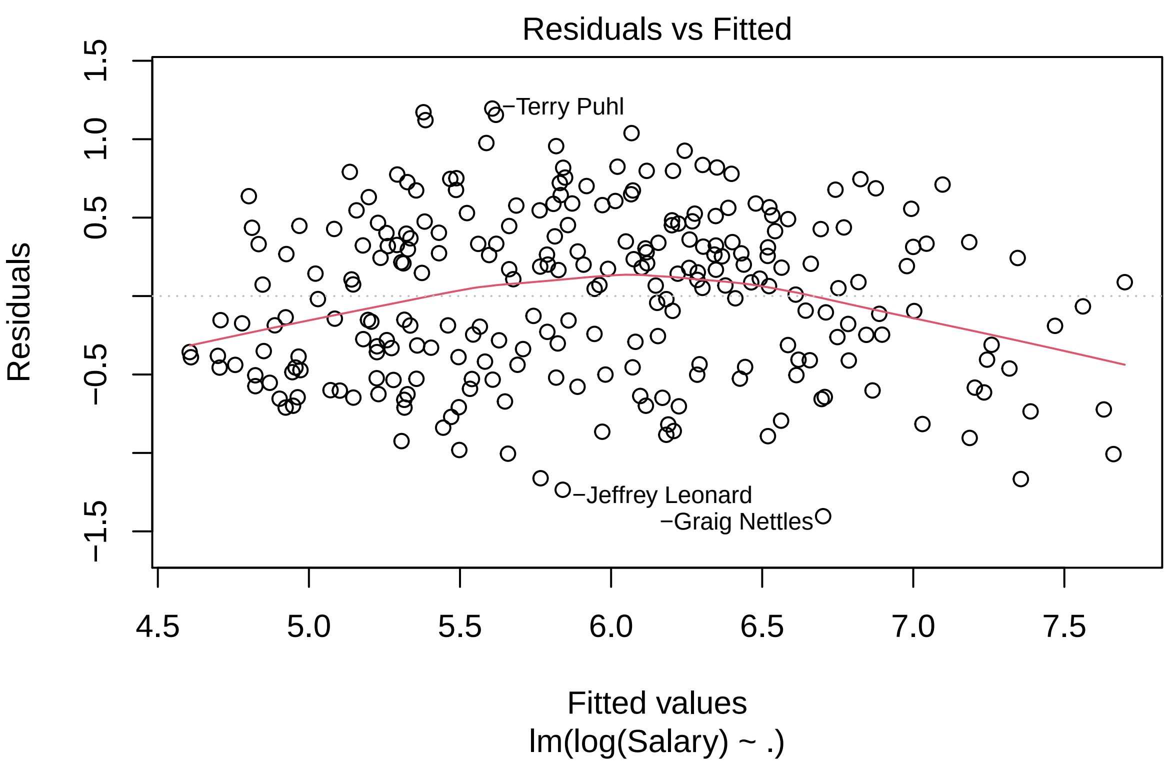
```
## [1] 260  20
```

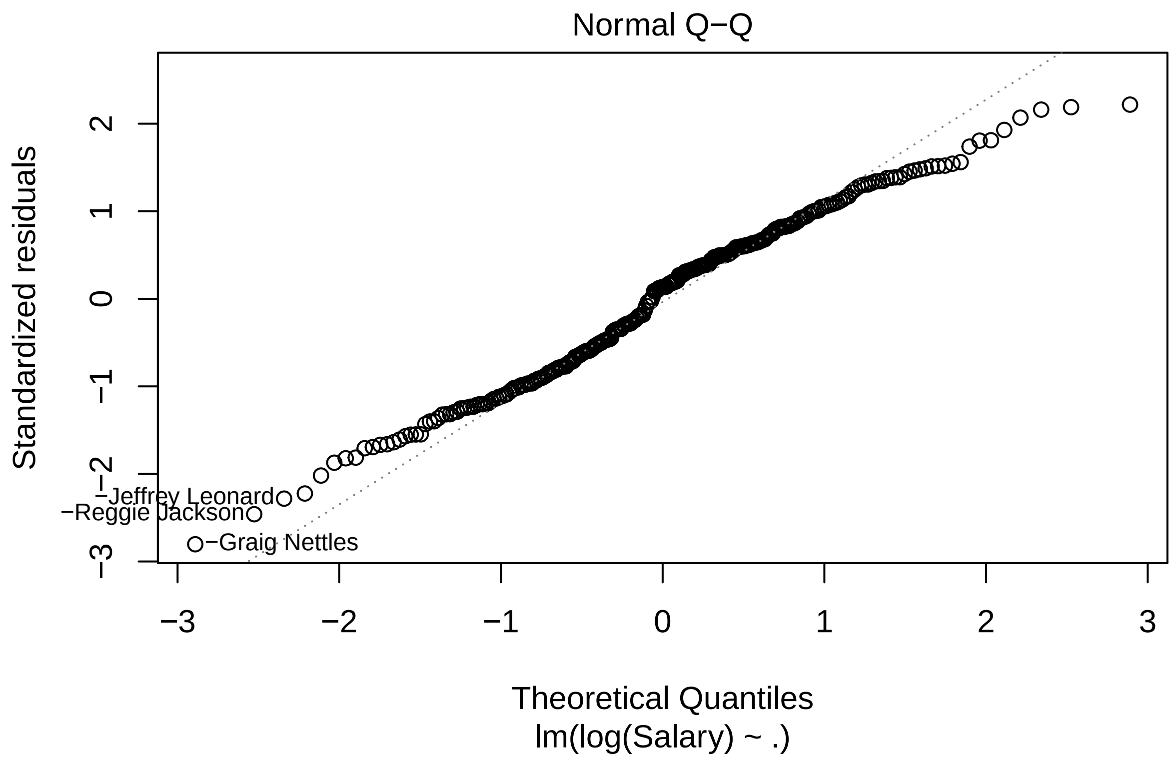
```
modeloRLM3 = lm(data = Hitters3, formula = log(Salary) ~ . )
summary(modeloRLM3)
```

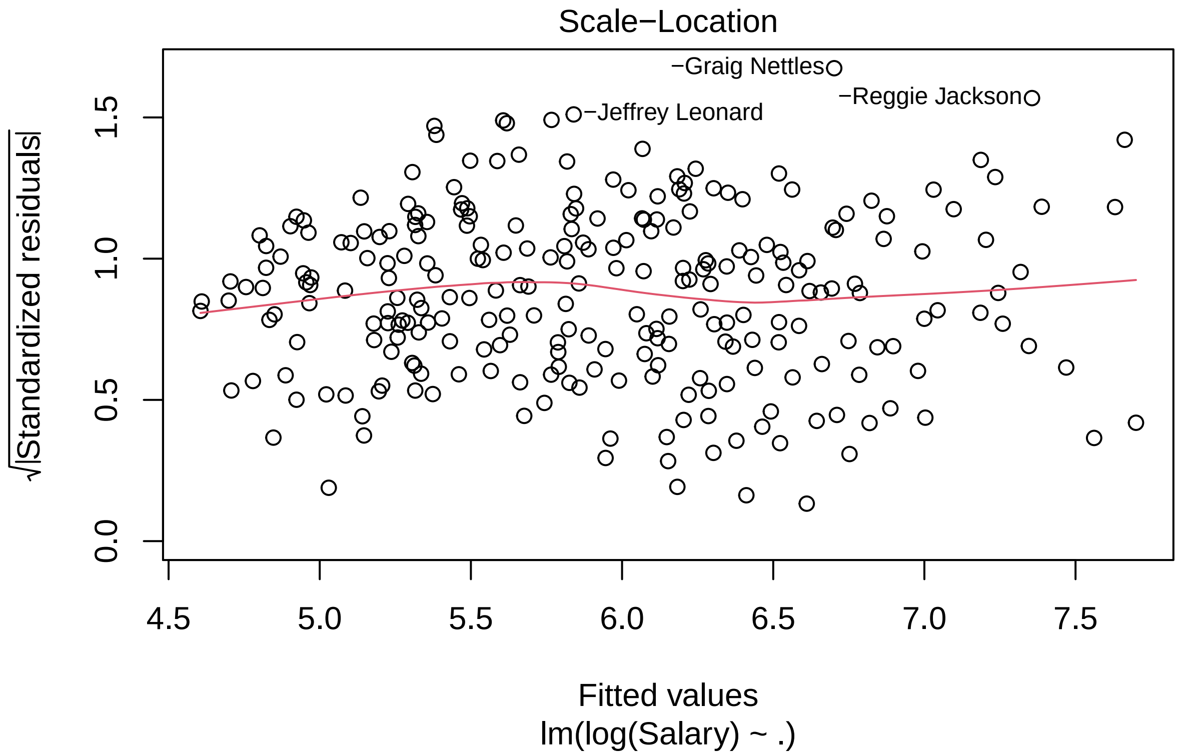
```
##
## Call:
## lm(formula = log(Salary) ~ ., data = Hitters3)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.40343 -0.43821  0.06915  0.40222  1.19551
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.3086377  0.1649525  26.120  < 2e-16 ***
## AtBat       -0.0027899  0.0011363  -2.455 0.014787 *
## Hits         0.0163153  0.0042537   3.836 0.000160 ***
## HmRun        0.0106397  0.0108651   0.979 0.328439
## Runs        -0.0038013  0.0052288  -0.727 0.467937
## RBI         -0.0026616  0.0045590  -0.584 0.559890
## Walks        0.0118608  0.0032036   3.702 0.000265 ***
```

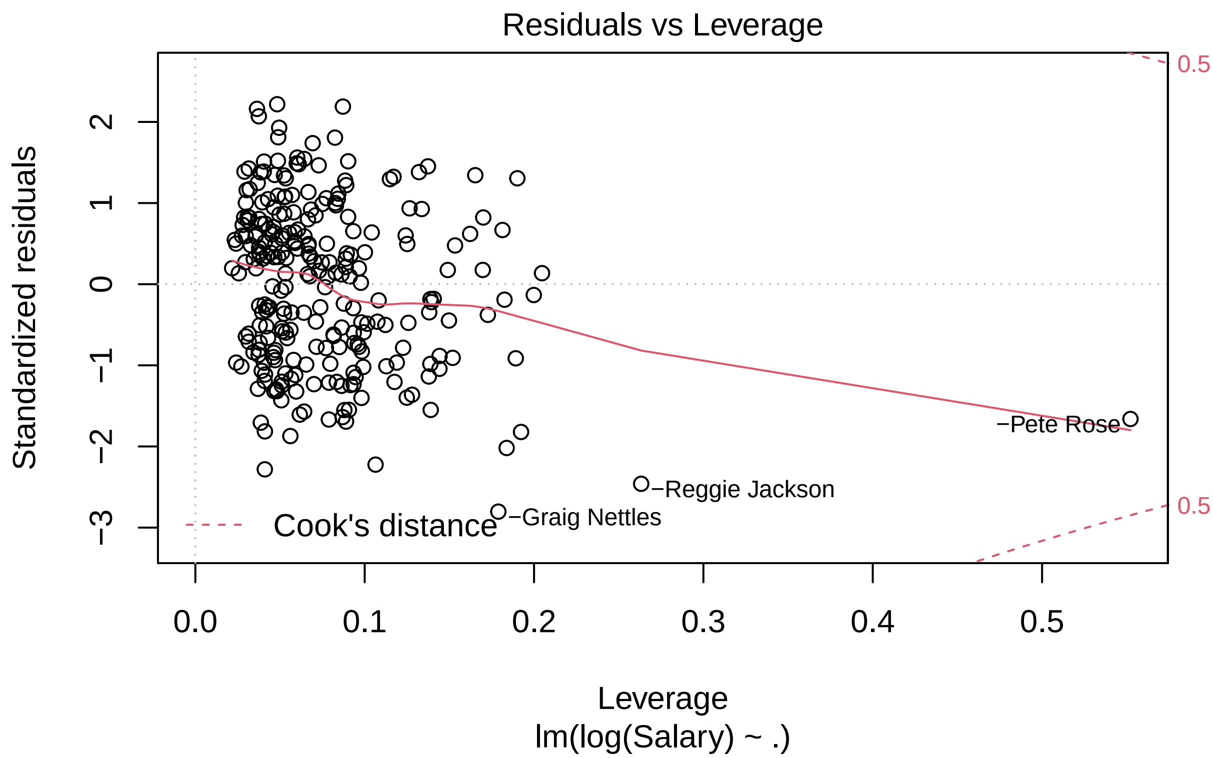
```
## Years        0.0691209  0.0218739   3.160 0.001781 **
## CAtBat       0.0002366  0.0002374   0.997 0.319989
## CHits       -0.0007743  0.0011820  -0.655 0.513039
## CHmRun       0.0001581  0.0028354   0.056 0.955588
## CRuns        0.0014210  0.0013143   1.081 0.280701
## CRBI        -0.0001596  0.0012148  -0.131 0.895611
## CWalks      -0.0013707  0.0005747  -2.385 0.017847 *
## LeagueN      0.1965060  0.1416566   1.387 0.166668
## DivisionW   -0.1328851  0.0712302  -1.866 0.063321 .
## PutOuts      0.0002470  0.0001375   1.797 0.073661 .
## Assists      0.0003085  0.0003939   0.783 0.434308
## Errors      -0.0090674  0.0077114  -1.176 0.240823
## NewLeagueN  -0.0796926  0.1416948  -0.562 0.574352
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5525 on 240 degrees of freedom
## Multiple R-squared:  0.636,  Adjusted R-squared:  0.6072
## F-statistic: 22.07 on 19 and 240 DF,  p-value: < 2.2e-16
```

```
plot(modeloRLM3)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(log(Salary) ~ .)
−Jeffrey Leonard
−Reggie Jackson
−Graig Nettles


Scale−Location
√|Standardized residuals|
Fitted values
lm(log(Salary) ~ .)
−Graig Nettles
−Reggie Jackson
−Jeffrey Leonard



Ejercicio 5

Regresión Lineal Múltiple.

El fichero de datos "Advertising.csv"" contiene las ventas de un producto en 200 mercados diferentes junto con los presupuestos de publicidad en cada mercado en tres medios: televisión, radio y prensa.

El objetivo es construir un modelo de regresión lineal múltiple para predecir las ventas del producto en función de los gastos en publicidad.

Variables en el archivo: Caso; TV, Radio, Prensa (miles de dólares) y las Ventas (miles de unidades).

Apartado a

Incrementar en mil dólares el gasto publicitario en TV conlleva, por término medio, aumentar en 0.046*1000= 46 unidades las ventas del producto (suponiendo que el gasto publicitario en Radio no cambia).

Si se incrementa en mil dólares el gasto publicitario en la Radio, cabe esperar que las ventas aumenten en 188 unidades, suponiendo fijo el gasto en TV.

Los gastos publicitarios en TV y Radio explican el 89.72 % de la varianza de las Ventas del producto mediante este modelo.

Solución

```
datos = read.csv("Advertising.csv",header = TRUE, sep = ",")
head(datos)
```

```
##   Caso    TV Radio Prensa Ventas
## 1    1 230.1  37.8   69.2   22.1
## 2    2  44.5  39.3   45.1   10.4
## 3    3  17.2  45.9   69.3    9.3
```

```
## 4    4 151.5  41.3   58.5   18.5
## 5    5 180.8  10.8   58.4   12.9
## 6    6   8.7  48.9   75.0    7.2
```

```
dim(datos)
```

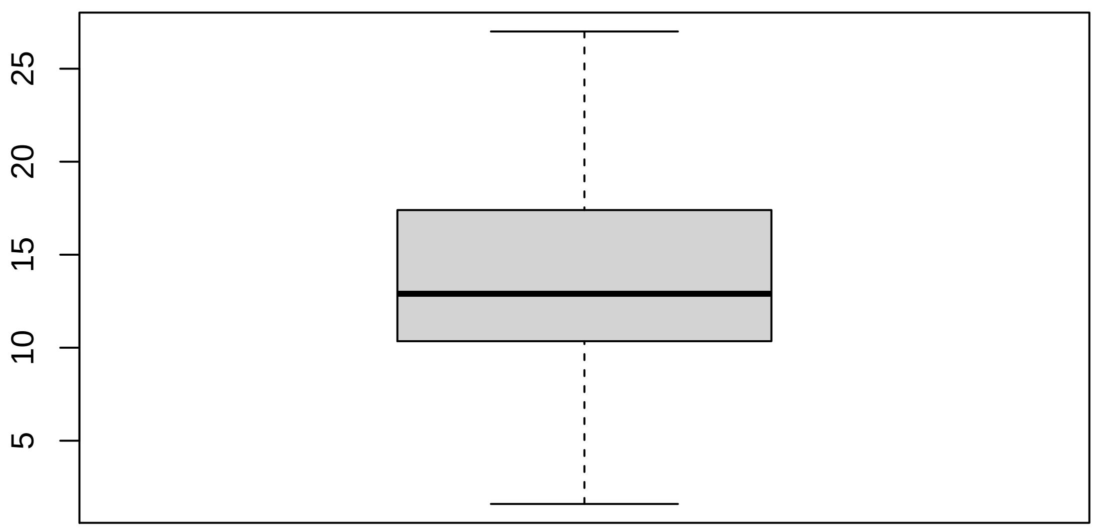
```
## [1] 200   5
```

```
summary(datos)
```

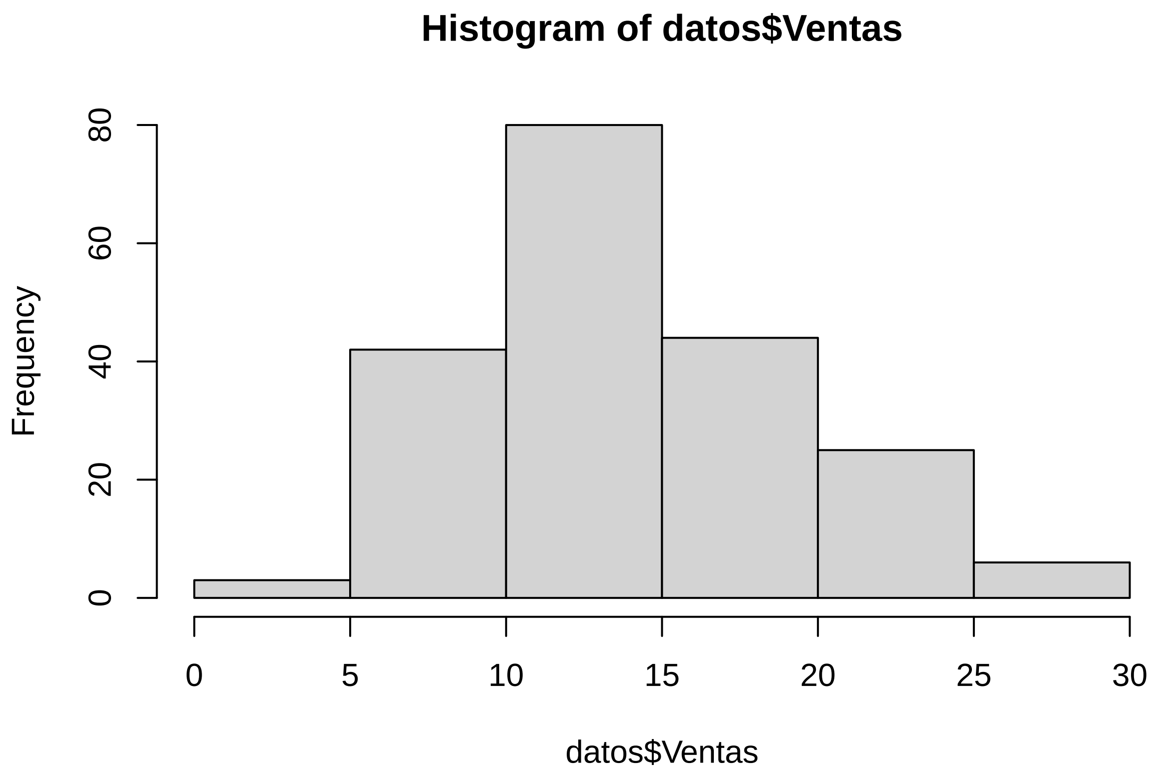
```
##       Caso              TV              Radio            Prensa      
##  Min.   :  1.00   Min.   :  0.70   Min.   : 0.000   Min.   :  0.30  
##  1st Qu.: 50.75   1st Qu.: 74.38   1st Qu.: 9.975   1st Qu.: 12.75  
##  Median :100.50   Median :149.75   Median :22.900   Median : 25.75  
##  Mean   :100.50   Mean   :147.04   Mean   :23.264   Mean   : 30.55  
##  3rd Qu.:150.25   3rd Qu.:218.82   3rd Qu.:36.525   3rd Qu.: 45.10  
##  Max.   :200.00   Max.   :296.40   Max.   :49.600   Max.   :114.00  
##      Ventas     
##  Min.   : 1.60  
##  1st Qu.:10.38  
##  Median :12.90  
##  Mean   :14.02  
##  3rd Qu.:17.40  
##  Max.   :27.00
```

En este dataset no existen valores faltantes o NA.

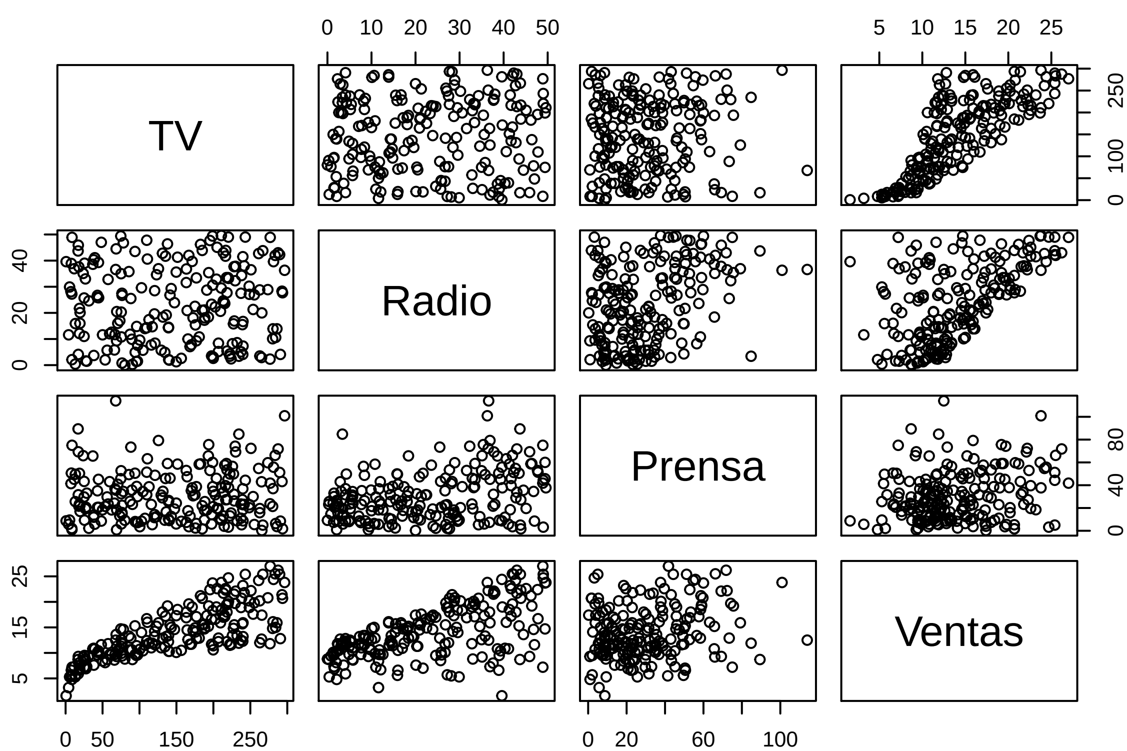
```
boxplot(datos$Ventas)
```




```
hist(datos$Ventas)
```




```
plot(datos[,-1])
```




```
cor(datos[,-1])
```

```
##                 TV      Radio     Prensa    Ventas
## TV      1.00000000 0.05480866 0.05664787 0.7822244
```

```
## Radio  0.05480866 1.00000000 0.35410375 0.5762226
## Prensa 0.05664787 0.35410375 1.00000000 0.2282990
## Ventas 0.78222442 0.57622257 0.22829903 1.0000000
```

No es necesario utilizar transformaciones de “Ventas”.

El modelo de regresión lineal múltiple que consideramos es

```
resMRL = lm(data = datos, formula = Ventas ~ TV+Radio+Prensa )
summary(resMRL)
```

```
##
## Call:
## lm(formula = Ventas ~ TV + Radio + Prensa, data = datos)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.8277 -0.8908  0.2418  1.1893  2.8292
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.938889   0.311908   9.422   <2e-16 ***
## TV           0.045765   0.001395  32.809   <2e-16 ***
## Radio        0.188530   0.008611  21.893   <2e-16 ***
## Prensa      -0.001037   0.005871  -0.177     0.86
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.686 on 196 degrees of freedom
## Multiple R-squared:  0.8972, Adjusted R-squared:  0.8956
## F-statistic: 570.3 on 3 and 196 DF,  p-value: < 2.2e-16
```

```
resMRL2 = lm(data = datos, formula = Ventas ~ TV+Radio )
summary(resMRL2)
```

```
##
## Call:
## lm(formula = Ventas ~ TV + Radio, data = datos)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.7977 -0.8752  0.2422  1.1708  2.8328
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.92110    0.29449   9.919   <2e-16 ***
## TV           0.04575    0.00139  32.909   <2e-16 ***
## Radio        0.18799    0.00804  23.382   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.681 on 197 degrees of freedom
## Multiple R-squared:  0.8972, Adjusted R-squared:  0.8962
## F-statistic: 859.6 on 2 and 197 DF,  p-value: < 2.2e-16
```

Para comparar modelos lineales no es conveniente utilizar el R^2. El R^2 ajustado es mejor ya que tiene en cuenta el número de variables predictoras.

Pero aún así, se recomienda utilizar los criterios de información:

- AIC (de Akaike)
- BIC (Bayesiano)

```
AIC(resMRL)
```

```
## [1] 782.3622
```

```
AIC(resMRL2)
```

```
## [1] 780.3941
```

```
BIC(resMRL)
```

```
## [1] 798.8538
```

```
BIC(resMRL2)
```

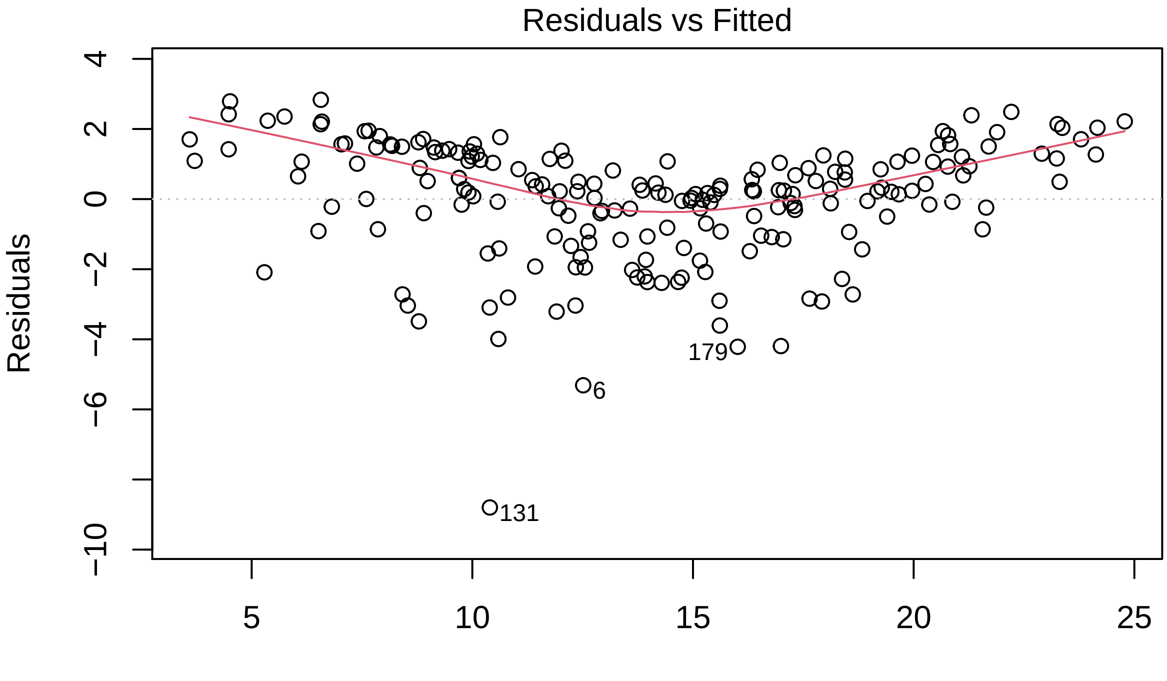
```
## [1] 793.5874
```

Se observa que el modelo "resMRL2" se comporta mejor según los dos criterios, al presentar un valor menor.

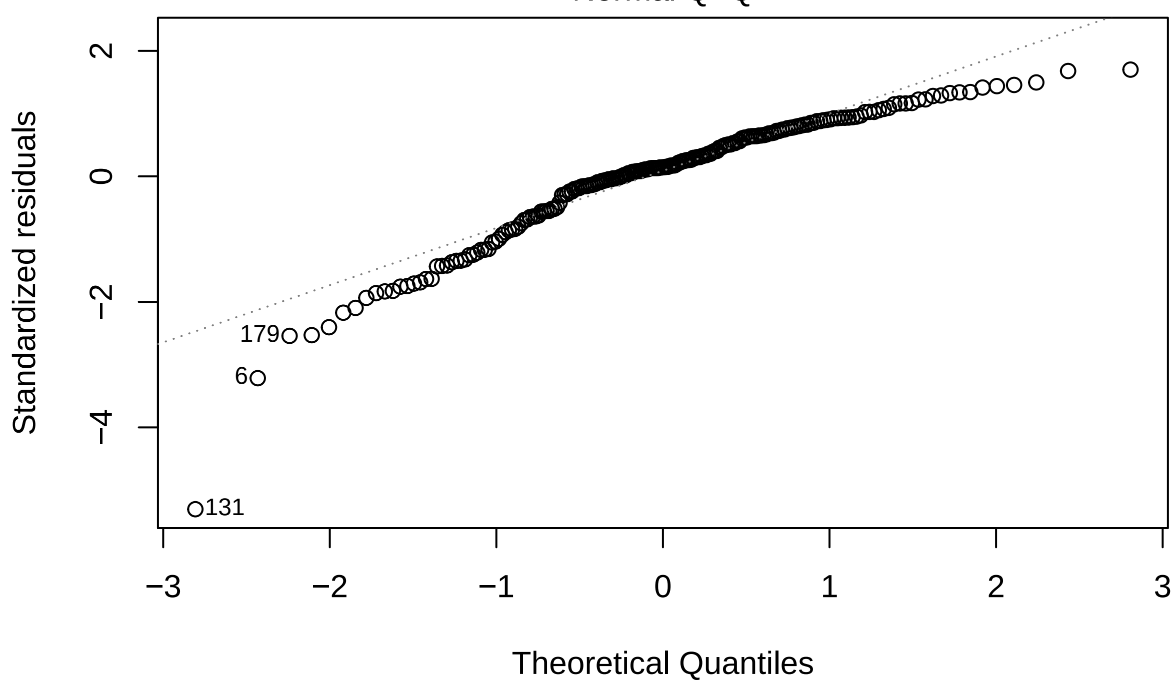
```
summary(resMRL2)
```

```
##
## Call:
## lm(formula = Ventas ~ TV + Radio, data = datos)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.7977 -0.8752  0.2422  1.1708  2.8328
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.92110    0.29449   9.919   <2e-16 ***
## TV           0.04575    0.00139  32.909   <2e-16 ***
## Radio        0.18799    0.00804  23.382   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.681 on 197 degrees of freedom
## Multiple R-squared:  0.8972, Adjusted R-squared:  0.8962
## F-statistic: 859.6 on 2 and 197 DF,  p-value: < 2.2e-16
```

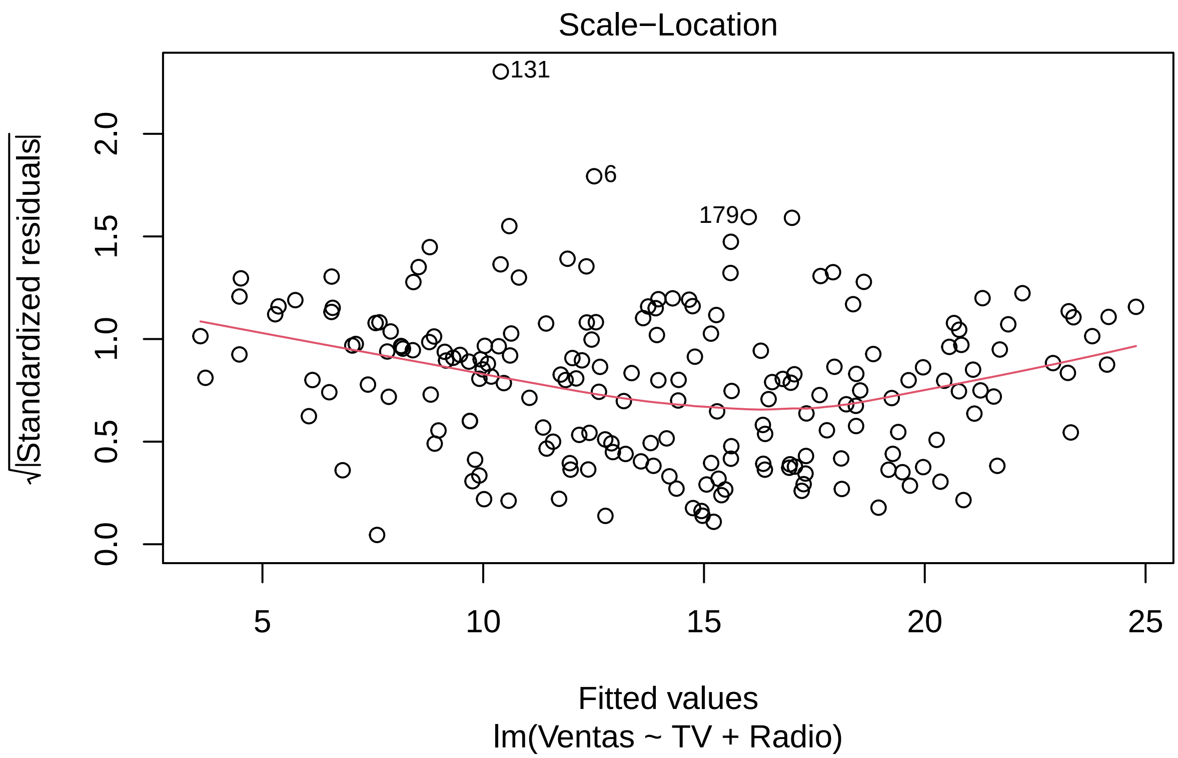
```
plot(resMRL2)
```

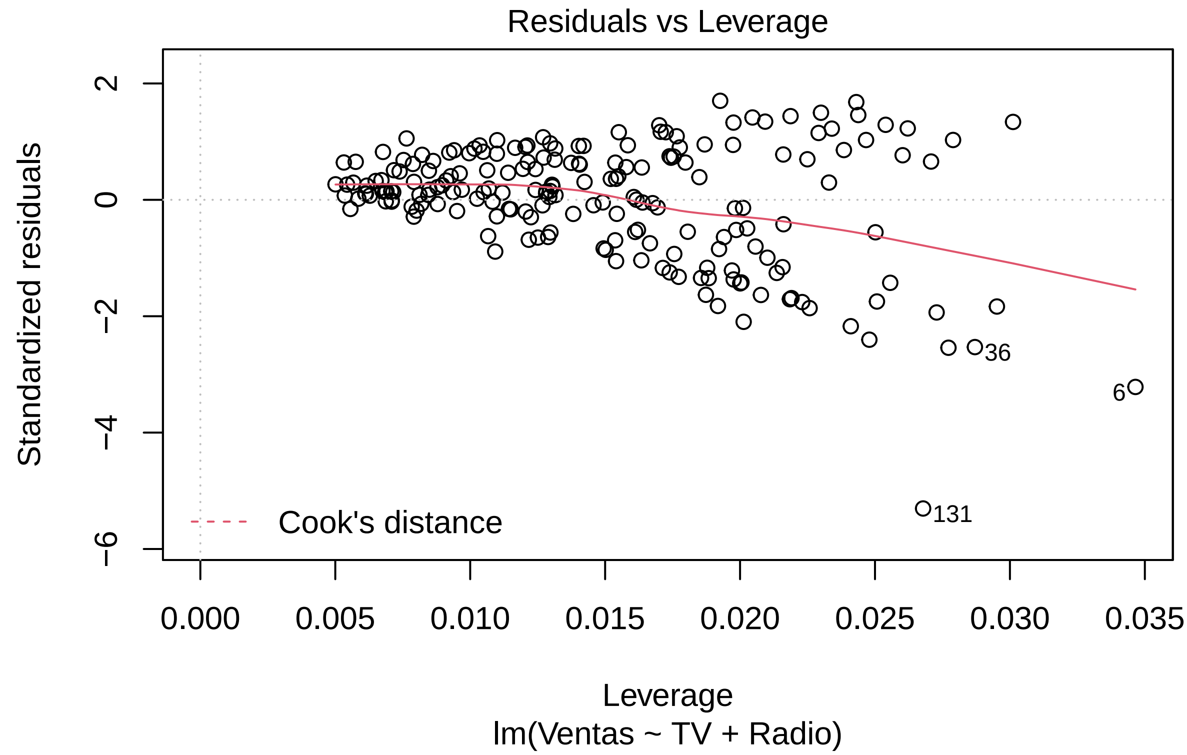


lm(Ventas ~ TV + Radio)



lm(Ventas ~ TV + Radio)





Se pueden obtener intervalos de confianza para los coeficientes de regresión:

```
confint(resMRL2)
```

```
##                   2.5 %     97.5 %
## (Intercept) 2.34034299 3.50185683
## TV          0.04301292 0.04849671
```

```
## Radio       0.17213877 0.20384969
```

Para realizar predicciones

$$Ventas = Intercept +$$

```r
predict(resMRL2)
```

```
##         1         2         3         4         5         6         7         8
## 20.555465 12.345362 12.337018 17.617116 13.223908 12.512084 11.718212 12.105516
##         9        10        11        12        13        14        15        16
##  3.709379 12.551697  7.035860 17.256520 10.608662  8.810951 18.444668 20.828915
##        17        18        19        20        21        22        23        24
## 12.903865 23.241076  9.941215 14.153846 18.121392 14.742064  6.514172 16.544027
##        25        26        27        28        29        30        31        32
##  8.140352 15.608021 14.967694 17.046335 19.399541  9.159297 21.642922 11.357918
##        33        34        35        36        37        38        39        40
##  7.650459 18.833463  7.563028 16.992801 23.367207 15.625899  9.912578 20.440580
##        41        42        43        44        45        46        47        48
## 16.378721 17.298709 21.562154 13.966923  8.900997 15.162638  8.886450 21.699440
##        49        50        51        52        53        54        55        56
## 16.286903  8.181629 12.645694  9.319628 20.661801 19.961262 20.355124 21.308647
##        57        58        59        60        61        62        63        64
##  8.537748 12.762395 21.890729 18.107469  5.744971 22.904187 16.784138 13.184749
##        65        66        67        68        69        70        71        72
## 16.965709  7.826528  8.987035 12.020662 18.953134 21.093690 17.783507 10.633296
##        73        74        75        76        77        78        79        80
## 10.351138  9.913340 17.309835 11.909704  4.480148 13.792391  8.789203  9.676214
##        81        82        83        84        85        86        87        88
## 11.436214 14.663881 10.182720 14.416472 20.773505 15.220024 11.582034 15.618724
##        89        90        91        92        93        94        95        96
## 11.755103 16.931103  9.987143  4.511679 19.179730 21.262772 10.467086 16.333479
##        97        98        99       100       101       102       103       104
## 12.620231 15.329044 24.128426 16.946510 13.905346 23.307018 17.640341 14.751930
##       105       106       107       108       109       110       111       112
## 20.268099 17.953621  6.132907  7.113733  3.595686 19.663924 14.794090 21.123819
##       113       114       115       116       117       118       119       120
## 13.855332 16.383990 15.297256 12.937084 11.978488  6.567163 15.609467  6.816651
##       121       122       123       124       125       126       127       128
## 14.424501  7.860765 13.621365 15.058118 19.494043  9.129252 10.590963  6.590636
##       129       130       131       132       133       134       135       136
## 22.212603  7.904018 10.397700 15.600460  8.418883 19.275815 11.866030 13.966786
##       137       138       139       140       141       142       143       144
## 11.424198 20.877226  9.757607 19.634112  9.475405 18.438803 19.251445  8.778621
##       145       146       147       148       149       150       151       152
## 10.105028  9.697690 15.279189 23.260388 12.235950  9.816591 18.377596 10.036584
##       153       154       155       156       157       158       159       160
## 16.342517 18.222271 15.480532  5.289428 15.395226 10.019564 10.393418 12.406103
##       161       162       163       164       165       166       167       168
## 14.216501 13.572481 14.944003 17.320200 11.047079 14.289784 10.808694 13.360766
##       169       170       171       172       173       174       175       176
## 17.213351 17.921933  7.389574 14.376846  7.596578 11.960970 13.736151 24.783526
##       177       178       179       180       181       182       183       184
## 19.964022 12.174924 16.013844 12.378040 10.575089 13.933696  6.564088 24.163936
```

```
##       185       186       187       188       189       190       191       192
## 18.537949 20.779377  9.698684 17.060279 18.620097  6.051445 12.454978  8.405926
##       193       194       195       196       197       198       199       200
##  4.478859 18.448761 16.463190  5.364512  8.152375 12.768048 23.792923 15.157543
```

Hacer predicciones sobre datos nuevos

```
datos.nuevos = data.frame(
  TV = c(rep(seq(100,300,by=50),5)),
  Radio = c(  rep(1:5,each=5)*10  )
)
datos.nuevos
```

```
##     TV Radio
## 1  100    10
## 2  150    10
## 3  200    10
## 4  250    10
## 5  300    10
## 6  100    20
## 7  150    20
## 8  200    20
## 9  250    20
## 10 300    20
## 11 100    30
## 12 150    30
## 13 200    30
## 14 250    30
## 15 300    30
## 16 100    40
## 17 150    40
## 18 200    40
## 19 250    40
## 20 300    40
## 21 100    50
## 22 150    50
## 23 200    50
## 24 250    50
## 25 300    50
```

Las predicciones:

```
predict(resMRL2, datos.nuevos, interval = "confidence",level = 0.95 )
```

```
##          fit      lwr       upr
## 1   9.376524  9.04057  9.712477
## 2  11.664264 11.34890 11.979627
## 3  13.952005 13.60039 14.303619
## 4  16.239746 15.80916 16.670328
## 5  18.527487 17.99386 19.061115
## 6  11.256466 10.98525 11.527677
## 7  13.544207 13.30387 13.784544
## 8  15.831947 15.54989 16.114008
## 9  18.119688 17.74694 18.492440
## 10 20.407429 19.92171 20.893145
## 11 13.136408 12.84567 13.427142
## 12 15.424149 15.16657 15.681733
```

```
## 13 17.711890 17.41904 18.004743
## 14 19.999630 19.62179 20.377474
## 15 22.287371 21.80018 22.774566
## 16 15.016350 14.63455 15.398156
## 17 17.304091 16.95023 17.657950
## 18 19.591832 19.21468 19.968988
## 19 21.879573 21.43588 22.323270
## 20 24.167314 23.62965 24.704973
## 21 16.896293 16.38904 17.403544
## 22 19.184034 18.69992 19.668143
## 23 21.471774 20.97277 21.970783
## 24 23.759515 23.21065 24.308375
## 25 26.047256 25.42190 26.672616
```

Para datos nuevos mejor usar el valor del argumento "prediction":

```r
predict(resMRL2, datos.nuevos, interval = "prediction",level = 0.95 )
```

```
##          fit       lwr      upr
## 1   9.376524  6.043771 12.70928
## 2  11.664264  8.333525 14.99500
## 3  13.952005 10.617638 17.28637
## 4  16.239746 12.896129 19.58336
## 5  18.527487 15.169044 21.88593
## 6  11.256466  7.929616 14.58332
## 7  13.544207 10.219731 16.86868
## 8  15.831947 12.504196 19.15970
## 9  18.119688 14.783025 21.45635
## 10 20.407429 17.056266 23.75859
## 11 13.136408  9.807910 16.46491
## 12 15.424149 12.098382 18.74992
## 13 17.711890 14.383206 21.04057
## 14 19.999630 16.662395 23.33687
## 15 22.287371 18.935993 25.63875
## 16 15.016350 11.678664 18.35404
## 17 17.304091 13.969486 20.63870
## 18 19.591832 16.254674 22.92899
## 19 21.879573 18.534241 25.22490
## 20 24.167314 20.808229 27.52640
## 21 16.896293 13.541941 20.25064
## 22 19.184034 15.833103 22.53496
## 23 21.471774 18.118659 24.82489
## 24 23.759515 20.398619 27.12041
## 25 26.047256 22.673023 29.42149
```

Ejercicio 6

Regresión cuadrática.

En 1609 Galileo demostró que la trayectoria de un cuerpo cayendo con una componente horizontal es una parábola. En el curso de ganar conocimiento de este hecho, estableció un experimento que midió dos variables, una altura y una distancia, produciendo los siguientes datos.

```r
dist = c(253, 337,395,451,495,534,574)
height = c(100,200,300,450,600,800,1000)
```

Solución

Regresión Cuadrática

```
lm.2 = lm(dist ~ height + I(height^2))
summary(lm.2)
```

```
##
## Call:
## lm(formula = dist ~ height + I(height^2))
##
## Residuals:
##       1       2       3       4       5       6       7
## -14.420   9.192  13.624   2.060  -6.158 -12.912   8.614
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.002e+02  1.695e+01  11.811 0.000294 ***
## height       7.062e-01  7.568e-02   9.332 0.000734 ***
## I(height^2) -3.410e-04  6.754e-05  -5.049 0.007237 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.79 on 4 degrees of freedom
## Multiple R-squared:  0.9902, Adjusted R-squared:  0.9852
## F-statistic: 201.1 on 2 and 4 DF,  p-value: 9.696e-05
```

Regresión cúbica o de grado 3:

```
lm.3 = lm(dist ~ height + I(height^2) + I(height^3))
summary(lm.3)
```

```
##
## Call:
## lm(formula = dist ~ height + I(height^2) + I(height^3))
##
## Residuals:
##        1        2        3        4        5        6        7
## -2.35639  3.52782  1.83769 -4.43416  0.01945  2.21560 -0.81001
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.555e+02  8.182e+00  19.003 0.000318 ***
## height       1.119e+00  6.454e-02  17.332 0.000419 ***
## I(height^2) -1.254e-03  1.360e-04  -9.220 0.002699 **
## I(height^3)  5.550e-07  8.184e-08   6.782 0.006552 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.941 on 3 degrees of freedom
## Multiple R-squared:  0.9994, Adjusted R-squared:  0.9988
## F-statistic:  1658 on 3 and 3 DF,  p-value: 2.512e-05
```

```
lm.1 = lm(dist ~ height)
summary(lm.1)
```

```
##
```

```
## Call:
## lm(formula = dist ~ height)
##
## Residuals:
##        1        2        3        4        5        6        7 
## -49.8788   0.7086  25.2959  31.1769  25.0578  -2.7675 -29.5929 
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 269.46607   24.18421  11.142 0.000102 ***
## height        0.33413    0.04181   7.992 0.000495 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 33.5 on 5 degrees of freedom
## Multiple R-squared:  0.9274, Adjusted R-squared:  0.9129 
## F-statistic: 63.88 on 1 and 5 DF,  p-value: 0.0004951
```

```
AIC(lm.1)
```

```
## [1] 72.67186
```

```
AIC(lm.2)
```

```
## [1] 60.68759
```

```
AIC(lm.3)
```

```
## [1] 43.13532
```