UNIVERSITAT DE BARCELONA

Universitat Oberta de Catalunya

Diagnosis del modelo
Soluciones a los ejercicios opcionales

Francesc Carmona

7 de abril de 2018

Ejercicios del libro de Faraway. Capítulo 6.

Los ejercicios 6, 7 y 8 son similares a los propuestos.

Ejercicio 6

Using the `happy` *data, fit a model with* `happy` *as the response and the other four variables as predictors. Answer the questions posed in the first question.*

Ejercicio 7

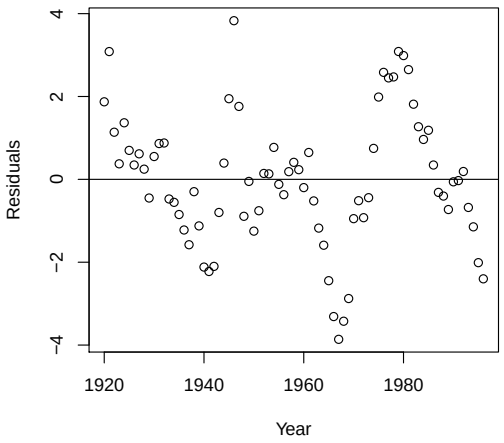
Using the `tvdoctor` *data, fit a model with* `life` *as the response and the other two variables as predictors. Answer the questions posed in the first question.*

Ejercicio 8

For the `divusa` *data, fit a model with* `divorce` *as the response and the other variables, except* `year` *as predictors. Check for serial correlation.*

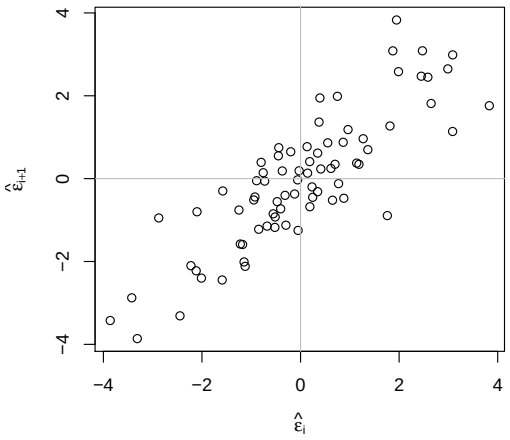
Vamos a comprobar si hay correlación temporal.

```
> library(faraway)
> lmod <- lm(divorce ~ ., data=divusa[,-1])
> plot(residuals(lmod) ~ divusa$year,xlab="Year",ylab="Residuals")
> abline(h=0)
```



La correlación temporal es absolutamente clara.

```
> n <- length(residuals(lmod))
> plot(tail(residuals(lmod),n-1) ~ head(residuals(lmod),n-1),
+      xlab=expression(hat(epsilon)[i]),
+      ylab=expression(hat(epsilon)[i+1]))
> abline(h=0,v=0,col=grey(0.75))
```



También así queda clara la correlación entre residuos consecutivos.

```
> require(lmtest)
> dwtest(divorce ~ ., data=divusa[,-1])


Durbin-Watson test

data:  divorce ~ .
DW = 0.29988, p-value < 2.2e-16
alternative hypothesis: true autocorrelation is greater than 0
```

Finalmente, el test de Durbin-Watson es significativo.

Ejercicios del libro de Faraway. Capítulo 7.

Ejercicio 8

Use the `fat` *data, fitting the model described in Section 4.2.*

(a) *Compute the condition numbers and variance inflation factors. Comment on the degree of collinearity observed in the data.*

```
> lmod <- lm(brozek ~ age + weight + height + neck + chest + abdom +
+            hip + thigh + knee + ankle + biceps + forearm + wrist, data=fat)
> X <- model.matrix(lmod)
> va <- eigen(t(X) %*% X)$values
> sqrt(max(va)/va)
```

```
 [1]     1.00000    17.47132    25.30453    58.60647    83.59171   100.63278
 [7]   137.89789   175.28697   192.61522   213.00868   228.15833   268.20747
[13]   555.67004 17837.52765
```

Hay números de condición mucho mayores que 30. Tenemos un problema grave de multicolinealidad.

```
> vif(lmod)

      age    weight    height      neck     chest     abdom       hip
 2.250450 33.509320  1.674591  4.324463  9.460877 11.767073 14.796520
    thigh      knee     ankle    biceps   forearm     wrist
 7.777865  4.612147  1.907961  3.619744  2.192492  3.377515
```

Hay muchos vif's superiores a 10 e incluso también a 4, lo que confirma el problema de multicolinealidad.

(b) *Cases 39 and 42 are unusual. Refit the model without these two cases and recompute the collinearity diagnostics. Comment on the differences observed from the full data fit.*

Se entiende por "observación inusual" aquella que está alejada del núcleo principal de las observaciones. Para obtenerlas utilizaremos el *leverage*.

```
> k <- 13 # variables regresoras
> n <- length(fat$brozek) # observaciones
> hatv <- hatvalues(lmod)
> which(hatv > 2 * (k + 1)/n)

  5  31  36  39  41  42  54  86 106 159 175 206 216
  5  31  36  39  41  42  54  86 106 159 175 206 216

> head(sort(hatv, decreasing=T))

       42        39        86        31       175        41
0.7400257 0.3751201 0.3560554 0.3090124 0.2713452 0.2139662
```

Vemos que hay varias observaciones que superan el criterio habitual para tener un alto leverage, pero las dos mayores son para las observaciones 42 y 39.

Ahora recalculamos el modelo sin estas dos observaciones.

```
> lmod2 <- lm(brozek ~ age + weight + height + neck + chest + abdom +
+             hip + thigh + knee + ankle + biceps + forearm + wrist,
+             data=fat[-c(39,42),])
> X <- model.matrix(lmod2)
> va <- eigen(t(X) %*% X)$values
> sqrt(max(va)/va)

 [1]     1.00000    18.39781    26.21503    61.53260    91.07685   114.44853
 [7]   148.72593   178.80985   202.08836   211.78489   240.69602   276.35187
[13]   554.79882 24128.42169
```

```
> vif(lmod2)

      age    weight    height      neck     chest     abdom       hip
 2.278191 45.298843  3.439587  3.978898 10.712505 11.967580 12.146249
    thigh      knee     ankle    biceps   forearm     wrist
 7.153711  4.441752  1.810253  3.409524  2.422878  3.263677
```

Es evidente que el problema se ha agravado. La solución a la multicolinealidad no es eliminar observaciones inusuales.

(c) *Fit a model with* `brozek` *as the response and just* `age`, `weight` *and* `height` *as predictors. Compute the collinearity diagnostics and compare to the full data fit.*

```
> lmod3 <- lm(brozek ~ age + weight + height, data=fat)
> X <- model.matrix(lmod3)
> va <- eigen(t(X) %*% X)$values
> sqrt(max(va)/va)

[1]    1.00000   13.51191   22.67135 4072.27833

> vif(lmod3)

     age   weight   height
1.032253 1.107050 1.140470
```

El número de condición es superior a 30, lo que indica un problema de multicolinealidad, pero los vif's son todos inferiores a 4. El problema de multicolinealidad existe, pero se ha reducido.

(d) *Compute a 95 % prediction interval for* `brozek` *for the median values of* `age`, `weight` *and* `height`.

```
> p1 <- predict(lmod3, newdata = data.frame(age=median(fat$age),
+                                           weight=median(fat$weight),
+                                           height=median(fat$height)),
+                       interval = "confidence"); p1

       fit      lwr     upr
1 18.28132 17.60315 18.9595
```

(e) *Compute a 95 % prediction interval for brozek for* `age`*=40,* `weight`*=200 and* `height`*=73. How does the interval compare to the previous prediction?*

```
> p2 <- predict(lmod3, newdata = data.frame(age=40, weight=200, height=73),
+                       interval = "confidence"); p2

       fit      lwr      upr
1 20.47854 19.54863 21.40845

> c(p1[3]-p1[2], p2[3]-p2[2])

[1] 1.356343 1.859825
```

El intervalo es mayor en el segundo caso puesto que la observación que se pretende predecir se aleja de la centralidad de los datos.

(f) *Compute a 95 % prediction interval for* `brozek` *for* `age`*=40,* `weight`*=130 and* `height`*=73. Are the values of predictors unusual? Comment on how the interval compares to the previous two answers.*

Como se sabe (ecuación 9.2 del libro de Carmona), el leverage de una observación es equivalente a la distancia de Mahalanobis al centro de los datos. Podemos comparar las distancias de Mahalanobis (al cuadrado) de las tres observaciones de las que se pide su predicción.

```
> datos <- matrix(c(median(fat$age),median(fat$weight),median(fat$height),
+                   40,200,73,
+                   40,130,73), nrow = 3, byrow = T)
> mahalanobis(datos, center=colMeans(fat[,4:6]), cov = cov(fat[,4:6]))

[1] 0.03136358 0.93568074 4.64432854
```

Vemos que la tercera fila que corresponde a los datos de este apartado tiene una distancia de Mahalanobis (al cuadrado) mucho mayor que en los otros dos casos. Es una observación inusual. Eso provocará un intervalo de predicción mucho más ancho.

```
> p3 <- predict(lmod3, newdata = data.frame(age=40, weight=130, height=73),
+                       interval = "confidence"); p3

       fit      lwr     upr
1 7.617419 6.028418 9.20642

> p3[3]-p3[2]

[1] 3.178002
```

Ejercicios del libro de Carmona

Ejercicio 9.1

Realizar el análisis completo de los residuos del modelo de regresión parabólico propuesto en la sección 1.2 con los datos de tráfico.

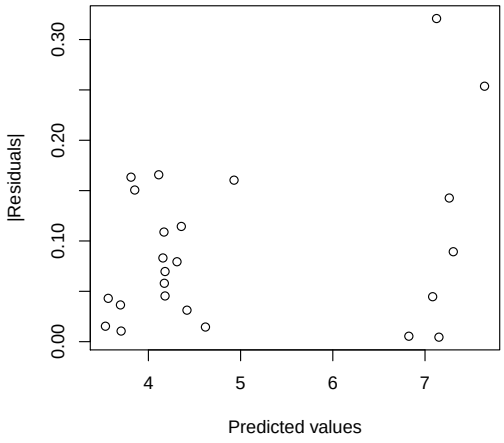
El modelo parabólico es

```
> densidad <- c(12.7,17,66,50,87.8,81.4,75.6,66.2,81.1,62.8,77,89.6,18.3,
+               19.1,16.5,22.2,18.6,66,60.3,56,66.3,61.7,66.6,67.8)
> velocidad <- c(62.4,50.7,17.1,25.9,12.4,13.4,13.7,17.9,13.8,17.9,15.8,
+                12.6,51.2,50.8 ,54.7,46.5,46.3,16.9,19.8,21.2,18.3,18,16.6,18.3)
> rvelocidad <- sqrt(velocidad)
> model <- lm(rvelocidad ~ densidad + I(densidad^2))
```

Siguiremos el esquema de diagnosis que propone Faraway en sus ejercicios.

(a) **Varianza constante.**

```
> plot(fitted(model),abs(residuals(model)),
+         xlab="Predicted values",ylab="|Residuals|")
```



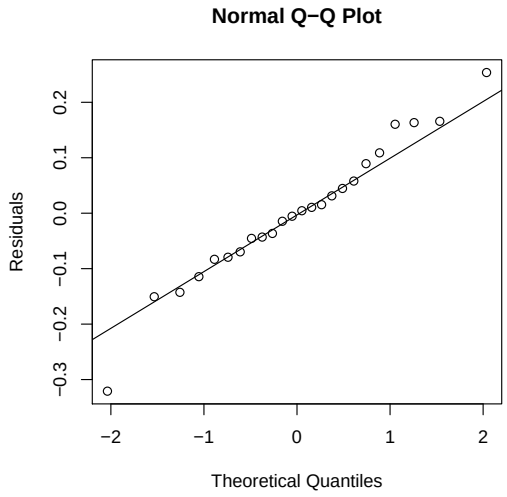
Observamos que la varianza parece mayor en las predicciones más altas. Podemos comprobar si es así con un test:

```
> sumary(lm(sqrt(abs(residuals(model))) ~ fitted(model)))

              Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.183425   0.098286  1.8662   0.0754
fitted(model) 0.018098   0.018859  0.9596   0.3477

n = 24, p = 2, Residual SE = 0.13357, R-Squared = 0.04
```

El p-valor indica que la pendiente no es significativa y podemos asumir que la varianza es constante.

(b) **Normalidad.**

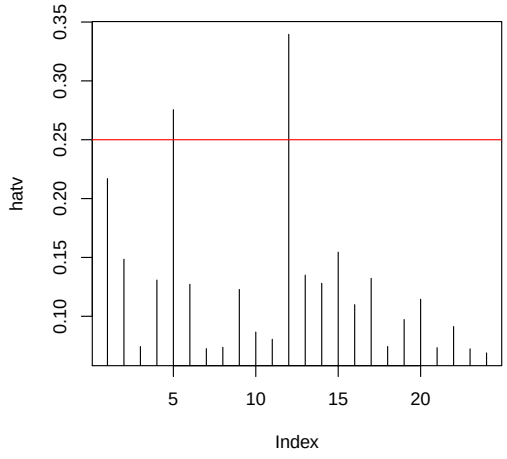
```
> qqnorm(residuals(model),ylab="Residuals")
> qqline(residuals(model))
> shapiro.test(residuals(model))


Shapiro-Wilk normality test

data:  residuals(model)
W = 0.97667, p-value = 0.8275
```



Tanto el gráfico, como el test no muestran ningún problema con la normalidad de los residuos.

(c) **Leverage** Observaciones inusuales lejanas de la media y con influencia potencial.

```
> hatv <- hatvalues(model)
> p <- length(model$coefficients) # k+1
> n <- length(model$fitted.values)
> leverage.mean <- p/n # (k+1)/n
> which(hatv > 2*leverage.mean)

 5 12
 5 12

> plot(hatv, type="h")
> abline(h=2*leverage.mean, col="red")
```



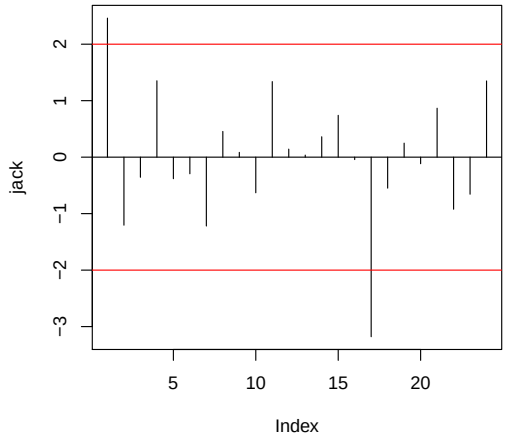
(d) **Valores atípicos** (outliers)

Los residuos studentizados externamente son:

```
> jack <- rstudent(model) # jacknife residuals
```

Podemos utilizar el criterio naíf de considerar outlier todo residuo con valor absoluto superior a 2.

```
> which(abs(jack)>2)

 1 17
 1 17

> plot(jack, type="h")
> abline(h=-2, col="red"); abline(h=0); abline(h=2, col="red")
```



O ser más sofisticados teniendo en cuenta que, en condiciones normales, estos residuos siguen una distribució t-Student de $n - p - 1$ grados de libertad:

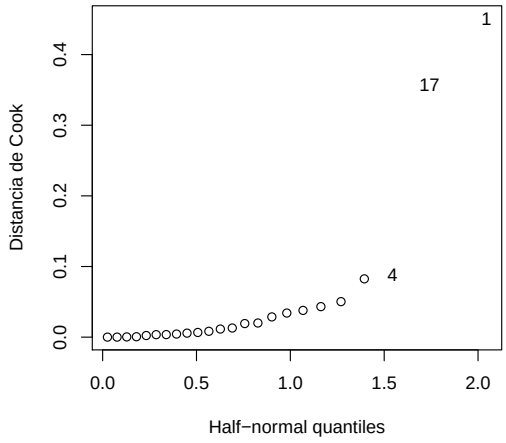
```
> grlib <- n-p-1
> which(abs(jack) > abs(qt(0.05/(2*n),grlib)))

named integer(0)
```

Con la corrección de Bonferroni para comparaciones múltiples y según este criterio, no hay residuos atípicos (outliers).

(e) **Observaciones influyentes**

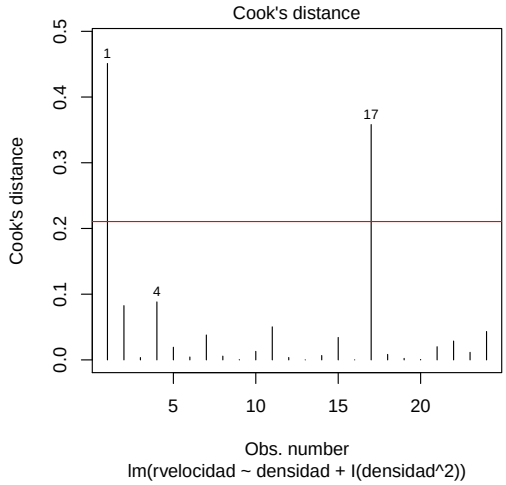
Calculamos la distancia de Cook como medida de la influencia de los puntos y la representamos contra los cuartiles de una distribución seminormal.

```
> cook <- cooks.distance(model)
> halfnorm(cook, nlab=3, ylab="Distancia de Cook")
```



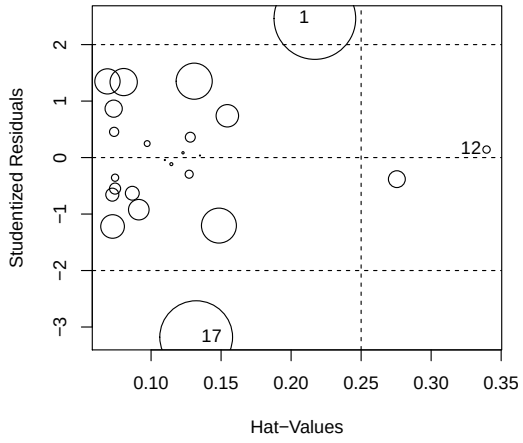
En el siguiente gráfico se muestra un criterio de selección:

```
> # Cook's D plot
> # identify D values > 4/(n-k-1)
> plot(model, which=4)
> abline(h=4/((n-p-2)), col="red")
```



Finalmente, en el paquete `car` tenemos la función `influencePlot` que resume la situación:

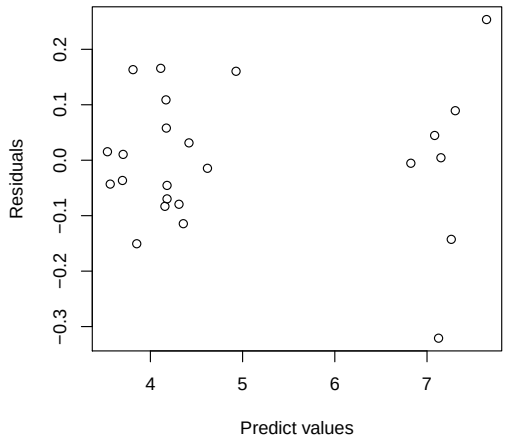
```
> library(car)
> influencePlot(model)

      StudRes       Hat       CookD
1   2.4617222 0.2169701 0.451047078
12  0.1421674 0.3395720 0.003633589
17 -3.1795129 0.1322228 0.358110216
```



Los puntos destacados son el 1 y el 17. Estos puntos son outliers e influyentes. Tal vez deberíamos recalcular el modelo sin estas dos observaciones y ver si los coeficientes se modifican sustancialmente o no.

(f) **Estructura del modelo**

```
> plot(fitted(model),residuals(model),xlab="Predict values",ylab="Residuals")
```



No aparece haber ningún problema con la estructura del modelo.

Ejercicio 9.2

Realizar el análisis completo de los residuos de los modelos de regresión simple y parabólico propuestos en la sección 1.2 con los datos de tráfico, pero tomando como variable respuesta la velocidad (sin raíz cuadrada). Este análisis debe justificar la utilización de la raíz cuadrada de la velocidad como variable dependiente.

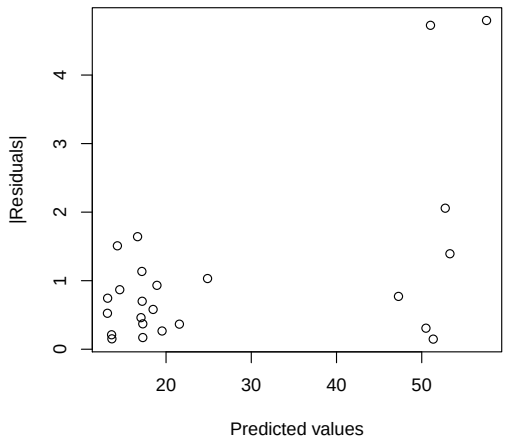
El modelo parabólico es

```
> model <- lm(velocidad ~ densidad + I(densidad^2))
```

Seguiremos el mismo esquema del ejercicio anterior para poder comparar.

(a) **Varianza constante.**

```
> plot(fitted(model),abs(residuals(model)),xlab="Predicted values",ylab="|Residuals|")
```



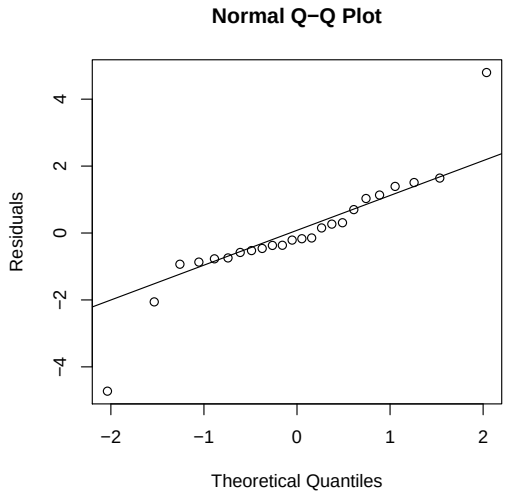
Observamos que la varianza es mucho mayor en las predicciones más altas. Podemos comprobar si es así con un test:

```
> sumary(lm(sqrt(abs(residuals(model))) ~ fitted(model)))

               Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.5307324  0.1727442  3.0724 0.005573
fitted(model) 0.0144171  0.0054602  2.6404 0.014943

n = 24, p = 2, Residual SE = 0.43331, R-Squared = 0.24
```

El p-valor indica que la pendiente es significativa y la varianza no es constante.

(b) **Normalidad.**

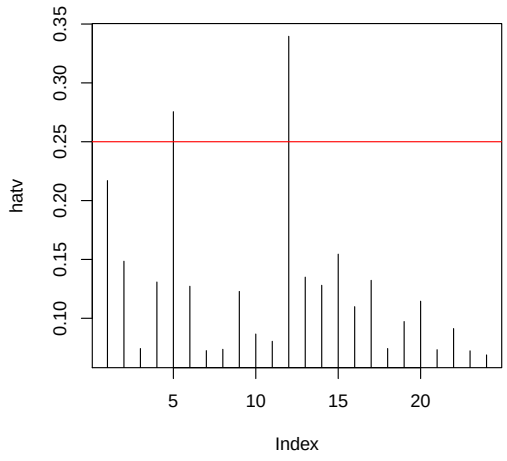
```
> qqnorm(residuals(model),ylab="Residuals")
> qqline(residuals(model))
> shapiro.test(residuals(model))


Shapiro-Wilk normality test

data:  residuals(model)
W = 0.87969, p-value = 0.008192
```



Tanto el gráfico, como el test muestran la no normalidad de los residuos.

(c) **Leverage** Observaciones inusuales lejanas de la media y con influencia potencial.

```
> hatv <- hatvalues(model)
> p <- length(model$coefficients) # k+1
> n <- length(model$fitted.values)
> leverage.mean <- p/n # (k+1)/n
> which(hatv > 2*leverage.mean)

 5 12
 5 12

> plot(hatv, type="h")
> abline(h=2*leverage.mean, col="red")
```



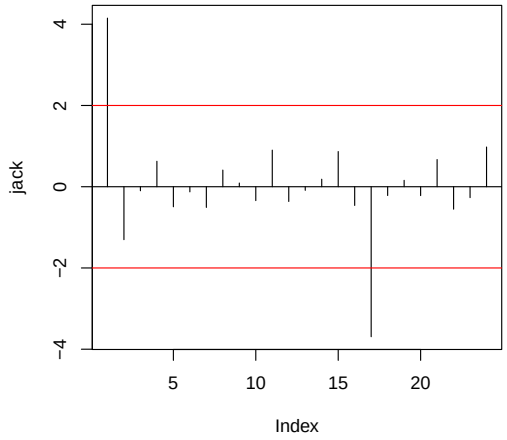
(d) **Valores atípicos** (outliers)

Los residuos studentizados externamente son:

```
> jack <- rstudent(model) # jacknife residuals
```

Podemos utilizar el criterio naíf de considerar outlier todo residuo con valor absoluto superior a 2.

```
> which(abs(jack)>2)

 1 17
 1 17

> plot(jack, type="h")
> abline(h=-2, col="red"); abline(h=0); abline(h=2, col="red")
```



O ser más sofisticados teniendo en cuenta que, en condiciones normales, estos residuos siguen una distribució t-Student de $n - p - 1$ grados de libertad:

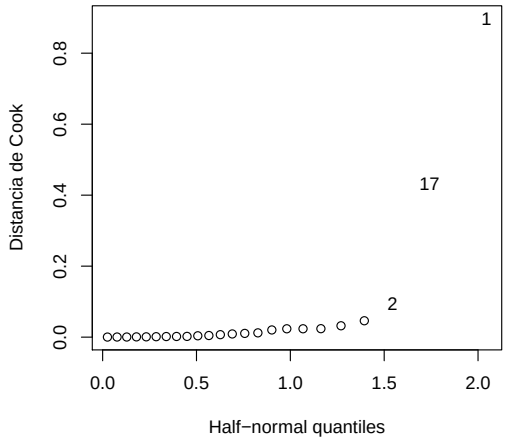
```
> grlib <- n-p-1
> which(abs(jack) > abs(qt(0.05/(2*n),grlib)))

 1 17
 1 17
```

Con la corrección de Bonferroni para comparaciones múltiples y según este criterio, hay dos residuos atípicos (outliers).

(e) **Observaciones influyentes**

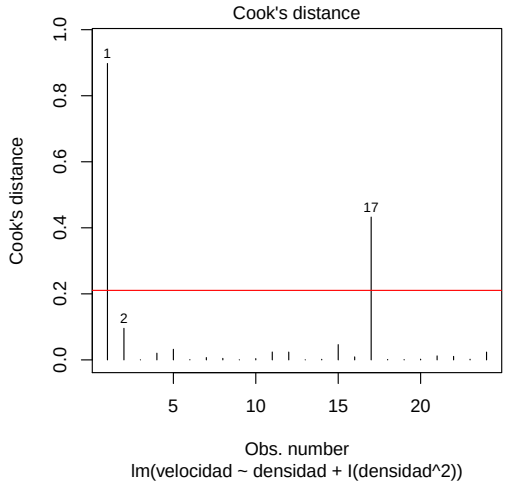
Calculamos la distancia de Cook como medida de la influencia de los puntos y la representamos contra los cuartiles de una distribución seminormal.

```
> cook <- cooks.distance(model)
> halfnorm(cook, nlab=3, ylab="Distancia de Cook")
```



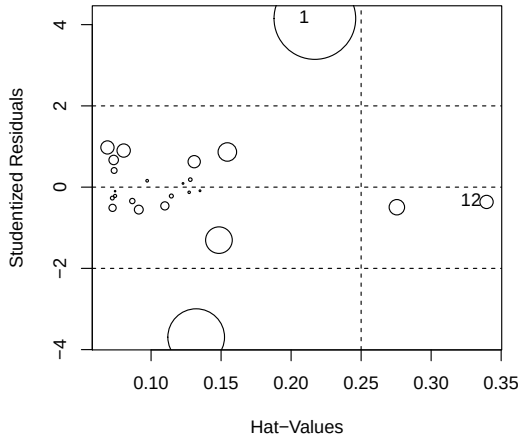
En el siguiente gráfico se muestra un criterio de selección:

```
> # Cook's D plot
> # identify D values > 4/(n-k-1)
> plot(model, which=4)
> abline(h=4/((n-p-2)), col="red")
```



Finalmente, en el paquete `car` tenemos la función `influencePlot` que resume la situación:

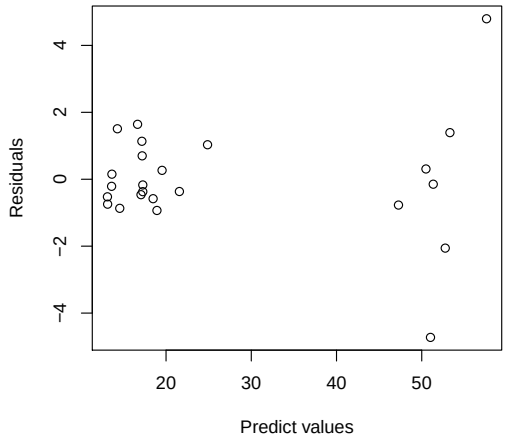
```
> influencePlot(model)

      StudRes       Hat      CookD
1   4.1507056 0.2169701 0.89761365
12 -0.3636947 0.3395720 0.02364751
```



Los puntos destacados son el 1 y el 17. Estos puntos son outliers e influyentes.

(f) **Estructura del modelo**

```
> plot(fitted(model),residuals(model),xlab="Predict values",ylab="Residuals")
```



Básicamente el problema es la heterocedasticidad.

Como el crecimiento de la varianza respecto a la respuesta no es muy grande, se propone la transformación raíz cuadrada para corregir los problemas detectados como se ha visto en el ejercicio anterior.