Práctica_Predicción_01

Alejandro Gómez de Miguel

26/10/2017

Primeros pasos

```
setwd("~/Documents/CUNEF - Data Science/1º Semestre/Predicción/Clase 2")
```

```
data <- read.csv2('Fondos.csv', header= T)
```

Guardamos una copia y cambiamos el nombre de las columnas al orginal.

```
fondos_DF_original <- data
colnames(data) <- c('Rent_1año', 'Nombre', 'CustomDelayToBuy', 'Dias_Depl_Reemb', 'ISIN', 'Gestora', 'I
head(data)
```

```
##   Rent_1año                                Nombre CustomDelayToBuy
## 1     27.22                 Merch-Oportunidades FI                0
## 2     24.65            Renta 4 Latinoam\xe9rica FI                0
## 3     21.19 Santander Acciones Latinoamericanas FI                0
## 4     20.95                    BBVA Bolsa Latam FI                0
## 5     17.20           Eurovalor Iberoam\xe9rica FI                0
## 6     11.08                  Global Allocation FI                1
##   Dias_Depl_Reemb         ISIN                              Gestora
## 1               2 ES0162305033                       Merchbanc SGIIC
## 2               1 ES0173320039                  Renta 4 Gestora SGIIC
## 3               0 ES0105930038       Santander Asset Management SGIIC
## 4               1 ES0142332032            BBVA Asset Management SGIIC
## 5               2 ES0133576035 Allianz Popular Asset Management SGIIC
## 6               1 ES0116848005                  Renta 4 Gestora SGIIC
##   Inv_Min_Inicial Rent_1dia Rent_1semana Rent_1mes Rent_3meses Rent_6meses
## 1               0      2.18         8.04      9.17        6.41       12.89
## 2              10     -3.31         0.38      4.67        8.91       25.84
## 3            3000     -3.39         0.30      3.80        3.76       22.82
## 4             600     -2.54         2.96      5.08        3.71       23.45
## 5             600     -1.79         1.52      3.03        2.44       19.41
## 6              10      2.01         4.45      6.30       12.10        3.53
##   Rent_en_el_año Rent_3años Rent_5años Rent_10años Estilo_Inv_RV
## 1          36.53       3.91       4.09        4.50             1
## 2          37.48      -2.59      -2.72        1.78             1
## 3          30.12      -0.27      -2.38       -0.24             2
## 4          30.02       0.03      -1.27        0.80             2
## 5          29.19      -2.07      -2.51        0.97             2
## 6          10.97      10.23      15.53        9.18             1
##   Estilo_Inv_RF Cap_Media_Bursatil Patrimonio Rating_MorningStar
## 1            NA          19759.989    4.97245                 3
## 2            NA           9358.149    2.24107                 3
## 3            NA          10096.463   24.14945                 3
## 4            NA          10984.282   20.20589                 3
## 5            NA          11387.819    5.29536                 3
## 6            NA          30409.994   46.21474                 5
##   Volatilidad Sharpe Ratio_Informacion Media Comision_Gestion CustomBuyFee
```

```
## 1       18.04   0.20            -0.37  0.30            1.25           0
## 2       25.01   0.04            -0.19  0.08            1.35           0
## 3       20.13   0.10            -0.12  0.16            2.25           0
## 4       20.53   0.08            -0.13  0.14            2.25           0
## 5       19.63  -0.03            -0.45 -0.05            1.35           0
## 6       18.44   0.57             0.03  0.87            1.35           0
##   Comision_Suscripcion Comision_Deposito
## 1                    0             0.10
## 2                    0             0.12
## 3                    0             0.10
## 4                    0             0.20
## 5                    0             0.15
## 6                    0             0.15
```

Busqueda del modelo

En primer lugar, estimamos la regresion con todas las variables de la tabla y analizamos los resultados.

```
modelo_completo <- lm(Rent_1año ~ Rent_1dia + Rent_1semana + Rent_1mes + Rent_3meses + Rent_6meses  + R
summary(modelo_completo)
```

```
##
## Call:
## lm(formula = Rent_1año ~ Rent_1dia + Rent_1semana + Rent_1mes +
##     Rent_3meses + Rent_6meses + Rent_en_el_año + Rent_3años +
##     Rent_5años + Rent_10años + CustomDelayToBuy + Cap_Media_Bursatil +
##     Patrimonio + Rating_MorningStar + Volatilidad + Sharpe +
##     Ratio_Informacion + Media + Comision_Gestion + CustomBuyFee +
##     Comision_Suscripcion + Comision_Deposito, data = data, na.action = na.omit)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.69683 -0.51823  0.08895  0.50471  2.48655
##
## Coefficients: (1 not defined because of singularities)
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -4.877e-01  7.534e-01  -0.647   0.5187
## Rent_1dia             2.333e-02  1.107e-01   0.211   0.8334
## Rent_1semana         -2.011e-01  1.071e-01  -1.878   0.0629 .
## Rent_1mes            -8.328e-02  1.049e-01  -0.794   0.4290
## Rent_3meses           1.604e-01  7.683e-02   2.088   0.0390 *
## Rent_6meses          -2.386e-01  3.594e-02  -6.640 1.09e-09 ***
## Rent_en_el_año        9.589e-01  2.939e-02  32.628  < 2e-16 ***
## Rent_3años            2.870e-01  3.748e-01   0.766   0.4454
## Rent_5años           -2.843e-02  6.332e-02  -0.449   0.6543
## Rent_10años           1.120e-02  7.611e-02   0.147   0.8833
## CustomDelayToBuy      1.211e-01  2.920e-01   0.415   0.6792
## Cap_Media_Bursatil    2.471e-06  5.141e-06   0.481   0.6316
## Patrimonio            5.108e-05  2.260e-04   0.226   0.8216
## Rating_MorningStar    3.420e-02  1.445e-01   0.237   0.8134
## Volatilidad          -2.052e-01  7.877e-02  -2.605   0.0104 *
## Sharpe               -6.624e-01  7.869e-01  -0.842   0.4016
## Ratio_Informacion     7.099e-02  1.746e-01   0.407   0.6851
## Media                 9.186e-01  4.776e+00   0.192   0.8478
```

```
## Comision_Gestion      2.292e-01  1.995e-01   1.149   0.2531
## CustomBuyFee         -2.796e-01  7.427e-01  -0.376   0.7073
## Comision_Suscripcion         NA         NA      NA       NA
## Comision_Deposito    -1.555e+00  1.009e+00  -1.540   0.1263
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9577 on 115 degrees of freedom
##   (364 observations deleted due to missingness)
## Multiple R-squared:  0.9752, Adjusted R-squared:  0.9709
## F-statistic: 226.2 on 20 and 115 DF,  p-value: < 2.2e-16
```

Los modelos de regresion deben tener una base teorica en cuanto a variables explicativas se refiere. Para explicar la rentabilidad anual, personalmente, no considero necesarias algunas de las variables que se han provisto.

Tras realizar distintas pruebas y analizando la significacion de las variables explicativas se han escogido dos modelos:

```
modelo1 <- lm(Rent_1año ~ + Rent_1semana
              + Rent_3meses + Rent_6meses
               + Rent_en_el_año + Rent_3años + Rent_10años
               + Cap_Media_Bursatil
               + Volatilidad,
               data = data, na.action = na.omit)
summary(modelo1)
```

```
##
## Call:
## lm(formula = Rent_1año ~ +Rent_1semana + Rent_3meses + Rent_6meses +
##     Rent_en_el_año + Rent_3años + Rent_10años + Cap_Media_Bursatil +
##     Volatilidad, data = data, na.action = na.omit)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.59040 -0.52265  0.09385  0.51556  2.36222
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -7.075e-01  2.110e-01  -3.353  0.00103 **
## Rent_1semana        -2.586e-01  5.698e-02  -4.540 1.21e-05 ***
## Rent_3meses          1.296e-01  5.015e-02   2.584  0.01080 *
## Rent_6meses         -2.290e-01  2.938e-02  -7.792 1.39e-12 ***
## Rent_en_el_año       9.442e-01  1.955e-02  48.288  < 2e-16 ***
## Rent_3años           2.705e-01  3.723e-02   7.266 2.42e-11 ***
## Rent_10años          2.782e-02  5.884e-02   0.473  0.63713
## Cap_Media_Bursatil   1.988e-06  4.232e-06   0.470  0.63930
## Volatilidad         -1.679e-01  2.694e-02  -6.232 5.16e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.926 on 139 degrees of freedom
##   (352 observations deleted due to missingness)
## Multiple R-squared:  0.9723, Adjusted R-squared:  0.9707
## F-statistic:   610 on 8 and 139 DF,  p-value: < 2.2e-16
```

```
modelo2 <- lm(Rent_1año ~ Rent_1dia + Rent_1semana
              + Rent_1mes + Rent_3meses + Rent_6meses
              + Rent_en_el_año + Rent_3años + Rent_5años + Rent_10años
              + Cap_Media_Bursatil
              + Volatilidad,
              data = data, na.action = na.omit)
summary(modelo2)
```

```
##
## Call:
## lm(formula = Rent_1año ~ Rent_1dia + Rent_1semana + Rent_1mes +
##     Rent_3meses + Rent_6meses + Rent_en_el_año + Rent_3años +
##     Rent_5años + Rent_10años + Cap_Media_Bursatil + Volatilidad,
##     data = data, na.action = na.omit)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5690 -0.4604  0.1157  0.5068  2.5432
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -6.760e-01  2.153e-01  -3.139  0.00208 **
## Rent_1dia           2.252e-02  1.005e-01   0.224  0.82302
## Rent_1semana       -2.278e-01  7.235e-02  -3.149  0.00202 **
## Rent_1mes          -8.786e-02  9.408e-02  -0.934  0.35199
## Rent_3meses         1.710e-01  6.681e-02   2.560  0.01156 *
## Rent_6meses        -2.355e-01  3.078e-02  -7.651 3.28e-12 ***
## Rent_en_el_año      9.498e-01  2.544e-02  37.340  < 2e-16 ***
## Rent_3años          2.831e-01  5.868e-02   4.824 3.71e-06 ***
## Rent_5años         -1.247e-02  5.844e-02  -0.213  0.83137
## Rent_10años         1.469e-02  6.462e-02   0.227  0.82055
## Cap_Media_Bursatil  2.466e-06  4.341e-06   0.568  0.57096
## Volatilidad        -1.669e-01  3.349e-02  -4.983 1.87e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9331 on 136 degrees of freedom
##   (352 observations deleted due to missingness)
## Multiple R-squared:  0.9725, Adjusted R-squared:  0.9703
## F-statistic:   437 on 11 and 136 DF,  p-value: < 2.2e-16
```

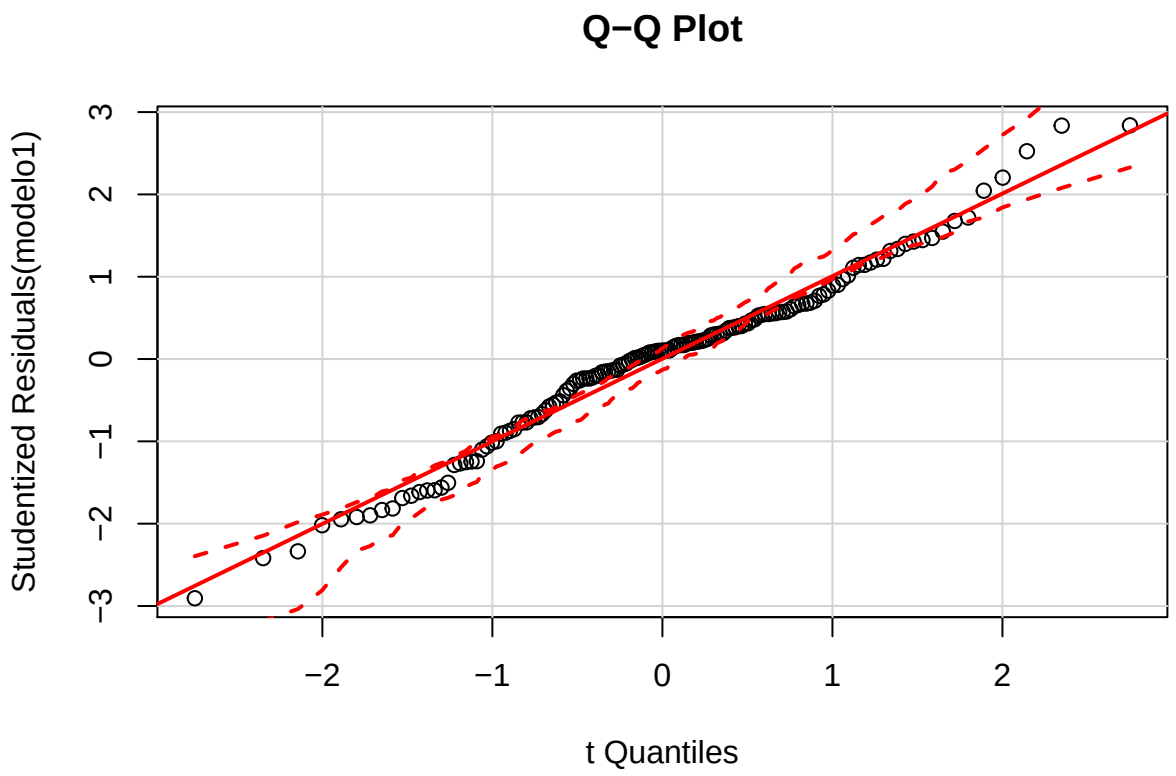
Normalidad

```
library(car)
```

```
## Warning: package 'car' was built under R version 3.2.5
```

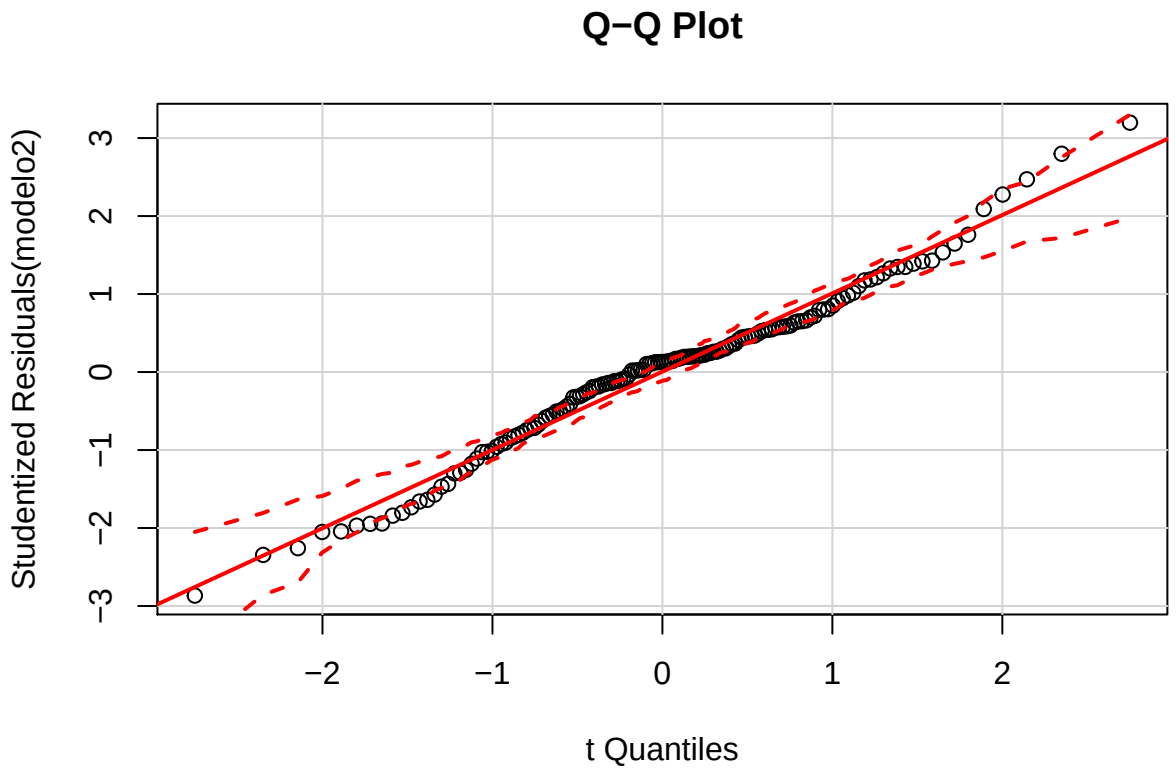
Para el primer modelo:

```
qqPlot(modelo1, labels=row.names(data), id.method="identify",
       simulate=TRUE, main="Q-Q Plot")
```



Para el segundo modelo:

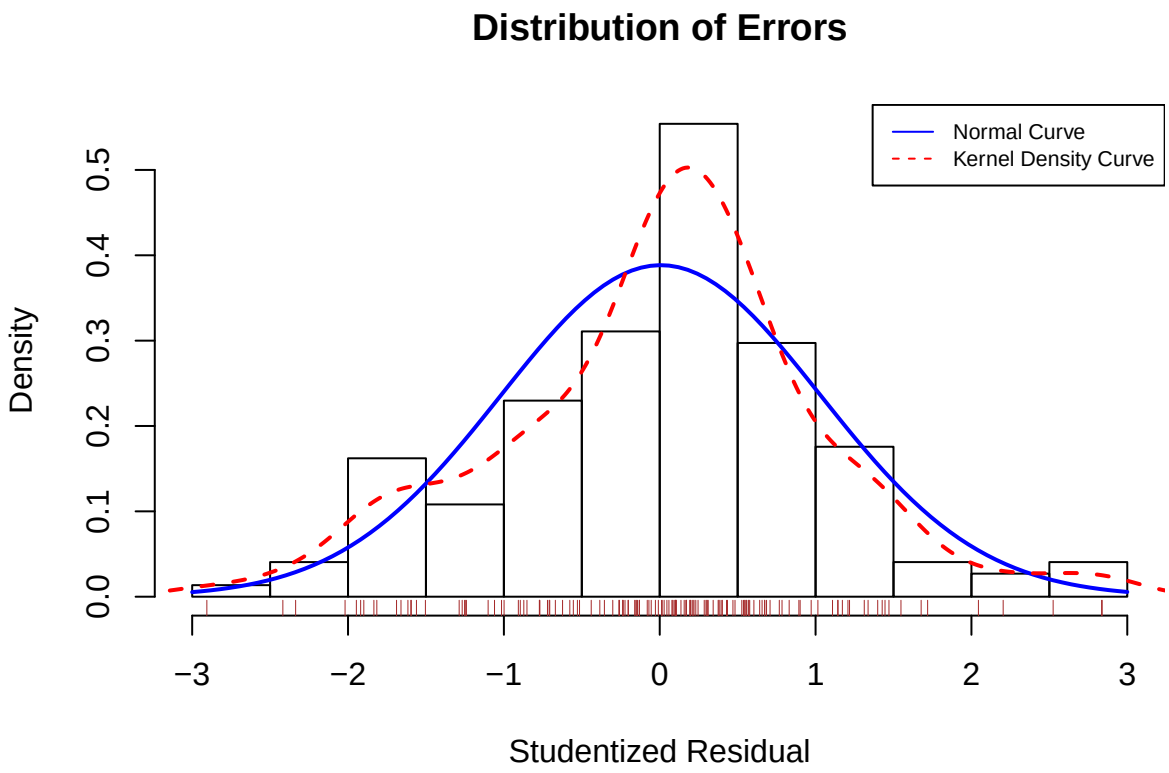
```
qqPlot(modelo2, labels=row.names(data), id.method="identify",
       simulate=TRUE, main="Q-Q Plot")
```



Para el primer modelo:

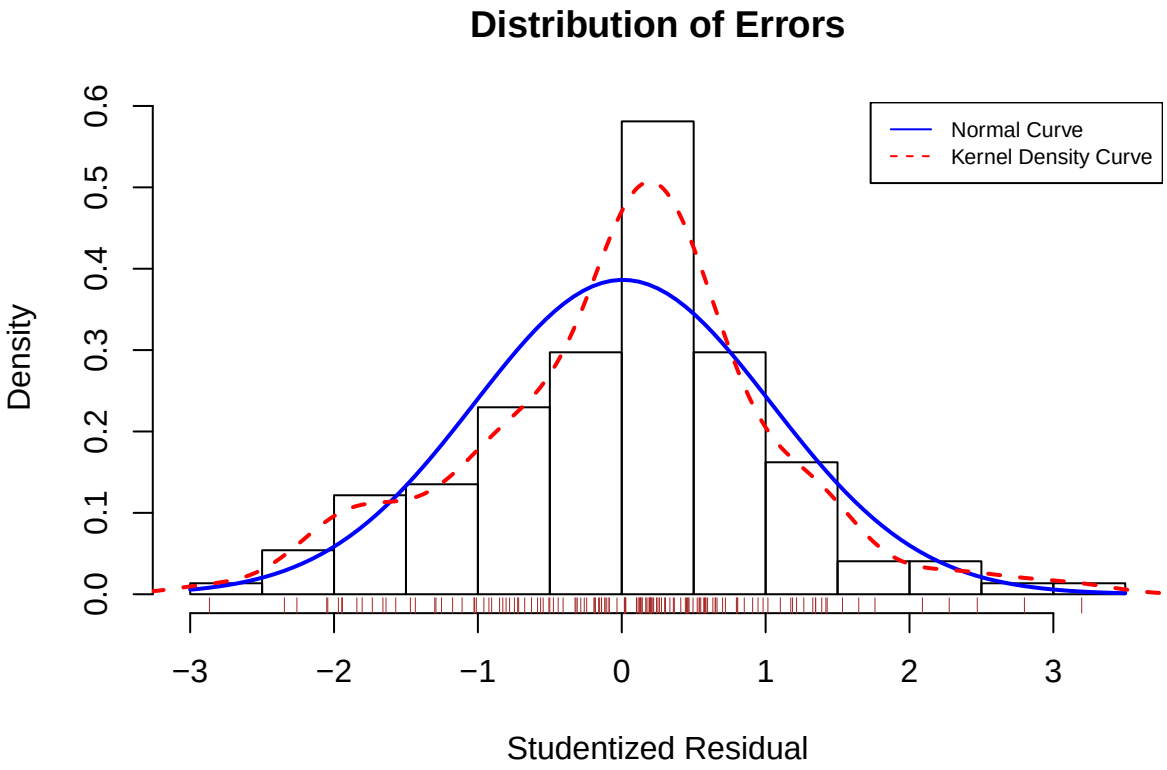
```r
residplot <- function(fit, nbreaks=10) { #n breaks, cuantas barras de histograma
  z <- rstudent(fit) #calcula los residuos estandandarizados
  hist(z, breaks=nbreaks, freq=FALSE,
       xlab="Studentized Residual",
       main="Distribution of Errors")
  rug(jitter(z), col="brown") #jitter los pone pegados, no uno arriba de otro
  curve(dnorm(x, mean=mean(z), sd=sd(z)),
        add=TRUE, col="blue", lwd=2)
  lines(density(z)$x, density(z)$y,
        col="red", lwd=2, lty=2) #calcula la linea de densidad
  legend("topright",
         legend = c( "Normal Curve", "Kernel Density Curve"),
         lty=1:2, col=c("blue","red"), cex=.7) #la leyenda
}


residplot(modelo1)
```



Para el segundo modelo:

```r
residplot(modelo2)
```



Estos graficos permiten hacer un contraste de hipotesis visual, comparando los residuos estandarizados con su supuesta posicion en una distribucion normal. En el grafico de la funcion de densidad, en ambos casos, los errores parecen estar algo sesgados, y las colas se situan fuera de la normal. Existen otros contrastes mas especificos:

Jarque Bera

```
library(fBasics)
```

```
## Loading required package: timeDate
```

```
## Warning in as.POSIXlt.POSIXct(Sys.time()): unknown timezone 'default/
## Europe/Madrid'
```

```
## Loading required package: timeSeries
```

```
##
```

```
## Rmetrics Package fBasics
```

```
## Analysing Markets and calculating Basic Statistics
```

```
## Copyright (C) 2005-2014 Rmetrics Association Zurich
```

```
## Educational Software for Financial Engineering and Computational Science
```

```
## Rmetrics is free software and comes with ABSOLUTELY NO WARRANTY.
```

```
## https://www.rmetrics.org --- Mail to: info@rmetrics.org
```

```
##
## Attaching package: 'fBasics'
```

```
## The following object is masked from 'package:car':
##
##     densityPlot
```

```
library(akima)
```

```
## Warning: package 'akima' was built under R version 3.2.5
```

Para el primer modelo:

```
vResid1 <- resid(modelo1)
jbTest(vResid1)
```

```
## Warning in interpp.old(x, y, z, xo, yo, ncp = 0, extrap = FALSE, duplicate
## = "median", : interpp.old() is deprecated, future versions will only
## provide interpp()

## Warning in interpp.old(x, y, z, xo, yo, ncp = 0, extrap = FALSE, duplicate
## = "median", : interpp.old() is deprecated, future versions will only
## provide interpp()
```

```
##
## Title:
##  Jarque - Bera Normality Test
##
## Test Results:
##   PARAMETER:
##     Sample Size: 148
##   STATISTIC:
##     LM: 1.587
##     ALM: 1.796
##   P VALUE:
##     LM p-value: 0.393
##     ALM p-value: 0.356
##     Asymptotic: 0.452
##
## Description:
##  Fri Oct 27 02:01:03 2017 by user:
```

Para el segundo modelo:

```
vResid2 <- resid(modelo2)
jbTest(vResid2)
```

```
## Warning in interpp.old(x, y, z, xo, yo, ncp = 0, extrap = FALSE, duplicate
## = "median", : interpp.old() is deprecated, future versions will only
## provide interpp()

## Warning in interpp.old(x, y, z, xo, yo, ncp = 0, extrap = FALSE, duplicate
## = "median", : interpp.old() is deprecated, future versions will only
## provide interpp()
```

```
##
## Title:
##  Jarque - Bera Normality Test
##
## Test Results:
##   PARAMETER:
```

```
##     Sample Size: 148
##   STATISTIC:
##     LM: 1.594
##     ALM: 1.857
##   P VALUE:
##     LM p-value: 0.391
##     ALM p-value: 0.343
##     Asymptotic: 0.451
##
## Description:
##  Fri Oct 27 02:01:03 2017 by user:
```

En ambos casos se aceptan las hipotesis nulas: siguen una distribucion normal.

Shapiro-Wilk

Para el primer modelo:

```
shapiro.test(vResid1)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  vResid1
## W = 0.98494, p-value = 0.1066
```

Para el segundo modelo:

```
shapiro.test(vResid2)
```

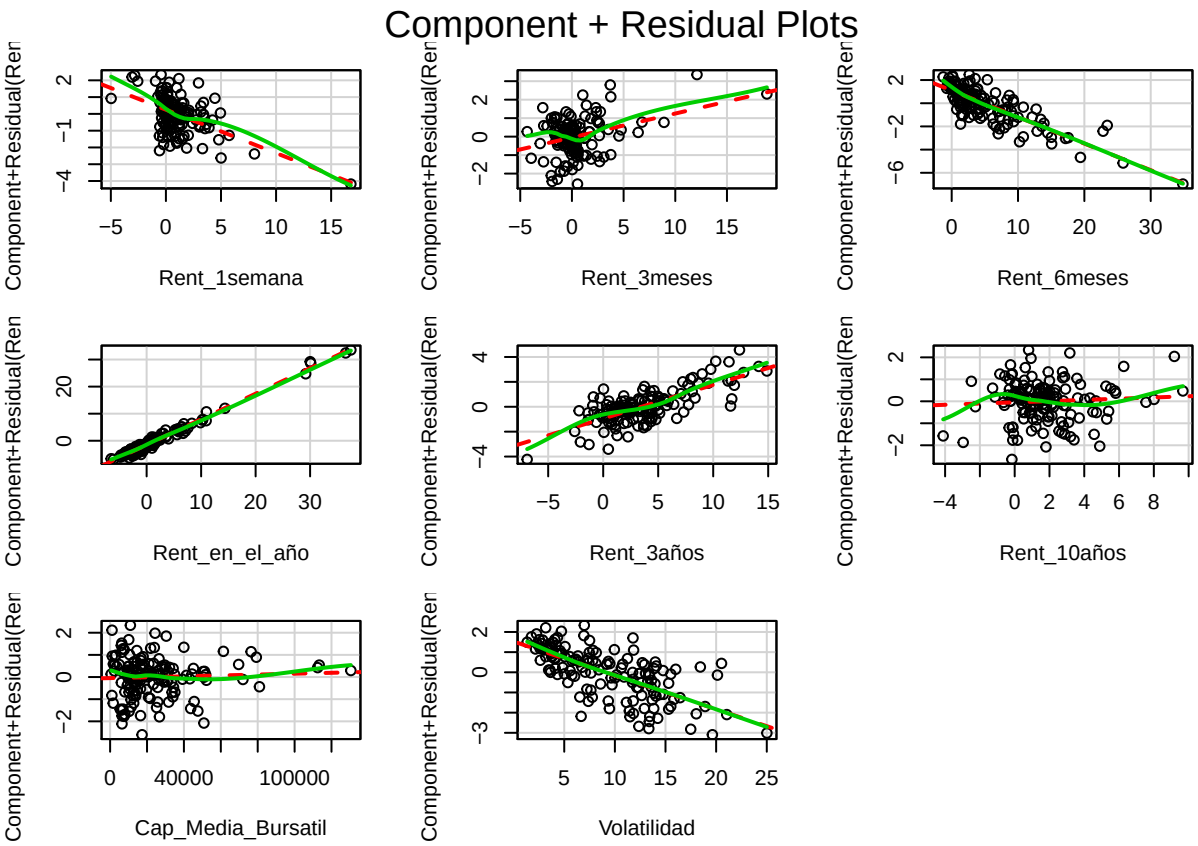
```
##
##  Shapiro-Wilk normality test
##
## data:  vResid2
## W = 0.98419, p-value = 0.08769
```

En ambos casos el test demuestra que siguen una distribucion normal.

Linealidad

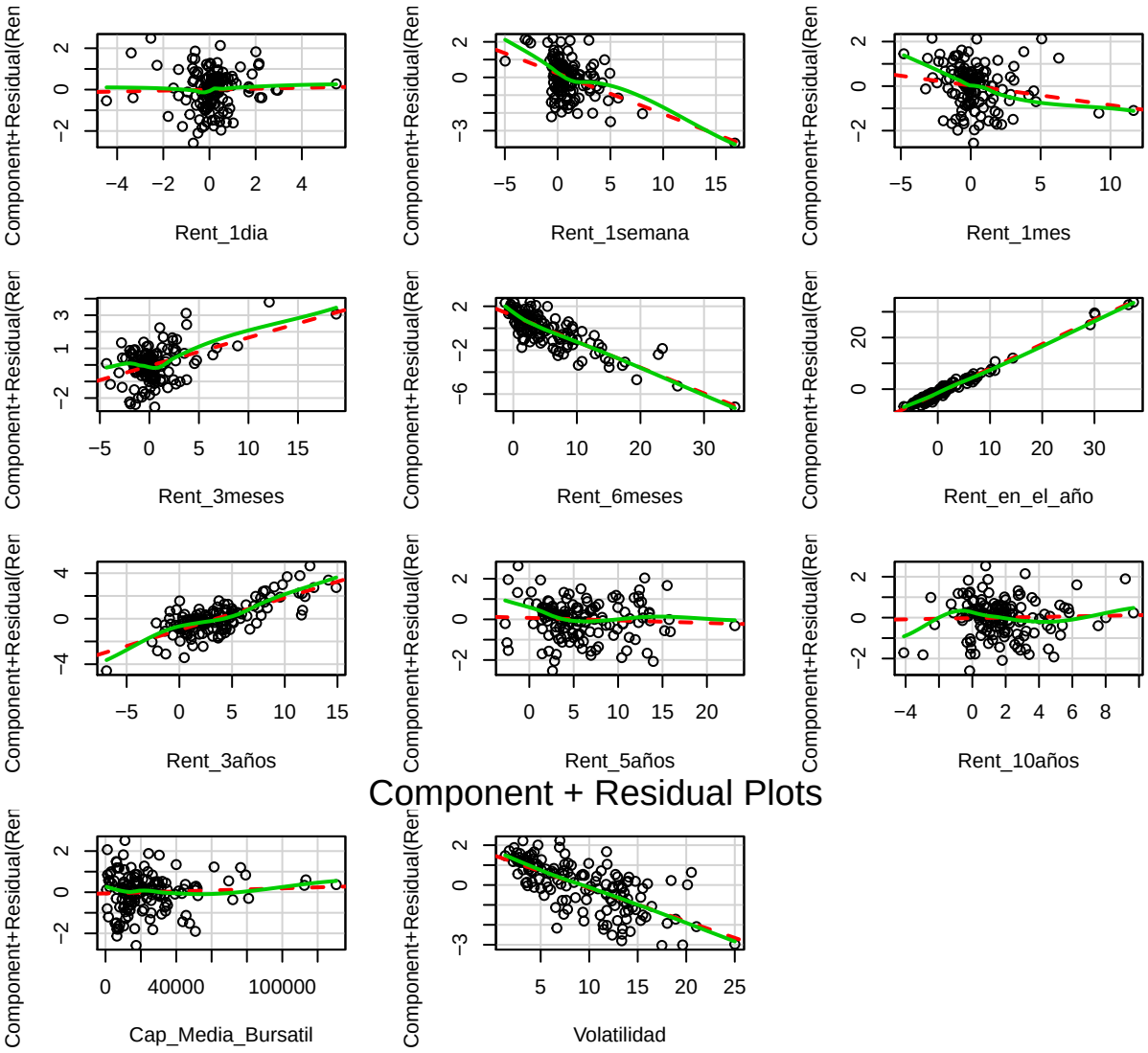
Para el primer modelo:

```
crPlots(modelo1)
```



Para el segundo modelo:

```
crPlots(modelo2)
```



En los graficos superiores se oberva variable a variable la relacion lineal. En general, resulta exisistir bastante dispersion y un efecto no lineal.

Varianza constante - Homocedasticidad

El test de varanzia es homocedasticidad, y si no se cumple, se llama heterocedasticidad. Indica si existen zonas donde los errores son mas grandes, debido probablemente a que alguna variable tiene desperfectos. Se utiliza el test de Breusch-Pagan.

Para el primer modelo:

```
ncvTest(modelo1)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 6.828105    Df = 1     p = 0.00897344
```

Para el segundo modelo:

```
ncvTest(modelo2)
```

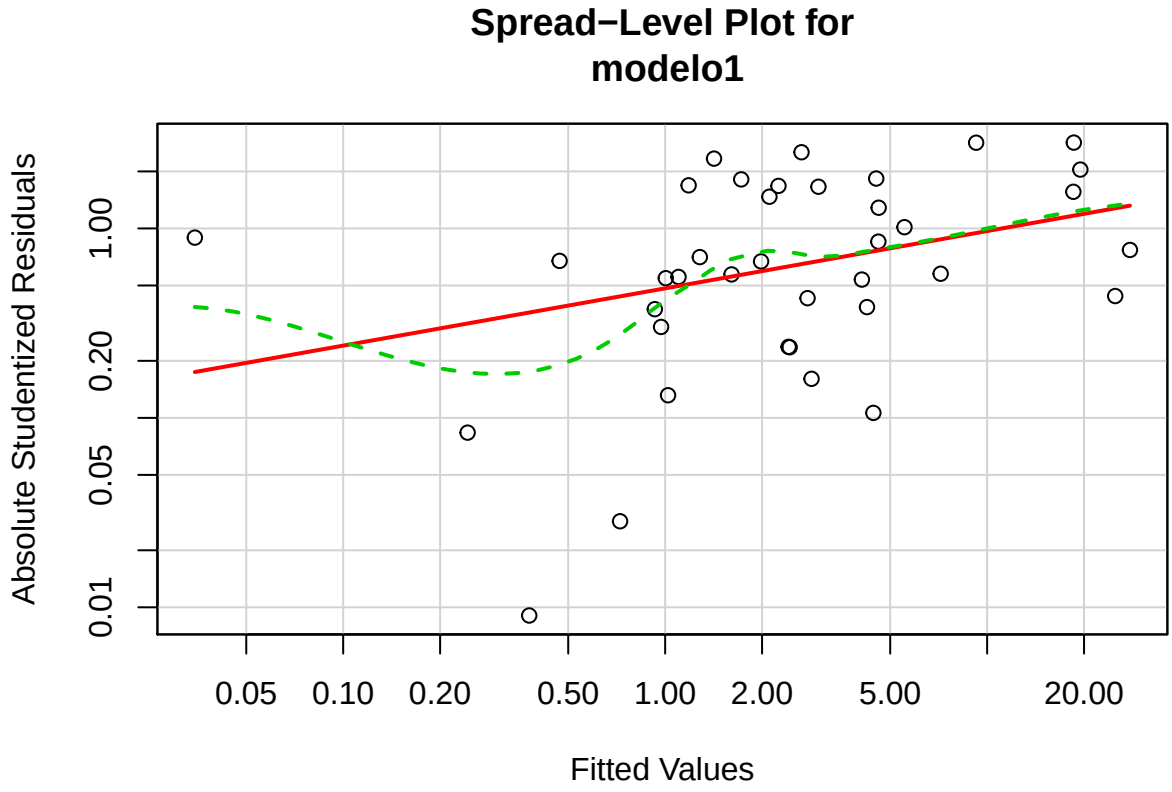
```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 8.319091    Df = 1     p = 0.003923054
```

Con valores tan pequeños se rechaza la hipotesis nula de que los residuos sean homocedasticos. No tiene varianza constante.

Graficamente para el primer modelo:

```
spreadLevelPlot(modelo1)
```

```
## Warning in spreadLevelPlot.lm(modelo1): 110 negative fitted values removed
```

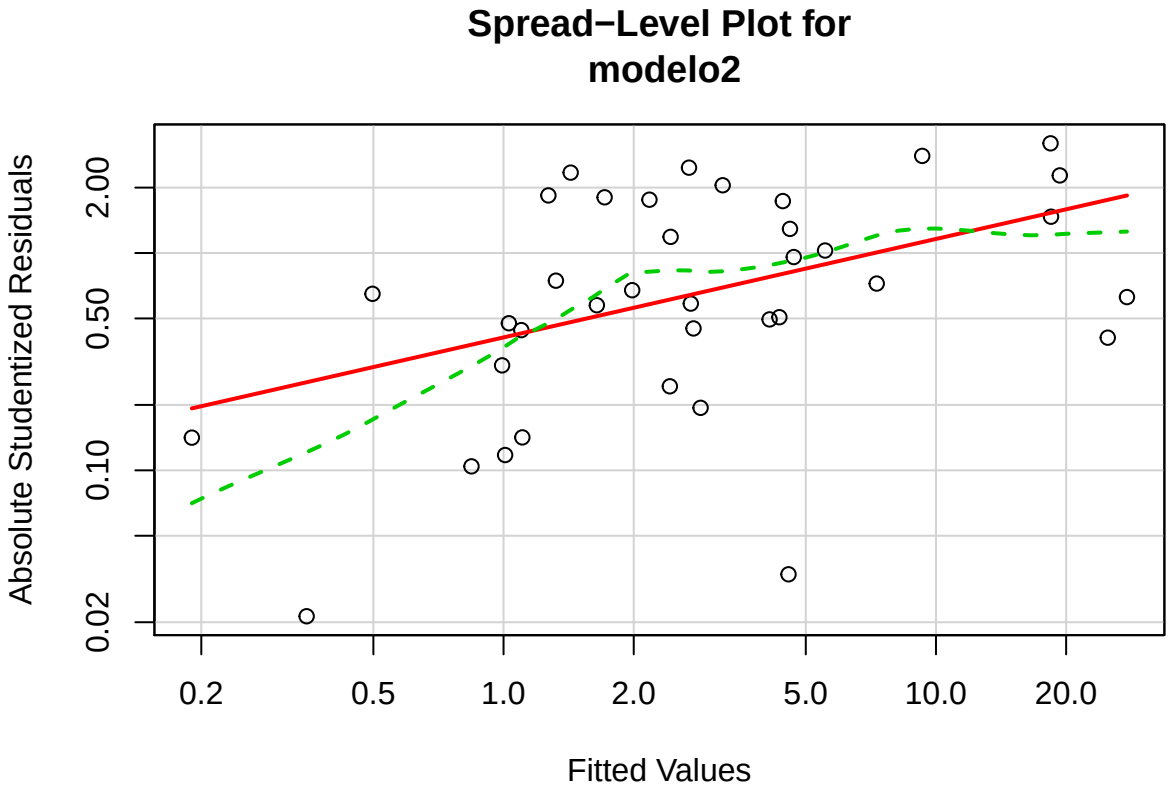



```
##
## Suggested power transformation:  0.6976421
```

Para el segundo modelo:

```
spreadLevelPlot(modelo2)
```

```
## Warning in spreadLevelPlot.lm(modelo2): 111 negative fitted values removed
```




```
##
## Suggested power transformation:  0.5469303
```

Validacion Global

```
library(gvlma)
```

```
gvmodel1 <- gvlma(modelo1)
summary(gvmodel1)
```

```
##
## Call:
## lm(formula = Rent_1año ~ +Rent_1semana + Rent_3meses + Rent_6meses +
##     Rent_en_el_año + Rent_3años + Rent_10años + Cap_Media_Bursatil +
##     Volatilidad, data = data, na.action = na.omit)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.59040 -0.52265  0.09385  0.51556  2.36222
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -7.075e-01  2.110e-01  -3.353  0.00103 **
## Rent_1semana      -2.586e-01  5.698e-02  -4.540 1.21e-05 ***
## Rent_3meses        1.296e-01  5.015e-02   2.584  0.01080 *
## Rent_6meses       -2.290e-01  2.938e-02  -7.792 1.39e-12 ***
## Rent_en_el_año     9.442e-01  1.955e-02  48.288  < 2e-16 ***
## Rent_3años         2.705e-01  3.723e-02   7.266 2.42e-11 ***
```

```
## Rent_10años         2.782e-02  5.884e-02   0.473  0.63713
## Cap_Media_Bursatil  1.988e-06  4.232e-06   0.470  0.63930
## Volatilidad        -1.679e-01  2.694e-02  -6.232 5.16e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.926 on 139 degrees of freedom
##   (352 observations deleted due to missingness)
## Multiple R-squared:  0.9723, Adjusted R-squared:  0.9707
## F-statistic:   610 on 8 and 139 DF,  p-value: < 2.2e-16
##
##
## ASSESSMENT OF THE LINEAR MODEL ASSUMPTIONS
## USING THE GLOBAL TEST ON 4 DEGREES-OF-FREEDOM:
## Level of Significance =  0.05
##
## Call:
##  gvlma(x = modelo1)
##
##                     Value p-value                Decision
## Global Stat        3.4525  0.4851 Assumptions acceptable.
## Skewness           1.3178  0.2510 Assumptions acceptable.
## Kurtosis           0.2697  0.6036 Assumptions acceptable.
## Link Function      1.2795  0.2580 Assumptions acceptable.
## Heteroscedasticity 0.5856  0.4441 Assumptions acceptable.
```

```
gvmodel2 <- gvlma(modelo2)
summary(gvmodel2)
```

```
##
## Call:
## lm(formula = Rent_1año ~ Rent_1dia + Rent_1semana + Rent_1mes +
##     Rent_3meses + Rent_6meses + Rent_en_el_año + Rent_3años +
##     Rent_5años + Rent_10años + Cap_Media_Bursatil + Volatilidad,
##     data = data, na.action = na.omit)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5690 -0.4604  0.1157  0.5068  2.5432
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -6.760e-01  2.153e-01  -3.139  0.00208 **
## Rent_1dia           2.252e-02  1.005e-01   0.224  0.82302
## Rent_1semana       -2.278e-01  7.235e-02  -3.149  0.00202 **
## Rent_1mes          -8.786e-02  9.408e-02  -0.934  0.35199
## Rent_3meses         1.710e-01  6.681e-02   2.560  0.01156 *
## Rent_6meses        -2.355e-01  3.078e-02  -7.651 3.28e-12 ***
## Rent_en_el_año      9.498e-01  2.544e-02  37.340  < 2e-16 ***
## Rent_3años          2.831e-01  5.868e-02   4.824 3.71e-06 ***
## Rent_5años         -1.247e-02  5.844e-02  -0.213  0.83137
## Rent_10años         1.469e-02  6.462e-02   0.227  0.82055
## Cap_Media_Bursatil  2.466e-06  4.341e-06   0.568  0.57096
## Volatilidad        -1.669e-01  3.349e-02  -4.983 1.87e-06 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9331 on 136 degrees of freedom
##   (352 observations deleted due to missingness)
## Multiple R-squared:  0.9725, Adjusted R-squared:  0.9703
## F-statistic:   437 on 11 and 136 DF,  p-value: < 2.2e-16
##
##
## ASSESSMENT OF THE LINEAR MODEL ASSUMPTIONS
## USING THE GLOBAL TEST ON 4 DEGREES-OF-FREEDOM:
## Level of Significance =  0.05
##
## Call:
##  gvlma(x = modelo2)
##
##                     Value p-value                Decision
## Global Stat        5.9657 0.20172 Assumptions acceptable.
## Skewness           1.1042 0.29335 Assumptions acceptable.
## Kurtosis           0.4897 0.48407 Assumptions acceptable.
## Link Function      3.6596 0.05575 Assumptions acceptable.
## Heteroscedasticity 0.7123 0.39868 Assumptions acceptable.
```

Mediante este test, se contrastan todas las hipotesis a la vez. Tanto para el grafico 1 como para el 2:

- Se aceptan las hipotesis de normalidad.
- Se considera aceptable el test de heterocedasticidad, contrariamente a lo que se habia visto en el apartado anterior.

Deteccion de multicolinealidad

Cuando la raiz cuadrada del Factor de Inflacion de Varianza es mayor que 2, se considera que existen problemas de multicolinealidad.

Para el primer modelo:

```
sqrt(vif(modelo1)) > 2
```

```
##       Rent_1semana        Rent_3meses        Rent_6meses
##              FALSE              FALSE               TRUE
##     Rent_en_el_año         Rent_3años        Rent_10años
##              FALSE              FALSE              FALSE
## Cap_Media_Bursatil        Volatilidad
##              FALSE              FALSE
```

Para el segundo modelo:

```
sqrt(vif(modelo2)) > 2
```

```
##          Rent_1dia       Rent_1semana          Rent_1mes
##              FALSE              FALSE               TRUE
##        Rent_3meses        Rent_6meses     Rent_en_el_año
##               TRUE               TRUE               TRUE
##         Rent_3años         Rent_5años        Rent_10años
##               TRUE               TRUE              FALSE
## Cap_Media_Bursatil        Volatilidad
##              FALSE               TRUE
```

En este aspecto resulta mejor el modelo 1 que el 2. En el modelo 1 existe multicolinealidad tan solo en la variable 'Rentabilidad a 6 meses', mientras que en el modelo 2 se da en distintas variables explicativas.

Observaciones anomalas

Mediante el test de Bonferonni.

Para el primer modelo:

```
outlierTest(modelo1)
```

```
##
## No Studentized residuals with Bonferonni p < 0.05
## Largest |rstudent|:
##      rstudent unadjusted p-value Bonferonni p
## 458 -2.905868          0.0042676       0.6316
```

Para el segundo modelo:

```
outlierTest(modelo2)
```

```
##
## No Studentized residuals with Bonferonni p < 0.05
## Largest |rstudent|:
##   rstudent unadjusted p-value Bonferonni p
## 4  3.19715           0.001729      0.25589
```

En ambos casos encontramos valores atipicos, en la posicion 458 y 4 para cada uno de los modelos respectivamente. Se puede representar en un grafico:

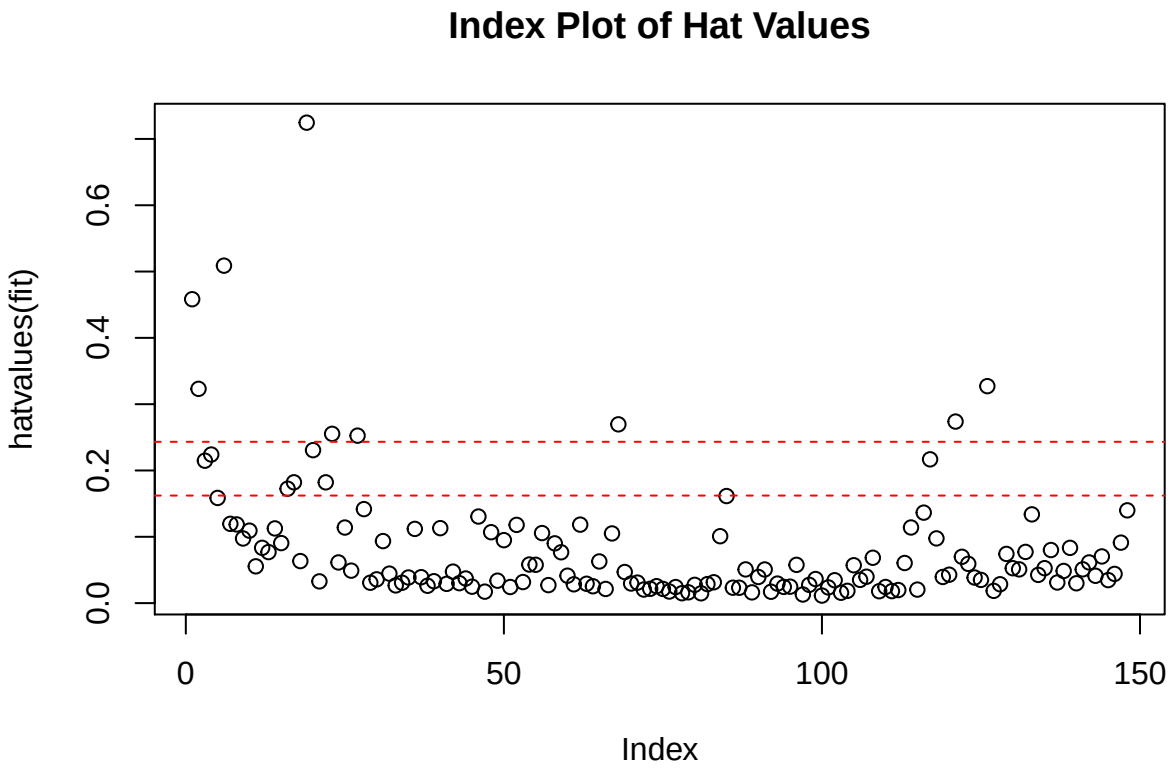
Para el modelo 1:

```
hat.plot <- function(fit) {
  p <- length(coefficients(fit))
  n <- length(fitted(fit))
  plot(hatvalues(fit), main="Index Plot of Hat Values")
  abline(h=c(2,3)*p/n, col="red", lty=2)
  identify(1:n, hatvalues(fit), names(hatvalues(fit)))
}
hat.plot(modelo1)
```


```
## integer(0)
```

Para el modelo 2:

```
hat.plot(modelo2)
```

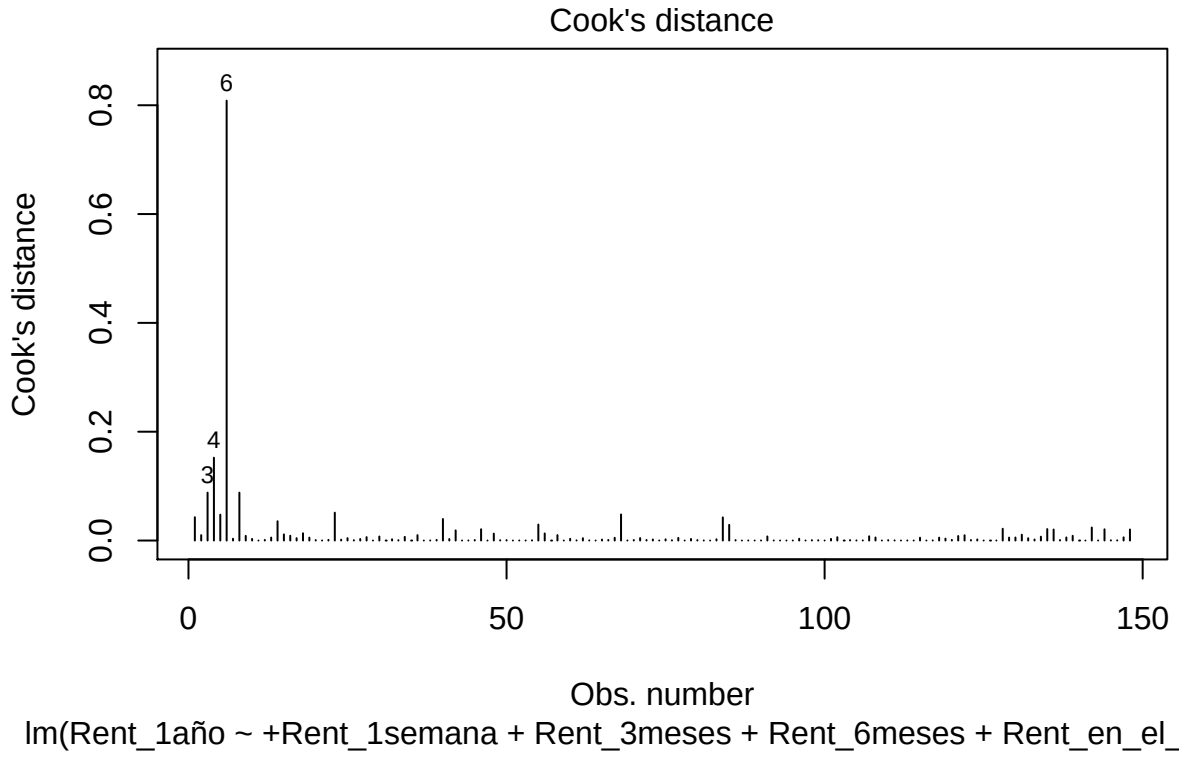



```
## integer(0)
```

Graficos de la distancia de Cook:

Para el modelo 1:

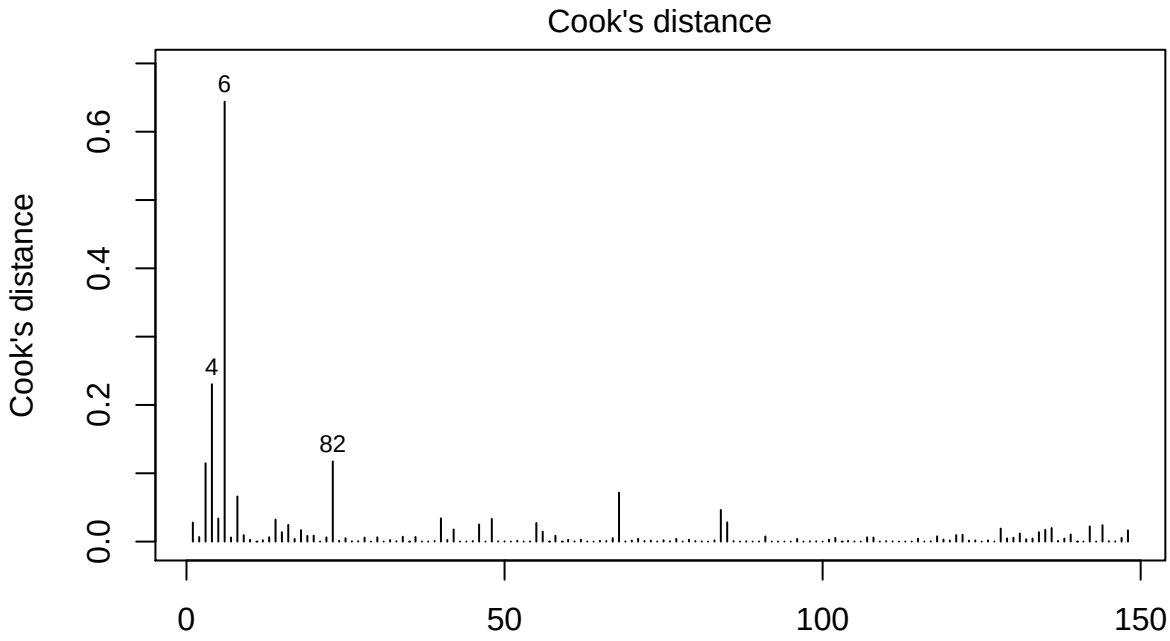
```
cutoff <- 4/(nrow(data)-length(modelo1$coefficients)-2)
plot(modelo1, which=4, cook.levels=cutoff)
```




```
# abline(h=cutoff, lty=1, col="red") # no representa la linea por algun motivo
```

Para el modelo 2:

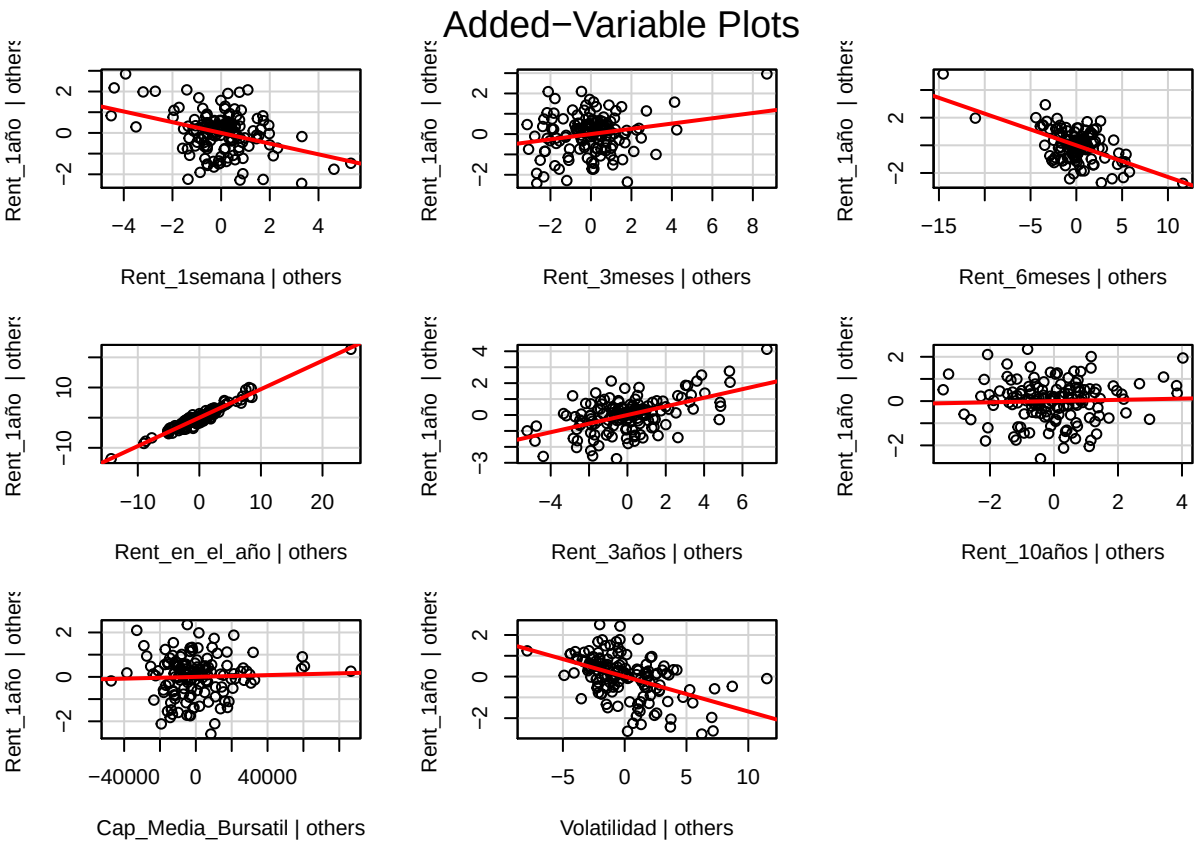
```
cutoff <- 4/(nrow(data)-length(modelo2$coefficients)-2)
plot(modelo2, which=4, cook.levels=cutoff)
```




```
# abline(h=cutoff, lty=1, col="red") # no representa la linea por algun motivo
```

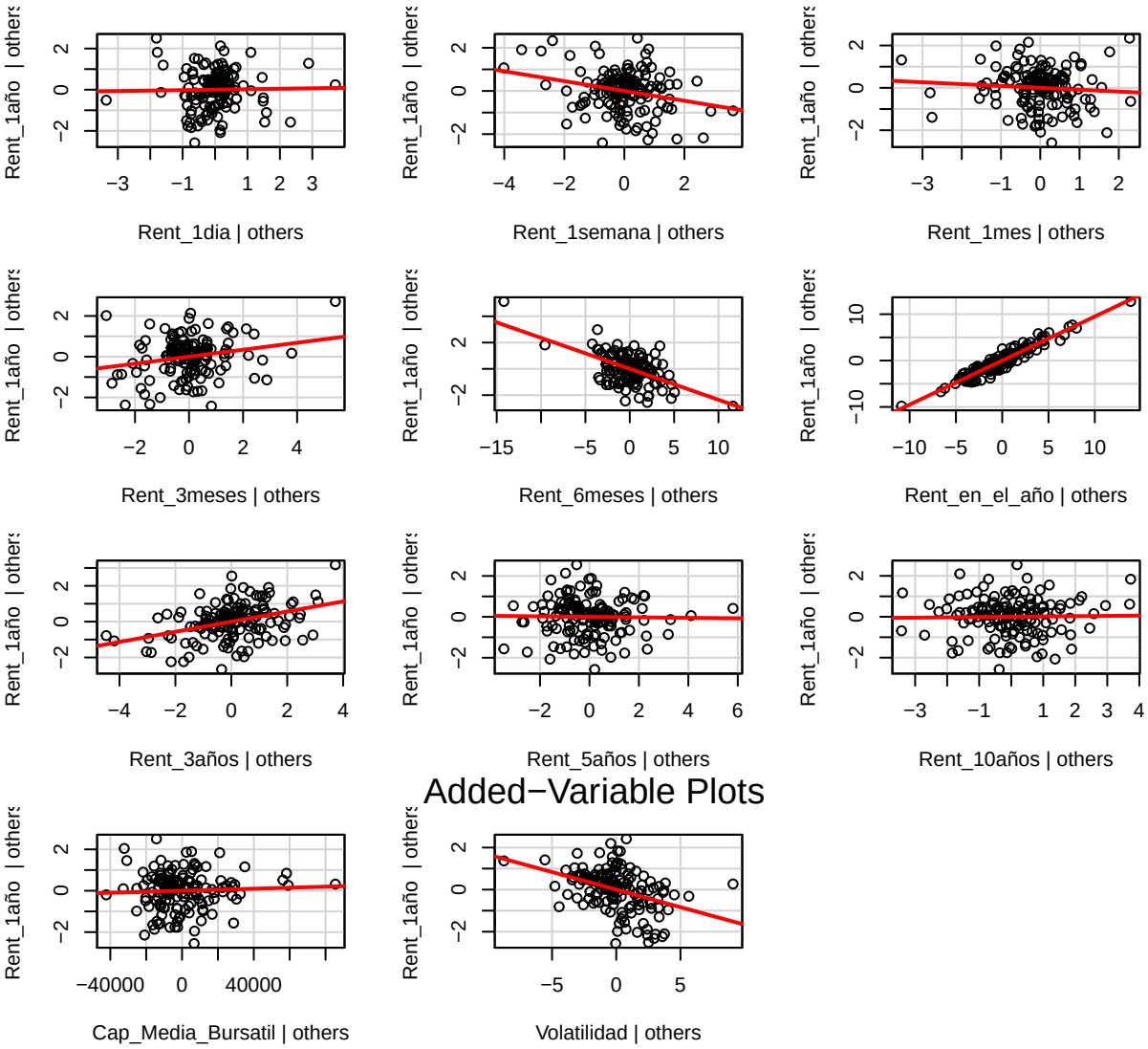
En ambos casos además aparecen valores influyentes. Habria que analizarlos.

```
avPlots(modelo1, ask=FALSE, id.method="identify")
```



Para el modelo 2:

```
avPlots(modelo2, ask=FALSE, id.method="identify")
```

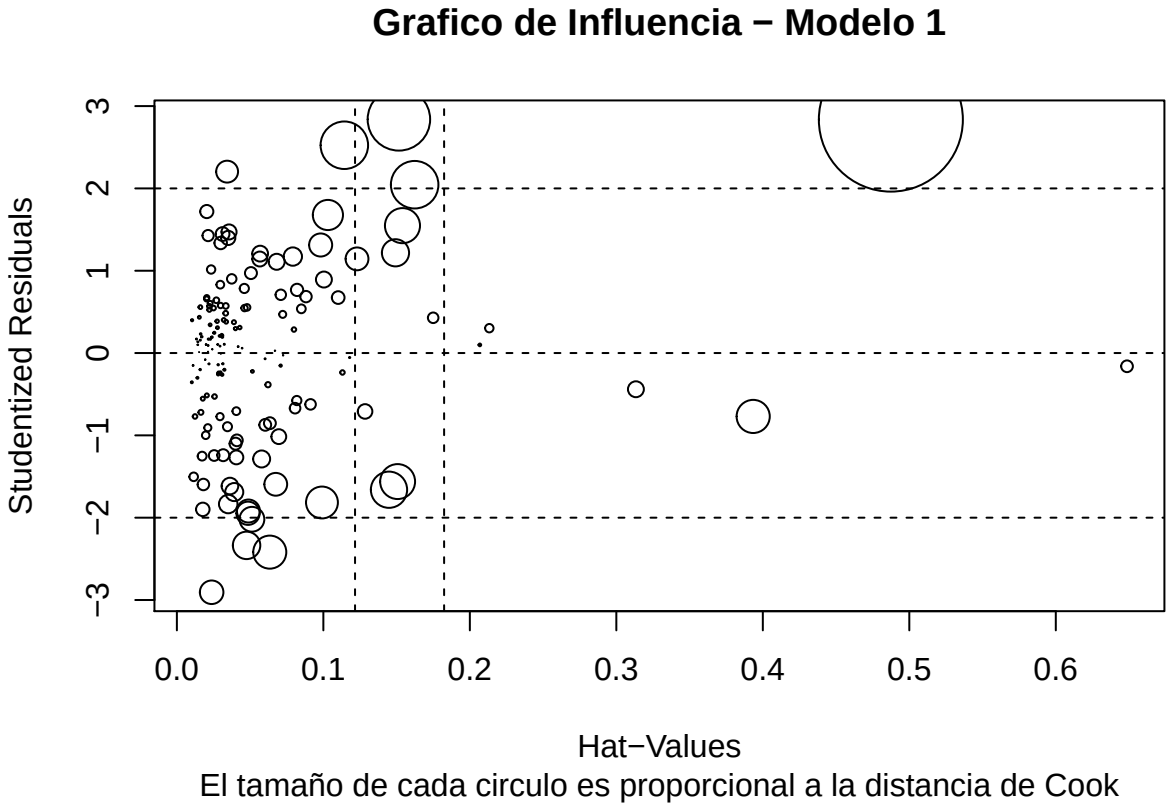


Bastante dispersion en todas las variables, a excepcion de la 'Rentabilidad en el año', que por si misma, guarda bastante relacion con la variable explicada 'Rentabilidad anual'.

Se puede realizar para cada modelo un grafico de influencia, para conocer los valores que estan distorsionando el modelo.

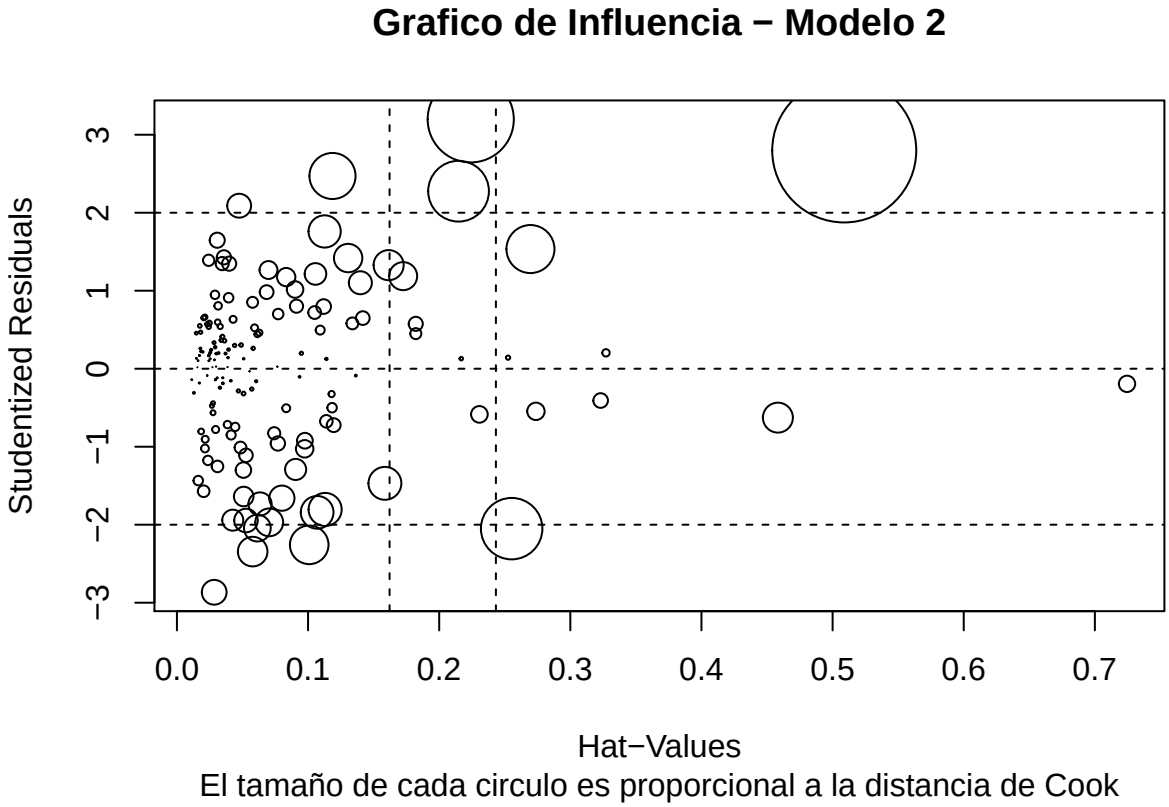
Para el primer modelo:

```
influencePlot(modelo1, id.method="identify", main="Grafico de Influencia - Modelo 1",
              sub="El tamaño de cada circulo es proporcional a la distancia de Cook" )
```



Para el segundo modelo:

```
influencePlot(modelo2, id.method="identify", main="Grafico de Influencia - Modelo 2",
              sub="El tamaño de cada circulo es proporcional a la distancia de Cook" )
```



Notese en ambos casos la existencia de valores extremos que cuentan con gran influencia (mayor tamaño del circulo segun la distancia de Cook). Cerca de los circulos mas concentrados, tambien se encuentran valores atipicos.

Seleccion de Variables

El modelo 1 esta anidado dentro del 2, es decir, contiene alguna de las variables explicativas del 2.

```
anova(modelo1, modelo2)
```

```
## Analysis of Variance Table
##
## Model 1: Rent_1año ~ +Rent_1semana + Rent_3meses + Rent_6meses + Rent_en_el_año +
##     Rent_3años + Rent_10años + Cap_Media_Bursatil + Volatilidad
## Model 2: Rent_1año ~ Rent_1dia + Rent_1semana + Rent_1mes + Rent_3meses +
##     Rent_6meses + Rent_en_el_año + Rent_3años + Rent_5años +
##     Rent_10años + Cap_Media_Bursatil + Volatilidad
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    139 119.20
## 2    136 118.42  3   0.78446 0.3003 0.8251
```

La hipotesis nula es que el modelo es valido, es decir, que las variables 'extra' que tiene el modelo 2 frente al 1 no son necesarias, y por lo tanto se acepta el modelo 1.

Para cointinuar con la seleccion de modelos se utilizan, ademas, dos criterios:

AIC: se escoge el modelo 1 con un valor AIC menor.

```
AIC(modelo1, modelo2)
```

```
##         df      AIC
```

```
## modelo1 10 407.9795
## modelo2 13 413.0023
```

BIC: se escoge el modelo 1 con un valor BIC menor.

```
BIC(modelo1, modelo2)
```

```
##         df      BIC
## modelo1 10 437.9517
## modelo2 13 451.9661
```

Metodos de Seleccion

Best Subset:

Consiste en estimar todas las regresiones posibles con las combinaciones de los p regresores.

```
library (leaps)
```

```
## Warning: package 'leaps' was built under R version 3.2.5
```

Modelo 1:

```
regfit.full=regsubsets(Rent_1año ~ Rent_1semana
             + Rent_3meses + Rent_6meses
              + Rent_en_el_año + Rent_3años + Rent_10años
              + Cap_Media_Bursatil
              + Volatilidad,
              data = data, na.action = na.omit)
reg.summary=summary(regfit.full)
reg.summary
```

```
## Subset selection object
## Call: regsubsets.formula(Rent_1año ~ Rent_1semana + Rent_3meses +
##     Rent_6meses + Rent_en_el_año + Rent_3años + Rent_10años +
##     Cap_Media_Bursatil + Volatilidad, data = data, na.action = na.omit)
## 8 Variables  (and intercept)
##                     Forced in Forced out
## Rent_1semana            FALSE      FALSE
## Rent_3meses             FALSE      FALSE
## Rent_6meses             FALSE      FALSE
## Rent_en_el_año          FALSE      FALSE
## Rent_3años              FALSE      FALSE
## Rent_10años             FALSE      FALSE
## Cap_Media_Bursatil      FALSE      FALSE
## Volatilidad             FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: exhaustive
##          Rent_1semana Rent_3meses Rent_6meses Rent_en_el_año Rent_3años
## 1  ( 1 ) " "          " "         " "         "*"            " "
## 2  ( 1 ) " "          " "         "*"         "*"            " "
## 3  ( 1 ) " "          " "         "*"         "*"            "*"
## 4  ( 1 ) " "          " "         "*"         "*"            "*"
## 5  ( 1 ) "*"          " "         "*"         "*"            "*"
## 6  ( 1 ) "*"          "*"         "*"         "*"            "*"
## 7  ( 1 ) "*"          "*"         "*"         "*"            "*"
## 8  ( 1 ) "*"          "*"         "*"         "*"            "*"
```

```
##          Rent_10años Cap_Media_Bursatil Volatilidad
## 1  ( 1 ) " "         " "                " "
## 2  ( 1 ) " "         " "                " "
## 3  ( 1 ) " "         " "                " "
## 4  ( 1 ) " "         " "                "*"
## 5  ( 1 ) " "         " "                "*"
## 6  ( 1 ) " "         " "                "*"
## 7  ( 1 ) "*"         " "                "*"
## 8  ( 1 ) "*"         "*"                "*"
```

El resultado del metodo subset es una matriz que incluye las mejores variables predictoras para 8 modelos diferentes. Cuando el modelo tiene una variable, la mejor es la 'Rentabilidad en el año', y esta se repite para los demas modelos. Cuando el modelo tiene dos variables, las dos mejores predictoras son la 'Rentabilidad en el año' y la 'Rentabilidad a 6 meses'. Asi continuaria para el resto de modelos de este primer modelo.

Modelo 2:

```
regfit.full=regsubsets(Rent_1año ~ Rent_1dia + Rent_1semana
              + Rent_1mes + Rent_3meses + Rent_6meses
              + Rent_en_el_año + Rent_3años + Rent_5años + Rent_10años
              + Cap_Media_Bursatil
              + Volatilidad,
              data = data, na.action = na.omit)
reg.summary=summary(regfit.full)
reg.summary
```

```
## Subset selection object
## Call: regsubsets.formula(Rent_1año ~ Rent_1dia + Rent_1semana + Rent_1mes +
##     Rent_3meses + Rent_6meses + Rent_en_el_año + Rent_3años +
##     Rent_5años + Rent_10años + Cap_Media_Bursatil + Volatilidad,
##     data = data, na.action = na.omit)
## 11 Variables  (and intercept)
##                     Forced in Forced out
## Rent_1dia               FALSE      FALSE
## Rent_1semana            FALSE      FALSE
## Rent_1mes               FALSE      FALSE
## Rent_3meses             FALSE      FALSE
## Rent_6meses             FALSE      FALSE
## Rent_en_el_año          FALSE      FALSE
## Rent_3años              FALSE      FALSE
## Rent_5años              FALSE      FALSE
## Rent_10años             FALSE      FALSE
## Cap_Media_Bursatil      FALSE      FALSE
## Volatilidad             FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: exhaustive
##          Rent_1dia Rent_1semana Rent_1mes Rent_3meses Rent_6meses
## 1  ( 1 ) " "       " "          " "       " "         " "
## 2  ( 1 ) " "       " "          " "       " "         "*"
## 3  ( 1 ) " "       " "          " "       " "         "*"
## 4  ( 1 ) " "       " "          " "       " "         "*"
## 5  ( 1 ) " "       "*"          " "       " "         "*"
## 6  ( 1 ) " "       "*"          " "       "*"         "*"
## 7  ( 1 ) " "       "*"          "*"       "*"         "*"
## 8  ( 1 ) " "       "*"          "*"       "*"         "*"
##          Rent_en_el_año Rent_3años Rent_5años Rent_10años
```

```
## 1  ( 1 ) "*"            " "         " "         " "
## 2  ( 1 ) "*"            " "         " "         " "
## 3  ( 1 ) "*"            "*"         " "         " "
## 4  ( 1 ) "*"            "*"         " "         " "
## 5  ( 1 ) "*"            "*"         " "         " "
## 6  ( 1 ) "*"            "*"         " "         " "
## 7  ( 1 ) "*"            "*"         " "         " "
## 8  ( 1 ) "*"            "*"         " "         " "
##          Cap_Media_Bursatil Volatilidad
## 1  ( 1 ) " "                " "
## 2  ( 1 ) " "                " "
## 3  ( 1 ) " "                " "
## 4  ( 1 ) " "                "*"
## 5  ( 1 ) " "                "*"
## 6  ( 1 ) " "                "*"
## 7  ( 1 ) " "                "*"
## 8  ( 1 ) "*"                "*"
```

El resultado en este segundo modelo es similar. Las dos mejores predictoras son la 'Rentabilidad en el año' y la 'Rentabilidad en 6 meses'. Por el contrario, en las 8 variantes de modelos, las variables 'Rentabilidad en 3 años' y 'Rentabilidad en 10 años' no se incluyen en ningun caso. Probablemente seria necesario preguntarse si son necesarias estas variables para una finalidad predictiva del modelo.

Forward Stepwise

```
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 3.2.5
```

Modelo 1:

```
regfit.fwd=regsubsets(Rent_1año ~ Rent_1semana
             + Rent_3meses + Rent_6meses
              + Rent_en_el_año + Rent_3años + Rent_10años
              + Cap_Media_Bursatil
              + Volatilidad,
              data = data, na.action = na.omit, method = "forward")
summary (regfit.fwd )
```

```
## Subset selection object
## Call: regsubsets.formula(Rent_1año ~ Rent_1semana + Rent_3meses +
##     Rent_6meses + Rent_en_el_año + Rent_3años + Rent_10años +
##     Cap_Media_Bursatil + Volatilidad, data = data, na.action = na.omit,
##     method = "forward")
## 8 Variables  (and intercept)
##                    Forced in Forced out
## Rent_1semana           FALSE      FALSE
## Rent_3meses            FALSE      FALSE
## Rent_6meses            FALSE      FALSE
## Rent_en_el_año         FALSE      FALSE
## Rent_3años             FALSE      FALSE
## Rent_10años            FALSE      FALSE
## Cap_Media_Bursatil     FALSE      FALSE
## Volatilidad            FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: forward
```

```
##          Rent_1semana Rent_3meses Rent_6meses Rent_en_el_año Rent_3años
## 1  ( 1 ) " "          " "         " "         "*"            " "
## 2  ( 1 ) " "          " "         "*"         "*"            " "
## 3  ( 1 ) " "          " "         "*"         "*"            "*"
## 4  ( 1 ) " "          " "         "*"         "*"            "*"
## 5  ( 1 ) "*"          " "         "*"         "*"            "*"
## 6  ( 1 ) "*"          "*"         "*"         "*"            "*"
## 7  ( 1 ) "*"          "*"         "*"         "*"            "*"
## 8  ( 1 ) "*"          "*"         "*"         "*"            "*"
##          Rent_10años Cap_Media_Bursatil Volatilidad
## 1  ( 1 ) " "         " "               " "
## 2  ( 1 ) " "         " "               " "
## 3  ( 1 ) " "         " "               " "
## 4  ( 1 ) " "         " "               "*"
## 5  ( 1 ) " "         " "               "*"
## 6  ( 1 ) " "         " "               "*"
## 7  ( 1 ) "*"         " "               "*"
## 8  ( 1 ) "*"         "*"               "*"
```

Modelo 2:

```
regfit.fwd=regsubsets(Rent_1año ~ Rent_1dia + Rent_1semana
              + Rent_1mes + Rent_3meses + Rent_6meses
              + Rent_en_el_año + Rent_3años + Rent_5años + Rent_10años
              + Cap_Media_Bursatil
              + Volatilidad,
              data = data, na.action = na.omit, method = "forward")
summary (regfit.fwd )
```

```
## Subset selection object
## Call: regsubsets.formula(Rent_1año ~ Rent_1dia + Rent_1semana + Rent_1mes +
##     Rent_3meses + Rent_6meses + Rent_en_el_año + Rent_3años +
##     Rent_5años + Rent_10años + Cap_Media_Bursatil + Volatilidad,
##     data = data, na.action = na.omit, method = "forward")
## 11 Variables  (and intercept)
##                    Forced in Forced out
## Rent_1dia              FALSE      FALSE
## Rent_1semana           FALSE      FALSE
## Rent_1mes              FALSE      FALSE
## Rent_3meses            FALSE      FALSE
## Rent_6meses            FALSE      FALSE
## Rent_en_el_año         FALSE      FALSE
## Rent_3años             FALSE      FALSE
## Rent_5años             FALSE      FALSE
## Rent_10años            FALSE      FALSE
## Cap_Media_Bursatil     FALSE      FALSE
## Volatilidad            FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: forward
##          Rent_1dia Rent_1semana Rent_1mes Rent_3meses Rent_6meses
## 1  ( 1 ) " "       " "          " "       " "         " "
## 2  ( 1 ) " "       " "          " "       " "         "*"
## 3  ( 1 ) " "       " "          " "       " "         "*"
## 4  ( 1 ) " "       " "          " "       " "         "*"
## 5  ( 1 ) " "       "*"          " "       " "         "*"
```

```
## 6  ( 1 ) " "       "*"           " "       "*"          "*"
## 7  ( 1 ) " "       "*"           "*"       "*"          "*"
## 8  ( 1 ) " "       "*"           "*"       "*"          "*"
##          Rent_en_el_año Rent_3años Rent_5años Rent_10años
## 1  ( 1 ) "*"            " "        " "        " "
## 2  ( 1 ) "*"            " "        " "        " "
## 3  ( 1 ) "*"            "*"        " "        " "
## 4  ( 1 ) "*"            "*"        " "        " "
## 5  ( 1 ) "*"            "*"        " "        " "
## 6  ( 1 ) "*"            "*"        " "        " "
## 7  ( 1 ) "*"            "*"        " "        " "
## 8  ( 1 ) "*"            "*"        " "        " "
##          Cap_Media_Bursatil Volatilidad
## 1  ( 1 ) " "                " "
## 2  ( 1 ) " "                " "
## 3  ( 1 ) " "                " "
## 4  ( 1 ) " "                "*"
## 5  ( 1 ) " "                "*"
## 6  ( 1 ) " "                "*"
## 7  ( 1 ) " "                "*"
## 8  ( 1 ) "*"                "*"
```

Aunque los resultados son iguales que con el metodo Best Subset, este sistema comienza con un modelo que no incluye ningún regresor y se van añadiendo regresores de uno en uno. En cada etapa, la variable adicional que más mejora el modelo se incluye en el.

Backward Stepwise

```
library(MASS)
```

```
#stepAIC(modelo1, direction="backward")
```

A partir de este momento tengo problemas para continuar con los modelos que habia estimado inicialmente. El error que aparece al llamar la funcion stepAIC se debe a que encuentra missing values (NA) durante el proceso, en las variables 'Rentabilidad a 3 años', 'Rentabilidad a 5 años', 'Rentabilidad a 10 años', 'Capitalizacion Media Bursatil' y 'Volatilidad'.

Error in stepAIC(modelo1, direction = "backward") : number of rows in use has changed: remove missing values?

La solucion para continuar con el proceso es llamando a la libreria 'rminer'. Con ella se podria realizar el metodo de 'Nearest Neighbor Hot-Dock Imputation', mediante el cual se rellenan los NA con datos que podrian ser equivalentes.

Sin embargo y por desgracia, al importar la libreria 'rminer' no se instala una dependencia necesaria llamada 'xgboost'. Al intentar instalar la dependencia por separado salta otro error refiriendose que esa dependencia ya no esta disponible para mi version de R.

He realizado consultas en distintas paginas web que comentaban este problema y ninguna me ha sido de ayuda. Por lo visto, esta dependencia ya no existe en el repositorio CRAN y dificulta su uso en versiones mas recientes de R. He probado 'soluciones' que recomendaban usuarios y no me han resuelto el problema.

Frente a este inconveniente, pense en rellenar los NA con medias de las columnas en cuestion mediante bucles, como por ejemplo, el bucle para el caso de la variable 'Rentabilidad en 3 años' seria:

```
media3años <- mean(na.omit(data$Rent_3años))

for (i in 1:length(data$Rent_3años)) {
```

```
  if (is.na(data$Rent_3años[i] == TRUE)) {
    data$Rent_3años[i] = media3años
  }
}
```

El resto de los bucles se programarian simplemente cambiando el nombre a las variables y tendrian la misma estructura.

El resultado de su aplicacion ha sido malo, los modelos empeoraban mucho, y en realidad carece de sentido aplicar medias de distintos fondos de inversion a la rentabilidad de uno, cuando en teoria, entendemos que son independientes unos de otros.

La unica opcion que me ha quedado para terminar la practica ha sido buscar otros modelos que no incluyeran variables con NA. Para ello es necesario volver al incio de esta practica.

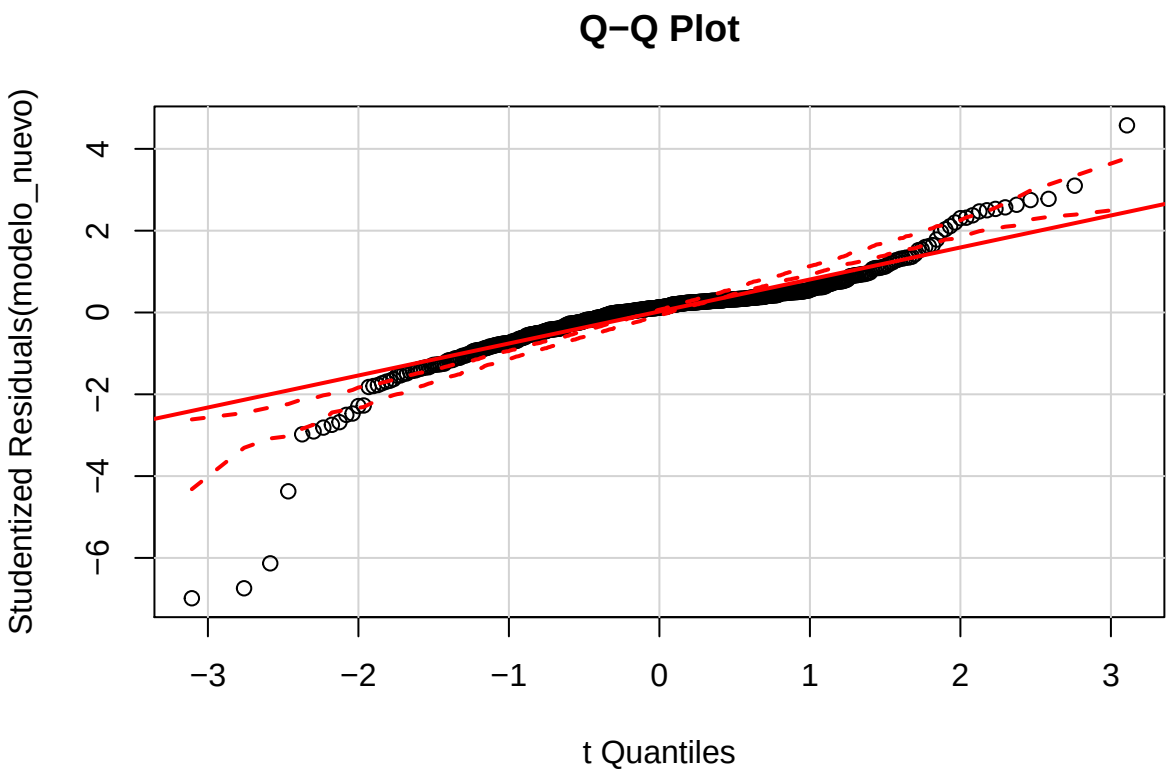
Resumire los test y graficos mas importantes del nuevo modelo con la finalidad de no repetir y explicar los mismos pasos:

Seleccion, demostracion y validacion del nuevo modelo

```
modelo_nuevo <- lm(Rent_1año ~ Rent_1semana
              + Rent_3meses + Rent_6meses
              + Rent_en_el_año,
              data = data, na.action = na.omit)
summary(modelo_nuevo)
```

```
##
## Call:
## lm(formula = Rent_1año ~ Rent_1semana + Rent_3meses + Rent_6meses +
##     Rent_en_el_año, data = data, na.action = na.omit)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.1337 -0.5498  0.1723  0.5241  6.1771
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.44797    0.07746  -5.783 1.30e-08 ***
## Rent_1semana   -0.25272    0.05977  -4.228 2.81e-05 ***
## Rent_3meses     0.10231    0.04652   2.199   0.0283 *
## Rent_6meses    -0.31127    0.02402 -12.959  < 2e-16 ***
## Rent_en_el_año  0.94686    0.02005  47.231  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.385 on 495 degrees of freedom
## Multiple R-squared:  0.8683, Adjusted R-squared:  0.8672
## F-statistic: 815.7 on 4 and 495 DF,  p-value: < 2.2e-16
```

```
qqPlot(modelo_nuevo, labels=row.names(data), id.method="identify",
       simulate=TRUE, main="Q-Q Plot")
```




```
vResid_nuevo <- resid(modelo_nuevo)
jbTest(vResid1)
```

```
## Warning in interpp.old(x, y, z, xo, yo, ncp = 0, extrap = FALSE, duplicate
## = "median", : interpp.old() is deprecated, future versions will only
## provide interpp()

## Warning in interpp.old(x, y, z, xo, yo, ncp = 0, extrap = FALSE, duplicate
## = "median", : interpp.old() is deprecated, future versions will only
## provide interpp()

##
## Title:
##  Jarque - Bera Normality Test
##
## Test Results:
##   PARAMETER:
##     Sample Size: 148
##   STATISTIC:
##     LM: 1.587
##     ALM: 1.796
##   P VALUE:
##     LM p-value: 0.393
##     ALM p-value: 0.356
##     Asymptotic: 0.452
##
## Description:
##  Fri Oct 27 02:01:10 2017 by user:
```

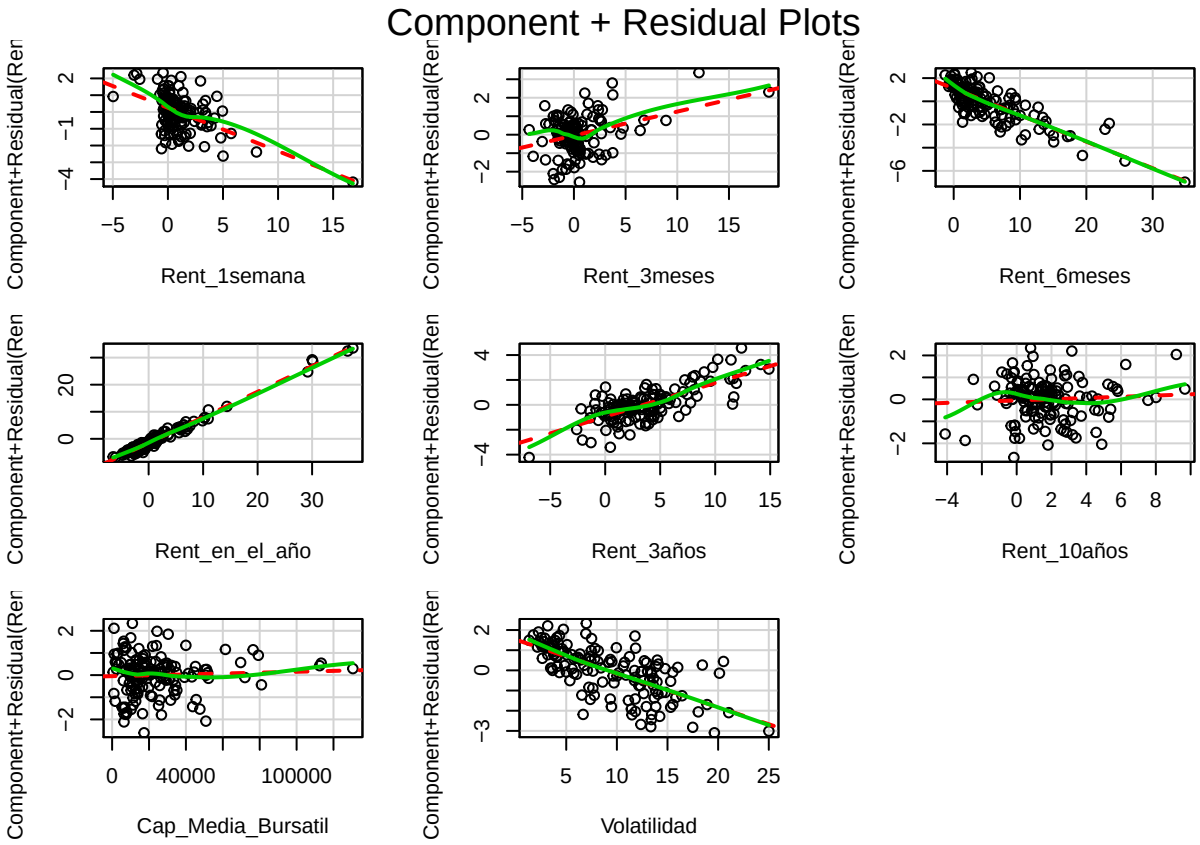
Se rechaza la hipotesis nula en el test de Jarque Bera, el modelo no sigue una distribucion normal.

```
shapiro.test(vResid_nuevo)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  vResid_nuevo
## W = 0.85037, p-value < 2.2e-16
```

Se rechaza la hipotesis nula en el test de Shapiro-Wilk, el modelo no sigue una distribucion normal.

```
crPlots(modelo1)
```



Se encuentra dispersion y no linealidad en las variables explicativas, pero si es necesario comparar, los resultados en cuanto a la linealidad son mejores en este modelo frente a los dos anteriores.

```
ncvTest(modelo_nuevo)
```

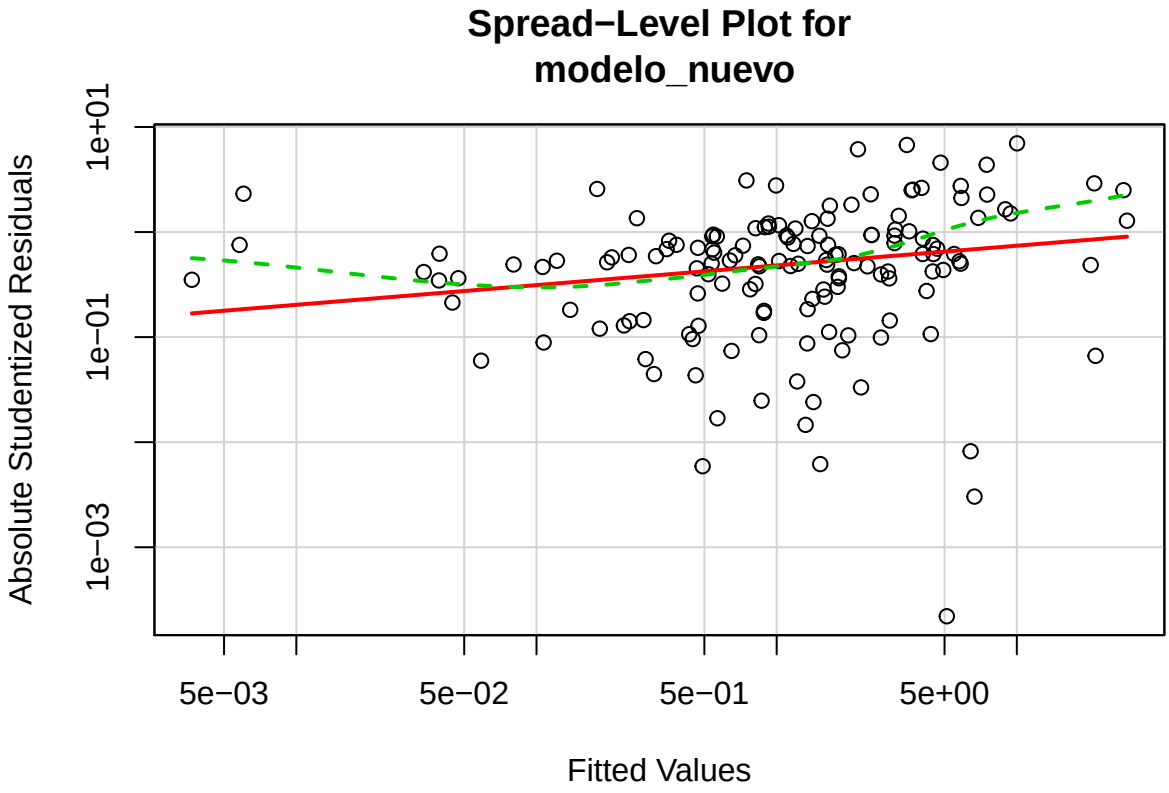
```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 142.708    Df = 1     p = 6.808955e-33
```

Se rechaza la hipotesis nula, no tiene varianza constante.

Graficamente:

```
spreadLevelPlot(modelo_nuevo)
```

```
## Warning in spreadLevelPlot.lm(modelo_nuevo): 352 negative fitted values
## removed
```




```
##
## Suggested power transformation:  0.8125221
```

Veamos la validacion global:

```
gvmodel_nuevo <- gvlma(modelo_nuevo)
summary(gvmodel_nuevo)
```

```
##
## Call:
## lm(formula = Rent_1año ~ Rent_1semana + Rent_3meses + Rent_6meses +
##     Rent_en_el_año, data = data, na.action = na.omit)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.1337 -0.5498  0.1723  0.5241  6.1771
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.44797    0.07746  -5.783 1.30e-08 ***
## Rent_1semana   -0.25272    0.05977  -4.228 2.81e-05 ***
## Rent_3meses     0.10231    0.04652   2.199   0.0283 *
## Rent_6meses    -0.31127    0.02402 -12.959  < 2e-16 ***
## Rent_en_el_año  0.94686    0.02005  47.231  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.385 on 495 degrees of freedom
## Multiple R-squared:  0.8683, Adjusted R-squared:  0.8672
## F-statistic: 815.7 on 4 and 495 DF,  p-value: < 2.2e-16
```

```
##
##
## ASSESSMENT OF THE LINEAR MODEL ASSUMPTIONS
## USING THE GLOBAL TEST ON 4 DEGREES-OF-FREEDOM:
## Level of Significance =  0.05
##
## Call:
##  gvlma(x = modelo_nuevo)
##
##                      Value   p-value                   Decision
## Global Stat        2625.7563 0.000e+00 Assumptions NOT satisfied!
## Skewness            186.9754 0.000e+00 Assumptions NOT satisfied!
## Kurtosis           2408.5023 0.000e+00 Assumptions NOT satisfied!
## Link Function        30.0422 4.227e-08 Assumptions NOT satisfied!
## Heteroscedasticity    0.2365 6.268e-01    Assumptions acceptable.
```

En este caso solo se acepta el test de heterocedasticidad. No es una distribucion normal.

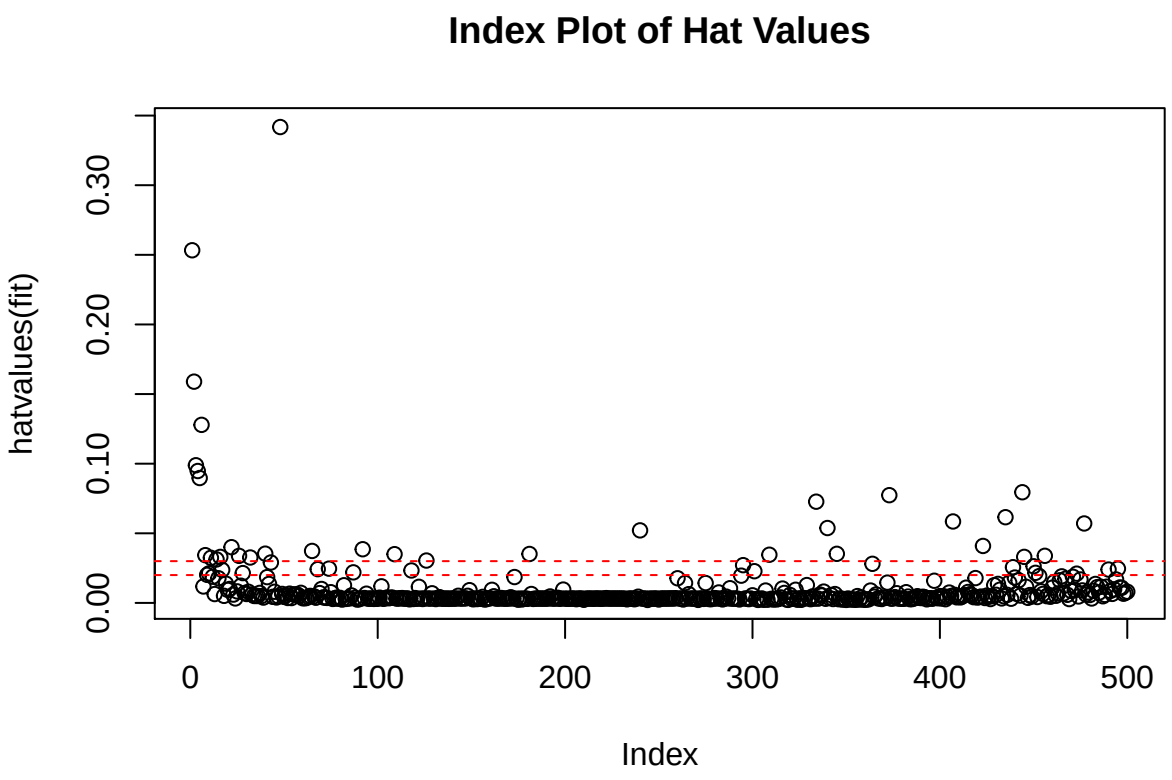
En cuanto a la multicolinealidad:

```
sqrt(vif(modelo_nuevo)) > 2
```

```
##   Rent_1semana    Rent_3meses    Rent_6meses Rent_en_el_año
##          FALSE          FALSE          FALSE          FALSE
```

No existe multicolinealidad, la muestra es suficientemente variada y grande. No hay informacion comun entre las variables. En lo relativo a los valores atipicos:

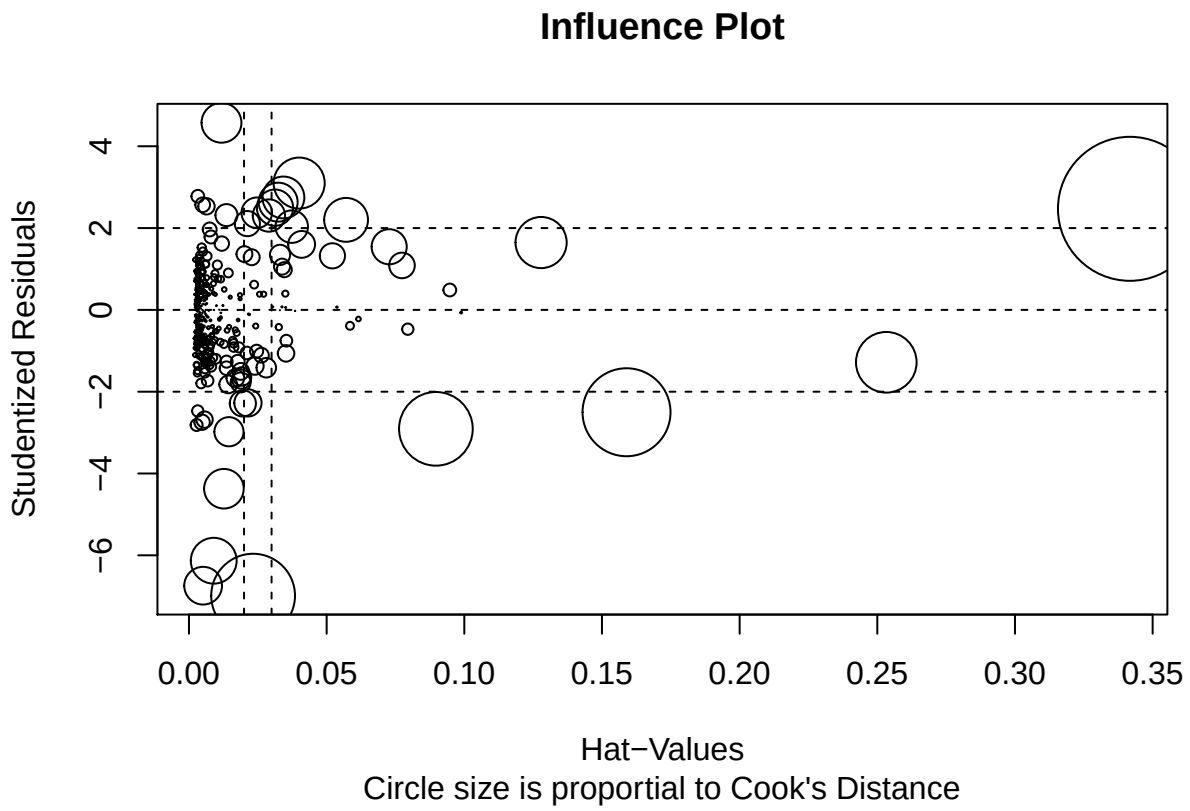
```
hat.plot(modelo_nuevo)
```




```
## integer(0)
```

La muestra esta sorprendentemente concentrada, pero mantiene valores atipicos e influyentes y extremos.

```
influencePlot(modelo_nuevo, id.method="identify", main="Influence Plot",
              sub="Circle size is proportial to Cook's Distance" )
```



Entremos con los modelos de seleccion:

Best Subset:

```
regfit.full_nuevo=regsubsets(Rent_1año ~ Rent_1semana
              + Rent_3meses + Rent_6meses
              + Rent_en_el_año,
              data = data, na.action = na.omit)
reg.summary=summary(regfit.full_nuevo)
reg.summary
```

```
## Subset selection object
## Call: regsubsets.formula(Rent_1año ~ Rent_1semana + Rent_3meses +
##     Rent_6meses + Rent_en_el_año, data = data, na.action = na.omit)
## 4 Variables  (and intercept)
##                Forced in Forced out
## Rent_1semana       FALSE      FALSE
## Rent_3meses        FALSE      FALSE
## Rent_6meses        FALSE      FALSE
## Rent_en_el_año     FALSE      FALSE
## 1 subsets of each size up to 4
## Selection Algorithm: exhaustive
##          Rent_1semana Rent_3meses Rent_6meses Rent_en_el_año
## 1  ( 1 ) " "          " "         " "         "*"
## 2  ( 1 ) " "          " "         "*"         "*"
## 3  ( 1 ) "*"          " "         "*"         "*"
```

```
## 4  ( 1 ) "*"          "*"         "*"         "*"
```

Forward Stepwise:

```
regfit.fwd_nuevo=regsubsets(Rent_1año ~ Rent_1semana
              + Rent_3meses + Rent_6meses
              + Rent_en_el_año,
              data = data, na.action = na.omit, method ="forward")
summary (regfit.fwd_nuevo)
```

```
## Subset selection object
## Call: regsubsets.formula(Rent_1año ~ Rent_1semana + Rent_3meses +
##     Rent_6meses + Rent_en_el_año, data = data, na.action = na.omit,
##     method = "forward")
## 4 Variables  (and intercept)
##                Forced in Forced out
## Rent_1semana       FALSE      FALSE
## Rent_3meses        FALSE      FALSE
## Rent_6meses        FALSE      FALSE
## Rent_en_el_año     FALSE      FALSE
## 1 subsets of each size up to 4
## Selection Algorithm: forward
##          Rent_1semana Rent_3meses Rent_6meses Rent_en_el_año
## 1  ( 1 ) " "          " "         " "         "*"
## 2  ( 1 ) " "          " "         "*"         "*"
## 3  ( 1 ) "*"          " "         "*"         "*"
## 4  ( 1 ) "*"          "*"         "*"         "*"
```

Variable predictora mas importante: 'Rentabilidad en el año' Variable predictora menos importante: 'Rentabilidad en 3 meses'

Ahora volvemos con lo que no se pudo realizar antes con los modelos iniciales:

Backward Stepwise

Empieza con un modelo que incluye todos los regresores y se van eliminando regresores de uno en uno. En cada etapa la variable que menos aporta al modelo se excluye y se vuelve a estimar.

```
library(MASS)
```

```
stepAIC(modelo_nuevo, direction = "backward")
```

```
## Start:  AIC=330.92
## Rent_1año ~ Rent_1semana + Rent_3meses + Rent_6meses + Rent_en_el_año
##
##                  Df Sum of Sq    RSS     AIC
## <none>                         950.0  330.92
## - Rent_3meses     1       9.3  959.3  333.78
## - Rent_1semana    1      34.3  984.3  346.66
## - Rent_6meses     1     322.3 1272.3  474.98
## - Rent_en_el_año  1    4281.1 5231.1 1181.89
```

```
##
## Call:
## lm(formula = Rent_1año ~ Rent_1semana + Rent_3meses + Rent_6meses +
##     Rent_en_el_año, data = data, na.action = na.omit)
##
```

```
## Coefficients:
##    (Intercept)    Rent_1semana     Rent_3meses     Rent_6meses
##        -0.4480         -0.2527          0.1023         -0.3113
## Rent_en_el_año
##         0.9469
```

Caulquier modificacion del modelo hace aumentar el AIC, por lo tanto, no necesita hacer simulaciones, dado que con la composicion actual de variables minimiza el AIC.

```
stepAIC(modelo_nuevo, direction = "both")
```

```
## Start:  AIC=330.92
## Rent_1año ~ Rent_1semana + Rent_3meses + Rent_6meses + Rent_en_el_año
##
##                  Df Sum of Sq    RSS     AIC
## <none>                         950.0  330.92
## - Rent_3meses     1       9.3  959.3  333.78
## - Rent_1semana    1      34.3  984.3  346.66
## - Rent_6meses     1     322.3 1272.3  474.98
## - Rent_en_el_año  1    4281.1 5231.1 1181.89
```

```
##
## Call:
## lm(formula = Rent_1año ~ Rent_1semana + Rent_3meses + Rent_6meses +
##     Rent_en_el_año, data = data, na.action = na.omit)
##
## Coefficients:
##    (Intercept)    Rent_1semana     Rent_3meses     Rent_6meses
##        -0.4480         -0.2527          0.1023         -0.3113
## Rent_en_el_año
##         0.9469
```

No se realizan modificaciones.

Cross-Validation

Validation Set

Se divide la muestra en 2 submuestras de forma aleatoria. Una de ellas se utilizara para el entrenamiento del modelo, y la otra para probar y verificar los resultados de la prediccion.

```
library(ISLR)
```

```
set.seed(123)
numData = nrow(data)
train = sample(numData ,numData/2)

regres.train = lm(Rent_1año ~ + Rent_1semana
                  + Rent_3meses + Rent_6meses
                  + Rent_en_el_año,
                  data = data, na.action = na.omit ,subset = train)

attach(data)

mean((Rent_1año - predict(regres.train, data))[-train ]^2)
```

```
## [1] 2.242588
```

Leave-One-Out Cross-Validation

Este metodo consiste en tomar una muestra con todos los datos menos uno. Se estima el modelo y se predice sobre el dato que se ha dejado fuera. Se repetira tantas veces como datos para estimar haya.

```
glm.fit_1 = glm(Rent_1año ~ + Rent_1semana
             + Rent_3meses + Rent_6meses
             + Rent_en_el_año,
             data = data,family = gaussian())

coef(glm.fit_1)
```

```
##    (Intercept)   Rent_1semana    Rent_3meses    Rent_6meses Rent_en_el_año
##     -0.4479713     -0.2527171      0.1023051     -0.3112663      0.9468637
```

```
library(boot)
```

```
## Warning: package 'boot' was built under R version 3.2.5
```

```
##
## Attaching package: 'boot'
```

```
## The following object is masked from 'package:car':
##
##     logit
```

```
cv.err =cv.glm(data ,glm.fit_1)
cv.err$delta
```

```
## [1] 1.993047 1.992948
```

Vector Delta: - Primer valor: Error cuadratico medio (promedio de los errores al cuadrado) - Segundo valor: Estimacion ajustada y corregida del sesgo

K-Fold Cross-Validation

El metodo consiste en dividir la muestra en k grupos o 'folds'.

Cada grupo constituye un conjunto de validación, de tal forma que se estima el modelo con los datos que no están en el grupo actual (resto de grupos) y se predicen en el grupo. Se repite entonces K veces para cada uno de los distintos grupos.

```
glm.fit2 = glm(Rent_1año ~ + Rent_1semana
             + Rent_3meses + Rent_6meses
             + Rent_en_el_año,
             data = data,family = gaussian())

cv.err = cv.glm(data, glm.fit2, K = 10) #numero de Ks mas utilizado
cv.err$delta
```

```
## [1] 1.984980 1.980191
```

Vector Delta: - Primer valor: Error cuadratico medio (promedio de los errores al cuadrado) - Segundo valor: Estimacion ajustada y corregida del sesgo

En todos los casos anteriores, el ECM podria interpretarse como la media de las varianzas de las predicciones frente a los verdaderos valores. Es decir, interesa que el error sea el minimo posible, lo cual garantiza que las predicciones sean cada vez mas ajustadas a la realidad.

Si se hubiera podido continuar con los modelos iniciales habria sido interesante comparar su capacidad predictiva, dado que cumplian las condiciones iniciales de los tests. Pero hay que tener en cuenta que aun

habiendo funcionado el metodo 'Nearest Neighbor Hot-Dock Imputation', los modelos hubieran cambiado al rellenar los NA que antes se omitian.

En definitiva, para obtener unos resultados adecuados, el tratamiento de datos ausentes debe incluirse en los primeros pasos antes de empezar a estimar los modelos, comprobando que estos toman valores coherentes y con soporte teorico.