PRACTICA 1 PREDICCION

Federico Ros

24 de octubre de 2017

- Se pide el mejor modelo de regresion para estimar la rentabilidad a 1 año:

1. Carga de datos:

En primer lugar, abrimos el csv y visualizamos por encima el mismo intuyendo que problemas podremos tener en cuanto a limpieza del archivo.

```
DS<-read.csv2("Fondos.csv")
getwd()
```

```
## [1] "/Users/fede/Desktop/DS/Prediccion"
```

```
vY<-DS$rent_1
vX<-cbind(1,DS[,])
```

Observamos que nuestro data frame de 500 observaciones y 30 variables, hay algunas que no resultan de interes para nuestro estudio y otras que desafortunadamente carecen de datos y no aparecen o aparececen como NA y que habra que eliminar para que no perturbe nuestro modelo.

2. Limpieza del fichero:

```
# Eliminamos las columnas que no son validas para nuestro estudio, y dejamos las que, a priori, nos van
DS<-DS[,-3]
DS<-DS[,-3]
DS<-DS[,-3]
DS<-DS[,-11]
DS<-DS[,-11]
DS<-DS[,-13]
DS<-DS[,-18]
DS<-DS[,-18]
DS<-DS[,-12]
DS<-DS[,-14]
DS<-DS[,-17]
DS<-DS[,-17]
```

```r
# Tras esto, seguimos teniendo un problema clave que podria alterar nuestro modelo, que son los valores
library(rminer)
library(kknn)
DS=na.omit(DS)
cat("NA values:",sum(is.na(DS)),"\n")
```

```
## NA values: 0
```

3. Selecion del modelo:

```r
regresion01<-lm(rent_1~ Inv_minima_inicial+  X1_Day_Return + X1_Week_Return + rent_1_mes + rent_3_meses
    rent_6_meses + rent_en_el_anio +  rent_10_anios +
    Capitaliz_media_bursatil + Sharpe_.3 + Volatilidad_3 + Com_Gestion + Com_deposito, data= DS)
summary(regresion01)
```

```
##
## Call:
## lm(formula = rent_1 ~ Inv_minima_inicial + X1_Day_Return + X1_Week_Return +
##     rent_1_mes + rent_3_meses + rent_6_meses + rent_en_el_anio +
##     rent_10_anios + Capitaliz_media_bursatil + Sharpe_.3 + Volatilidad_3 +
##     Com_Gestion + Com_deposito, data = DS)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6529 -0.4908 -0.0068  0.5268  3.1810
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -1.748e+00  3.202e-01  -5.458 2.25e-07 ***
## Inv_minima_inicial       -4.523e-07  3.514e-07  -1.287 0.200323
## X1_Day_Return             6.234e-02  1.098e-01   0.568 0.571195
## X1_Week_Return           -2.281e-01  7.780e-02  -2.931 0.003968 **
## rent_1_mes               -6.759e-02  1.042e-01  -0.649 0.517508
## rent_3_meses              1.647e-01  7.322e-02   2.249 0.026144 *
## rent_6_meses             -2.033e-01  3.283e-02  -6.192 6.79e-09 ***
## rent_en_el_anio           9.046e-01  2.345e-02  38.581  < 2e-16 ***
## rent_10_anios             1.785e-01  6.211e-02   2.874 0.004715 **
## Capitaliz_media_bursatil  4.902e-06  4.785e-06   1.025 0.307411
## Sharpe_.3                 1.430e+00  3.462e-01   4.131 6.31e-05 ***
## Volatilidad_3            -1.064e-01  3.073e-02  -3.463 0.000718 ***
## Com_Gestion               3.025e-01  2.017e-01   1.500 0.136089
## Com_deposito             -7.415e-01  1.036e+00  -0.716 0.475471
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.028 on 134 degrees of freedom
## Multiple R-squared:  0.9671, Adjusted R-squared:  0.9639
## F-statistic: 303.2 on 13 and 134 DF,  p-value: < 2.2e-16
```

```r
# Nos quedamos con las variable mas significativas y construimos nuestro nuevo model con rent_6_meses,r
```

Nueva Regresion:

```r
regresion02<-lm(rent_1~rent_6_meses + rent_en_el_anio + Sharpe_.3 + Volatilidad_3, DS)
summary(regresion02)
```

```
##
## Call:
## lm(formula = rent_1 ~ rent_6_meses + rent_en_el_anio + Sharpe_.3 +
##     Volatilidad_3, data = DS)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8204 -0.5614  0.0270  0.5119  4.0533
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -1.63982    0.24769  -6.620 6.72e-10 ***
## rent_6_meses    -0.18316    0.03194  -5.734 5.60e-08 ***
## rent_en_el_anio  0.90117    0.01960  45.989  < 2e-16 ***
## Sharpe_.3        2.15354    0.28443   7.571 4.17e-12 ***
## Volatilidad_3   -0.09763    0.02873  -3.398 0.000879 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.087 on 143 degrees of freedom
## Multiple R-squared:  0.9608, Adjusted R-squared:  0.9597
## F-statistic: 875.3 on 4 and 143 DF,  p-value: < 2.2e-16
```

```r
# Utilizamos el Best Subset para comprobar que hemos elegido bien nuestras variables:
library (leaps)
regfit.full=regsubsets(rent_1~ Inv_minima_inicial+  X1_Day_Return + X1_Week_Return + rent_1_mes + rent_3
    rent_6_meses + rent_en_el_anio +  rent_10_anios +
    Capitaliz_media_bursatil + Sharpe_.3 + Volatilidad_3 + Com_Gestion + Com_deposito, data= DS)
reg.summary=summary(regfit.full)
reg.summary
```

```
## Subset selection object
## Call: regsubsets.formula(rent_1 ~ Inv_minima_inicial + X1_Day_Return +
##     X1_Week_Return + rent_1_mes + rent_3_meses + rent_6_meses +
##     rent_en_el_anio + rent_10_anios + Capitaliz_media_bursatil +
##     Sharpe_.3 + Volatilidad_3 + Com_Gestion + Com_deposito, data = DS)
## 13 Variables  (and intercept)
##                          Forced in Forced out
## Inv_minima_inicial           FALSE      FALSE
## X1_Day_Return                FALSE      FALSE
## X1_Week_Return               FALSE      FALSE
## rent_1_mes                   FALSE      FALSE
## rent_3_meses                 FALSE      FALSE
## rent_6_meses                 FALSE      FALSE
## rent_en_el_anio              FALSE      FALSE
## rent_10_anios                FALSE      FALSE
## Capitaliz_media_bursatil     FALSE      FALSE
## Sharpe_.3                    FALSE      FALSE
## Volatilidad_3                FALSE      FALSE
```

```
## Com_Gestion                   FALSE      FALSE
## Com_deposito                  FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: exhaustive
##          Inv_minima_inicial X1_Day_Return X1_Week_Return rent_1_mes
## 1  ( 1 ) " "                " "           " "            " "
## 2  ( 1 ) " "                " "           " "            " "
## 3  ( 1 ) " "                " "           " "            " "
## 4  ( 1 ) " "                " "           " "            " "
## 5  ( 1 ) " "                " "           " "            " "
## 6  ( 1 ) " "                " "           "*"            " "
## 7  ( 1 ) " "                " "           "*"            " "
## 8  ( 1 ) " "                " "           "*"            " "
##          rent_3_meses rent_6_meses rent_en_el_anio rent_10_anios
## 1  ( 1 ) " "          " "          "*"             " "
## 2  ( 1 ) " "          "*"          "*"             " "
## 3  ( 1 ) " "          "*"          "*"             " "
## 4  ( 1 ) " "          "*"          "*"             " "
## 5  ( 1 ) " "          "*"          "*"             "*"
## 6  ( 1 ) " "          "*"          "*"             "*"
## 7  ( 1 ) "*"          "*"          "*"             "*"
## 8  ( 1 ) "*"          "*"          "*"             "*"
##          Capitaliz_media_bursatil Sharpe_.3 Volatilidad_3 Com_Gestion
## 1  ( 1 ) " "                      " "       " "           " "
## 2  ( 1 ) " "                      " "       " "           " "
## 3  ( 1 ) " "                      "*"       " "           " "
## 4  ( 1 ) " "                      "*"       "*"           " "
## 5  ( 1 ) " "                      "*"       "*"           " "
## 6  ( 1 ) " "                      "*"       "*"           " "
## 7  ( 1 ) " "                      "*"       "*"           " "
## 8  ( 1 ) " "                      "*"       "*"           "*"
##          Com_deposito
## 1  ( 1 ) " "
## 2  ( 1 ) " "
## 3  ( 1 ) " "
## 4  ( 1 ) " "
## 5  ( 1 ) " "
## 6  ( 1 ) " "
## 7  ( 1 ) " "
## 8  ( 1 ) " "
```

```r
reg.summary$bic
```

```
## [1] -351.5088 -401.1186 -447.7473 -454.2439 -455.0070 -458.4547 -458.0226
## [8] -456.2119
```

```r
summary(regresion02)
```

```
##
## Call:
## lm(formula = rent_1 ~ rent_6_meses + rent_en_el_anio + Sharpe_.3 +
##     Volatilidad_3, data = DS)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

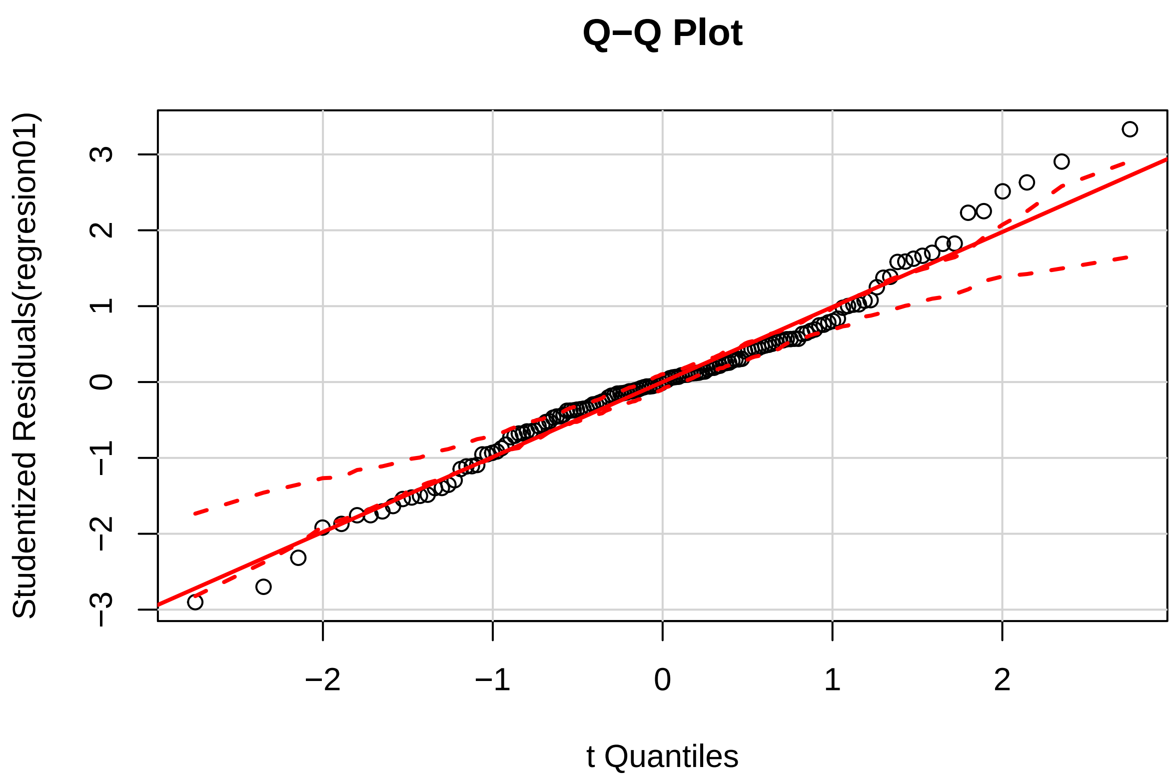
```
## -2.8204 -0.5614  0.0270  0.5119  4.0533
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -1.63982    0.24769  -6.620 6.72e-10 ***
## rent_6_meses     -0.18316    0.03194  -5.734 5.60e-08 ***
## rent_en_el_anio   0.90117    0.01960  45.989  < 2e-16 ***
## Sharpe_.3         2.15354    0.28443   7.571 4.17e-12 ***
## Volatilidad_3    -0.09763    0.02873  -3.398 0.000879 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.087 on 143 degrees of freedom
## Multiple R-squared:  0.9608, Adjusted R-squared:  0.9597
## F-statistic: 875.3 on 4 and 143 DF,  p-value: < 2.2e-16
```

```
# Finalmente nos decantamos por el modelo dos basandonos en signnificatividad de las variables, el alto
```

3. Diagnosis:

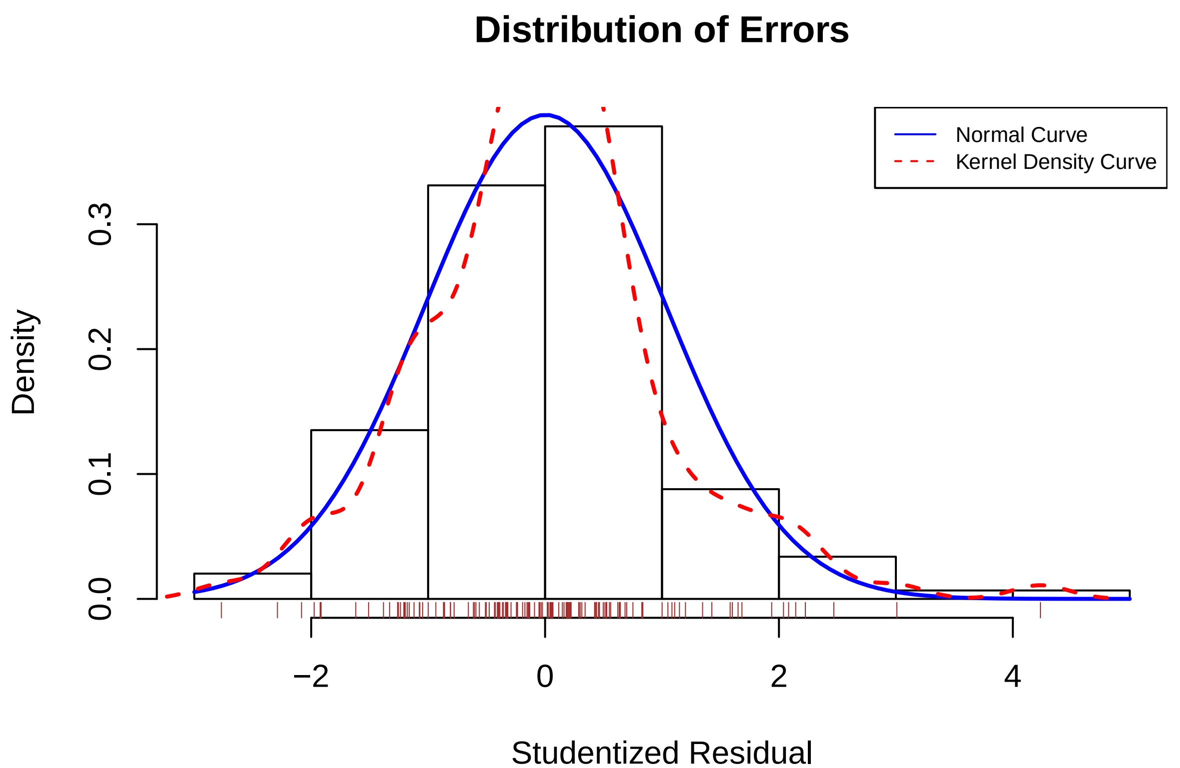
Ahora comprobamos normalidad:

```
library(car)
qqPlot(regresion01, labels=row.names(DS), id.method="identify",
       simulate=TRUE, main="Q-Q Plot")
```




```
residplot <- function(fit, nbreaks=10) {
  z <- rstudent(fit)
  hist(z, breaks=nbreaks, freq=FALSE,
```

```r
       xlab="Studentized Residual",
       main="Distribution of Errors")
  rug(jitter(z), col="brown")
  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)
  legend("topright",
         legend = c( "Normal Curve", "Kernel Density Curve"),
         lty=1:2, col=c("blue","red"), cex=.7)
}

residplot(regresion02)
```



#
podemos observar como los residuos obtenidos de la regresion02 se distribuyen normalmente. Seguidamente realizamos el test de Jarque Bera para corroborarlo.

```r
library(fBasics)
```

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

```r
vResid=resid(regresion02)
library(timeDate)
library(timeSeries)
jbTest(vResid)
```

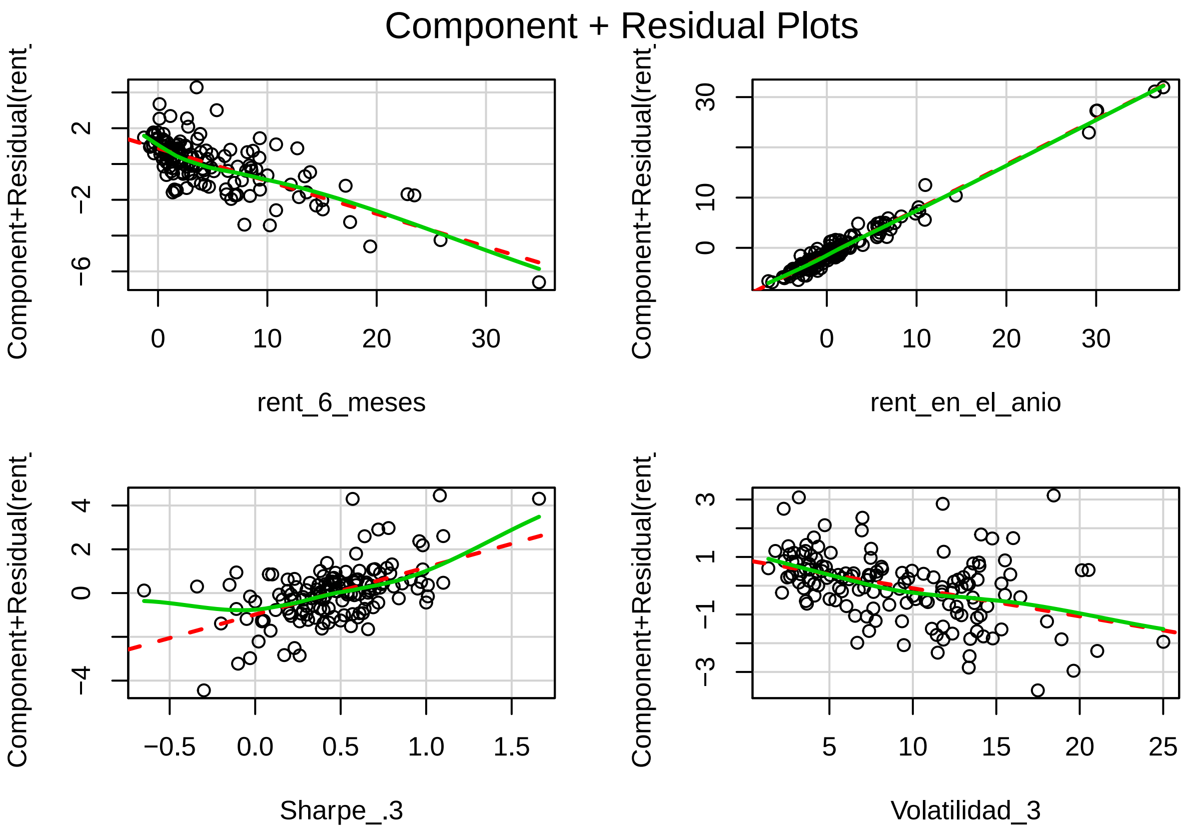
```
## Warning in interpp.old(x, y, z, xo, yo, ncp = 0, extrap = FALSE, duplicate
## = "median", : interpp.old() is deprecated, future versions will only
## provide interpp()

## 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: 15.1
##     ALM: 17.128
##   P VALUE:
##     LM p-value: 0.006
##     ALM p-value: 0.006
##     Asymptotic: 0.001
##
## Description:
##  Sat Oct 28 16:08:44 2017 by user:
```

```r
shapiro.test(vResid)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  vResid
## W = 0.97605, p-value = 0.01085
```

Realizando ambos Test comprobamos que no podemos rechazar la Hipotesis Nula y por tanto los residuos de esta regresion se distribuyen siguiendo una normal.

```r
crPlots(regresion02)
```



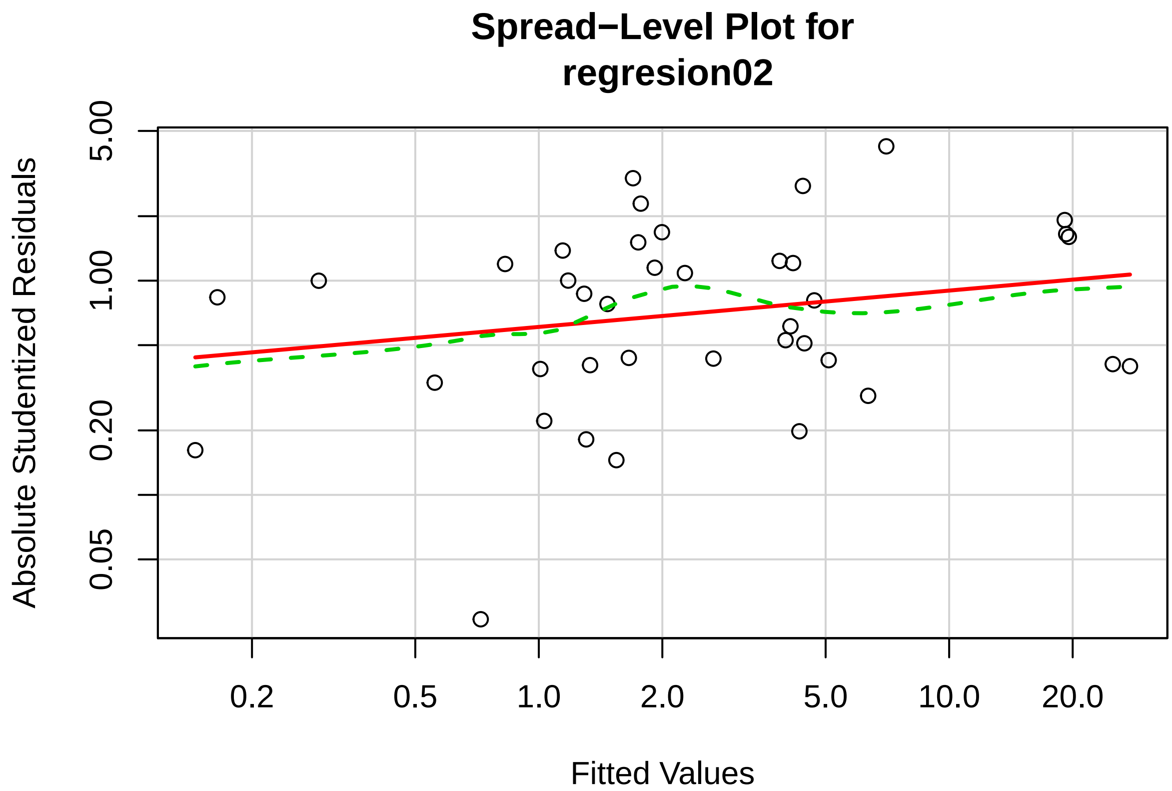
el proximo paso es comprobar la varianza de las observaciones es o no constante (homocedasticidad o heterocedasticidad). Para ello utilizamos el test de Breusch-Pagan en donde la H0 implica homocedasticidad.

```
ncvTest(regresion02)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 6.017495    Df = 1     p = 0.01416474
```

```
spreadLevelPlot(regresion02)
```

```
## Warning in spreadLevelPlot.lm(regresion02): 109 negative fitted values
## removed
```




```
##
## Suggested power transformation:  0.8303391
```

En este primer caso de regresion02 podemos concluir que la varianza de las observaciones es constante, ya que no podemos rechazar la H0.

Seguidamente relizamos el test de validacion global en nuestro modelo:

```
# Regresion3
library(gvlma)
gvmodel <- gvlma(regresion02)
summary(gvmodel)
```

```
##
## Call:
## lm(formula = rent_1 ~ rent_6_meses + rent_en_el_anio + Sharpe_.3 +
##     Volatilidad_3, data = DS)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8204 -0.5614  0.0270  0.5119  4.0533
##
## Coefficients:
```

```
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -1.63982    0.24769  -6.620 6.72e-10 ***
## rent_6_meses    -0.18316    0.03194  -5.734 5.60e-08 ***
## rent_en_el_anio  0.90117    0.01960  45.989  < 2e-16 ***
## Sharpe_.3        2.15354    0.28443   7.571 4.17e-12 ***
## Volatilidad_3   -0.09763    0.02873  -3.398 0.000879 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.087 on 143 degrees of freedom
## Multiple R-squared:  0.9608, Adjusted R-squared:  0.9597
## F-statistic: 875.3 on 4 and 143 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 = regresion02)
##
##                       Value  p-value                   Decision
## Global Stat        22.28287 0.000176 Assumptions NOT satisfied!
## Skewness            4.63147 0.031391 Assumptions NOT satisfied!
## Kurtosis           10.46829 0.001214 Assumptions NOT satisfied!
## Link Function       0.06065 0.805473    Assumptions acceptable.
## Heteroscedasticity  7.12246 0.007612 Assumptions NOT satisfied!
```

4. Multicolinealidad:

Seguimos comprobando la validez de nuestro modelo, comprobando si existe correlacion entre las variables explicativas (mulicolinealidad) mediante el Test VIF (Factor de inflacion de la varianza).

```
# Regresion3
vif(regresion02)
```

```
##    rent_6_meses rent_en_el_anio       Sharpe_.3   Volatilidad_3
##        4.158433        2.296332        1.037553        2.555556
```

```
sqrt(vif(regresion02)) > 2
```

```
##    rent_6_meses rent_en_el_anio       Sharpe_.3   Volatilidad_3
##            TRUE           FALSE           FALSE           FALSE
```

Con el resultado del Test VIF concluimos que, en regresion02, la variable rentabilidad 6 meses presenta multicolinealidad.

5. Identificacion de observaciones anomalas:

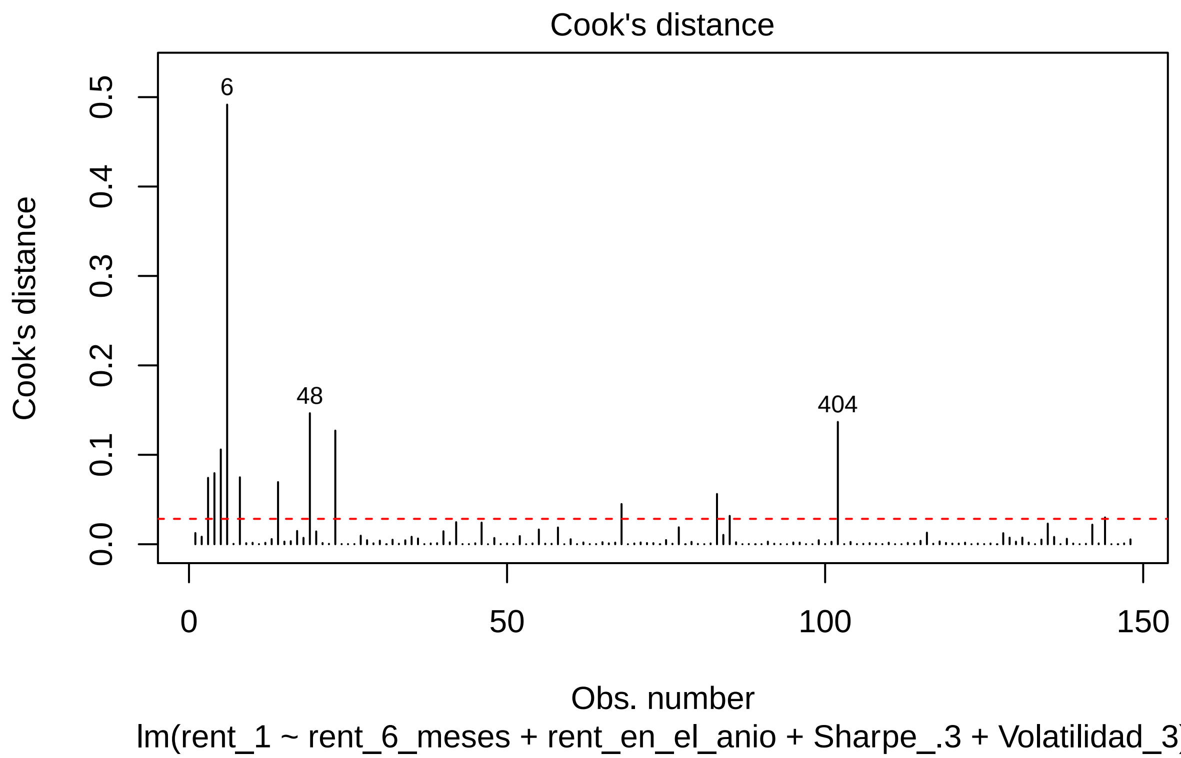
```r
outlierTest(regresion02)
```

```
##   rstudent unadjusted p-value Bonferonni p
## 6  4.23603         4.0695e-05    0.0060229
```

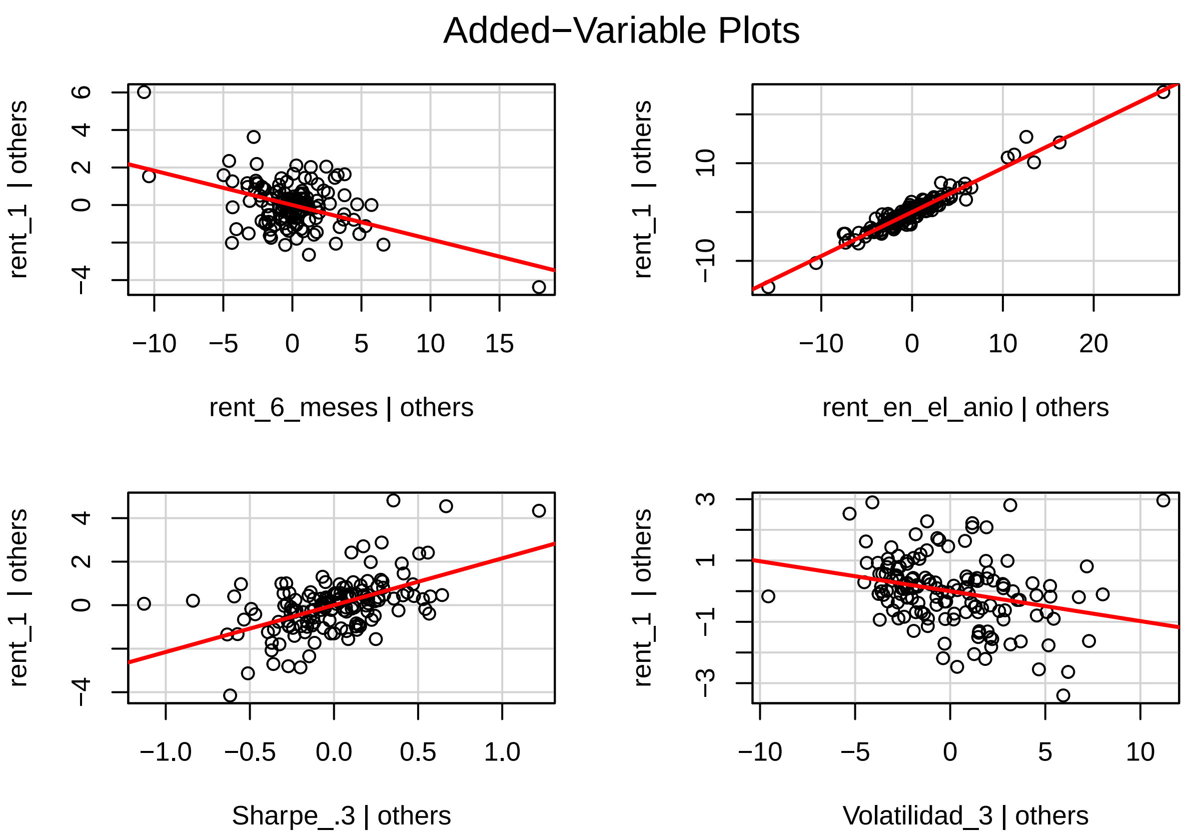
```r
hat.plot <- function(fit) {
  p <- length(coefficients(fit))
  n <- length(fitted(fit))
  plot(hatvalues(fit), main="Index Plot of Hat Values")
  abline(h=c(2,3)*p/n, col="red", lty=2)
  identify(1:n, hatvalues(fit), names(hatvalues(fit)))
}
hat.plot(regresion02)
```


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

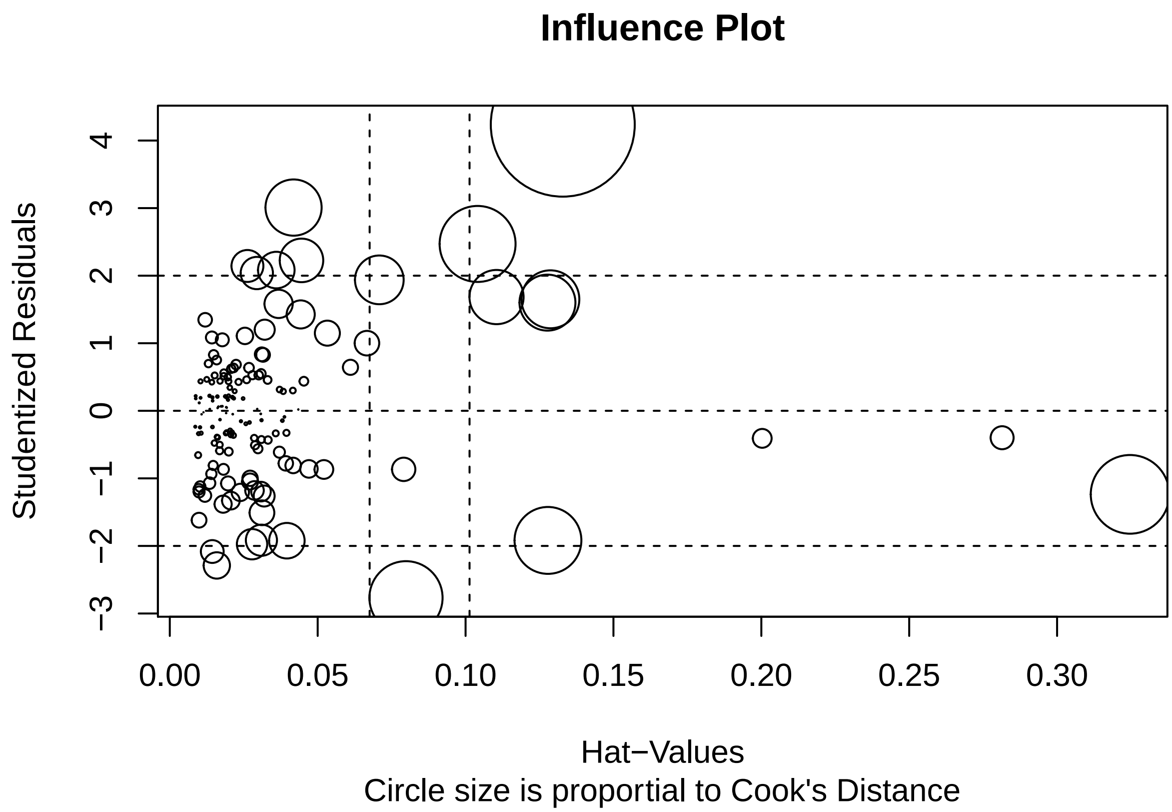
```r
#Tambien observamos los valores influyentes:
cutoff <- 4/(nrow(DS)-length(regresion02$coefficients)-2)
plot(regresion02, which=4, cook.levels=cutoff)
abline(h=cutoff, lty=2, col="red")
```




```r
avPlots(regresion02, ask=FALSE, id.method="identify")
```




```r
influencePlot(regresion02, id.method="identify", main="Influence Plot",
              sub="Circle size is proportial to Cook's Distance" )
```



Observamos que tenemos una observacion anomala y la elimnamos del modelo:

```
DS<-DS[-6,]
regresion02<-lm(rent_1~rent_6_meses + rent_en_el_anio + Sharpe_.3 + Volatilidad_3, DS)
summary(regresion02)
```

```
##
## Call:
## lm(formula = rent_1 ~ rent_6_meses + rent_en_el_anio + Sharpe_.3 +
##     Volatilidad_3, data = DS)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5230 -0.5764 -0.0059  0.5263  3.3260
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -1.46471    0.23782  -6.159 7.12e-09 ***
## rent_6_meses    -0.13979    0.03189  -4.383 2.26e-05 ***
## rent_en_el_anio  0.88205    0.01907  46.253  < 2e-16 ***
## Sharpe_.3        2.04046    0.27027   7.550 4.82e-12 ***
## Volatilidad_3   -0.13424    0.02851  -4.709 5.86e-06 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.028 on 142 degrees of freedom
## Multiple R-squared:  0.9639, Adjusted R-squared:  0.9629
## F-statistic: 948.8 on 4 and 142 DF,  p-value: < 2.2e-16
```

```
# Aunque el modelo no mejora demasiado ya que tan solo hemos eliminado una observacion. Comprobamos de
gvmodel <- gvlma(regresion02)
summary(gvmodel)
```

```
##
## Call:
## lm(formula = rent_1 ~ rent_6_meses + rent_en_el_anio + Sharpe_.3 +
##     Volatilidad_3, data = DS)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5230 -0.5764 -0.0059  0.5263  3.3260
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -1.46471    0.23782  -6.159 7.12e-09 ***
## rent_6_meses    -0.13979    0.03189  -4.383 2.26e-05 ***
## rent_en_el_anio  0.88205    0.01907  46.253  < 2e-16 ***
## Sharpe_.3        2.04046    0.27027   7.550 4.82e-12 ***
## Volatilidad_3   -0.13424    0.02851  -4.709 5.86e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.028 on 142 degrees of freedom
## Multiple R-squared:  0.9639, Adjusted R-squared:  0.9629
## F-statistic: 948.8 on 4 and 142 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 = regresion02)
##
##                     Value p-value                   Decision
## Global Stat        9.726 0.04531 Assumptions NOT satisfied!
## Skewness           1.354 0.24455    Assumptions acceptable.
## Kurtosis           1.605 0.20519    Assumptions acceptable.
## Link Function      2.638 0.10431    Assumptions acceptable.
## Heteroscedasticity 4.128 0.04217 Assumptions NOT satisfied!
```

De esta manera sin el outlier se cumple los supuestos de asimetria y kurtosis.

Como penultimo paso comparamos los modelos mediante los criterios AIC y de nuevo BIC:

```
AIC(regresion01,regresion02)
```

```
## Warning in AIC.default(regresion01, regresion02): models are not all fitted
## to the same number of observations
```

```
##             df      AIC
## regresion01 15 443.3684
## regresion02  6 432.0759
```

```
BIC(regresion01,regresion02)
```

```
## Warning in BIC.default(regresion01, regresion02): models are not all fitted
## to the same number of observations
```

```
##             df      BIC
## regresion01 15 488.3266
## regresion02  6 450.0185
```

```
# Comprobamos la mejora del modelo 02 con respecto al primero bajo ambos criterios de seleccion.
```

6. Cross Validation:

```
library(ISLR)
set.seed(250)
numData=nrow(DS)
train=sample(numData ,numData/2)

regres.train =lm(rent_1~rent_6_meses + rent_en_el_anio + Sharpe_.3 + Volatilidad_3, DS ,subset =train )
attach(DS)
mean((rent_1-predict(regres.train ,Auto))[-train ]^2)
```

```
## Warning: 'newdata' had 392 rows but variables found have 147 rows
```

```
## [1] 1.401156
```

```
set.seed(251)
regres.train2 =lm(rent_1~rent_6_meses + rent_en_el_anio + Sharpe_.3 + Volatilidad_3, DS ,subset =train
mean((rent_1-predict(regres.train2 ,Auto))[-train ]^2)
```

```
## Warning: 'newdata' had 392 rows but variables found have 147 rows
```

```
## [1] 1.401156
```

Leave-one-out Cross Validation:

```
glm.fit1=glm(rent_1~rent_6_meses + rent_en_el_anio + Sharpe_.3 + Volatilidad_3, DS ,family = gaussian())
coef(glm.fit1)
```

```
##     (Intercept)    rent_6_meses rent_en_el_anio       Sharpe_.3
##      -1.4647074      -0.1397874       0.8820467       2.0404648
##   Volatilidad_3
##      -0.1342384
```

```
library (boot)
```

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

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

```
cv.err =cv.glm(DS,glm.fit1)
cv.err$delta
```

```
## [1] 1.144061 1.143608
```

```
glm.fit2=glm(rent_1~rent_en_el_anio + Sharpe_.3 + Volatilidad_3, DS ,family = gaussian())
cv.err2 =cv.glm(DS,glm.fit2)
cv.err2$delta
```

```
## [1] 1.261463 1.261096
```

K-Fold Cross-Validation:

```
library (boot)
cv.err =cv.glm(DS,glm.fit1,K=10)
cv.err$delta
```

```
## [1] 1.167098 1.158525
```

```
glm.fit2=glm(rent_1~rent_en_el_anio + Sharpe_.3 + Volatilidad_3, DS ,family = gaussian())
cv.err2 =cv.glm(DS,glm.fit2,K=10)
cv.err2$delta
```

```
## [1] 1.283470 1.276506
```