

Taller 2 Regresión lineal Multiple

Andrés Felipe Palomino - David Stiven Rojas

2023-04-21

1 Introducción

La base de datos "yarn" obtenida de la librería (PLS) contiene información sobre espectros NIR y mediciones de densidad de hilos de PET, consta de 28 individuos (hilos de PET), 268 variables predictoras (NIRS) y una variable de respuesta (densidad). Se ajustará un modelo lineal múltiple para estimar la densidad del hilo PET, mediante mediciones NIR

```
#Importación de librerías necesarias
library(car)
library(glmnet)
library(MASS)
library(xtable)
library(lmtest)
library(readxl)
library(lmridge)
library(pls)
library(olsrr)
library(effects)
```

1.1 Base de datos

En la siguiente tabla se encuentra un encabezado de la base de datos que se trabajara, esta consta de 30 covariables predictoras, las cuales estarán desde NIR1 hasta NIR30. De primera mano se observa que los valores de los NIR disminuyen a medida que la covariable aumenta

```
X <- data.frame(matrix(c(yarn$NIR[,1:30],yarn$density),nrow =28, ncol= 31))
colnames(X) <- c(paste("NIR",1:30,sep=""), "density")
```

1.2 Funciones creadas

Antes de empezar con el proceso de seleccionar las variables para ajustar el modelo se crean funciones para optimizar el proceso de validación de supuestos, debido a que constantemente se deben realizar, estas funciones estan diseñadas para objetos lm.

```
##Validacion grafica para homocedasticidad y normalidad y pruebas formales
validaciongrafica<- function(model,cor=F){

  par(mfrow=c(1,2))
```

```

plot(fitted.values(model),studres(model),panel.first=grid(),
     pch=19,ylab='Residuos Estudentizados',xlab='Valores ajustados',main='A',col='aquamarine4')
abline(h=c(-2,0,2),lty=2)
qqPlot(model,pch=19,ylab='Residuos Estudentizados',
        xlab='Cuantiles Teóricos',col=carPalette()[1],
        col.lines=carPalette()[3],main='B')
print('Shapiro Test; H0: Normalidad vs H1: No Normalidad')
print(shapiro.test(studres(model)))
print('Breusch Pagan Test;H0: Homocedasticidad vs H1: No Homocedasticidad')
print(bptest(model))
if(cor==T){
  par(mfrow=c(1,2))
  plot(studres(model),type="b",xlab="Tiempo",ylab="Residuos Estudentizados",main="A",
       pch=19,panel.first=grid())
  plot(studres(model)[-length(fitted.values(model))],
       studres(model)[-1],pch=19,panel.first = grid(),col="turquoise3",
       xlab=TeX("$Residuos_{t-1}$"),ylab=TeX("$Residuos_{t}$"),main="B")
  abline(lm(studres(model)[-1]~studres(model)[-length(fitted.values(model))]))
  print('Durbin Watson Test')
  print(durbinWatsonTest(model,
                          method='resample',reps=10000))
}
par(mfrow=c(1,1))
}

## Calculo de lambda optimo para boxcox
lambda<- function(model,a,b){
  par(mfrow=c(1,1))
  box.cox<-boxcox(model,lambda=seq(a,b,length.out = 1000),
                  ylab='log-verosimilitud')
  bc<-round(box.cox$x[box.cox$y ==max(box.cox$y)],2)
  print(bc)
}

```

2 Selección de variables

En el proceso de selección de variables se procede a realizar la Regresión de LASSO para identificar las posibles variables que tengan un aporte poco relevante, Por ultimo se ajustara el modelo cuyas variables tengan buenos indicadores y se pueda realizar corrección de supuestos

2.1 Regresión de LASSO

Este es un método de regularización que se implementa cuando se tiene muchas covariables disponibles y se cree que pocas tienen un aporte relevante.

Se asume el modelo de regresión usual, donde :

$$E(y|x)=X^T\beta, \text{ y } V(y|x)=\sigma^2$$

Donde se asume que algunos β son cero. El objetivo del estimador es seleccionar los coeficientes que tienen valores diferentes de cero. El cual se obtiene minimizando la siguiente expresión:

$$S_{lasso}(\beta) = \sum_{i=1}^n (y_i - x^T \beta)^2 + \lambda \sum_{j=1}^{p-1} |\beta_j|$$

Esta es la suma de cuadrados del estimador por MCO más una penalización (λ), a la suma del valor absoluto de los coeficientes. A medida que λ aumenta la penalización tendrá mas peso sobre la estimación de los coeficientes, es decir que si la penalización es muy grande, todas las estimaciones serán cero. No hay solución analítica para $\hat{\beta}_{lasso}$ por lo que se usan algoritmos para la estimación, como lo es la función de `glmnet` de la librería `glmnet`.

2.1.1 Modelo a realizar regresión LASSO

Como se estableció anteriormente, se asume un modelo de regresión usual, el cual debe cumplir los siguientes supuestos: $E(y|x) = x^T \beta$, y $V(y|x) = \sigma^2$, es decir, varianza constante y $E(\varepsilon) = 0$. Por ende es necesario proponer un modelo con $p < n$, en el cual se eliminarán las variables con menor correlación con la variable `density`. Dicho modelo se expresa a continuación y se evalúan los supuestos:

```
model <- lm(density ~ .-NIR1-NIR8-NIR9-NIR10-NIR11-NIR7, data=X)
car::vif(model)[1:5]
```

```
##      NIR2      NIR3      NIR4      NIR5      NIR6
## 1664.742 39841.312 361180.493 623252.746 254014.080
```

```
car::vif(model)[6:10]
```

```
##      NIR12      NIR13      NIR14      NIR15      NIR16
## 8859704 76280641 79779605 53664069 80678689
```

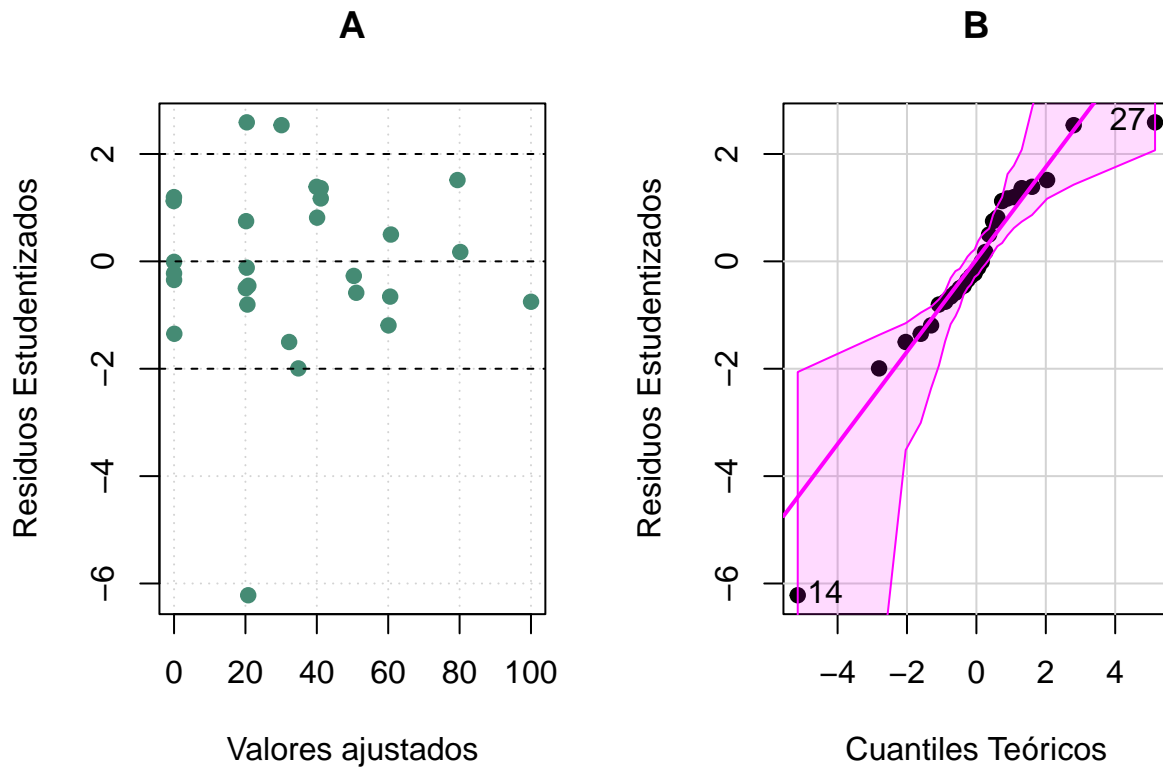
```
car::vif(model)[11:16]
```

```
##      NIR17      NIR18      NIR19      NIR20      NIR21      NIR22
## 99398936 163539712 308758508 360036276 277176858 369337309
```

```
car::vif(model)[17:24]
```

```
##      NIR23      NIR24      NIR25      NIR26      NIR27      NIR28      NIR29      NIR30
## 475476198 461114852 385039558 205007436 70428398 37122348 20001839 1522304
```

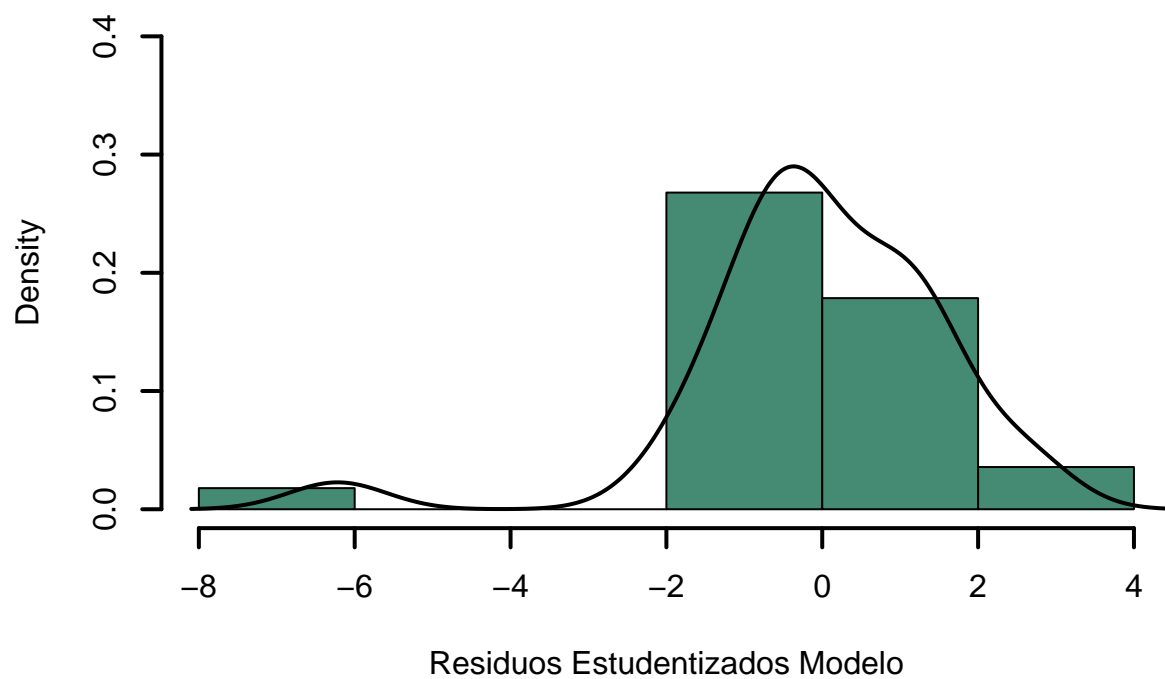
```
validaciongrafica(model)
```



```
## [1] "Shapiro Test; H0: Normalidad vs H1: No Normalidad"
##
##  Shapiro-Wilk normality test
##
## data:  studres(model)
## W = 0.86458, p-value = 0.001868
##
## [1] "Breusch Pagan Test;H0: Homocedasticidad vs H1: No Homocedasticidad"
##
##  studentized Breusch-Pagan test
##
## data:  model
## BP = 27.288, df = 24, p-value = 0.2912
```

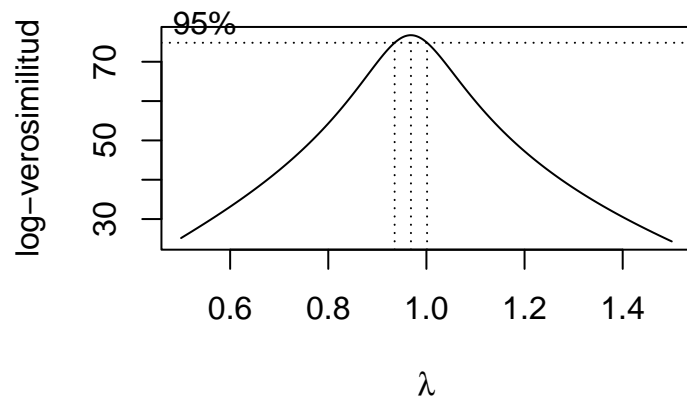
Mediante el grafico y los valores P asociados a la homocedasticidad y normalidad de residuos se evidencia el incumplimiento de la normalidad de los residuos.

```
hist(studres(model),lwd=2,col='aquamarine4',freq=F,ylim=c(0,0.4),
     xlab='Residuos Estudentizados Modelo',main='')
lines(density(studres(model)),lwd=2,col='black')
```



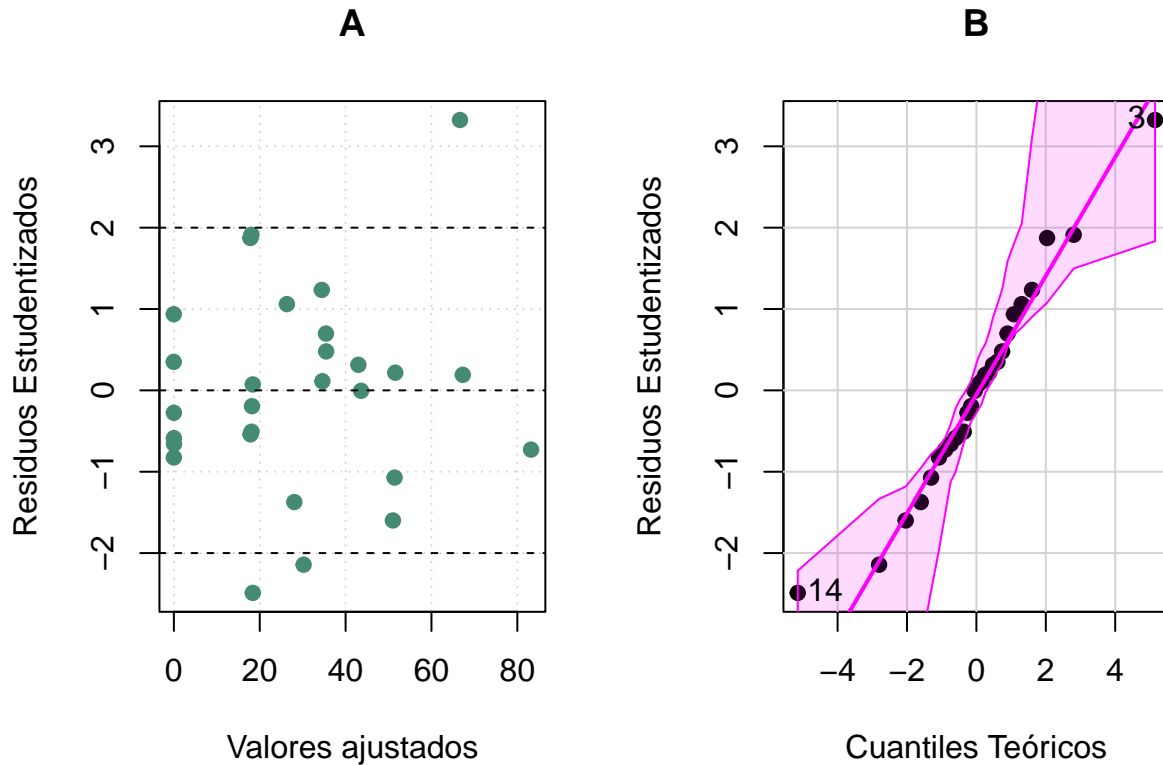
Como no se cumple el supuesto de normalidad se procede a corregir mediante el metodo de BoxCox y se verifica el cumplimiento de los mismos.

```
model <- lm(density+0.01 ~ .-NIR1-NIR8-NIR9-NIR10-NIR11-NIR7, data=X)
lambda(model,0.5,1.5)
```



```
## [1] 0.97
```

```
model.box <- lm(I(density^0.96) ~.-NIR1-NIR8-NIR9-NIR10-NIR11-NIR7,data=X)
validaciongrafica(model.box)
```



[1] “Shapiro Test; H0: Normalidad vs H1: No Normalidad”

Shapiro-Wilk normality test

data: studres(model) W = 0.97774, p-value = 0.7934

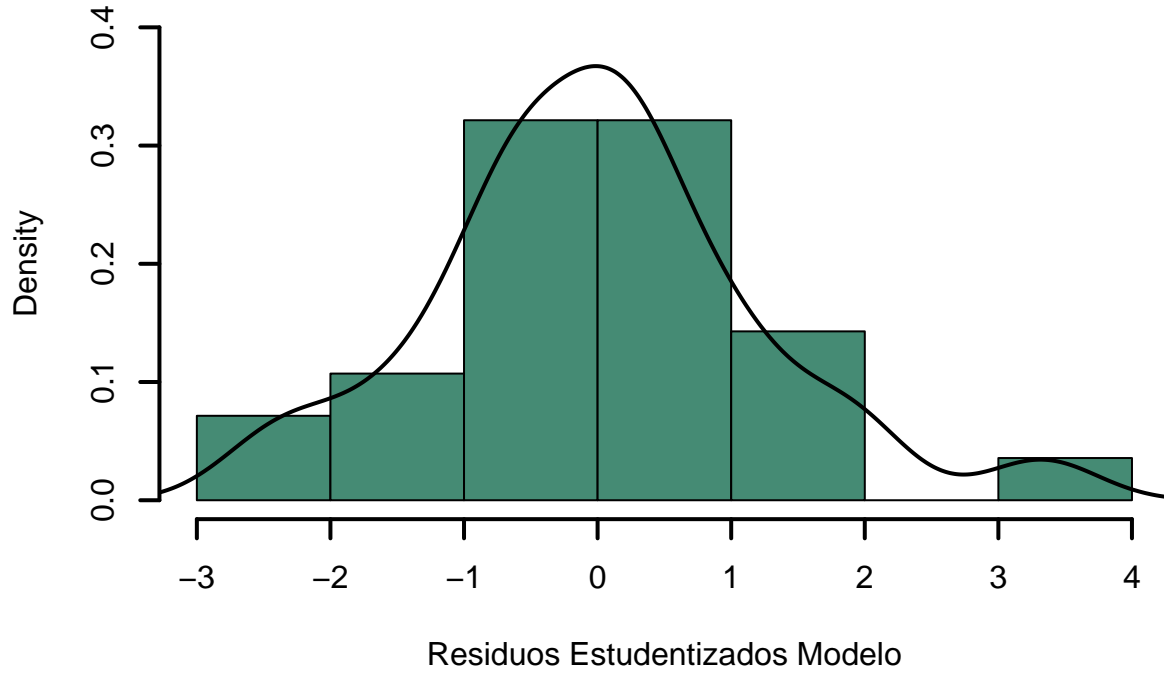
[1] “Breusch Pagan Test;H0: Homocedasticidad vs H1: No Homocedasticidad”

studentized Breusch-Pagan test

data: model BP = 23.94, df = 24, p-value = 0.4651

Mediante el grafico y los valores P asociados a la homocedasticidad y normalidad de residuos se evidencia el cumplimiento de ambos supuestos.

```
hist(studres(model.box),lwd=2,col='aquamarine4',
freq=F,ylim=c(0,0.4),xlab='Residuos Estudentizados Modelo',main='')
lines(density(studres(model.box)),lwd=2,col='black')
```



Ya con los requerimientos necesarios para realizar regresión de LASSO se procede a calcular el valor de λ optimo mediante la validación cruzada

2.1.2 Validación cruzada

Es un método para evaluar que tan bueno es un modelo para predecir observaciones futuras de la población objeto de estudio. La muestra se divide en dos grupos:

- Entrenamiento: Se usa para ajustar el modelo.
- Validación: Se utiliza para validar el modelo ajustado.

Se realiza K interacciones dividiendo los datos en K subconjuntos. En cada interacción uno de los subconjuntos es utilizado para validación, el el resto (K-1) como datos de entrenamiento. Para cada división, $k=1, \dots, K$ y para cada valor de λ se estima el modelo basado en la muestra de entrenamiento. Mientras que con cada muestra de validación y para cada valor de λ se utiliza para calcular el error cuadrático medio:

$$ECM_k(\lambda) = \sum_{i=1}^{n_k} \frac{(y_i^k - x_i^k * \hat{\beta}^k(\lambda))^2}{n}$$

Donde y_i^k son las observaciones de la muestra de validación k y $\hat{\beta}^k$ es la estimación utilizando la muestra de entrenamiento k. para cada λ se calcula:

$$CV(\lambda) = \frac{1}{K} \sum_{k=1}^K ECM_k(\lambda)$$

Y la desviación estándar:

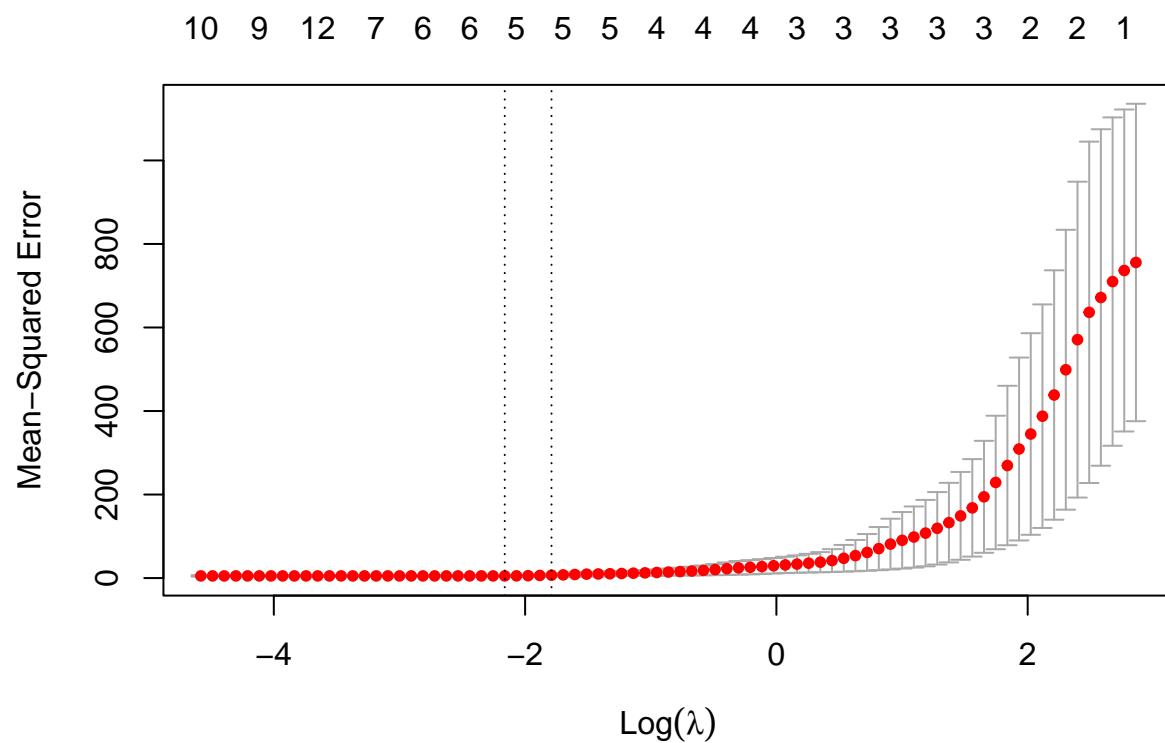
$$SD(\lambda) = \sqrt{\sum_{i=1}^K \frac{(ECM_k(\lambda) - CV(\lambda))^2}{K-1}}$$

λ optimo: minimiza $CV\lambda$ o

maximiza λ : $CV(\hat{\lambda}) < CV(\hat{\lambda}_{cv}) + SD(\hat{\lambda}_{cv})$

La maximización es mas optima ya que se tiene en cuenta la variabilidad debida a la selección de las submuestras.

```
X.<-model.matrix(model.box)[-1]
lasso.cv <-cv.glmnet(X., X$density, nfolds = 4, alpha = 1,
                    nlambda = 100)
plot(lasso.cv)
```



```
est = glmnet(X., X$density, alpha = 1, lambda = lasso.cv$lambda.1se)
est$beta
```

```
## 24 x 1 sparse Matrix of class "dgCMatrix"
##           s0
## NIR2  -13.817392
## NIR3    .
## NIR4    .
## NIR5    .
## NIR6   89.843734
## NIR12   .
## NIR13   .
```



```

## NIR14  .
## NIR15  .
## NIR16  .
## NIR17  .
## NIR18  -6.290143
## NIR19  .
## NIR20  .
## NIR21  .
## NIR22  .
## NIR23  .
## NIR24  .
## NIR25  .
## NIR26  .
## NIR27  .
## NIR28  -94.842367
## NIR29  -33.680130
## NIR30  .

```

La selección de variables por medio del estimador LASSO son: NIR2, NIR6, NIR18, NIR28, NIR29. Consecuente a eso se procede a realizar una prueba sobre subconjuntos para evaluar si podemos eliminar NIR29 para disminuir problemas de multicolinealidad.

2.2 Suma extra de cuadrados

Sirve para probar la significancia de un subconjunto de coeficientes.

Se tiene el siguiente modelo:

$$y = X\beta + \varepsilon$$

donde $\beta = \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix}$

donde β_1 es un vector (p-r)x1 y β_2 es un vector rx1, se quiere evaluar la siguiente hipótesis:

$$H_0 : \beta_2 = 0 \text{ vs } H_1 : \beta_2 \neq 0$$

Se tienen los siguientes modelos: Modelo completo : $y = X_1\beta_1 + X_2\beta_2 + \varepsilon$

$$SCR(B) = y^T(H - \frac{1}{n}11^T)y$$

Modelo reducido : $y = X_1\beta_1 + \varepsilon$

$$SCR(B_1) = y^T(H_1 - \frac{1}{n}11^T)y$$

La suma de cuadrados de la regresión debida a β_2 dado que β_1 ya está en el modelo es:

$$SSR(\beta_2|\beta_1) = SSR(\beta) - SSR(\beta_1)$$

Conocida como suma extra de cuadrados debido a β_2 , y dado que queremos probar $H_0 : \beta_2 = 0$ se construye el siguiente estadístico.

$$F_0 = \frac{\frac{SSR(\beta_2|\beta_1)}{r}}{\frac{SSE}{n-p}}$$

Si H_0 es cierta entonces $F_0 \sim F_{r,n-p}$

Se realiza la respectiva prueba con la funcion anova, asumiendo que el modelo reducido es aquel $\beta_5=0$ asociado al NIR29

Hipotesis: $H_0 : \beta_5 = 0$ VS $H_1 : \beta_5 \neq 0$

```
model.lasso1 <- lm(density~NIR2+NIR6+NIR18+NIR28+NIR29,data=X)
model.lasso2 <- lm(density~NIR2+NIR6+NIR18+NIR28,data=X)
anova(model.lasso2,model.lasso1)
```

```
## Analysis of Variance Table
##
## Model 1: density ~ NIR2 + NIR6 + NIR18 + NIR28
## Model 2: density ~ NIR2 + NIR6 + NIR18 + NIR28 + NIR29
##   Res.Df    RSS Df Sum of Sq    F Pr(>F)
## 1      23 31.435
## 2      22 30.610   1   0.82493 0.5929 0.4495
```

La prueba anova indica un valor P de 0.4495 lo que indica que el coeficiente asociado al NIR29 es significativamente 0, por ende, podemos retirarlo del modelo ya que no aporta a la estimación de la densidad y en cambio aumenta el VIF como se evidencia a continuación.

2.3 Factor de inflación de varianza (VIF)

Es una medida que detecta si hay problemas de multicolinealidad. Generalmente un VIF mayor a 10 indica problemas graves de multicolinealidad.

$$VIF_j = \frac{1}{1-R^2}$$

Donde R_j^2 es el coeficiente de determinación obtenido ajustando una regresión de x_j sobre las demás covariables.

$$VIF_j = \sum_{j=1}^{p-1} \frac{t_{ij}^2}{\lambda_j}$$

Donde $T = (t_1, \dots, t_{p-1})$ es una matriz ortogonal de vectores propios y λ_j los valores propios asociados a la descomposición de la matriz $R = Z^T Z$ y Z es la matrix de X estandarizada. $R = T \Lambda T^T$ (descomposición)

```
car::vif(model.lasso1)
```

```
##      NIR2      NIR6      NIR18      NIR28      NIR29
##  3.766765  5.643206 36.089199 269.277707 304.968458
```

```
car::vif(model.lasso2)
```

```
##      NIR2      NIR6      NIR18      NIR28
##  2.967327  4.203285 31.085734 26.983026
```

3 Modelo de regresión multiple

Con base en el proceso de selección de variables se ajusta el siguiente modelo y se realiza la respectiva validación de supuestos:

```
model.lasso1<- lm(density~NIR2+NIR6+NIR18+NIR28,data=X)
summary(model.lasso1)
```

```
##
## Call:
## lm(formula = density ~ NIR2 + NIR6 + NIR18 + NIR28, data = X)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.1312 -0.9776  0.1102  0.8381  2.0416
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   29.389     10.712   2.744  0.0116 *
## NIR2         -26.257      3.892  -6.747 6.99e-07 ***
## NIR6          96.140      1.741  55.211 < 2e-16 ***
## NIR18         -9.055      1.905  -4.753 8.62e-05 ***
## NIR28        -109.939     5.818 -18.896 1.66e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.169 on 23 degrees of freedom
## Multiple R-squared:  0.9984, Adjusted R-squared:  0.9981
## F-statistic: 3584 on 4 and 23 DF,  p-value: < 2.2e-16
```

3.1 Interpretación

- En el resumen del modelo contamos con un $R^2 = 0.9984$, es decir, el 99.84 de la variabilidad de la densidad del Hilo PET está siendo explicada por el modelo. Se cuenta con un R^2_{adj} de 0.9981.
- El valor del estadístico F es de 3584 con un valor p asociado de aproximadamente 0 lo que indica que por lo menos una de estas estimaciones de los β_i es diferente de 0. Por ende ajustar el modelo ayuda a la predicción de la densidad del Hilo de Pet.

```
Anova(model.lasso1)
```

3.1.0.1 ANOVA

```
## Anova Table (Type II tests)
##
## Response: density
##              Sum Sq Df F value    Pr(>F)
## NIR2           62.2  1  45.520 6.995e-07 ***
## NIR6          4166.2  1 3048.258 < 2.2e-16 ***
## NIR18           30.9  1  22.591 8.615e-05 ***
## NIR28           488.0  1  357.056 1.661e-15 ***
## Residuals       31.4 23
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

La tabla ANOVA realiza pruebas sobre subconjuntos de coeficientes haciendo uso de la suma de cuadrados extra.

Estadístico F asociado al NIR2 : $F_0 = \frac{SSR(\beta_1|\beta_0)}{\frac{SE}{26}}$

Modelo: $y = \beta_0 + NIR2\beta_1 + \varepsilon$

Hipotesis: $H_0 : \beta_1 = 0$

Estadístico F asociado al NIR6 : $F_0 = \frac{SSR(\beta_2|\beta_1, \beta_0)}{\frac{SE}{26}}$

Modelo: $y = \beta_0 + NIR2\beta_1 + NIR6\beta_2 + \varepsilon$

Hipotesis: $H_0 : \beta_2 = 0$

Estadístico F asociado al NIR18 : $F_0 = \frac{SSR(\beta_3|\beta_0, \beta_1, \beta_2)}{\frac{SE}{26}}$

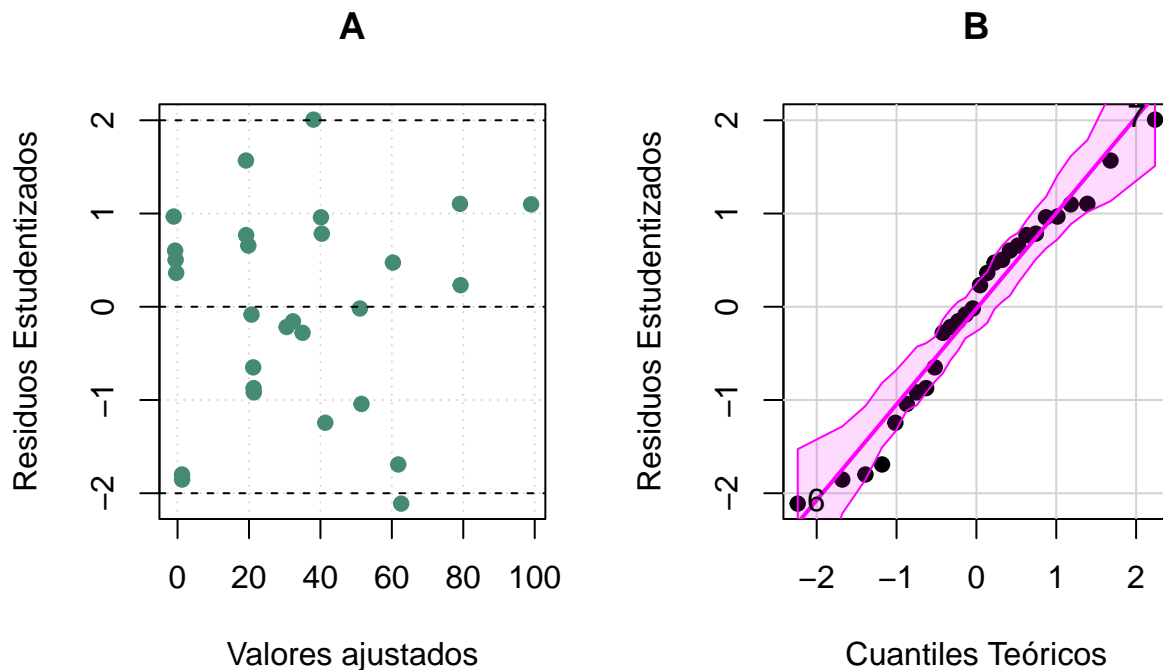
Modelo: $y = \beta_0 + NIR2\beta_1 + NIR6\beta_2 + NIR18\beta_3 + \varepsilon$

Hipotesis: $H_0 : \beta_3 = 0$

En cada hipotesis evaluada el valor p es menor a un nivel de significancia al 5% lo que indica que para cada caso, añadir el β_i es significativamente distinto de 0

3.2 Validación de supuestos

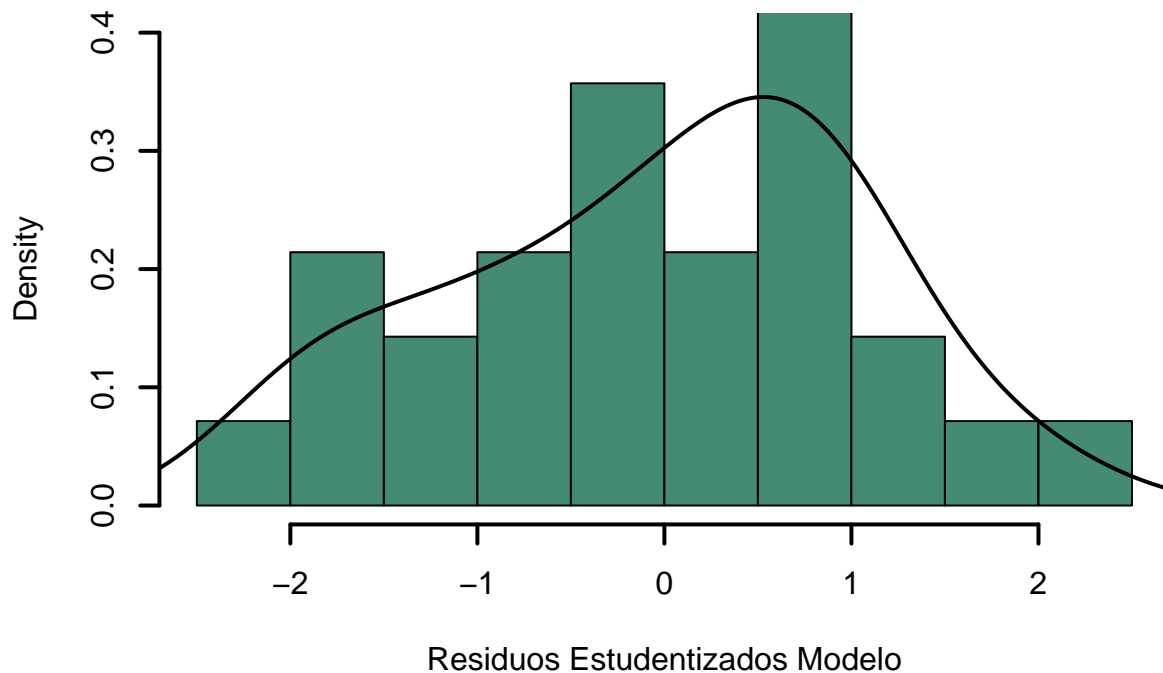
```
validaciongrafica(model.lasso1)
```



```
## [1] "Shapiro Test; H0: Normalidad vs H1: No Normalidad"
##
## Shapiro-Wilk normality test
```

```
##
## data: studres(model)
## W = 0.96468, p-value = 0.4471
##
## [1] "Breusch Pagan Test;H0: Homocedasticidad vs H1: No Homocedasticidad"
##
## studentized Breusch-Pagan test
##
## data: model
## BP = 1.6317, df = 4, p-value = 0.8031

hist(studres(model.lasso1),lwd=2,col='aquamarine4',
freq=F,ylim=c(0,0.4),xlab='Residuos Estudentizados Modelo',main='')
lines(density(studres(model.lasso1)),lwd=2,col='black')
```



Como se evidencia en los graficos y en las pruebas formales, ambos coinciden con sus respectivos resultados, es decir, varianza constante (Homocedasticidad) y distribución Normal en los errores.

3.3 Identificación de puntos atípicos e influyentes

Para esto utilizaremos la función `influence.measures()`

```
influence.measures(model.lasso1)$infmat[, -1]
```

```
##          dfb.NIR2      dfb.NIR6      dfb.NIR1      dfb.NIR28      dffit      cov.r
```

```

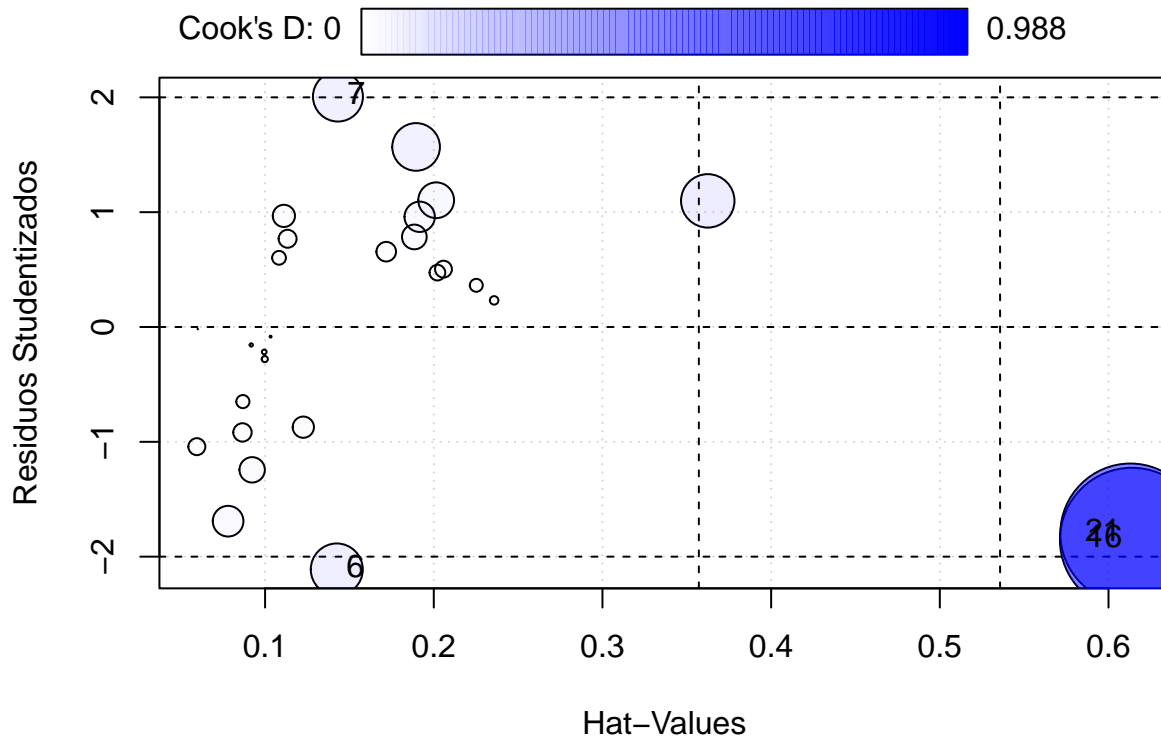
## 1 -0.373394566 0.6496848240 -0.331772536 0.184113316 0.828264429 1.4997618
## 2 -0.305492910 0.3720578646 0.073404994 -0.157310694 0.554365040 1.1944235
## 3 -0.019231535 0.0750594731 -0.081707189 0.061293691 0.128782237 1.6143881
## 4 -0.106547754 0.0838881352 0.142356089 -0.158122212 0.238509612 1.4878645
## 5 0.074553155 -0.1793449266 -0.047690711 0.118053390 -0.492162405 0.7352324
## 6 -0.191413380 -0.0858801572 0.403910765 -0.282271360 -0.860714944 0.5788966
## 7 -0.167776877 0.1601914874 0.433679945 -0.401078719 0.820454554 0.6288836
## 8 -0.069030633 0.0884179635 -0.268936383 0.265910872 -0.396733876 0.9794717
## 9 0.306437652 -0.1573248360 -0.132526148 0.132848593 0.377972302 1.3407617
## 10 0.271697306 -0.1981032511 -0.170864580 0.147302866 0.467326908 1.2584910
## 11 -0.157527270 0.1499879881 0.006592832 0.017816423 0.298765048 1.3687873
## 12 0.038781596 0.0435008484 -0.218669684 0.188489125 -0.326187278 1.2009151
## 13 -0.131949786 0.1833351054 -0.155967107 0.129606661 -0.282701090 1.1329332
## 14 -0.016192687 0.0213626824 -0.003712707 0.002586270 -0.028522311 1.3905338
## 15 0.157449597 -0.3162571007 -0.207031055 0.193692160 0.758360671 0.9070889
## 16 0.170444545 -0.9628595573 1.795953591 -1.991921206 -2.336700639 1.5662380
## 17 -0.012587784 0.0453219738 -0.099683412 0.126128085 0.195758464 1.5644498
## 18 -0.003506908 -0.0381099029 0.003559217 0.032787471 0.209923969 1.2904872
## 19 -0.061830397 -0.1112793632 0.061394764 -0.019920811 0.342151199 1.1407321
## 20 -0.100782895 -0.0516699098 0.034376668 -0.024073454 0.256303217 1.4848636
## 21 1.547939289 -0.3340483842 -0.015071485 0.067136669 -2.264225545 1.6244314
## 22 -0.000733778 -0.0003063477 -0.001200355 0.001528885 -0.004712816 1.3286357
## 23 -0.062384588 0.0207989090 -0.041460498 0.068343148 -0.262229787 1.0436235
## 24 -0.019265307 0.0171498437 -0.026120275 0.022702022 -0.049700844 1.3675029
## 25 -0.065310959 0.0562803062 -0.034684395 0.031194744 -0.092996640 1.3630398
## 26 -0.046421805 0.0530560130 -0.020592018 0.019655727 -0.072195305 1.3720953
## 27 0.009424286 0.0351500667 -0.115781227 0.095327904 -0.200308770 1.2437138
## 28 0.213025445 -0.2025624490 0.018407942 0.004972384 0.274777337 1.2338010
##      cook.d      hat
## 1 1.359813e-01 0.36242057
## 2 6.088590e-02 0.20142292
## 3 3.459292e-03 0.23577992
## 4 1.177436e-02 0.20216121
## 5 4.481989e-02 0.07807664
## 6 1.288001e-01 0.14249406
## 7 1.189678e-01 0.14319400
## 8 3.074773e-02 0.09232509
## 9 2.905892e-02 0.18848578
## 10 4.382906e-02 0.19164343
## 11 1.830555e-02 0.17180913
## 12 2.150305e-02 0.12266013
## 13 1.609381e-02 0.08659086
## 14 1.700455e-04 0.10329971
## 15 1.081585e-01 0.18951329
## 16 9.875246e-01 0.61389878
## 17 7.964918e-03 0.22520021
## 18 9.064674e-03 0.10826245
## 19 2.347849e-02 0.11113378
## 20 1.357895e-02 0.20575666
## 21 9.344561e-01 0.61296861
## 22 4.643968e-06 0.06009151
## 23 1.370158e-02 0.05954171
## 24 5.159177e-04 0.09178965
## 25 1.801908e-03 0.09981384

```

```
## 26 1.087484e-03 0.09948725
## 27 8.231598e-03 0.08683178
## 28 1.537417e-02 0.11334702
```

#Puntos de Balanceo, Influyentes y Atípicos

```
par(mfrow=c(1,1))
influencePlot(model.lasso1,panel.first=grid(),ylab='Residuos Studentizados')
```



```
##      StudRes      Hat      CookD
## 6  -2.111444 0.1424941 0.1288001
## 7   2.006935 0.1431940 0.1189678
## 16 -1.853127 0.6138988 0.9875246
## 21 -1.799176 0.6129686 0.9344561
```

Dónde observamos que las observaciones 16,21 son influyentes a nuestro modelo y las 6,7 atípicas. Los puntos dentro de la base de datos lucen así y procedemos a ilustrarlos para que cuando un experto en el tema pueda considerarlos y evaluar si fueron errores de mediciones o que ocurre realmente con ellos. A su vez mostraremos los efectos de cada covariable con densidad. En general observamos que la relación de NIR2, 18 y 28 describen relaciones lineales negativas, inversamente proporcionales, a diferencia de la relación con NIR6 que es positiva, directamente proporcional.

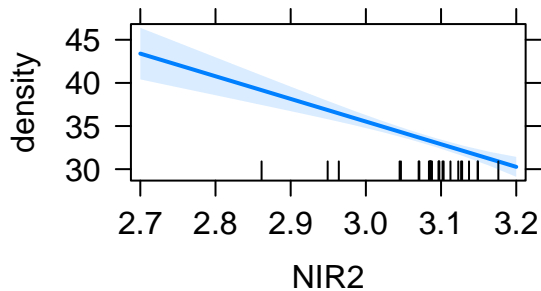
```
X[c(6,7,16,21),c(2,6,18,28,31)]
```

```
##      NIR2   NIR6  NIR18  NIR28 density
```

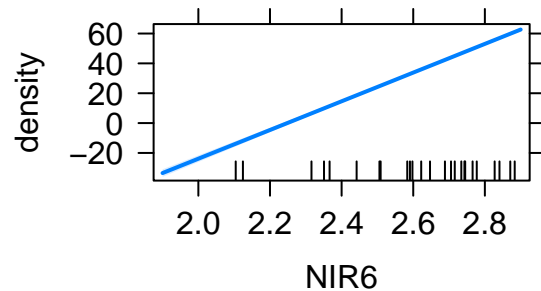
```
## 6  3.0849 2.5089 1.1999 1.0562  60.48
## 7  3.1372 2.9268 2.8934 1.4930  40.10
## 16 3.1229 2.9345 3.3254 1.8021   0.00
## 21 2.6803 1.8602 1.3031 1.1352   0.00
```

```
plot(allEffects(model.lasso1))
```

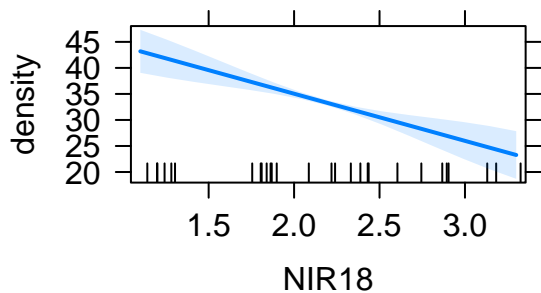
NIR2 effect plot



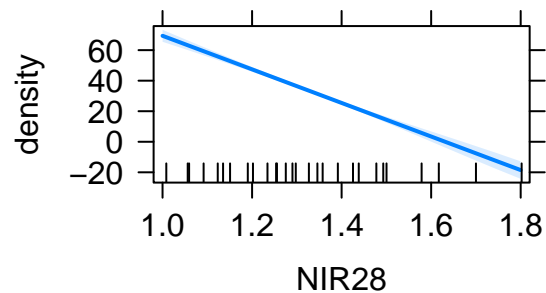
NIR6 effect plot



NIR18 effect plot



NIR28 effect plot



Generamos una predicción con el modelo seleccionado:

```
x.nuevo<- data.frame(NIR2=3.06,NIR6=2.55,NIR18=2.14,NIR28=1.32)
pred.media = predict(model.lasso1,x.nuevo,interval = "confidence")
pred.media
```

```
##      fit      lwr      upr
## 1 29.70163 29.21701 30.18625
```

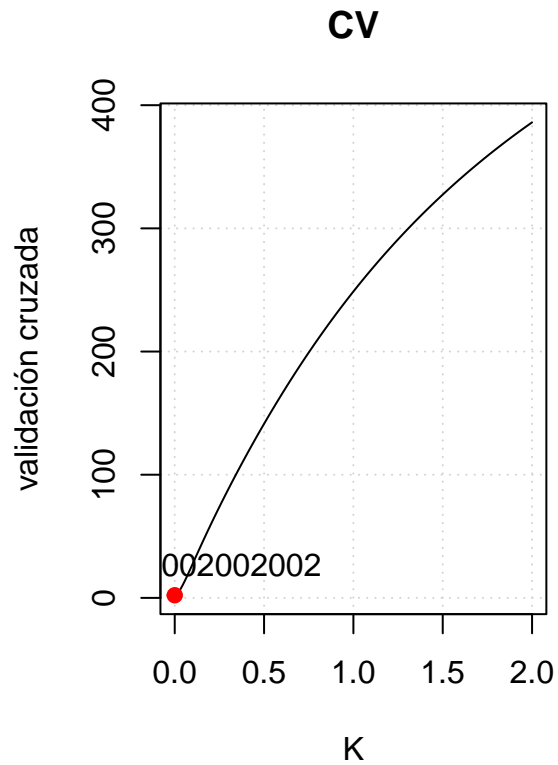
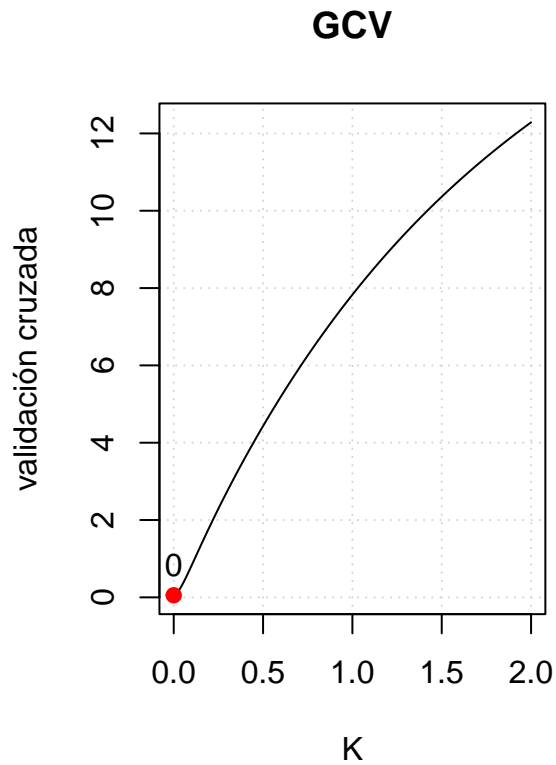
A pesar de que evidenciamos claras mejoras en los problemas de multicolinealidad dada la selección de variables, procederemos a realizar la regresión de ridge que propone la siguiente estimación:

```
# Regresión ridge
K = seq(from=0,to=2,length.out = 1000)
ridgedensity = lmridge(density~NIR2+NIR6+NIR18+NIR28,
                       data=X,K=K,scaling='sc')
#####
criterios<- kest(ridgedensity)
criterios
```



```
## Ridge k from different Authors
##
##
## k values
## Minimum CV at K 0.00000
## Minimum GCV at K 0.00000
## Thisted (1976): 0.00008
## LW (lm.ridge) 0.00374
## LW (1976) 0.00027
## HKB (1975) 0.00016
## Dwividi & Srivastava (1978): 0.00004
## Kibria (2003) (AM) 0.00216
## Kibria 2003 (GM): 0.00046
## Kibria 2003 (MED): 0.00044
## Muniz et al. 2009 (KM2): 110.61039
## Muniz et al. 2009 (KM3): 0.08692
## Muniz et al. 2009 (KM4): 46.46118
## Muniz et al. 2009 (KM5): 0.02152
## Muniz et al. 2009 (KM6): 69.37817
## Mansson et al. 2012 (KMN8): 110.65147
## Mansson et al. 2012 (KMN9): 0.08408
## Mansson et al. 2012 (KMN10): 46.90013
## Mansson et al. 2012 (KMN11): 0.02132
## Mansson et al. 2012 (KMN12): 69.46417
## Dorugade et al. 2010: 0.00000
## Dorugade et al. 2014: 0.02584
```

```
par(mfrow=c(1,2))
plot(K,criterios$GCV,panel.first=grid(),type='l',xlab='K',ylab='validación cruzada',main='GCV')
points(K[criterios$GCV==min(criterios$GCV)],
       criterios$GCV[criterios$GCV==min(criterios$GCV)],
       pch=19,col='red1')
text(K[criterios$GCV==min(criterios$GCV)],
     criterios$GCV[criterios$GCV==min(criterios$GCV)],
     labels=paste(K[1]),pos=3)
#####
plot(K,criterios$CV,panel.first=grid(),type='l',xlab='K',ylab='validación cruzada',main='CV')
points(K[criterios$CV==min(criterios$CV)],
       criterios$CV[criterios$CV==min(criterios$CV)],
       pch=19,col='red1')
text(K[criterios$CV==min(criterios$CV)],
     criterios$CV[criterios$CV==min(criterios$CV)],
     labels=paste(K[2]),pos=3)
```



```
#####
lambda<-c(K[ criterios$GCV==min(criterios$GCV)],
          K[ criterios$CV==min(criterios$CV)])
lambda
```

```
## [1] 0 0
```

```
#####
ridgedensity<-lmridge(density~NIR2+NIR6+NIR18+NIR28,
                      data=X,K=0.01,scaling='sc')
summary(ridgedensity)
```

```
##
## Call:
## lmridge.default(formula = density ~ NIR2 + NIR6 + NIR18 + NIR28,
## data = X, K = 0.01, scaling = "sc")
##
##
## Coefficients: for Ridge parameter K= 0.01
##      Estimate Estimate (Sc) StdErr (Sc) t-value (Sc) Pr(>|t|)
## Intercept    4.4054    -45.9862   16.1287    -2.8512   0.0089 **
## NIR2        -18.4144     -9.5288    2.2346    -4.2641   0.0003 ***
## NIR6         93.1911    128.2731    2.5280    50.7405   <2e-16 ***
## NIR18        -12.0153    -41.1087    4.8468    -8.4815   <2e-16 ***
## NIR28        -98.6616   -102.9801    4.5424   -22.6706   <2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Ridge Summary
##      R2      adj-R2    DF ridge      F      AIC      BIC
##    0.96830    0.96440    3.56407 2628.64660    20.40490   118.45470
## Ridge minimum MSE= 329.5864 at K= 0.01
## P-value for F-test ( 3.56407 , 24.13811 ) = 4.389333e-31
## -----
```

```
vif.lmridge(ridgedensity)
```

```
##      NIR2      NIR6      NIR18      NIR28
## k=0.01 2.67937 3.42909 12.60473 11.07122
```

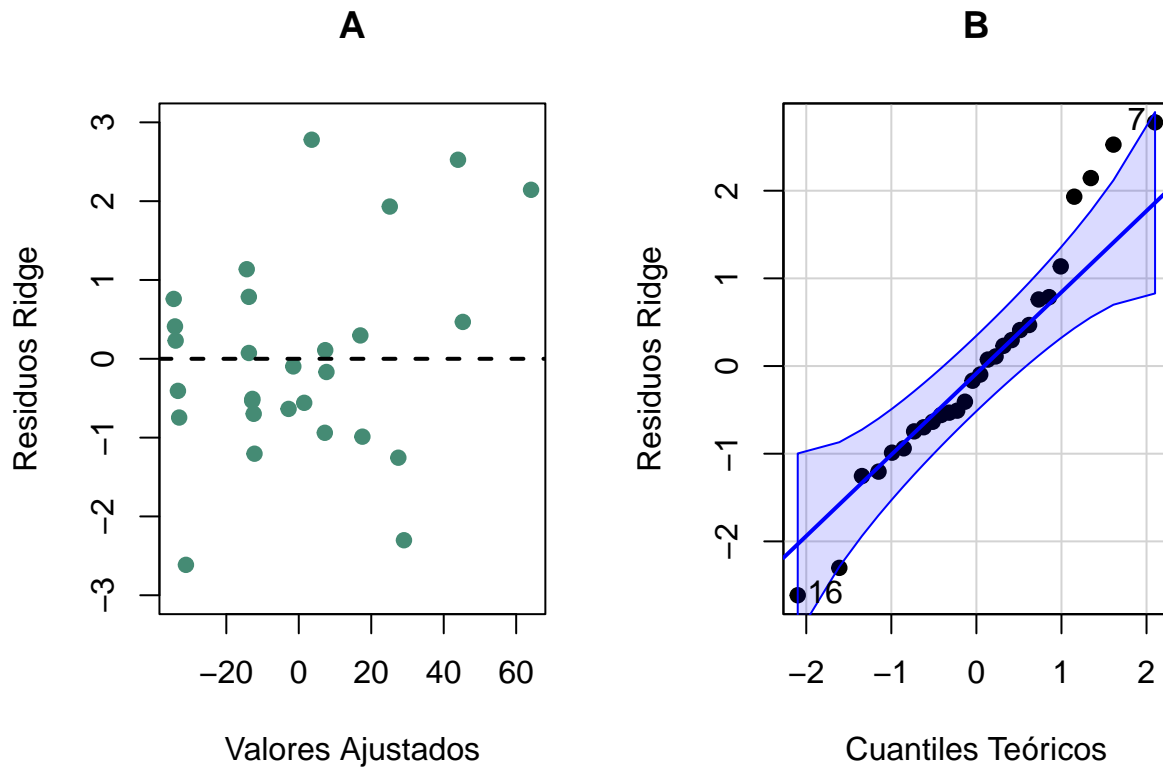
```
car::vif(model.lasso1)
```

```
##      NIR2      NIR6      NIR18      NIR28
## 2.967327 4.203285 31.085734 26.983026
```

```
lmridge::vif.lmridge(ridgedensity)
```

```
##      NIR2      NIR6      NIR18      NIR28
## k=0.01 2.67937 3.42909 12.60473 11.07122
```

```
par(mfrow=c(1,2))
plot(fitted.values(ridgedensity),residuals(ridgedensity),pch=19,
     ylab='Residuos Ridge',xlab='Valores Ajustados',main='A',col="aquamarine4",
     ylim=c(-3,3))
abline(h=0,lwd=2,lty=2)
car::qqPlot(residuals(ridgedensity),xlab="Cuantiles Teóricos",ylab="Residuos Ridge",main="B",pch=19)
```



```
## [1] 7 16
```

```
print('H0: Homocedasticidad vs H1: No hay homocedasticidad')
```

```
## [1] "H0: Homocedasticidad vs H1: No hay homocedasticidad"
```

```
bptest(ridgedensity)
```

```
##
## studentized Breusch-Pagan test
##
## data: ridgedensity
## BP = 1.6317, df = 4, p-value = 0.8031
```

```
print('H0: Normalidad vs H1: No Normalidad')
```

```
## [1] "H0: Normalidad vs H1: No Normalidad"
```

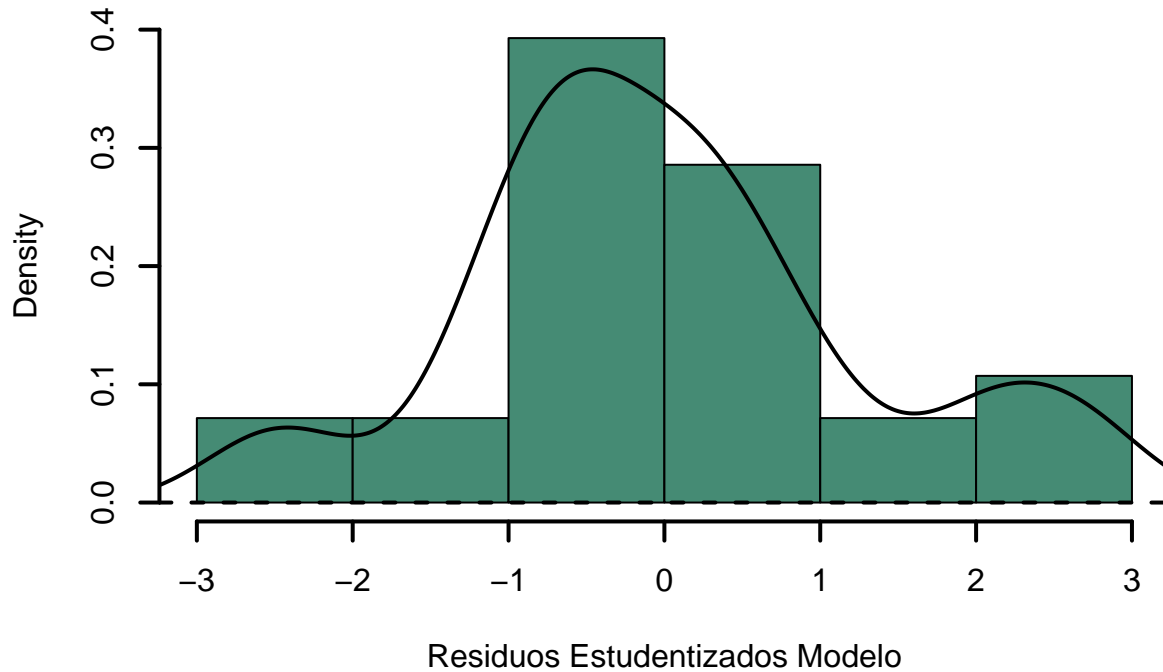
```
shapiro.test(residuals(ridgedensity))
```

```
##
## Shapiro-Wilk normality test
##
## data: residuals(ridgedensity)
## W = 0.96141, p-value = 0.3764
```

```

par(mfrow=c(1,1))
hist(residuals(ridgedensity),lwd=2,col='aquamarine4',
freq=F,ylim=c(0,0.4),xlab='Residuos Estudentizados Modelo',main='')
lines(density(residuals(ridgedensity)),lwd=2,col='black')
abline(h=0,lty=2,lwd=2)

```



```

#Para generar predicciones no es posible en una función en R
x.nuevo<- data.frame(NIR2=3.06,NIR6=2.55,NIR18=2.14,NIR28=1.32)
pred.media = predict(ridgedensity,x.nuevo,interval = "confidence")
pred.media

```

```
## [1] 29.74832
```

```

#Por lo cuál creamos la estimación de la varianza y matrix de covarianzas
# Obtener los coeficientes estimados y la matriz de diseño
beta_hat <- coef(ridgedensity)
X <- model.matrix(model.lasso1)

# Calcular la varianza del error
sigma2_hat <- sum(residuals(ridgedensity)^2) / (nrow(X)-length(coefficients(model.lasso1)))

# Calcular la matriz de covarianza de los coeficientes
V_beta_hat <- diag(residuals(ridgedensity)^2)*solve(t(X) %*% X
+0.01*diag(ncol(X))) %*% t(X) %*%X %*% solve(t(X) %*%

```

```

X + 0.01 * diag(ncol(X))
# Hacer una predicción para nuevas observaciones
x.nuevo<- data.frame(NIR2=3.06,NIR6=2.55,NIR18=2.14,NIR28=1.32)
X_new <- cbind(1, as.matrix(x.nuevo))
pred <- X_new %*% beta_hat
se_pred <- sqrt(diag(X_new %*% V_beta_hat %*% t(X_new))) * sqrt(sigma2_hat)

# Calcular los intervalos de confianza del 95%
lower <- pred - qt(0.975, nrow(iris)-4) * se_pred
upper <- pred + qt(0.975, nrow(iris)-4) * se_pred

# Mostrar los resultados
data.frame(lower, pred, upper)

```

```

##      lower      pred      upper
## 1 28.5759 29.74832 30.92074

```