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
lib_req<-c("glmnet","lmridge","scatterplot3d","plot3D","plotly","rgl","plot3Drgl",
           'effects','psych',
           'car','lmtest','MASS','latex2exp','orcutt',
           'nlme',"zoom",'ggfortify','readxl','pls')# Listado de librerias requeridas por el script
easypackages::packages(lib_req)
```

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

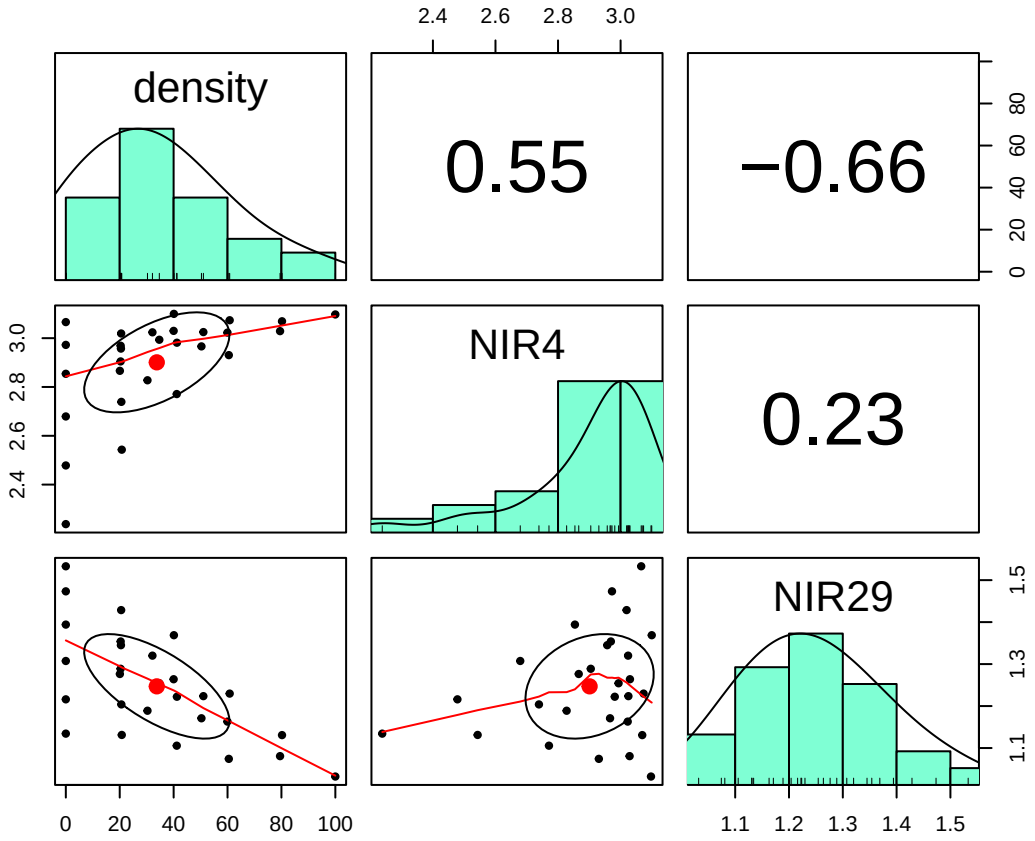
Antes de empezar con el proceso de seleccionar las variables para ajustar el modelo se crean funciones para optimizar el proceso de validación de supuestos, debido a que constantemente se deben realizar, estas funciones estan diseñadas para objetos lm.

```
##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)
}
```

Antes de generar el proceso de selección de variables recordaremos brevemente que una regresión lineal múltiple no es el conjunto de la suma de regresiones lineales simples, es decir estamos trabajando con espacios generados en función de un conjunto de covariables. Por lo cuál haremos un breve ejercicio ilustrativo de como se evidencia esto. En el diagrama de dispersión inicial vemos que la relación entre las covariables no se evidenciaba de una manera fuerte lineal, con coeficientes de correlación lineal débiles, pero en el diagrama de dispersión en 3D se nota relaciones fuertes.

```
par(mfrow=c(1,2))
psych::pairs.panels(X[,c(31,4,29)],
                    method = "pearson", # correlation method
                    hist.col = "aquamarine1",
                    density = TRUE,  # show density plots
                    ellipses = TRUE # show correlation ellipses
  )
```

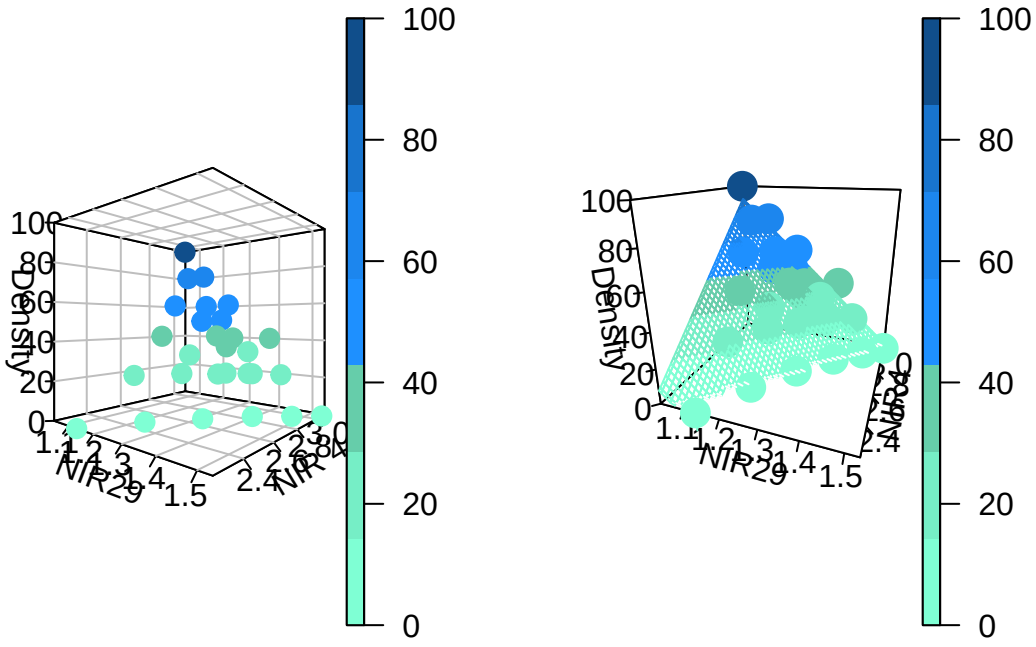



```
z<-X[,31];y<-X[,4];x<-X[,29];
scatter3D(x, y, z, phi = 0, bty = "b2",col = c('aquamarine','aquamarine2','aquamarine3','dodgerblue',
'dodgerblue2','dodgerblue3','dodgerblue4'),pch = 20, cex = 2,
ticktype = "detailed",xlab='NIR29',ylab='NIR 4', zlab='Density')
#Creamos un objeto para realizar las predicciones con elmodelo
objr<-lm(z ~ x+y)
summary(objr)
```

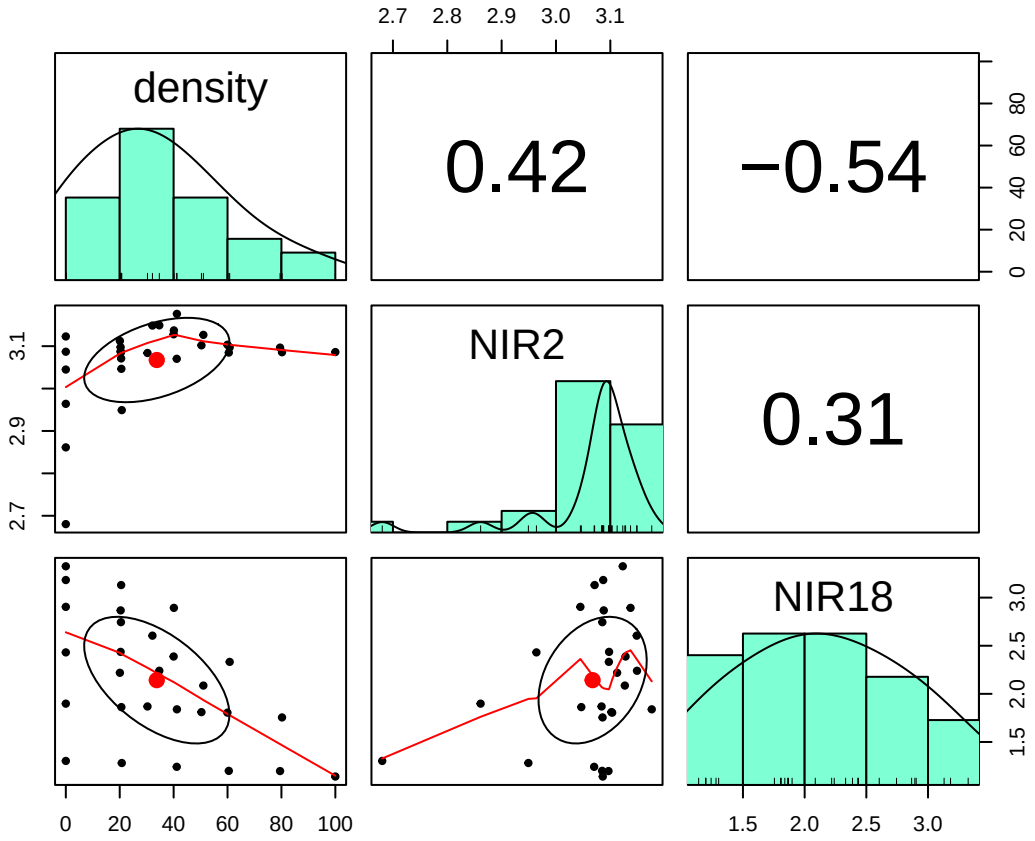
```
##
## Call:
## lm(formula = z ~ x + y)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.3050 -4.8871 -0.7899  2.2812 10.7069
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -24.845     17.179  -1.446    0.161
## x           -180.297      8.987 -20.063  < 2e-16 ***
## y             97.698      5.459  17.896 9.15e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.66 on 25 degrees of freedom
## Multiple R-squared:  0.9592, Adjusted R-squared:  0.9559
## F-statistic: 293.9 on 2 and 25 DF,  p-value: < 2.2e-16
```

```
#preparamos el modelado 3d
grid.lines = 42
x.pred <- seq(min(x), max(x), length.out = grid.lines)


y.pred <- seq(min(y), max(y), length.out = grid.lines)
xy <- expand.grid( x = x.pred, y = y.pred)
z.pred <- matrix(predict(objr, newdata = xy),
                 nrow = grid.lines, ncol = grid.lines)
# Marcamos las líneas de iteracción para que busquen la recta de regresión
fitpoints <- predict(objr)
#ploteamos la gráfica en 3d con recta de regresión
scatter3D(x, y, z, pch = 19, cex = 2,
theta = 20, phi = 20, ticktype = "detailed",
surf = list(x = x.pred, y = y.pred, z = z.pred,facets = NA, fit = fitpoints),
main = "",xlab='NIR29 ',zlab="Density",ylab='NIR4',
col = c('aquamarine','aquamarine2','aquamarine3',
'dodgerblue','dodgerblue2','dodgerblue3','dodgerblue4'))
```




```
par(mfrow=c(1,2))
psych::pairs.panels(X[,c(31,2,18)],
                    method = "pearson", # correlation method
                    hist.col = "aquamarine1",
                    density = TRUE,  # show density plots
                    ellipses = TRUE # show correlation ellipses
  )
```

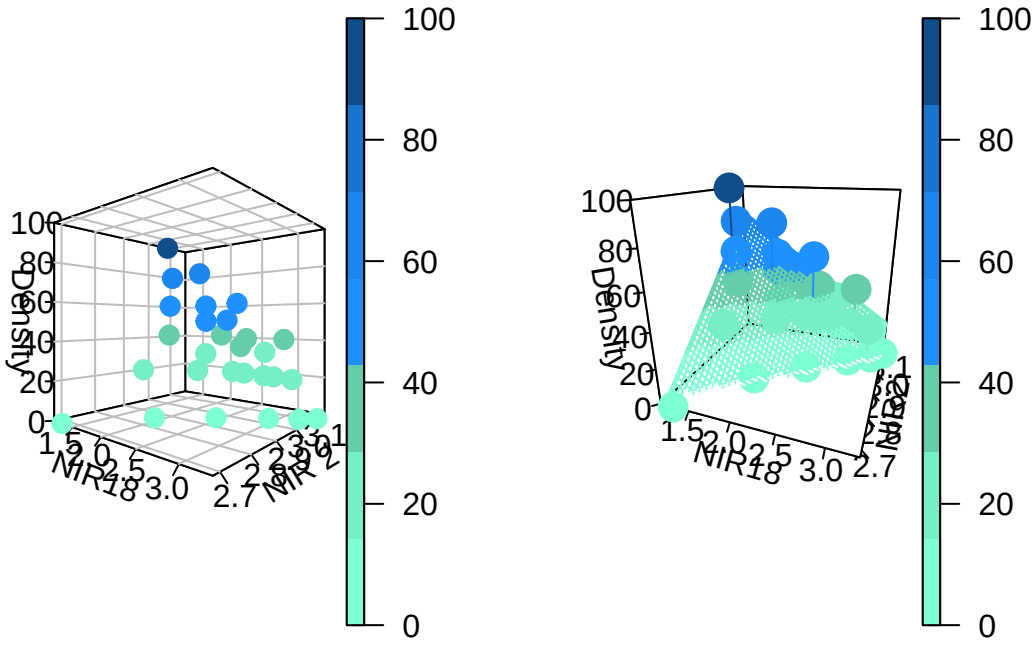



```
z<-X[,31];y<-X[,2];x<-X[,18];
scatter3D(x, y, z, phi = 0, bty = "b2",col = c('aquamarine','aquamarine2','aquamarine3','dodgerblue',
'dodgerblue2','dodgerblue3','dodgerblue4'),pch = 20, cex = 2,
ticktype = "detailed",
xlab='NIR18',ylab='NIR 2', zlab='Density')
#Creamos un objeto para realizar las predicciones con elmodelo
objr<-lm(z ~ x+y)
summary(objr)
```

```
##
## Call:
## lm(formula = z ~ x + y)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -21.003 -10.014  -3.323   9.239  32.444
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -443.361     96.022  -4.617    1e-04 ***
## x            -30.451      4.865  -6.259 1.51e-06 ***
## y            176.813     32.168   5.497 1.04e-05 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.84 on 25 degrees of freedom
## Multiple R-squared:  0.6805, Adjusted R-squared:  0.655
## F-statistic: 26.63 on 2 and 25 DF,  p-value: 6.389e-07
```

```
#preparamos el modelado 3d
grid.lines = 42
x.pred <- seq(min(x), max(x), length.out = grid.lines)


y.pred <- seq(min(y), max(y), length.out = grid.lines)
xy <- expand.grid( x = x.pred, y = y.pred)
z.pred <- matrix(predict(objr, newdata = xy),
                 nrow = grid.lines, ncol = grid.lines)
# Marcamos las líneas de iteracción para que busquen la recta de regresión
fitpoints <- predict(objr)
#ploteamos la gráfica en 3d con recta de regresión
scatter3D(x, y, z, pch = 19, cex = 2,
theta = 20, phi = 20, ticktype = "detailed",
surf = list(x = x.pred, y = y.pred, z = z.pred,
facets = NA, fit = fitpoints), main = "",
xlab='NIR18 ',zlab="Density",ylab='NIR2',
col = c('aquamarine','aquamarine2','aquamarine3',
'dodgerblue','dodgerblue2','dodgerblue3','dodgerblue4'))
```



2 Selección de variables

En el proceso de selección de variables se procede a realizar la Regresion 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 :

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

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 seran cero. No hay solución analitica para $\hat{\beta}_{lasso}$ por lo que se usan algoritmos para la estimación, como lo es la funcion de glmnet de la libreria glmnet.

2.1.1 Modelo a realizar regresión LASSO

Como se establecio 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)=σ^2, es decir, varianza constante y E(ε)=0 . Por ende es necesario proponer un modelo con p<n, en el cual se eliminaran las variables con menor correlación con la variable density. Dicho modelo se expresa acontinuación y se evaluan 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
```

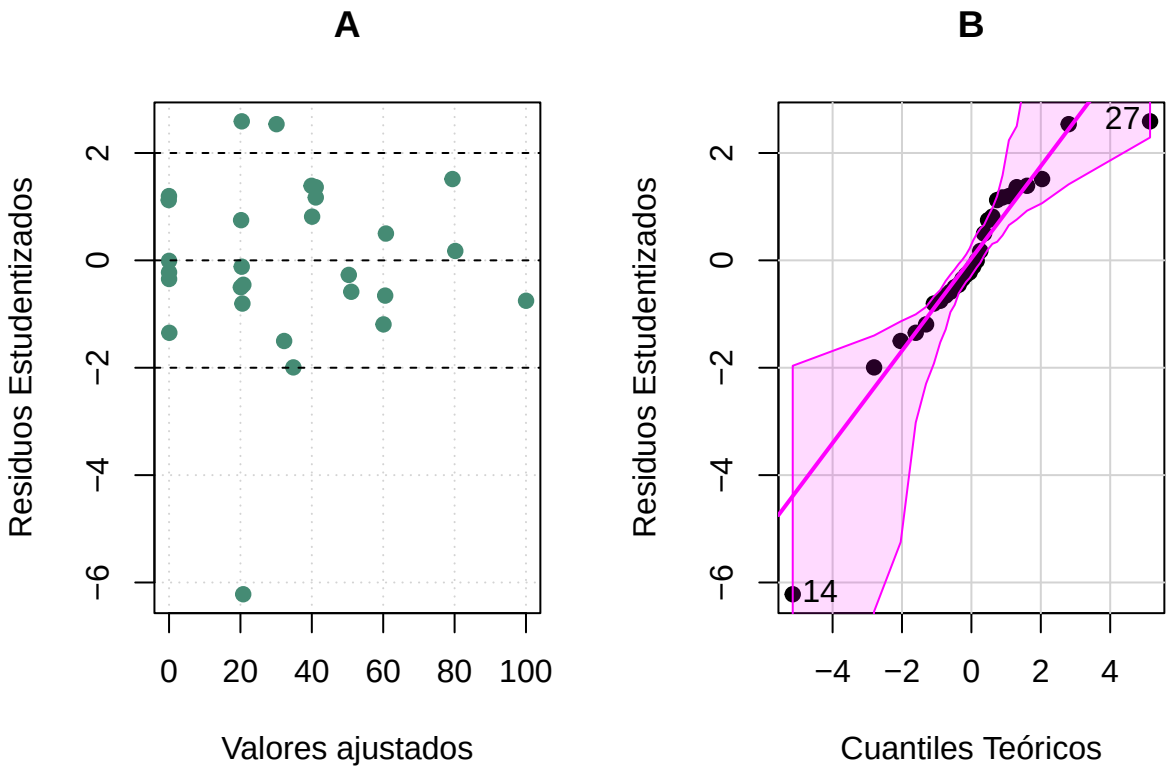
```
car::vif(model)[11:16]
```

```
##     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
```

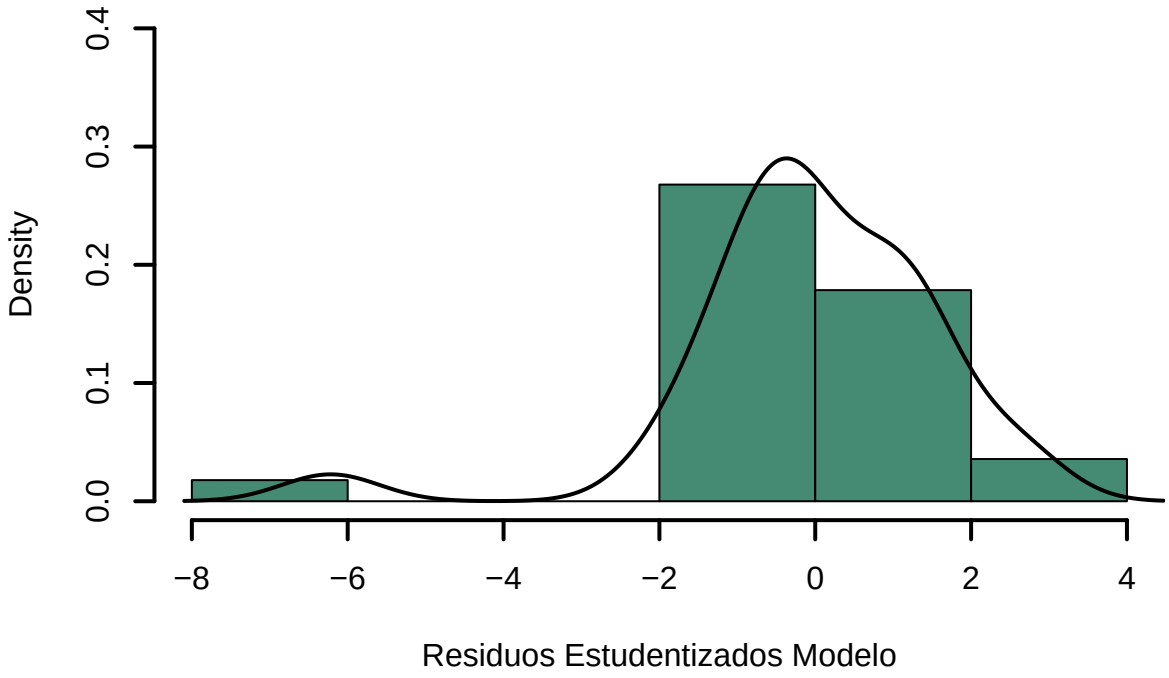
```
validaciongrafica(model)
```




```
## [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
```

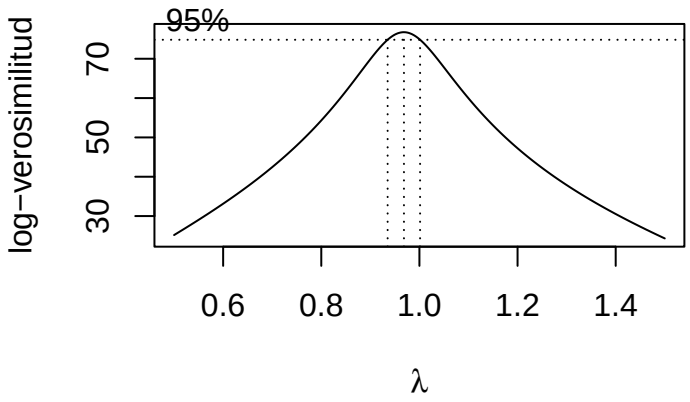
Mediante el grafico y los valores P asociados a la homocedasticidad y normalidad de residuos se evidencia el incumplimiento de la normalidad de los residuos.

```
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')
```



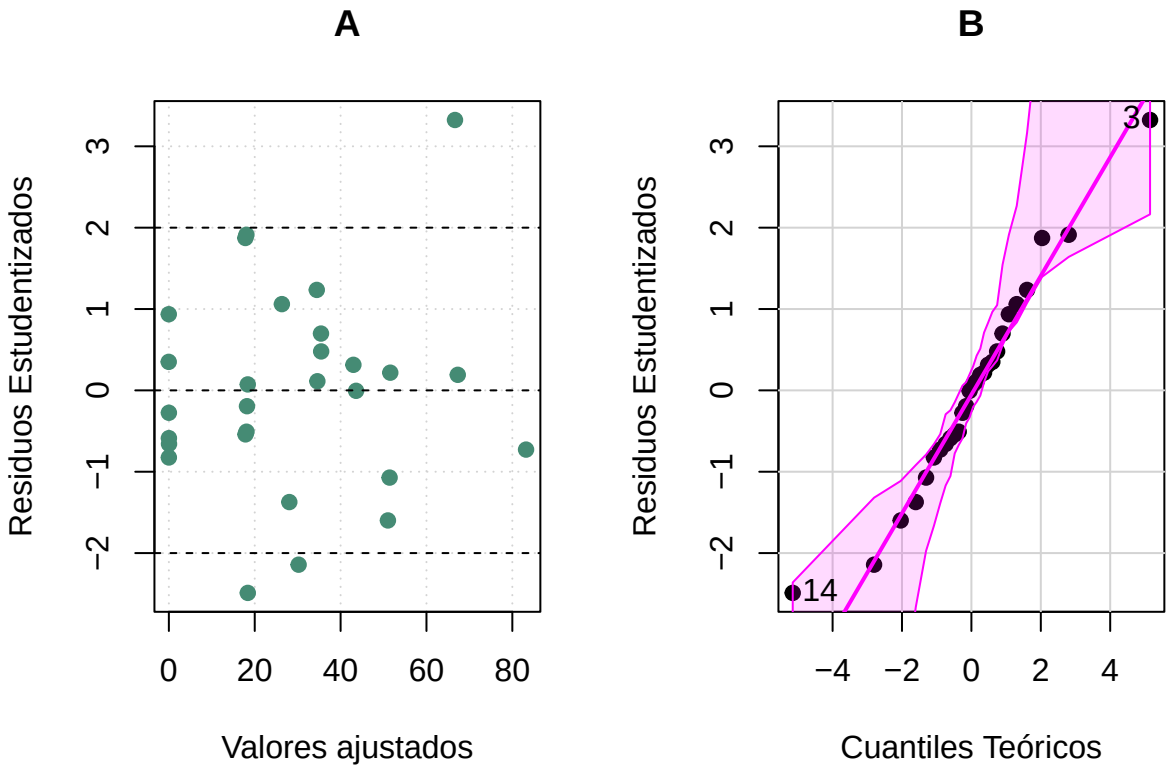
Como no se cumple el supuesto de normalidad se procede a corregir mediante el metodo de BoxCox y se verifica el cumplimiento de los mismos.

```
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

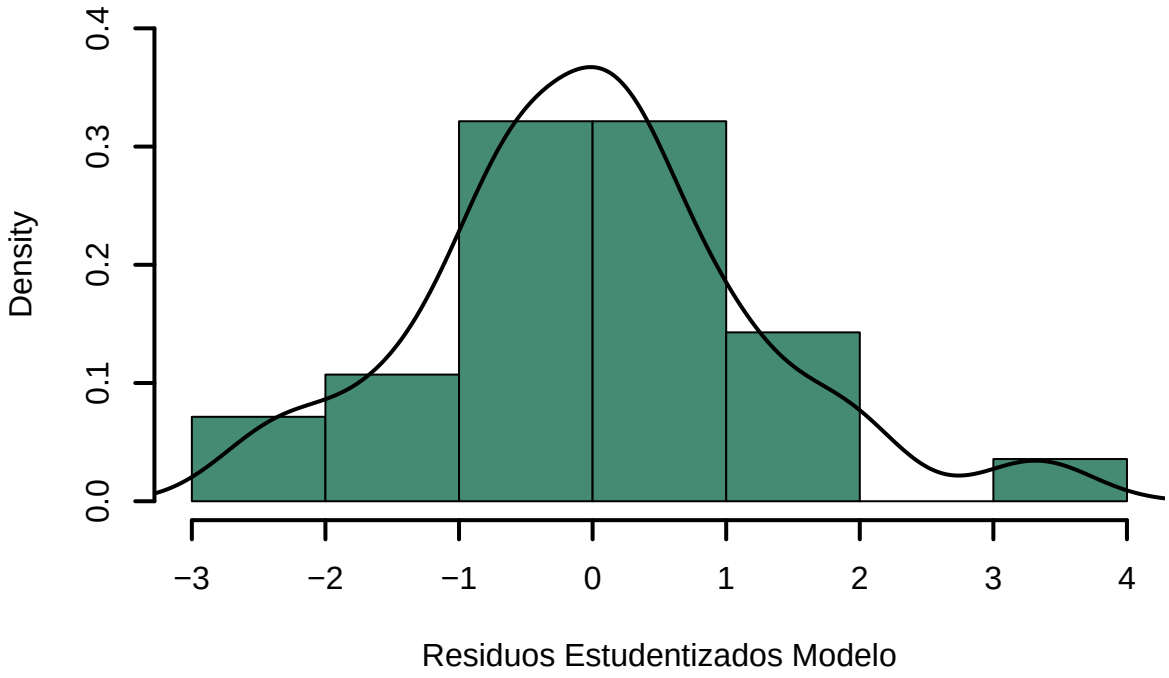
[1] "Breusch Pagan Test;H0: Homocedasticidad vs H1: No Homocedasticidad"

```
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 requerimentos 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,..,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 cuadratico 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 estimacion utilizando la muestra de entrenamiento k. para cada λ se calcula:

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

Y la desviación estándar:

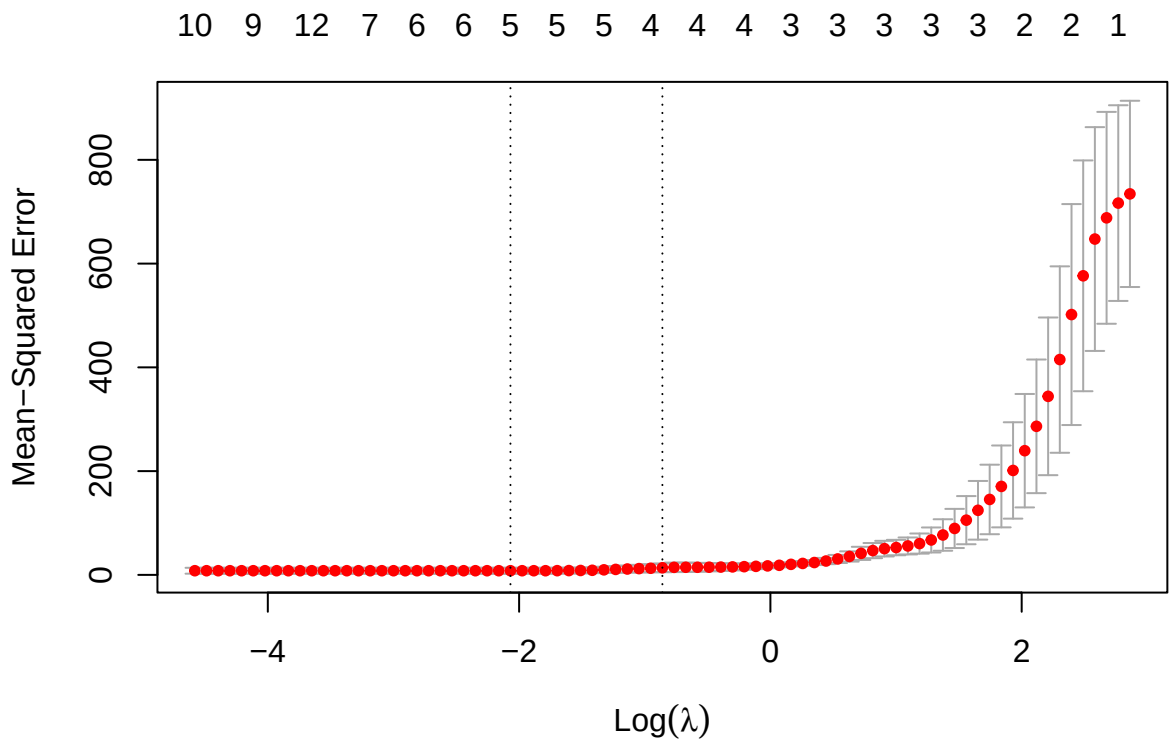
$SD(\lambda) = \sqrt{\sum_{i=1}^{K} \frac{(ECM_k(\lambda) - CV(\lambda))^2}{K-1}}$

λ optimo: minimiza $CV\lambda$ o

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

La maximisación 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    .
## NIR3    .
## NIR4    .
## NIR5    .
## NIR6   81.773107
## NIR12   .
## NIR13   .
```

```
## NIR14   .
## NIR15   .
## NIR16   .
## NIR17   .
## NIR18  -1.652552
## NIR19   .
## NIR20   .
## NIR21   .
## NIR22   .
## NIR23   .
## NIR24   .
## NIR25   .
## NIR26   .
## NIR27   .
## NIR28 -75.029928
## NIR29 -82.613375
## NIR30   .
```

La selección de variables por medio del estimador LASSO son: NIR2, NIR6, NIR18, NIR28, NIR29. Veremos el resumen del modelo para evaluar las pruebas individuales t a ver si podemos considerar descartar alguna covariable adicional. Cabe aclarar que a pesar de haber realizado la selección de variables, no podemos dejarlo todo a los métodos sino por el contrario seguir indagando e intetando llegar al mejor modelo posible.

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

```
##
## Call:
## lm(formula = density ~ NIR2 + NIR6 + NIR18 + NIR28 + NIR29, data = X)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.99054 -0.86304  0.07117  0.81460  1.63877
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   20.808     15.525   1.340 0.193832
## NIR2         -27.826      4.424  -6.290 2.49e-06 ***
## NIR6          96.932      2.036  47.615  < 2e-16 ***
## NIR18         -9.649      2.071  -4.659 0.000121 ***
## NIR28       -123.484     18.545  -6.659 1.08e-06 ***
## NIR29         24.496     31.814   0.770 0.449495
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.18 on 22 degrees of freedom
## Multiple R-squared:  0.9984, Adjusted R-squared:  0.9981
## F-statistic:  2817 on 5 and 22 DF,  p-value: < 2.2e-16
```

Consiguiente 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 β=$\begin{bmatrix}\beta_1\\ \beta_2\end{bmatrix}$

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

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

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 esta 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 estadistico.

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

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

Se realiza la respectiva prueba con la funcion anova, asumiento que el modelo reducido es aquel β_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 acontinuación.

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 $T = (t_1, ..., t_{p-1})$ 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. $R = TAT^T$ (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.
- Observamos relaciones negativas entre la densidad y el conjunto de covariables NIR2, NIR 18 y NIR28 SI asumimos que se presentan aumentos con las demas covariables constantes, y se presenta una relación positiva entre la densidad y el NIR6 si mantenemos las demas covariables constantes.

```
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.

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

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

Hipotesis: $H_0 : \beta_1 = 0$

Estadistico 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: $H_0 : \beta_2 = 0$

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

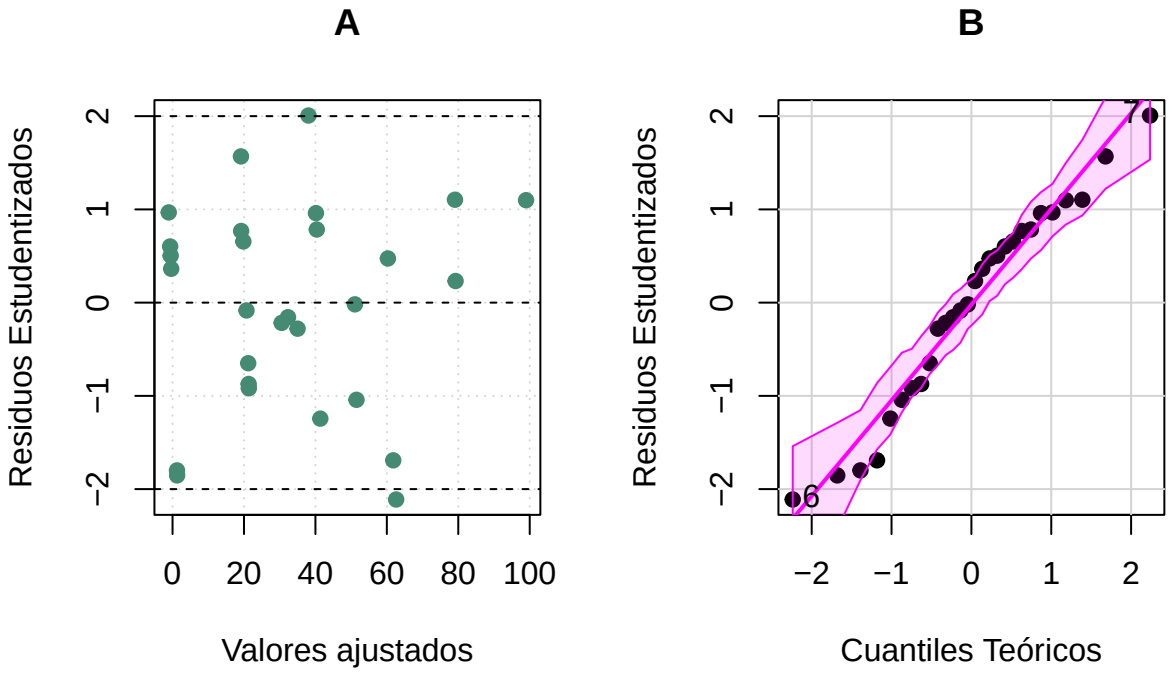
Modelo: $y = \beta_0 + NIR2\beta_1 + NIR6\beta_2 + NIR18\beta_3 + \varepsilon$

Hipotesis: $H_0 : \beta_3 = 0$

En cada hipoteis 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

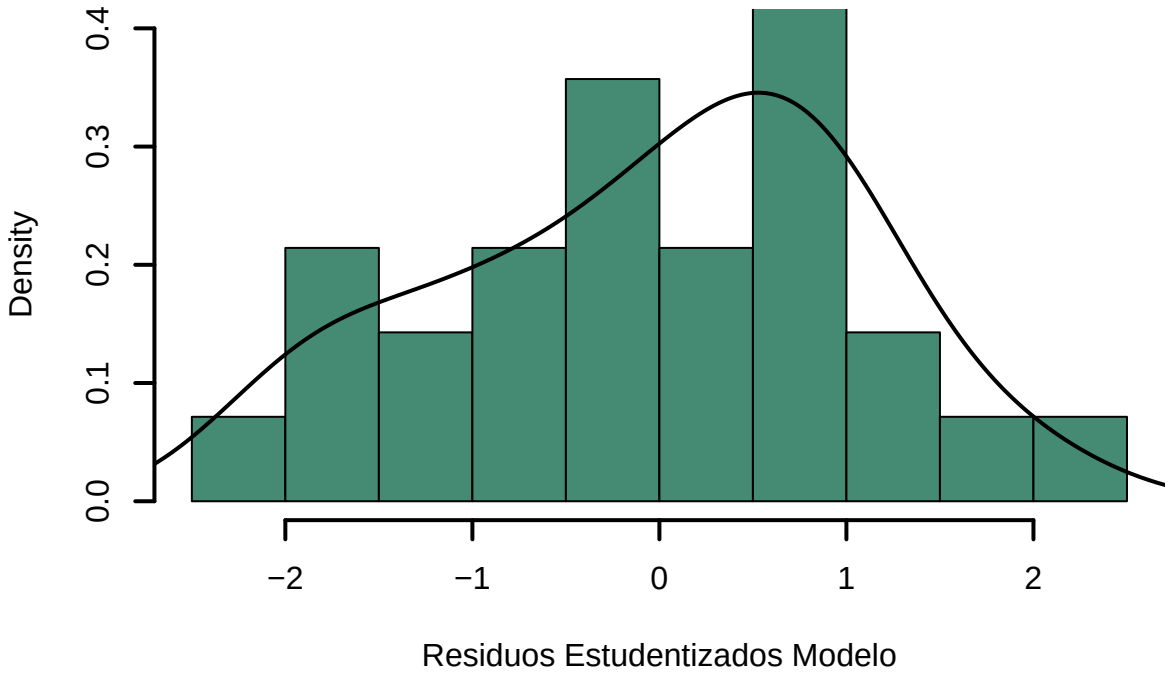
3.2 Validación de supuestos

```
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 puntosa atípicos e influyentes

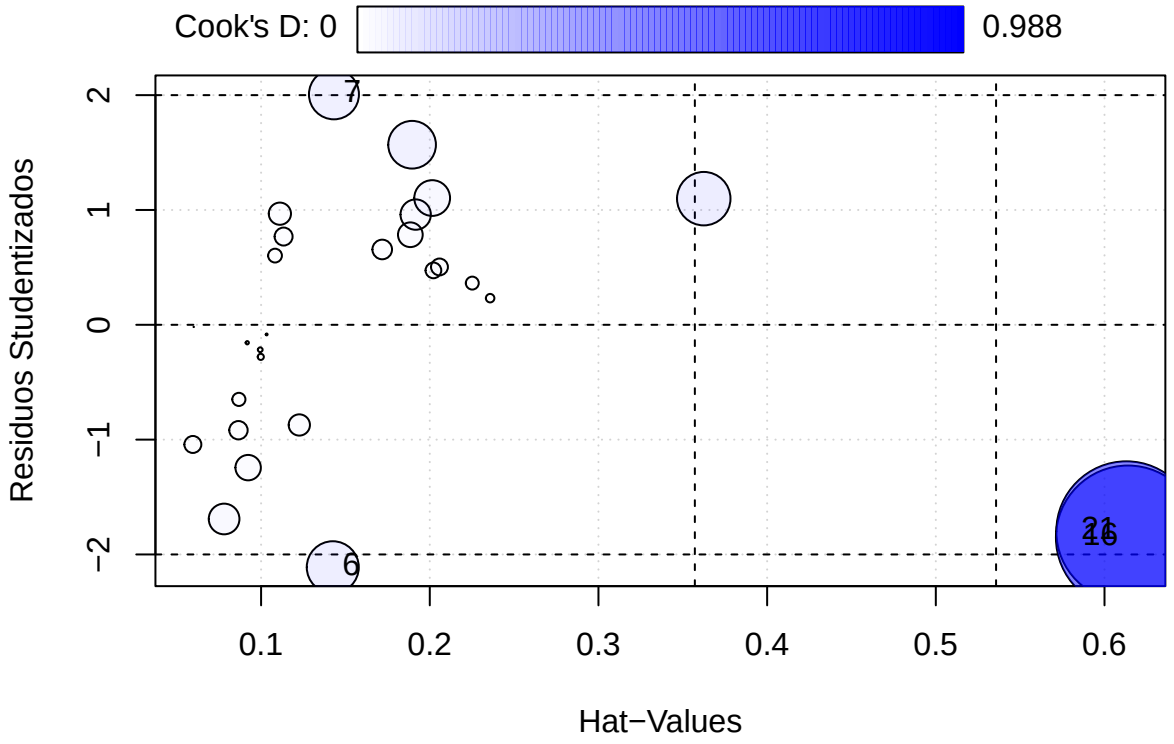
Para esto utilizaremos la función influence.measures()

```
influence.measures(model.lasso1)
```

```
## Influence measures of
##   lm(formula = density ~ NIR2 + NIR6 + NIR18 + NIR28, data = X) :
##
##      dfb.1_  dfb.NIR2  dfb.NIR6 dfb.NIR1 dfb.NIR28    dffit cov.r   cook.d
## 1   0.14096 -0.373395  0.649685 -0.33177   0.18411  0.82826 1.500 1.36e-01
## 2   0.27322 -0.305493  0.372058  0.07340  -0.15731  0.55437 1.194 6.09e-02
## 3  -0.02220 -0.019232  0.075059 -0.08171   0.06129  0.12878 1.614 3.46e-03
## 4   0.14484 -0.106548  0.083888  0.14236  -0.15812  0.23851 1.488 1.18e-02
## 5  -0.08091  0.074553 -0.179345 -0.04769   0.11805 -0.49216 0.735 4.48e-02
## 6   0.28982 -0.191413 -0.085880  0.40391  -0.28227 -0.86071 0.579 1.29e-01
## 7   0.25096 -0.167777  0.160191  0.43368  -0.40108  0.82045 0.629 1.19e-01
## 8  -0.05430 -0.069031  0.088418 -0.26894   0.26591 -0.39673 0.979 3.07e-02
## 9  -0.31667  0.306438 -0.157325 -0.13253   0.13285  0.37797 1.341 2.91e-02
## 10 -0.25575  0.271697 -0.198103 -0.17086   0.14730  0.46733 1.258 4.38e-02
## 11  0.09966 -0.157527  0.149988  0.00659   0.01782  0.29877 1.369 1.83e-02
## 12 -0.11748  0.038782  0.043501 -0.21867   0.18849 -0.32619 1.201 2.15e-02
```

```
## 13  0.03205 -0.131950  0.183335 -0.15597   0.12961 -0.28270 1.133 1.61e-02
## 14  0.00823 -0.016193  0.021363 -0.00371   0.00259 -0.02852 1.391 1.70e-04
## 15 -0.09547  0.157450 -0.316257 -0.20703   0.19369  0.75836 0.907 1.08e-01
## 16  0.95360  0.170445 -0.962860  1.79595  -1.99192 -2.33670 1.566 9.88e-01
## 17 -0.05625 -0.012588  0.045322 -0.09968   0.12613  0.19576 1.564 7.96e-03
## 18 -0.00244 -0.003507 -0.038110  0.00356   0.03279  0.20992 1.290 9.06e-03
## 19  0.11085 -0.061830 -0.111279  0.06139  -0.01992  0.34215 1.141 2.35e-02
## 20  0.14056 -0.100783 -0.051670  0.03438  -0.02407  0.25630 1.485 1.36e-02
## 21 -1.63766  1.547939 -0.334048 -0.01507   0.06714 -2.26423 1.624 9.34e-01
## 22  0.00023 -0.000734 -0.000306 -0.00120   0.00153 -0.00471 1.329 4.64e-06
## 23  0.02319 -0.062385  0.020799 -0.04146   0.06834 -0.26223 1.044 1.37e-02
## 24  0.00721 -0.019265  0.017150 -0.02612   0.02270 -0.04970 1.368 5.16e-04
## 25  0.03863 -0.065311  0.056280 -0.03468   0.03119 -0.09300 1.363 1.80e-03
## 26  0.02213 -0.046422  0.053056 -0.02059   0.01966 -0.07220 1.372 1.09e-03
## 27 -0.05245  0.009424  0.035150 -0.11578   0.09533 -0.20031 1.244 8.23e-03
## 28 -0.15921  0.213025 -0.202562  0.01841   0.00497  0.27478 1.234 1.54e-02
##       hat inf
## 1  0.3624
## 2  0.2014
## 3  0.2358
## 4  0.2022
## 5  0.0781
## 6  0.1425
## 7  0.1432
## 8  0.0923
## 9  0.1885
## 10 0.1916
## 11 0.1718
## 12 0.1227
## 13 0.0866
## 14 0.1033
## 15 0.1895
## 16 0.6139   *
## 17 0.2252
## 18 0.1083
## 19 0.1111
## 20 0.2058
## 21 0.6130   *
## 22 0.0601
## 23 0.0595
## 24 0.0918
## 25 0.0998
## 26 0.0995
## 27 0.0868
## 28 0.1133
```

```
#Puntos de Balanceo, Influyentes y Atípicos
par(mfrow=c(1,1))
influencePlot(model.lasso1,panel.first=grid(),ylab='Residuos Studentizados')
```



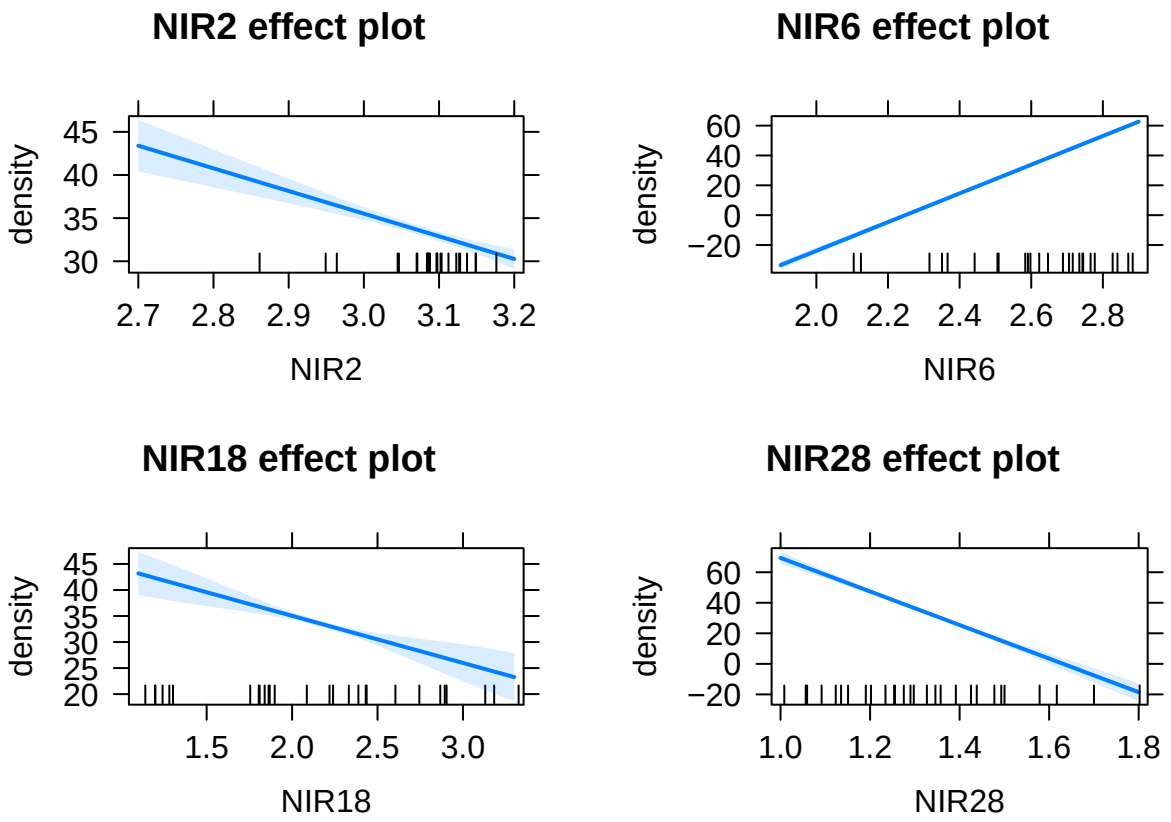

```
##      StudRes       Hat     CookD
## 6  -2.111444 0.1424941 0.1288001
## 7   2.006935 0.1431940 0.1189678
## 16 -1.853127 0.6138988 0.9875246
## 21 -1.799176 0.6129686 0.9344561
```

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))
```



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
```

4 Regresión ridge

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 tiene como objetivo minimizar la siguiente suma de cuadrados penalizada:

$$S_k(\beta) = \sum_{i=1}^{n}(y_i - z^T\beta)^2 + k\sum_{j=1}^{p-1}\beta_j^2$$

Al realizar derivada e igualar a 0 se obtienen las siguiente estimación:

$$\hat{\beta}_k = (R + kI)^{-1}Z^T y$$

Donde:

$$E(\hat{\beta}_k) = E((I + kR^{-1})^{-1}\hat{\beta}) = C\beta$$

y

$$V(\hat{\beta}_k) = \sigma^2 C^T R^{-1} C$$

Por ultimo se calcula el error cuadrático medio de $\hat{\beta}_k$

$$\begin{aligned} ECM(\hat{\beta}_k) &= trV(\hat{\beta}_k) + (E(\hat{\beta}_k) - \beta)^t (E(\hat{\beta}_k) - \beta) \\ &= \sigma^2 tr(R^{-1} C^T C) + \beta^T (C - I)^T (C - I)\beta \\ &= \sigma^2 \sum_{i=1}^{p-1} \frac{\lambda_j}{(\lambda_j + k)^2} + \sum_{i=1}^{p-1} \frac{\alpha_j^2 k^2}{(\lambda_j + k)^2} \end{aligned}$$

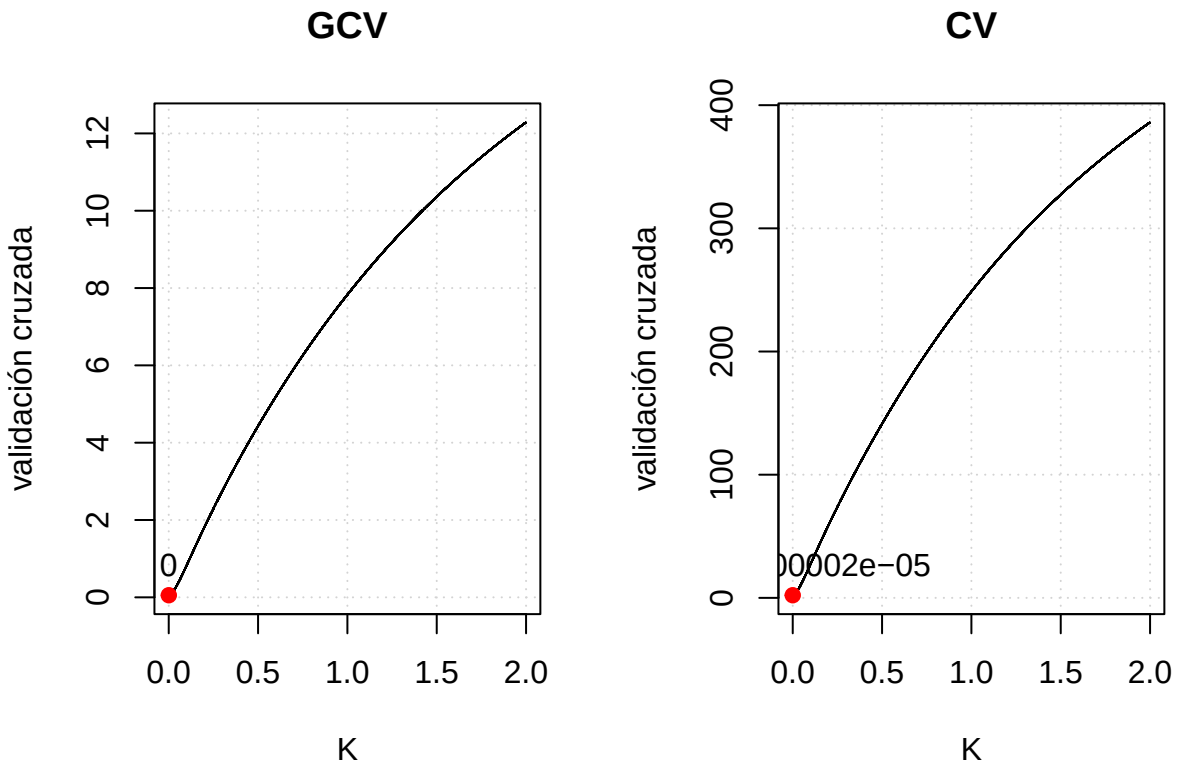
Se observa que si k crece, disminuye la varianza, pero aumenta el sesgo. Por lo que la idea es obtener un k que minimize el ECM.

Luego de tener claro estos conceptos procedemos a realizar la regresión de ridge utilizando la librería lmridge. Para poder realizar esto generamos una secuencia de 1000 valores en los limites de 0 a 2, recordemos que el valor de párametro K es estrictamente positivo, generamos un proceso de validación obteniendo diversos criterios y al final seleccionamos un valor de k=0.1

```
# Regresión ridge
K = seq(from=0,to=2,length.out = 100000)
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                 0.00042
## 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] 0.0004200042 0.0000000000
```

```
######
```

4.1 Modelo ajustado por estimación Ridge

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

4.1.1 Interpretación

procedemos a realizar las interpretaciones del modelo por regresión de ridge donde evidenciamos cambios en los valores p de las pruebas individuales de los β_j, lo cual se debe a la disminución en varianza de las estimaciones. Observamos relaciones negativas entre la densidad y el conjunto de covariables NIR2, NIR 18 y NIR28 SI asumimos que se presentan aumentos con las demas covariables constantes, y se presenta una relación positiva entre la densidad y el NIR6 si mantenemos las demas covariables constantes. Es decir se mantienen las relaciones con el modelo principal, con algunas variaciones en las estimaciones. Ademas se cuenta con un R^2_{adj} de 0.96440, el cual es 3% menor a comparacion del principal.

```
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
```

Observamos claras mejorar en los valores del VIF dónde a pesar de tener un par de valores por encima de 10, notamos claras mejoras, si aumentamos el valor de K, disminuira claramente estos factores de inflación de varianza pero aun así no es óptimo debido a que aumentamos el sesgo. Por último realizamos predicciones puntuales, dónde en R no había opciones para generar los intervalos de confianza por lo cual procedimos a realizar las estimaciones de la varianza de forma analítica para poder obtenerlos.

4.1.2 Predicción

```
#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
```

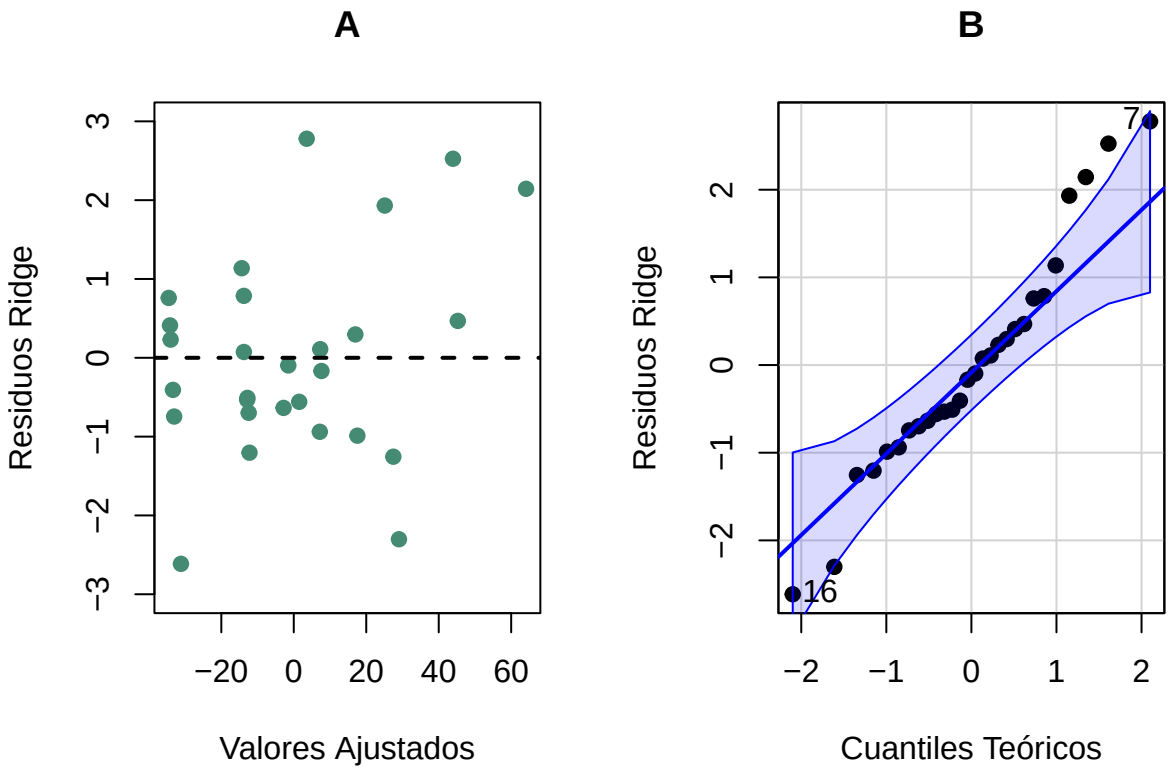
```
#Por lo cuál creamos la estimación de la varianza y matrix de covarianzas
# Obtener los coeficientes estimados y la matriz de diseño
beta_hat <- coef(ridgedensity)
X <- model.matrix(model.lasso1)
```

```r
# 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
+ridgedensity$K*diag(ncol(X))) %*% t(X) %*%X %*% solve(t(X) %*%
X + ridgedensity$K* 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
```

4.2 Validación de supuestos regresión ridge

```r
X <- data.frame(matrix(c(yarn$NIR[,1:30],yarn$density),nrow =28, ncol= 31))
colnames(X) <- c(paste("NIR",1:30,sep=""),"density")
ridgedensity<-lmridge(density~NIR2+NIR6+NIR18+NIR28,
                      data=X,K=0.01,scaling='sc')
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
```

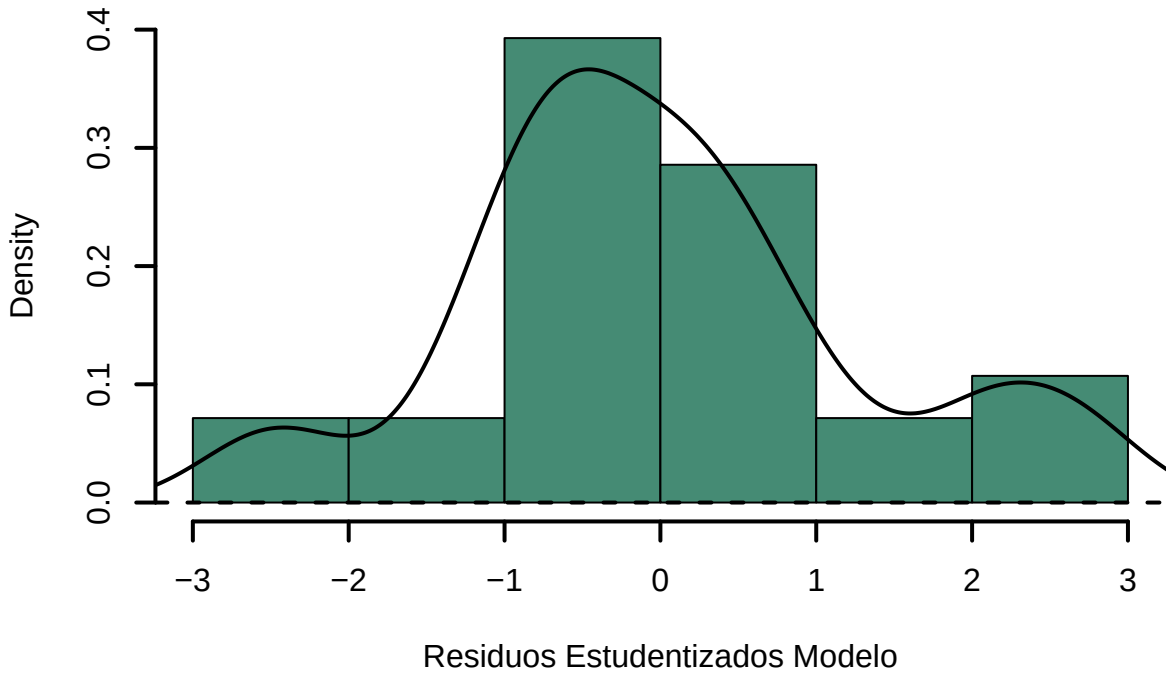
```
print('H0: Normalidad vs H1: No Normalidad')
```

```
## [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)
```



Al evaluar los supuestos se observa que las pruebas formales afirman que los supuestos se cumplen, en cuanto a las pruebas graficas, en la normalidad de los resudios, se observan unos cuantos puntos que estan por fuera de las bandas de confianza, y en el grafico de los residuos se observa que se presenta una varianza constante en los errores.