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)
```

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.

```
##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
  lines(lowess(studres(model)~fitted.values(model)), col = "red1")
```

```
  abline(h=c(-2,0,2),lty=2)
  qqPlot(model,pch=19,ylab='Residuos Estudentizados',xlab='Cuantiles Teóricos',col=carPalette()[1],col.l
  print('Shapiro Test')
  print(shapiro.test(studres(model)))
  print('Breusch Pagan Test')
  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.firs
    plot(studres(model)[-length(fitted.values(model))],studres(model)[-1],pch=19,panel.first = grid(),co
    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 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:

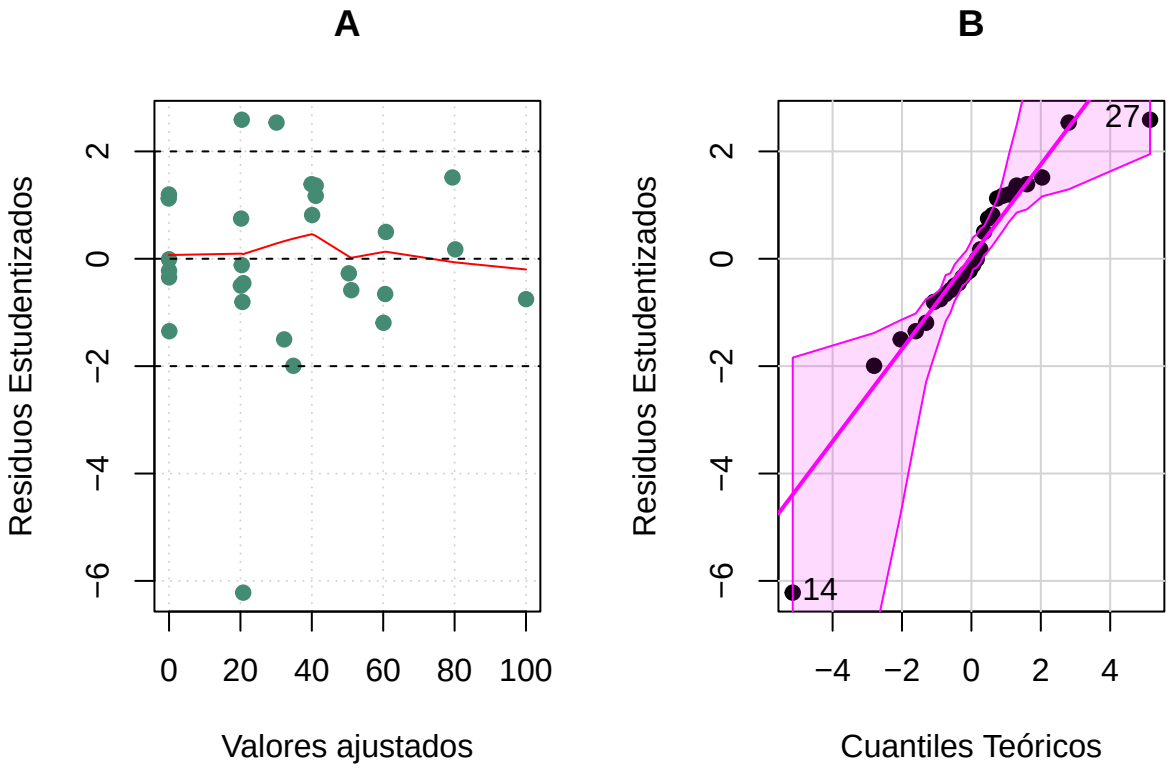
$$S_{lasso}(\beta) = \sum_{i=1}^{n}(y_i - x^T\beta)^2 + \lambda\sum_{j=1}^{p-1}|\beta_j|$$

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 y. Dicho modelo se expresa acontinuación y se evaluan los supuestos:

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



[1] “Shapiro Test”

```
Shapiro-Wilk normality test
```

data: studres(model) W = 0.86458, p-value = 0.001868

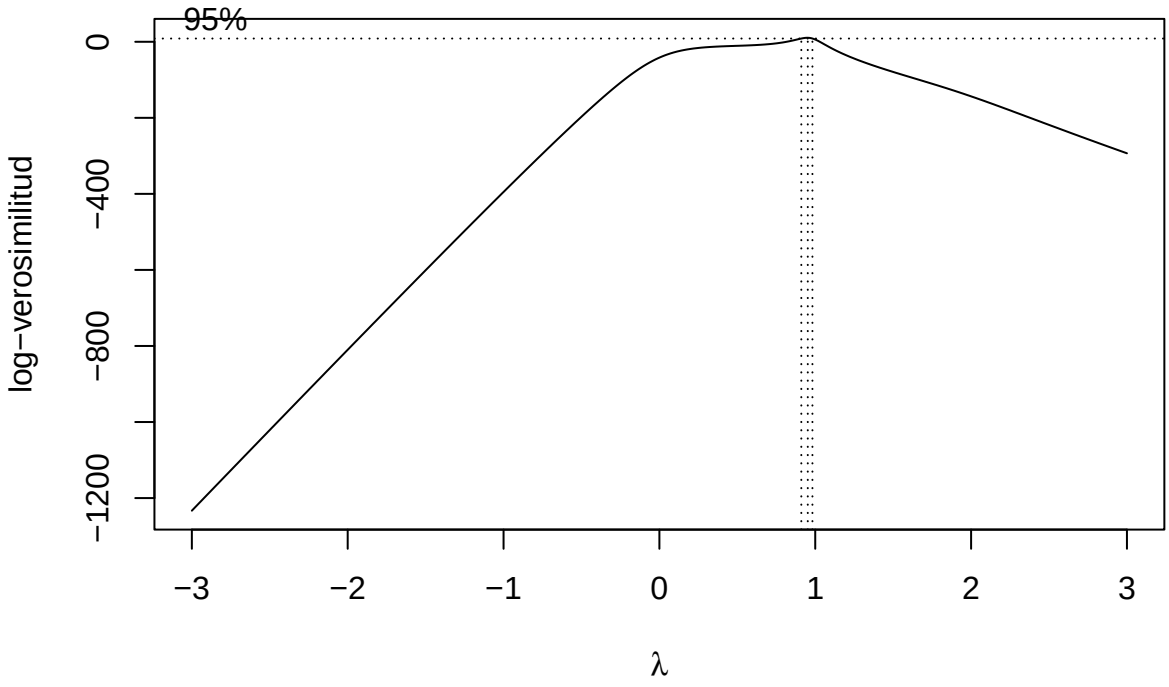
[1] “Breusch Pagan Test”

```
studentized Breusch-Pagan test
```

data: model BP = 27.288, df = 24, p-value = 0.2912

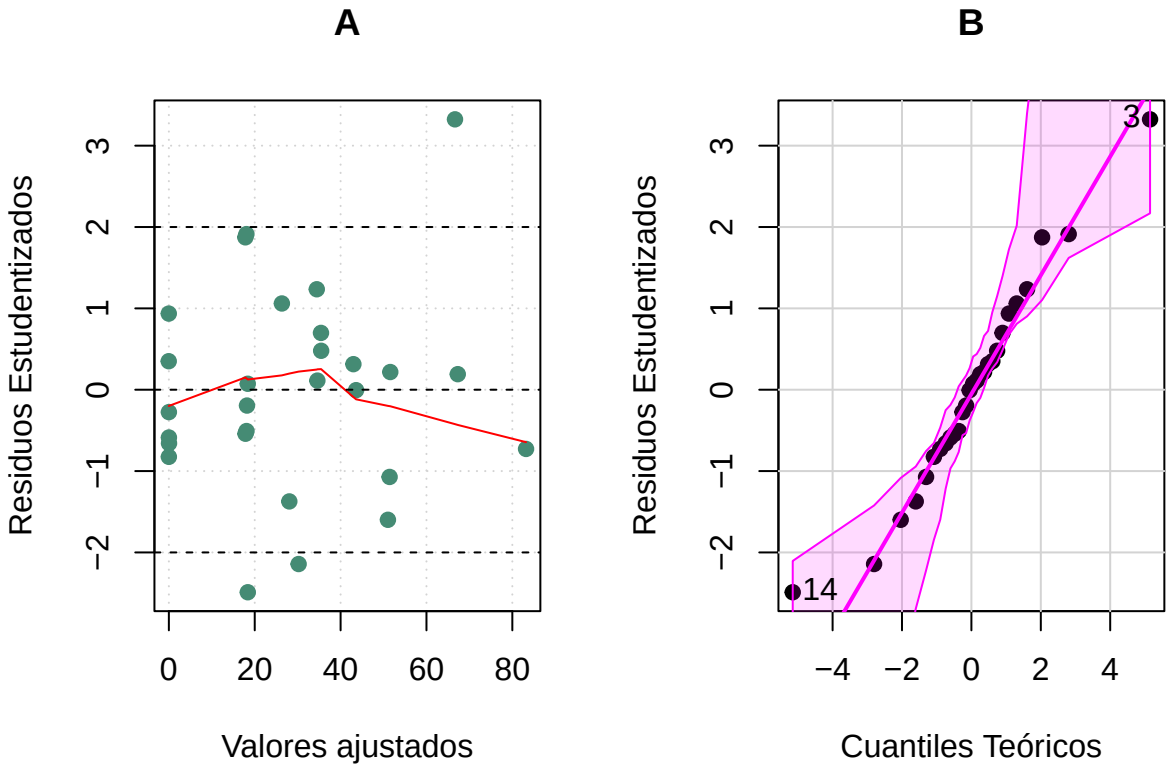
Como no se cumple el supuesto de normalidad se procede a corregir mediante el metodo de BoxCox y se verifica el cumplimiento de los mismos.

```
model <- lm(density+0.0000001 ~ .-NIR1-NIR8-NIR9-NIR10-NIR11-NIR7, data=X)
lambda(model,-3,3)
```



[1] 0.95

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



[1] “Shapiro Test”

```
Shapiro-Wilk normality test
```

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

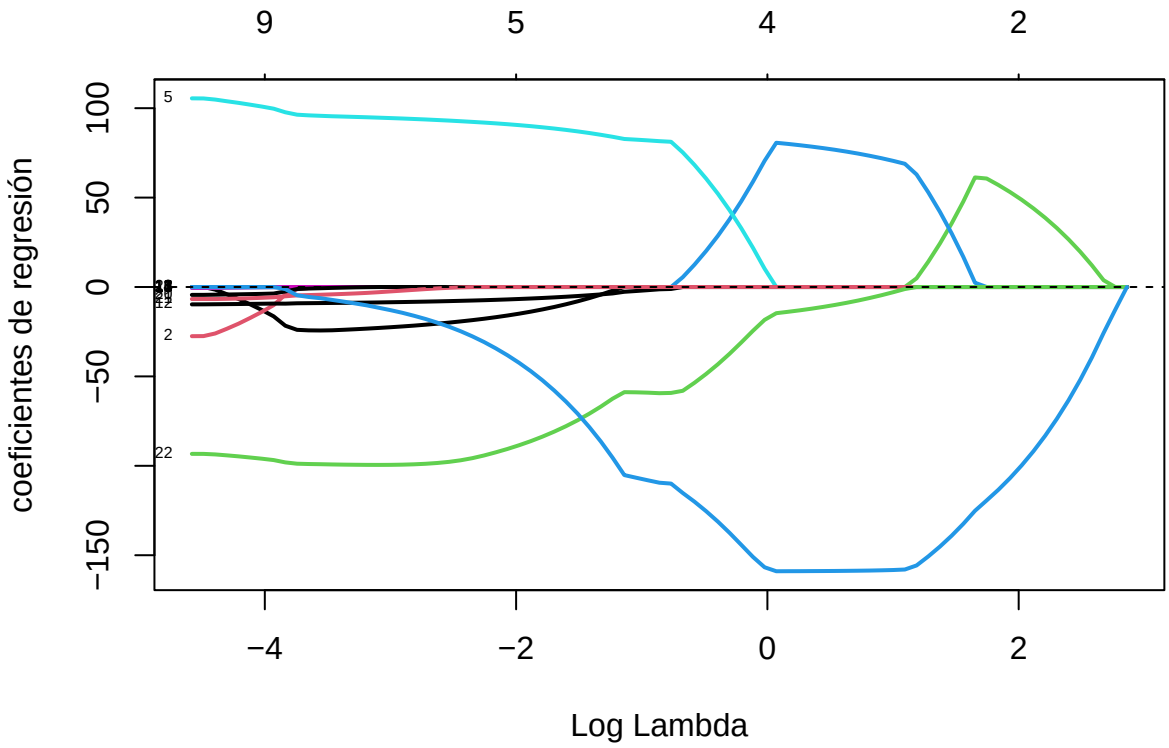
[1] “Breusch Pagan Test”

```
studentized Breusch-Pagan test
```

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

Ya con los requerimentos necesarios para realizar regresión de LASSO, se procede a calcular las estimaciones para distintos valores de λ que se muestran en la siguiente figura:

```
X.<-model.matrix(model.box)[,-1]
lasso.mod <- glmnet(X., X$density, alpha = 1,nlambda = 100)
plot(lasso.mod,xvar='lambda',label=T,lwd=2,ylab='coeficientes de regresión')
abline(h=0,lty=2)
```



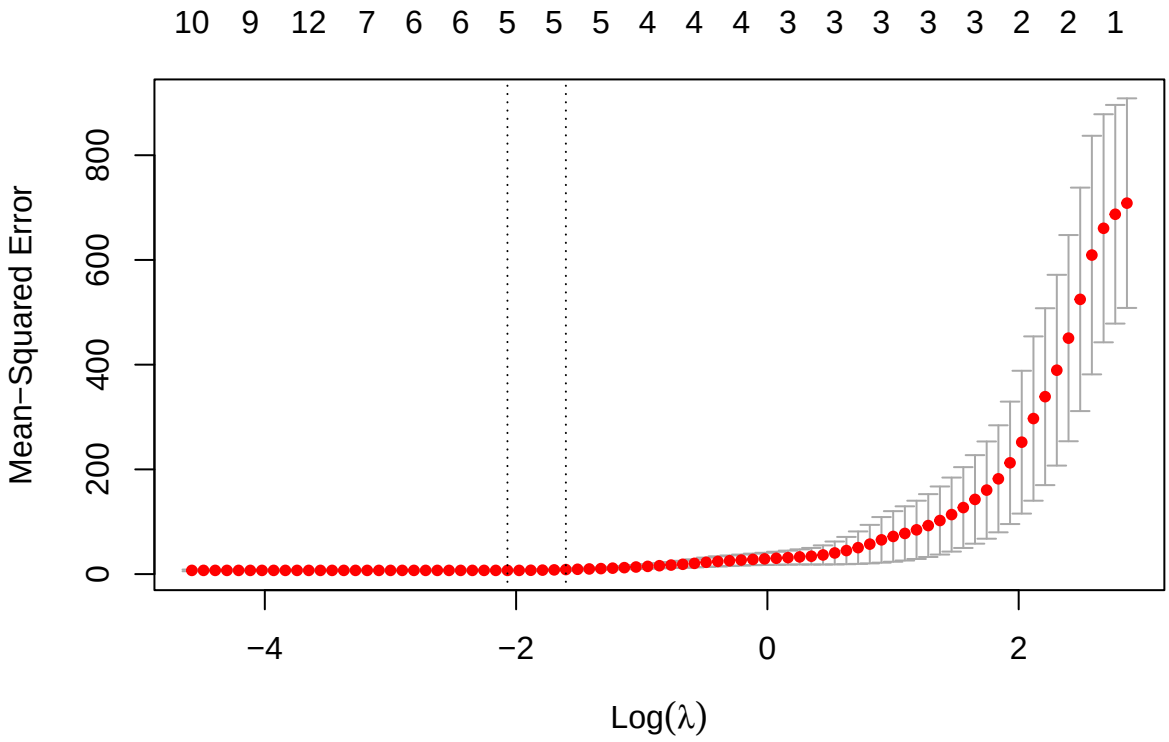
Para identificar el valor de λ optimo se procede a realizar validación cruzada.

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.

```
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 -11.314302 NIR3 .
NIR4 .
NIR5 .
NIR6 88.544252 NIR12 .
NIR13 .
NIR14 .
NIR15 .
NIR16 .
NIR17 .
NIR18 -5.643411 NIR19 .
NIR20 .
NIR21 .
NIR22 .
NIR23 .
NIR24 .
NIR25 .
NIR26 .
NIR27 .
NIR28 -92.202115 NIR29 -40.266344 NIR30 .

La selección de variables por medio del estimador LASSO son: NIR2, NIR6, NIR18, NIR28, NIR29. Consiguiente a eso se procede a realizar una suma extra de cuadrados para evaluar si podemos eliminar NIR29 para evitar 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 $\beta = \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix}$

```
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

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)
```

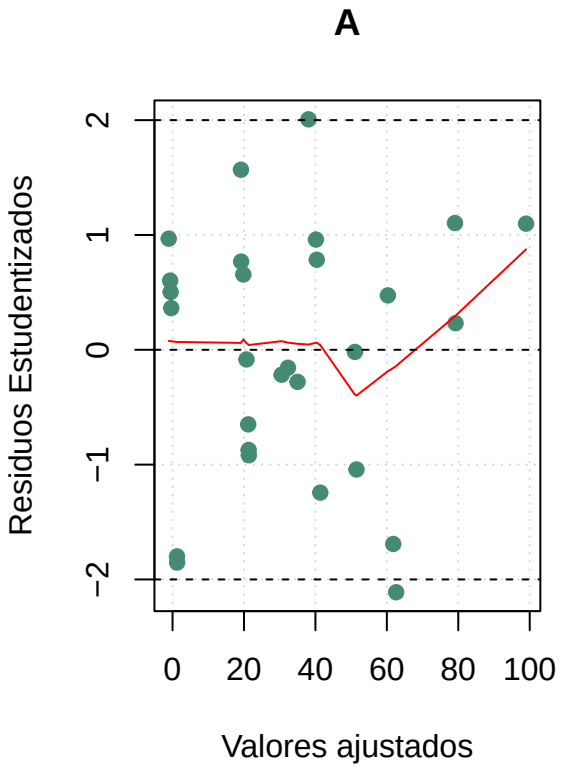
Call: lm(formula = density ~ NIR2 + NIR6 + NIR18 + NIR28, data = X)

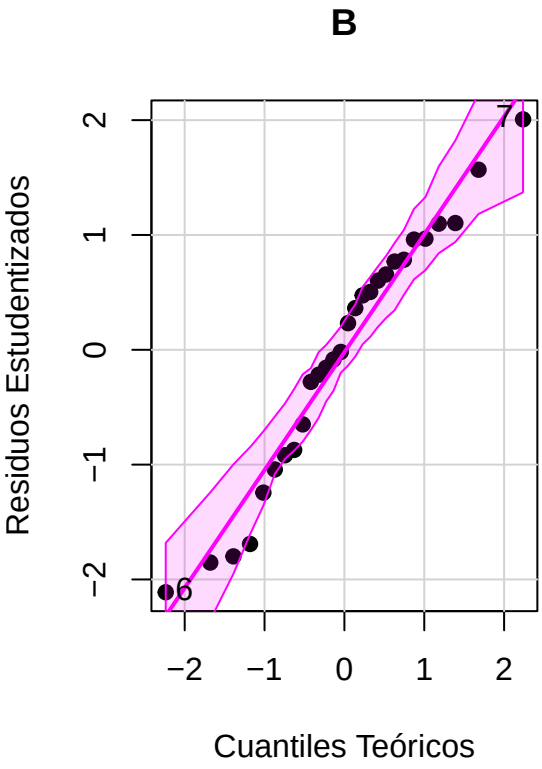
Residuals: Min 1Q Median 3Q Max -2.1312 -0.9776 0.1102 0.8381 2.0416

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

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





[1] “Shapiro Test”

```
Shapiro-Wilk normality test
```

data: studres(model) W = 0.96468, p-value = 0.4471

[1] “Breusch Pagan Test”

```
studentized Breusch-Pagan test
```

data: model BP = 1.6317, df = 4, p-value = 0.8031

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

```
 NIR2      NIR6     NIR18     NIR28
```

2.967327 4.203285 31.085734 26.983026

3.1