Tarea del módulo: Modelos Sparse y Regresión Penalizada

Master in Data Science & Bussines Analytics with R

Daniel Silva Gomes de Araújo.

30/03/2023

Conjunto de datos de trabajo

Los datos a utilizar corresponden al dataset **Boston** del paquete *ISLR2*. El dataset contiene datos sobre la tasa de criminalidad per cápita de 506 barrios de Boston, junto con 12 variables explicativas:

- `crim`: Tasa de criminalidad per cápita en cada barrio.
- `zn`: proporción de suelo residencial dividio en lotes de más de 25,000 pies cuadrados.
- `indus`: proporción de acres comerciales no minoristas por barrio.
- `chas`: Variable categórica, (= 1 si el barrio limita con el río Charles; 0 en caso contrario).
- `nox`: concentración de óxido de nitrógeno (partes por 10 millones).
- `rm`: número medio de habitaciones por vivienda.
- `age`: proporción de viviendas ocupadas por sus propietarios construidas antes de 1940.
- `dis`: media ponderada de las distancias a cinco centros de empleo de Boston
- `rad`: índice de accesibilidad a carreteras radiales
- `tax`: tasa de impuesto a la propiedad de valor total por 10,000$
- `ptratio`: ratio alumno/profesor
- `medv`: Valor mediano de viviendas ocupadas por sus propietarios (en miles de dólares)

El objetivo es predecir ta tasa de criminalidad (variable respuesta) a partir de las otras variables predictoras (usa solo las variables predictoras continuas). Para ello, lo primero que hay que hacer es dividir los datos en una muestra de entrenamiento y otra de testeo, para ello utiliza este código para seleccionar las filas del conjunto de datos que formarán parte de la muestra de entrenamiento y testeo.

```
library(ISLR2)
```

```
## Warning: package 'ISLR2' was built under R version 4.2.3
```

```
library(tidymodels)
```

```
## Warning: package 'tidymodels' was built under R version 4.2.2
```

```
## Warning: package 'dials' was built under R version 4.2.2
```

```
## Warning: package 'dplyr' was built under R version 4.2.2
```

```
## Warning: package 'ggplot2' was built under R version 4.2.2
```

```
## Warning: package 'infer' was built under R version 4.2.2

## Warning: package 'modeldata' was built under R version 4.2.2

## Warning: package 'parsnip' was built under R version 4.2.2

## Warning: package 'recipes' was built under R version 4.2.2

## Warning: package 'rsample' was built under R version 4.2.2

## Warning: package 'tune' was built under R version 4.2.2

## Warning: package 'workflows' was built under R version 4.2.2

## Warning: package 'workflowsets' was built under R version 4.2.2

## Warning: package 'yardstick' was built under R version 4.2.2
```

```
data(Boston, package = "ISLR2")
str(Boston)
```

```
## 'data.frame':    506 obs. of  13 variables:
##  $ crim   : num  0.00632 0.02731 0.02729 0.03237 0.06905 ...
##  $ zn     : num  18 0 0 0 0 0 12.5 12.5 12.5 12.5 ...
##  $ indus  : num  2.31 7.07 7.07 2.18 2.18 2.18 7.87 7.87 7.87 7.87 ...
##  $ chas   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ nox    : num  0.538 0.469 0.469 0.458 0.458 0.458 0.524 0.524 0.524 0.524 ...
##  $ rm     : num  6.58 6.42 7.18 7 7.15 ...
##  $ age    : num  65.2 78.9 61.1 45.8 54.2 58.7 66.6 96.1 100 85.9 ...
##  $ dis    : num  4.09 4.97 4.97 6.06 6.06 ...
##  $ rad    : int  1 2 2 3 3 3 5 5 5 5 ...
##  $ tax    : num  296 242 242 222 222 222 311 311 311 311 ...
##  $ ptratio: num  15.3 17.8 17.8 18.7 18.7 18.7 15.2 15.2 15.2 15.2 ...
##  $ lstat  : num  4.98 9.14 4.03 2.94 5.33 ...
##  $ medv   : num  24 21.6 34.7 33.4 36.2 28.7 22.9 27.1 16.5 18.9 ...
```

```
# formula = crim ~ zn + indus + nox + rm + age + dis + rad + tax + ptratio + lstat + medv

# División estratificada de datos entre 70% training y 30% test
set.seed(123456)
split <- initial_split(Boston, prop = 0.7, strata = crim)
Boston_train <- training(split)
Boston_test <- testing(split)
```

Pregunta 1

Ajusta un modelo de mínimos cuadrados con todas las variables a la muestra de entrenamiento e indica cual es el error de predicción en la muestra de testeo

```r
library(ISLR2)
library(tidymodels)
data(Boston, package = "ISLR2")
str(Boston)
```

```
## 'data.frame':    506 obs. of  13 variables:
##  $ crim   : num  0.00632 0.02731 0.02729 0.03237 0.06905 ...
##  $ zn     : num  18 0 0 0 0 0 12.5 12.5 12.5 12.5 ...
##  $ indus  : num  2.31 7.07 7.07 2.18 2.18 2.18 7.87 7.87 7.87 7.87 ...
##  $ chas   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ nox    : num  0.538 0.469 0.469 0.458 0.458 0.458 0.524 0.524 0.524 0.524 ...
##  $ rm     : num  6.58 6.42 7.18 7 7.15 ...
##  $ age    : num  65.2 78.9 61.1 45.8 54.2 58.7 66.6 96.1 100 85.9 ...
##  $ dis    : num  4.09 4.97 4.97 6.06 6.06 ...
##  $ rad    : int  1 2 2 3 3 3 5 5 5 5 ...
##  $ tax    : num  296 242 242 222 222 222 311 311 311 311 ...
##  $ ptratio: num  15.3 17.8 17.8 18.7 18.7 18.7 15.2 15.2 15.2 15.2 ...
##  $ lstat  : num  4.98 9.14 4.03 2.94 5.33 ...
##  $ medv   : num  24 21.6 34.7 33.4 36.2 28.7 22.9 27.1 16.5 18.9 ...
```

```r
# División estratificada de datos entre 70% training y 30% test
set.seed(123456)
split <- initial_split(Boston, prop = 0.7, strata = crim)
Boston_train <- training(split)
Boston_test <- testing(split)

# Residual Sum of Squares - muestra de entrenamiento
model_train <- lm(crim ~ zn + indus + nox + rm + age + dis + rad + tax + ptratio + lstat + medv, data =
deviance(model_train)
```

```
## [1] 18310.27
```

```r
# Residual Sum of Squares - muestra de testeo
model_test <- lm(crim ~ zn + indus + nox + rm + age + dis + rad + tax + ptratio + lstat + medv, data = 
deviance(model_test)
```

```
## [1] 2053.741
```

Pregunta 2

Utiliza el método del mejor subconjunto para elegir el mejor modelo desde el punto de vista del R^2 ajustado en la muestra de entrenamiento e indica cual es el error de predicción (con el modelo elegido) en la muestra de testeo.

```r
library(leaps)
```

```
## Warning: package 'leaps' was built under R version 4.2.2
```

```
regfit.full <- regsubsets(crim ~ zn + indus + nox + rm + age + dis + rad + tax + ptratio + lstat + medv
summary(regfit.full)
```

```
## Subset selection object
## Call: regsubsets.formula(crim ~ zn + indus + nox + rm + age + dis +
##     rad + tax + ptratio + lstat + medv, Boston)
## 11 Variables  (and intercept)
##         Forced in Forced out
## zn          FALSE      FALSE
## indus       FALSE      FALSE
## nox         FALSE      FALSE
## rm          FALSE      FALSE
## age         FALSE      FALSE
## dis         FALSE      FALSE
## rad         FALSE      FALSE
## tax         FALSE      FALSE
## ptratio     FALSE      FALSE
## lstat       FALSE      FALSE
## medv        FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: exhaustive
##          zn  indus nox rm  age dis rad tax ptratio lstat medv
## 1  ( 1 ) " " " "   " " " " " " " " " "*" " " " "     " "   " "
## 2  ( 1 ) " " " "   " " " " " " " " " "*" " " " "     "*"   " "
## 3  ( 1 ) " " " "   " " " " " " " " " "*" " " " "     "*"   "*"
## 4  ( 1 ) "*" " "   " " " " " " " " "*" "*" " " " "     " "   "*"
## 5  ( 1 ) "*" "*"   " " " " " " " " "*" "*" " " " "     " "   "*"
## 6  ( 1 ) "*" " "   "*" " " " " " " "*" "*" " " "*"     " "   "*"
## 7  ( 1 ) "*" " "   "*" " " " " " " "*" "*" " " "*"     "*"   "*"
## 8  ( 1 ) "*" "*"   "*" " " " " " " "*" "*" " " "*"     "*"   "*"
```

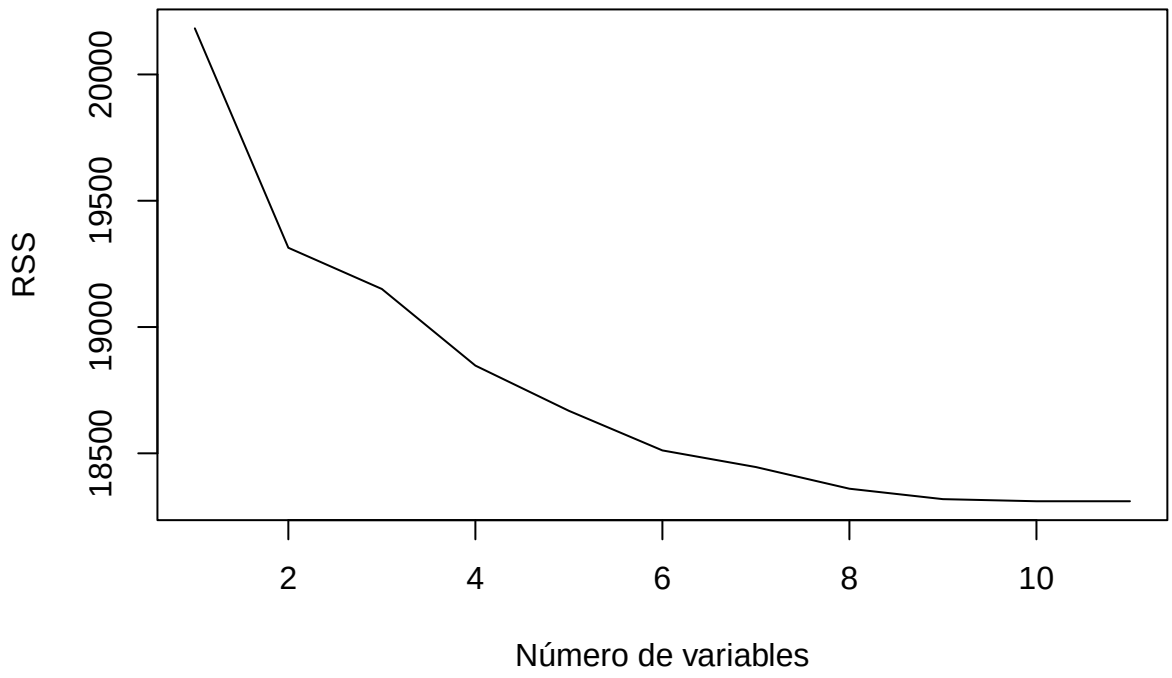
```
regfit.full <- regsubsets(crim ~ zn + indus + nox + rm + age + dis + rad + tax + ptratio + lstat + medv
    nvmax = 11)
reg.summary <- summary(regfit.full)
names(reg.summary)
```

```
## [1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"
```

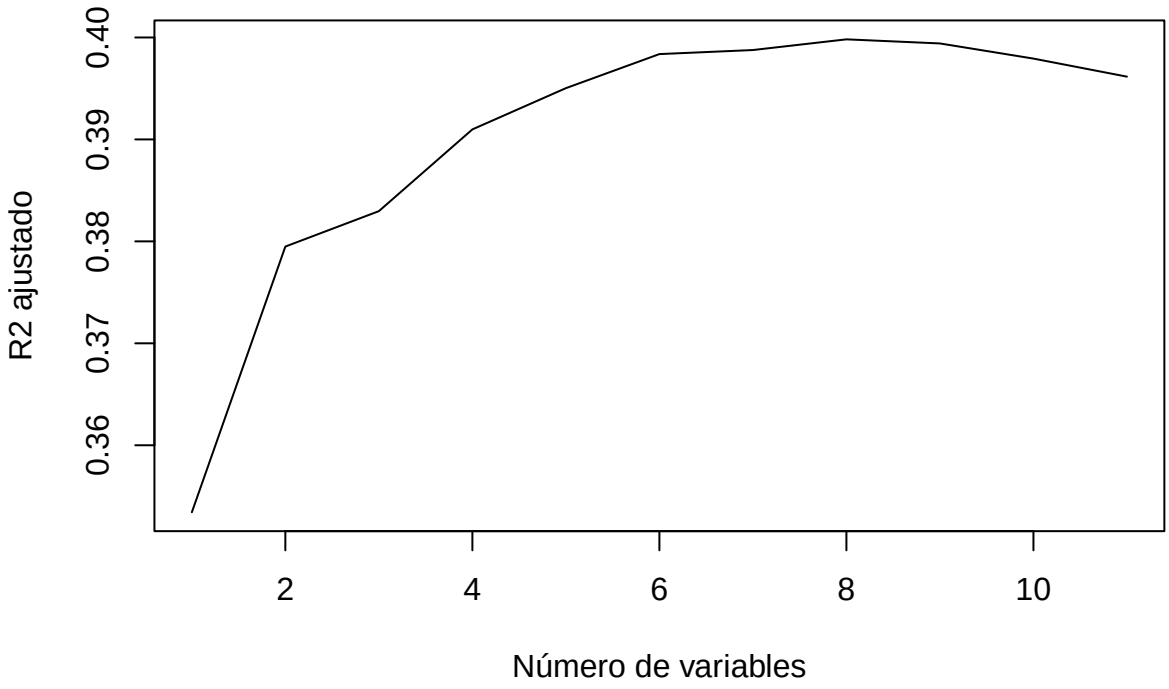
```
reg.summary$adjr2
```

```
##  [1] 0.3534298 0.3794905 0.3829634 0.3909843 0.3950269 0.3983701 0.3987699
##  [8] 0.3998168 0.3994152 0.3979281 0.3961576
```

```
plot(reg.summary$rss, xlab = "Número de variables",
    ylab = "RSS", type = "l")
```



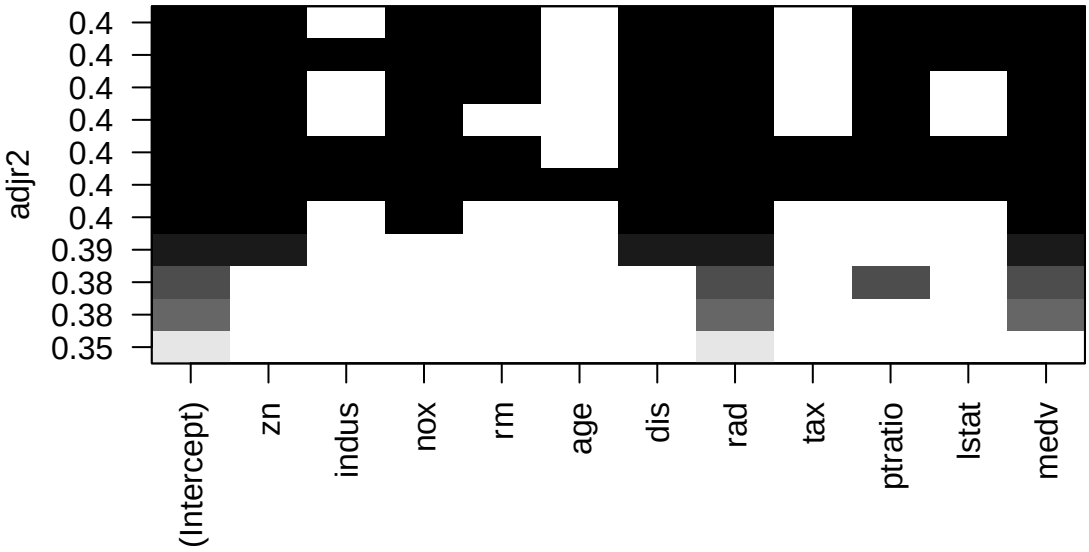

```
plot(reg.summary$adjr2, xlab = "Número de variables",
    ylab = "R2 ajustado", type = "l")
```




```
which.max(reg.summary$adjr2)
```

```
## [1] 8
```

```
# El mejor modelo es con 9 variables.

plot(regfit.full, scale = "adjr2")
```




```
coef(regfit.full, 9)
```

```
##  (Intercept)           zn        indus          nox           rm          dis
##  15.88439221   0.05212496  -0.08985772 -12.76344616   1.00174553  -1.13370688
##          rad      ptratio        lstat         medv
##   0.57572221  -0.38418186   0.12992041  -0.29674037
```

Pregunta 3

Utiliza el método stepwise forward y backward para elegir el mejor modelo desde el punto de vista del R^2 ajustado en la muestra de entrenamiento. Son los dos modelos iguales?. Indica cual es el error de predicción (con el modelo elegido) en la muestra de testeo.

```
regfit.fwd <- regsubsets(crim ~ zn + indus + nox + rm + age + dis + rad + tax + ptratio + lstat + medv,
    nvmax = 9, method = "forward")
summary(regfit.fwd)
```

```
## Subset selection object
## Call: regsubsets.formula(crim ~ zn + indus + nox + rm + age + dis +
##     rad + tax + ptratio + lstat + medv, data = Boston_test, nvmax = 9,
##     method = "forward")
## 11 Variables  (and intercept)
##         Forced in Forced out
## zn          FALSE      FALSE
```

```
## indus       FALSE      FALSE
## nox         FALSE      FALSE
## rm          FALSE      FALSE
## age         FALSE      FALSE
## dis         FALSE      FALSE
## rad         FALSE      FALSE
## tax         FALSE      FALSE
## ptratio     FALSE      FALSE
## lstat       FALSE      FALSE
## medv        FALSE      FALSE
## 1 subsets of each size up to 9
## Selection Algorithm: forward
##          zn  indus nox rm  age dis rad tax ptratio lstat medv
## 1  ( 1 ) " " " "   " " " " " " " " "*" " " " "     " "   " "
## 2  ( 1 ) " " " "   " " " " " " " " "*" " " " "     "*"   " "
## 3  ( 1 ) "*" " "   " " " " " " " " "*" " " " "     "*"   " "
## 4  ( 1 ) "*" " "   " " " " " " " "*" "*" " " " "     "*"   " "
## 5  ( 1 ) "*" " "   " " " " " " " "*" "*" "*" " "     "*"   " "
## 6  ( 1 ) "*" " "   " " " " " " " "*" "*" "*" " "     "*"   "*"
## 7  ( 1 ) "*" " "   "*" " " " " "*" "*" "*" " "     "*"   "*"
## 8  ( 1 ) "*" " "   "*" " " " " "*" "*" "*" "*"     "*"   "*"
## 9  ( 1 ) "*" " "   "*" " " "*" "*" "*" "*" "*"     "*"   "*"
```

```
regfit.bwd <- regsubsets(crim ~ zn + indus + nox + rm + age + dis + rad + tax + ptratio + lstat + medv,
    nvmax = 9, method = "backward")
summary(regfit.bwd)
```

```
## Subset selection object
## Call: regsubsets.formula(crim ~ zn + indus + nox + rm + age + dis +
##     rad + tax + ptratio + lstat + medv, data = Boston_test, nvmax = 9,
##     method = "backward")
## 11 Variables  (and intercept)
##          Forced in Forced out
## zn           FALSE      FALSE
## indus        FALSE      FALSE
## nox          FALSE      FALSE
## rm           FALSE      FALSE
## age          FALSE      FALSE
## dis          FALSE      FALSE
## rad          FALSE      FALSE
## tax          FALSE      FALSE
## ptratio      FALSE      FALSE
## lstat        FALSE      FALSE
## medv         FALSE      FALSE
## 1 subsets of each size up to 9
## Selection Algorithm: backward
##          zn  indus nox rm  age dis rad tax ptratio lstat medv
## 1  ( 1 ) " " " "   " " " " " " " " "*" " " " "     " "   " "
## 2  ( 1 ) " " " "   " " " " " " " " "*" " " " "     "*"   " "
## 3  ( 1 ) "*" " "   " " " " " " " " "*" " " " "     "*"   " "
## 4  ( 1 ) "*" " "   " " " " " " " "*" "*" " " " "     "*"   " "
## 5  ( 1 ) "*" " "   " " " " " " " "*" "*" "*" " "     "*"   " "
## 6  ( 1 ) "*" " "   " " " " " " " "*" "*" "*" " "     "*"   "*"
## 7  ( 1 ) "*" " "   "*" " " " " "*" "*" "*" " "     "*"   "*"
```

```
## 8  ( 1 ) "*" " "   "*" " " " " "*" "*" "*" "*"     "*"   "*"
## 9  ( 1 ) "*" " "   "*" " " "*" "*" "*" "*" "*"     "*"   "*"
```

```
coef(regfit.full, 2)
```

```
## (Intercept)         rad        medv
##   2.6836517   0.5636857  -0.1949976
```

```
coef(regfit.fwd, 2)
```

```
## (Intercept)         rad       lstat
##  -3.7804169   0.4747058   0.2012060
```

```
coef(regfit.bwd, 2)
```

```
## (Intercept)         rad       lstat
##  -3.7804169   0.4747058   0.2012060
```

```
which.max(summary(regfit.fwd)$adjr2)
```

```
## [1] 6
```

```
which.max(summary(regfit.bwd)$adjr2)
```

```
## [1] 6
```

```
# Los dos modelos no son iguales.
# Forward: el mejor modelo es el que tiene 7 variables
# Backward: el mejor modelo es el que tiene 6 variables

set.seed(1)
entreno <- sample(c(TRUE, FALSE), nrow(Boston_test),
    replace = TRUE)
test <- (!entreno)

regfit.best <- regsubsets(crim ~ zn + indus + nox + rm + age + dis + rad + tax + ptratio + lstat + medv
    data = Boston_test[entreno, ], nvmax = 9)

test.mat <- model.matrix(crim ~ zn + indus + nox + rm + age + dis + rad + tax + ptratio + lstat + medv,

val.errors <- rep(NA, 9)
for (i in 1:9) {
 coefi <- coef(regfit.best, id = i)
 pred <- test.mat[, names(coefi)] %*% coefi
 val.errors[i] <- mean((Boston_test$crim[test] - pred)^2)
}

val.errors
```

```
## [1] 11.55365 11.40441 11.17448 10.28247 11.77649 11.35610 11.31413 11.20303
## [9] 11.19878
```

```r
which.min(val.errors)
```

```
## [1] 4
```

```r
coef(regfit.best, 5)
```

```
## (Intercept)          zn          rm         dis         rad        medv
## -1.38890744  0.03359331  1.11370974 -0.60412184  0.46918880 -0.21280018
```

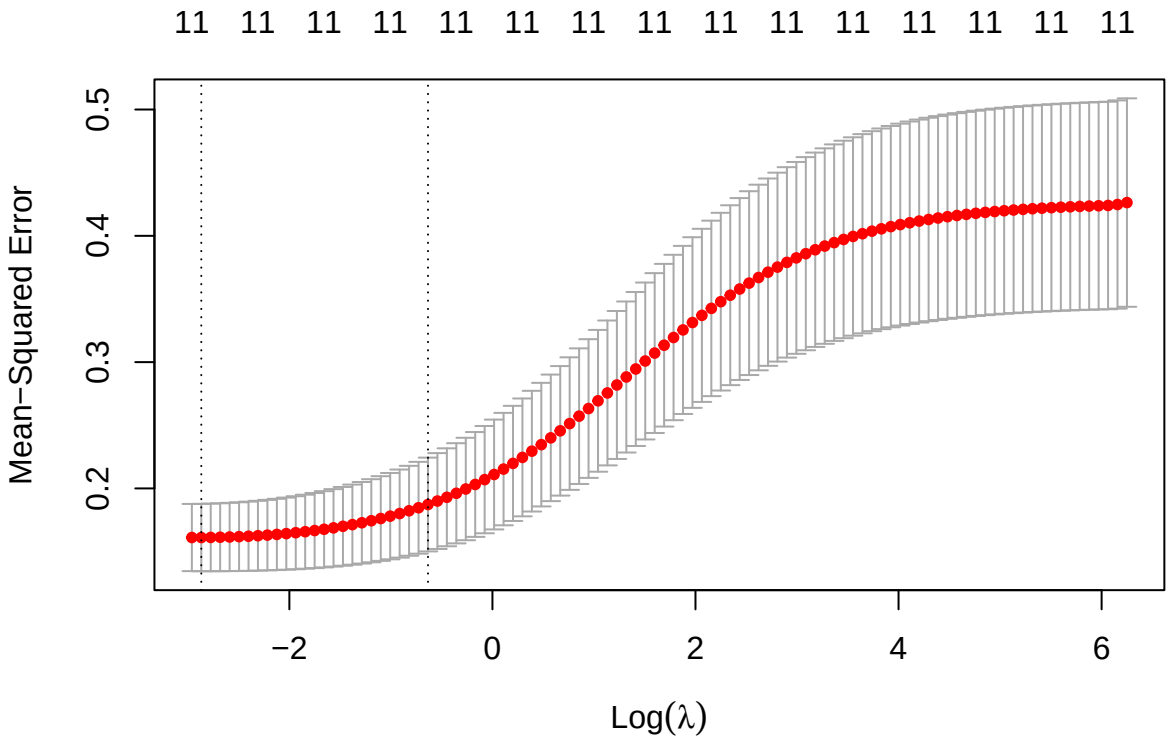
Pregunta 4

Ajusta un modelo de regresión ridge en la muestra de entrenamiento, con λ elegido mediante validación cruzada. Calcula el error de predicción en la muestra de testeo.

```r
library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 4.2.3
```

```r
x <- model.matrix(crim ~ zn + indus + nox + rm + age + dis + rad + tax + ptratio + lstat + medv, Boston
y <- Boston$crim
y.test <- y[test]

## entreno/test
set.seed(1)
entreno <- sample(1:nrow(Boston_train), nrow(Boston_train) / 2)
test <- (-entreno)

## mejor lambda
set.seed(1)
cv.out <- cv.glmnet(x[entreno, ], y[entreno], alpha = 0)
plot(cv.out)
```




```
mejorlam <- cv.out$lambda.min
mejorlam
```

```
## [1] 0.0568365
```

```
## modelo de regresión ridge
x <- model.matrix(crim ~ zn + indus + nox + rm + age + dis + rad + tax + ptratio + lstat + medv, Boston_
y <- Boston_train$crim
ridge.mod <- glmnet(x, y, alpha = 0, lambda = mejorlam)
```

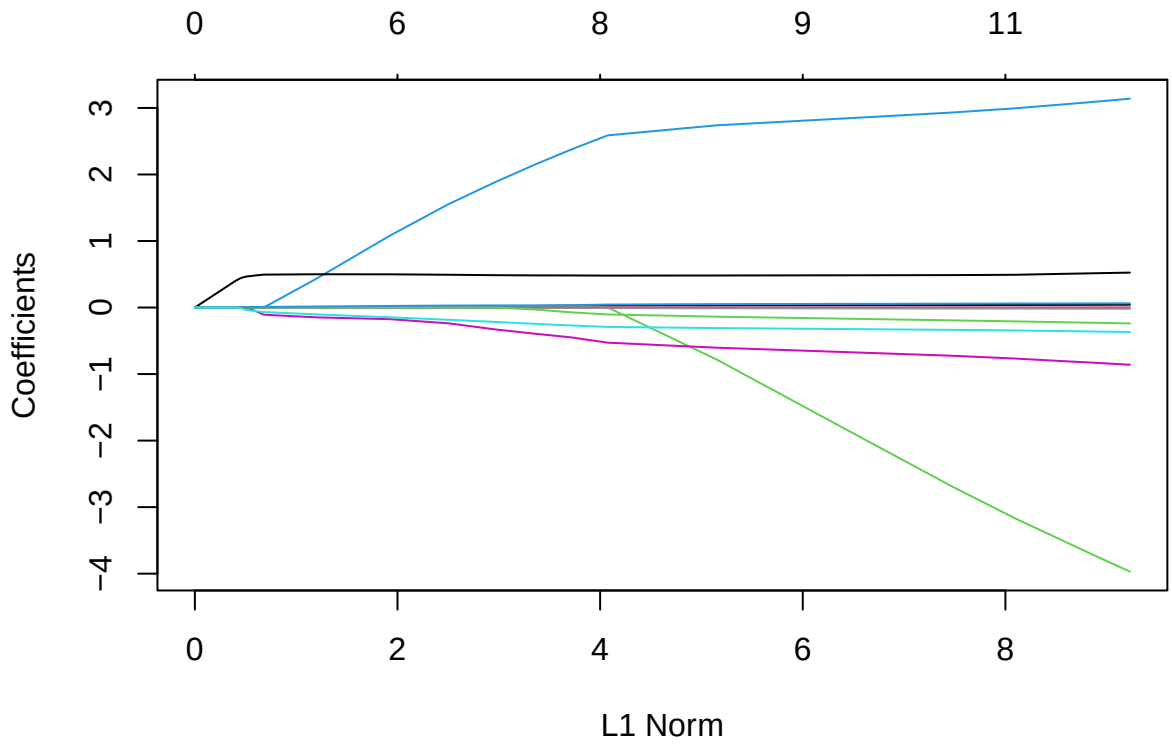
Pregunta 5

Ajusta un modelo de regresión lasso en la muestra de entrenamiento, con λ elegido mediante validación cruzada. Calcula el error de predicción en la muestra de testeo. ¿Cuántos coeficientes se han hecho cero?

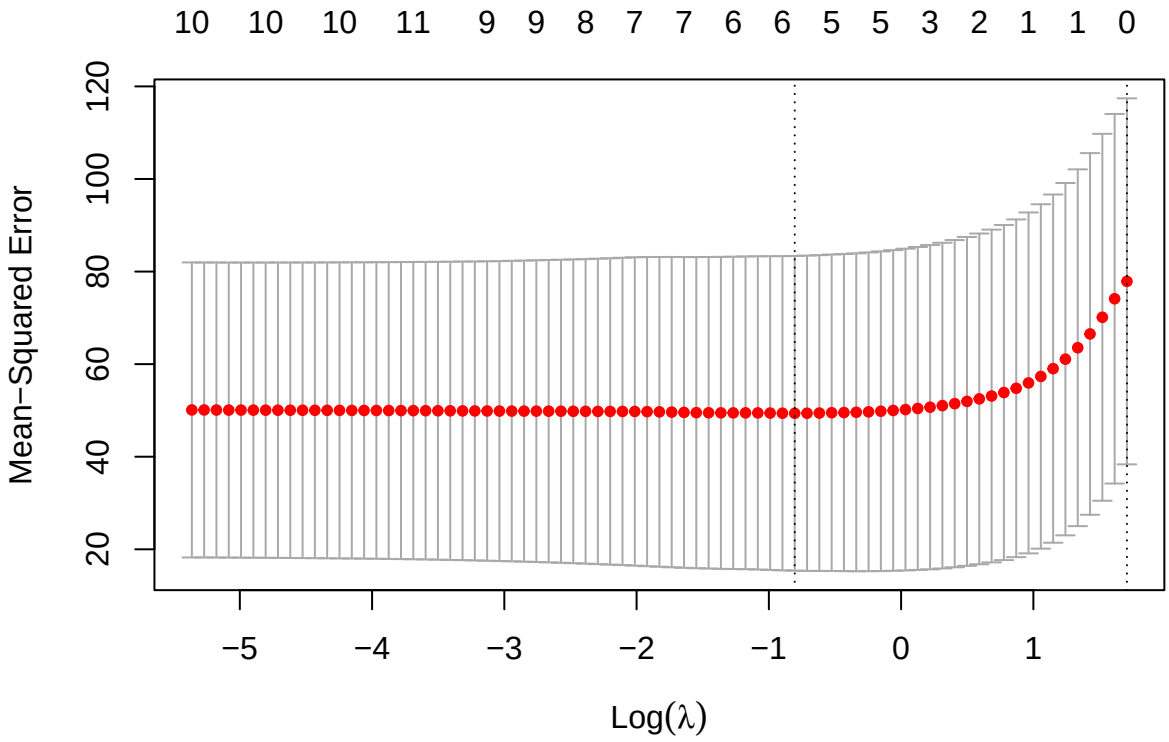
```
library(glmnet)
grid <- 10^seq(10, -2, length = 100)
x <- model.matrix(crim ~ zn + indus + nox + rm + age + dis + rad + tax + ptratio + lstat + medv, Boston_
y <- Boston_train$crim

## entreno/test
set.seed(1)
entreno <- sample(1:nrow(Boston_train), nrow(Boston_train) / 2)
test <- (-entreno)
```

```
# lasso
lasso.mod <- glmnet(x[entreno, ], y[entreno], alpha = 1,
    lambda = grid)
plot(lasso.mod)
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## colapsando para valores de 'x' únicos
```




```
set.seed(1)
cv.out <- cv.glmnet(x[entreno, ], y[entreno], alpha = 1)
plot(cv.out)
```




```
mejorlab <- cv.out$lambda.min
lasso.pred <- predict(lasso.mod, s = mejorlab,
    newx = x[test, ])

out <- glmnet(x, y, alpha = 1)
```

Pregunta 6

Ajusta un modelo elastic-net en la muestra de entrenamiento, con λ y α elegidos mediante validación cruzada. Calcula el error de predicción en la muestra de testeo. ¿Cuántos coeficientes se han echo cero?

Pregunta 7

¿Qué método da lugar a un menor error de predicción?. ¿Cón qué método te quedarías?,¿por qué?

Fin

Este es el final de la tarea.
Sube el archivo .Rmd y el informe (en .pdf) generado a la "tarea" de moodle. Recuerde que el profesor comprobará la reproducibilidad del fichero .Rmd.