

Práctica 6

Práctica 6

```
## Loading required package: Matrix
## Loaded glmnet 4.1-8
# Preparamos la base de datos eliminando los NA
Hitters <- na.omit(Hitters)

# Creamos una matriz para un modelo lineal en la que la variable
# dependiente sea Salary y el resto (.) sean independientes.
x <- model.matrix(Salary ~ ., Hitters)[, -1]
y <- Hitters$Salary

# Creamos un vector de valores de lambda de 10^10 a 10^-2.
grid <- 10^seq(10, -2, length = 100)

# Creamos una ridge regression con los valores de lambda.
ridge.mod <- glmnet(x, y, alpha = 0, lambda = grid)

# La regresión tiene 20 filas y 100 columnas, una
# fila por predictor y una columna por valor de lambda.
dim(coef(ridge.mod))

## [1] 20 100

# Para valores altos de lambda, los coeficientes deberían ser pequeños.
# Vamos a ver los coeficientes para valores altos y pequeños de lambda.
ridge.mod$lambda[50] # Valor lambda = 11498

## [1] 11497.57

coef(ridge.mod)[, 50]

##      (Intercept)      AtBat      Hits      HmRun      Runs
## 407.356050200    0.036957182    0.138180344    0.524629976    0.230701523
##           RBI      Walks      Years      CAtBat      CHits
##  0.239841459    0.289618741    1.107702929    0.003131815    0.011653637
##      CHmRun      CRuns      CRBI      CWalks      LeagueN
##  0.087545670    0.023379882    0.024138320    0.025015421    0.085028114
##  DivisionW      PutOuts      Assists      Errors      NewLeagueN
## -6.215440973    0.016482577    0.002612988   -0.020502690    0.301433531

sqrt(sum(coef(ridge.mod)[-1, 50]^2))

## [1] 6.360612
```

```

ridge.mod$lambda[60] # Valor lambda = 705

## [1] 705.4802

coef(ridge.mod)[, 60]

## (Intercept)      AtBat      Hits      HmRun      Runs      RBI
## 54.32519950  0.11211115  0.65622409  1.17980910  0.93769713  0.84718546
##      Walks      Years      CAtBat      CHits      CHmRun      CRuns
##  1.31987948  2.59640425  0.01083413  0.04674557  0.33777318  0.09355528
##      CRBI      CWalks      LeagueN      DivisionW      PutOuts      Assists
##  0.09780402  0.07189612  13.68370191 -54.65877750  0.11852289  0.01606037
##      Errors      NewLeagueN
## -0.70358655  8.61181213

sqrt(sum(coef(ridge.mod)[-1, 60]^2))

## [1] 57.11001

# Podemos usar la función predict() para obtener el valor de
# los coeficientes para un lambda concreto (por ejemplo 50).
predict(
  ridge.mod,
  s = 50,
  type = "coefficients"
)[1:20, ]

## (Intercept)      AtBat      Hits      HmRun      Runs
## 4.876610e+01 -3.580999e-01  1.969359e+00 -1.278248e+00  1.145892e+00
##      RBI      Walks      Years      CAtBat      CHits
## 8.038292e-01  2.716186e+00 -6.218319e+00  5.447837e-03  1.064895e-01
##      CHmRun      CRuns      CRBI      CWalks      LeagueN
## 6.244860e-01  2.214985e-01  2.186914e-01 -1.500245e-01  4.592589e+01
##      DivisionW      PutOuts      Assists      Errors      NewLeagueN
## -1.182011e+02  2.502322e-01  1.215665e-01 -3.278600e+00 -9.496680e+00

# Escogemos los grupos de entrenamiento y test
set.seed(1)

# Generamos una lista de len(x)/2 elementos con índices aleatorios de x.
train <- sample(seq_len(nrow(x)), nrow(x) / 2)

# Creamos los índices de test con los índices que no están en entrenamiento
# y guardamos los datos correspondientes de y en y.test.
test <- (-train)
y.test <- y[test]

ridge.mon <- glmnet(
  x[train, ], y[train],
  alpha = 0,
  lambda = grid,
  thresh = 1e-12
)

ridge.pred <- predict(ridge.mod, s = 4, newx = x[test, ])
mean((ridge.pred - y.test)^2)

```

[1] 97611.7