Practica2

Beltran

16/10/2019

LIBRARIES

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------------------------------------- tidyvers

## v ggplot2 3.2.1     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

## -- Conflicts ------------------------------------------------------------------------ tidyverse_confl
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(leaps)
library(ggplot2)
library(ISLR)
library(glmnet)
```

```
## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack

## Loading required package: foreach

##
## Attaching package: 'foreach'

## The following objects are masked from 'package:purrr':
##
##     accumulate, when

## Loaded glmnet 2.0-18
```

```r
library(caret)
```

```
## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift
```

```r
library(rsample)
library(dplyr)
library(gvlma)
library(car)
```

```
## Loading required package: carData

##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode

## The following object is masked from 'package:purrr':
##
##     some
```

```r
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2

##
## Attaching package: 'GGally'

## The following object is masked from 'package:dplyr':
##
##     nasa
```

```r
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```r
library(PerformanceAnalytics)
```

```
## Loading required package: xts

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Registered S3 method overwritten by 'xts':
##   method     from
##   as.zoo.xts zoo

##
## Attaching package: 'xts'

## The following objects are masked from 'package:dplyr':
##
##     first, last

##
## Attaching package: 'PerformanceAnalytics'

## The following object is masked from 'package:graphics':
##
##     legend
```

ANALISIS EXPLORATORIO

```r
mData <- read.csv("nba.csv")
```

Mi objetivo es elaborar un modelo predictivo que permita determinar con precisión el salario de un jugador de la NBA. Para ello dispongo de un fichero .csv llamado "nba".

```r
names(mData)
```

```
##  [1] "Player"          "Salary"          "NBA_Country"
##  [4] "NBA_DraftNumber" "Age"             "Tm"
##  [7] "G"               "MP"              "PER"
## [10] "TS."             "X3PAr"           "FTr"
## [13] "ORB."            "DRB."            "TRB."
## [16] "AST."            "STL."            "BLK."
## [19] "TOV."            "USG."            "OWS"
## [22] "DWS"             "WS"              "WS.48"
## [25] "OBPM"            "DBPM"            "BPM"
## [28] "VORP"
```

```r
str(mData)
```

```
## 'data.frame':    485 obs. of  28 variables:
##  $ Player         : Factor w/ 483 levels "Aaron Brooks",..: 483 482 481 480 479 478 477 476 475 474
##  $ Salary         : int  815615 3477600 12307692 3202217 3057240 1312611 74159 46080 12016854 1435750
##  $ NBA_Country    : Factor w/ 44 levels "Argentina","Australia",..: 10 19 44 44 44 44 44 9 44 37 ...
##  $ NBA_DraftNumber: int  43 42 19 13 10 62 62 62 23 35 ...
##  $ Age            : int  22 33 36 22 20 24 30 23 30 23 ...
##  $ Tm             : Factor w/ 31 levels "ATL","BOS","BRK",..: 11 10 26 4 25 7 2 15 8 29 ...
##  $ G              : int  16 66 59 24 62 79 2 5 70 45 ...
##  $ MP             : int  87 937 1508 656 979 2238 7 118 2200 430 ...
##  $ PER            : num  0.6 16.8 17.3 14.6 8.2 11.5 -4.9 0.9 11.1 20.6 ...
##  $ TS.            : num  0.303 0.608 0.529 0.499 0.487 0.543 0 0.315 0.543 0.592 ...
##  $ X3PAr          : num  0.593 0.004 0.193 0.346 0.387 0.489 0.667 0.333 0.39 0.075 ...
##  $ FTr            : num  0.37 0.337 0.14 0.301 0.146 0.141 0 0.214 0.186 0.555 ...
##  $ ORB.           : num  6.5 11 7 1.4 4.9 1.3 15.9 0 5 13.6 ...
##  $ DRB.           : num  16.8 25 23.8 14.4 18.3 11.3 15.4 5 14 25.2 ...
##  $ TRB.           : num  11.7 18.5 15 7.7 11.7 6.1 15.7 2.5 9.5 19.3 ...
##  $ AST.           : num  1.5 15.4 14.9 18.6 7.3 13.3 0 23.2 9.7 11 ...
##  $ STL.           : num  1.1 1.9 1.4 1.8 0.8 1.4 7.2 2.6 0.9 1.8 ...
##  $ BLK.           : num  6.8 1.3 0.6 0.5 2.5 0.3 0 2.4 1.4 2.8 ...
##  $ TOV.           : num  18.2 19.3 12.5 9.7 15.6 9.1 0 19.3 12 15.4 ...
##  $ USG.           : num  19.5 17.2 27.6 29.5 15.5 17 19.2 21.7 14.6 21.7 ...
##  $ OWS            : num  -0.4 1.7 0.3 -0.1 -0.4 1.6 -0.1 -0.5 2 0.8 ...
##  $ DWS            : num  0.1 1.4 1.1 0.5 1.2 1.6 0 0.1 1 0.6 ...
##  $ WS             : num  -0.2 3.1 1.4 0.4 0.8 3.1 0 -0.4 3.1 1.4 ...
##  $ WS.48          : num  -0.121 0.16 0.046 0.027 0.038 0.067 -0.251 -0.169 0.067 0.156 ...
##  $ OBPM           : num  -10.6 -0.6 -0.6 -0.7 -3.7 -0.4 -12.6 -8.7 -0.4 -0.1 ...
##  $ DBPM           : num  0.5 1.3 -1.3 -2 0.9 -0.5 -0.7 -1.9 -0.6 0.6 ...
##  $ BPM            : num  -10.1 0.8 -1.9 -2.6 -2.9 -0.9 -13.3 -10.6 -1 0.5 ...
##  $ VORP           : num  -0.2 0.7 0 -0.1 -0.2 0.6 0 -0.3 0.5 0.3 ...
```

El fichero cuenta con 485 observaciones correspondientes a 28 variables. Compruebo a que porcentaje de las observaciones les falta al menos una variable.

```r
1 - (sum(complete.cases(mData)) / nrow(mData))
```

```
## [1] 0.004123711
```

El 0,41% de las observaciones presentan NA´s, por ello procedo a excluirlas del data set.

```r
mData <- na.omit(mData)
```

Elimino la variables categórica nommbre del jugador y país.

```r
mData <- select(mData, -Player, -NBA_Country)
```

Analizo la dimensión del data set.

```r
dim(mData)
```

```
## [1] 483  26
```

Una vez excluidos los valores NA, el data set contiene 483 observaciones correspondientes a 25 variables. A continuación analizo la proporción observaciones / predictores.

```r
483 / 25
```

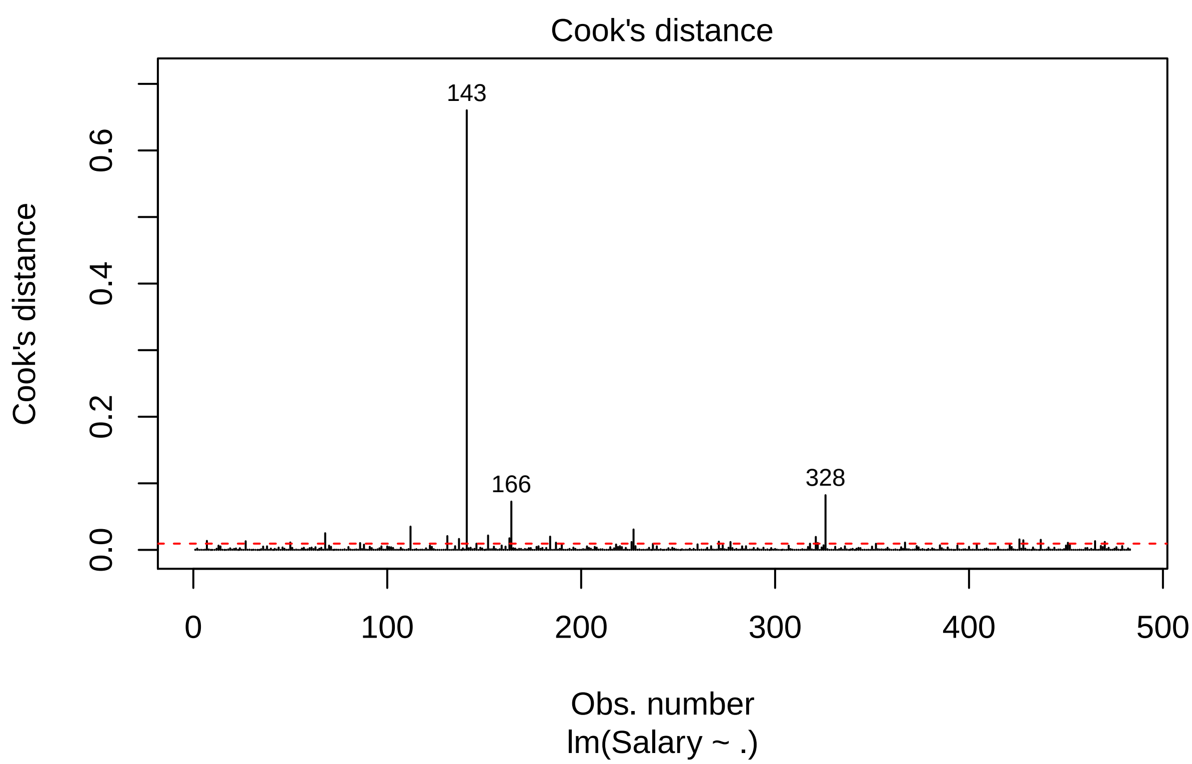
```
## [1] 19.32
```

El número de observaciones es superior en algo más de 19 veces al número de predictores. Por tanto, la regresión por mínimos cuadrados podría resultar adecuado. A continuación estudio los valores atípicos e influyentes y procedo a su eliminación de la muestra. ### Valores atípicos y/o influyentes.

```r
regres01 <- lm(Salary~., data = mData)
outlierTest(regres01)
```

```
##     rstudent unadjusted p-value Bonferroni p
## 328 4.464562         1.0282e-05     0.004966
## 114 4.205394         3.1773e-05     0.015346
```

```r
cutoff <- 4/(nrow(mData) - length(regres01$coefficients) - 2)
plot(regres01, which = 4, cook.levels = cutoff)
abline(h = cutoff, lty = 2, col = "red")
```



Los valores extraidos a través del outliertest y observados en la distancia de Cook corresponden con las observaciones 114, 143, 166 y 328. Procedo a la eliminación de la muestra de dichos observaciones,

```
mData <- mData[c(-114, -143, -166, -328)]
```

Establezco de nuevo la regresión sin los valores influyentes.

```
regres01 <- lm(Salary~., data = mData)
```

Matriz de correlaciones

Estudio la relación entre las variables del modelo.Para poder obtener la matriz de correlaciones he de eliminar la única variable categórica que queda en el modelo, el equipo.

```
mData <- select(mData, -Tm)
round(cor(mData),3)
```

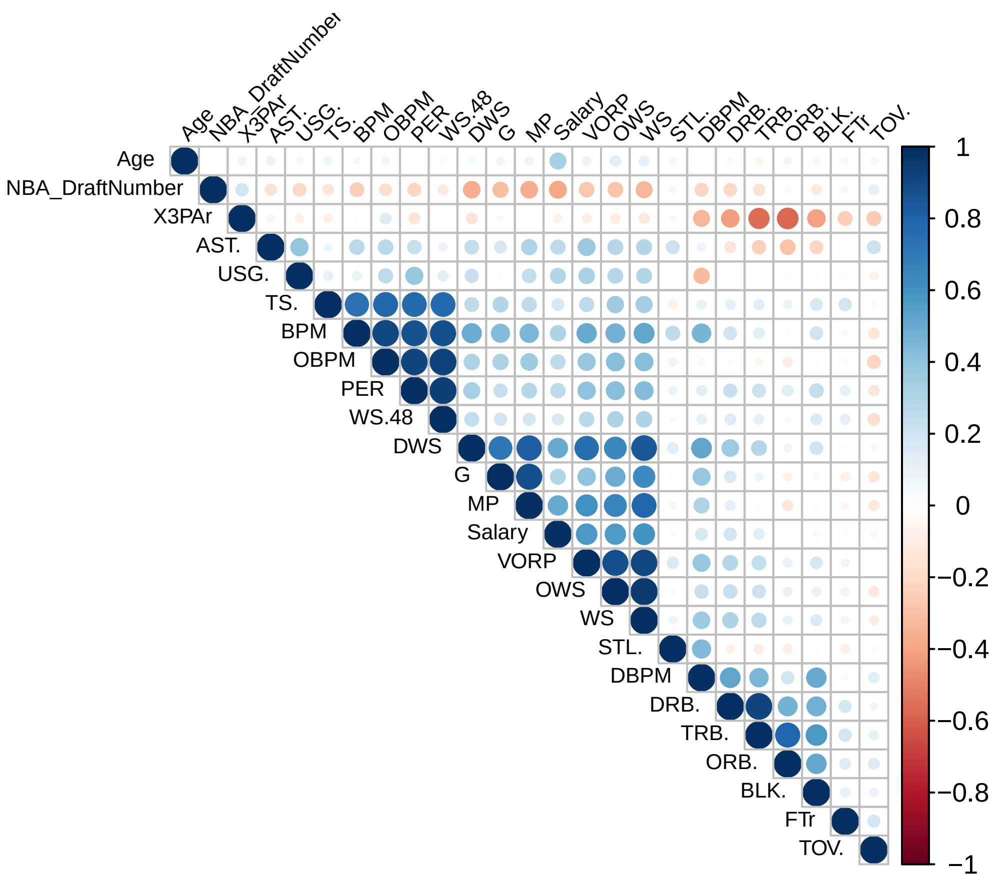
```
##                 Salary NBA_DraftNumber    Age      G     MP    PER    TS.
## Salary           1.000          -0.381  0.336  0.294  0.505  0.266  0.175
## NBA_DraftNumber -0.381           1.000  0.007 -0.303 -0.368 -0.216 -0.136
## Age              0.336           0.007  1.000  0.074  0.075  0.010  0.077
## G                0.294          -0.303  0.074  1.000  0.873  0.227  0.294
## MP               0.505          -0.368  0.075  0.873  1.000  0.289  0.264
## PER              0.266          -0.216  0.010  0.227  0.289  1.000  0.778
## TS.              0.175          -0.136  0.077  0.294  0.264  0.778  1.000
## X3PAr           -0.074           0.195  0.075 -0.033  0.002 -0.146 -0.086
## FTr              0.023           0.050 -0.043 -0.086 -0.056  0.114  0.190
## ORB.             0.001          -0.038 -0.064 -0.082 -0.131  0.136  0.086
## DRB.             0.191          -0.203 -0.034  0.160  0.120  0.224  0.117
## TRB.             0.135          -0.159 -0.052  0.073  0.022  0.219  0.124
## AST.             0.263          -0.155  0.085  0.178  0.306  0.234  0.063
## STL.             0.031          -0.060 -0.057  0.019  0.060  0.085 -0.086
## BLK.             0.042          -0.118 -0.058  0.049 -0.023  0.243  0.177
## TOV.            -0.043           0.116  0.059 -0.134 -0.127 -0.139  0.040
## USG.             0.295          -0.207 -0.057  0.022  0.242  0.390  0.105
## OWS              0.562          -0.280  0.134  0.499  0.658  0.428  0.351
## DWS              0.504          -0.360  0.046  0.711  0.821  0.339  0.251
## WS               0.591          -0.338  0.112  0.626  0.781  0.433  0.345
## WS.48            0.161          -0.120  0.030  0.191  0.189  0.934  0.780
## OBPM             0.264          -0.176  0.066  0.311  0.367  0.915  0.788
## DBPM             0.178          -0.219 -0.011  0.384  0.301  0.125  0.099
## BPM              0.309          -0.248  0.053  0.438  0.453  0.864  0.741
## VORP             0.573          -0.277  0.084  0.402  0.609  0.401  0.264
##                  X3PAr    FTr   ORB.   DRB.   TRB.   AST.   STL.   BLK.
## Salary          -0.074  0.023  0.001  0.191  0.135  0.263  0.031  0.042
## NBA_DraftNumber  0.195  0.050 -0.038 -0.203 -0.159 -0.155 -0.060 -0.118
## Age              0.075 -0.043 -0.064 -0.034 -0.052  0.085 -0.057 -0.058
## G               -0.033 -0.086 -0.082  0.160  0.073  0.178  0.019  0.049
## MP               0.002 -0.056 -0.131  0.120  0.022  0.306  0.060 -0.023
## PER             -0.146  0.114  0.136  0.224  0.219  0.234  0.085  0.243
## TS.             -0.086  0.190  0.086  0.117  0.124  0.063 -0.086  0.177
## X3PAr            1.000 -0.247 -0.578 -0.411 -0.552 -0.057 -0.049 -0.408
```

```
## FTr            -0.247  1.000  0.142  0.182  0.193 -0.004 -0.088  0.104
## ORB.           -0.578  0.142  1.000  0.479  0.797 -0.283 -0.095  0.510
## DRB.           -0.411  0.182  0.479  1.000  0.912 -0.147 -0.078  0.482
## TRB.           -0.552  0.193  0.797  0.912  1.000 -0.232 -0.098  0.572
## AST.           -0.057 -0.004 -0.283 -0.147 -0.232  1.000  0.213 -0.210
## STL.           -0.049 -0.088 -0.095 -0.078 -0.098  0.213  1.000 -0.022
## BLK.           -0.408  0.104  0.510  0.482  0.572 -0.210 -0.022  1.000
## TOV.           -0.250  0.184  0.141  0.061  0.108  0.213 -0.028  0.083
## USG.           -0.082  0.011  0.014 -0.004  0.005  0.397  0.003 -0.012
## OWS            -0.107  0.080  0.097  0.234  0.211  0.288  0.029  0.099
## DWS            -0.140  0.013  0.068  0.367  0.289  0.247  0.139  0.207
## WS             -0.130  0.063  0.096  0.307  0.261  0.298  0.075  0.151
## WS.48          -0.008  0.114  0.033  0.141  0.114  0.085  0.036  0.151
## OBPM            0.149  0.035 -0.108 -0.016 -0.059  0.274  0.076 -0.011
## DBPM           -0.339  0.036  0.193  0.525  0.453  0.077  0.447  0.508
## BPM            -0.011  0.046 -0.015  0.207  0.138  0.275  0.256  0.204
## VORP           -0.092  0.077  0.094  0.286  0.246  0.379  0.162  0.172
##                  TOV.   USG.    OWS    DWS     WS  WS.48   OBPM   DBPM
## Salary         -0.043  0.295  0.562  0.504  0.591  0.161  0.264  0.178
## NBA_DraftNumber  0.116 -0.207 -0.280 -0.360 -0.338 -0.120 -0.176 -0.219
## Age             0.059 -0.057  0.134  0.046  0.112  0.030  0.066 -0.011
## G              -0.134  0.022  0.499  0.711  0.626  0.191  0.311  0.384
## MP             -0.127  0.242  0.658  0.821  0.781  0.189  0.367  0.301
## PER            -0.139  0.390  0.428  0.339  0.433  0.934  0.915  0.125
## TS.             0.040  0.105  0.351  0.251  0.345  0.780  0.788  0.099
## X3PAr          -0.250 -0.082 -0.107 -0.140 -0.130 -0.008  0.149 -0.339
## FTr             0.184  0.011  0.080  0.013  0.063  0.114  0.035  0.036
## ORB.            0.141  0.014  0.097  0.068  0.096  0.033 -0.108  0.193
## DRB.            0.061 -0.004  0.234  0.367  0.307  0.141 -0.016  0.525
## TRB.            0.108  0.005  0.211  0.289  0.261  0.114 -0.059  0.453
## AST.            0.213  0.397  0.288  0.247  0.298  0.085  0.274  0.077
## STL.           -0.028  0.003  0.029  0.139  0.075  0.036  0.076  0.447
## BLK.            0.083 -0.012  0.099  0.207  0.151  0.151 -0.011  0.508
## TOV.            1.000 -0.081 -0.126 -0.044 -0.106 -0.173 -0.218  0.131
## USG.           -0.081  1.000  0.287  0.227  0.291  0.122  0.262 -0.316
## OWS            -0.126  0.287  1.000  0.648  0.956  0.311  0.427  0.233
## DWS            -0.044  0.227  0.648  1.000  0.843  0.247  0.310  0.522
## WS             -0.106  0.291  0.956  0.843  1.000  0.315  0.421  0.366
## WS.48          -0.173  0.122  0.311  0.247  0.315  1.000  0.929  0.116
## OBPM           -0.218  0.262  0.427  0.310  0.421  0.929  1.000  0.047
## DBPM            0.131 -0.316  0.233  0.522  0.366  0.116  0.047  1.000
## BPM            -0.138  0.098  0.477  0.495  0.528  0.873  0.907  0.463
## VORP           -0.015  0.321  0.870  0.755  0.906  0.275  0.384  0.383
##                   BPM   VORP
## Salary          0.309  0.573
## NBA_DraftNumber -0.248 -0.277
## Age             0.053  0.084
## G               0.438  0.402
## MP              0.453  0.609
## PER             0.864  0.401
## TS.             0.741  0.264
## X3PAr          -0.011 -0.092
## FTr             0.046  0.077
## ORB.           -0.015  0.094
```

```
## DRB.             0.207  0.286
## TRB.             0.138  0.246
## AST.             0.275  0.379
## STL.             0.256  0.162
## BLK.             0.204  0.172
## TOV.            -0.138 -0.015
## USG.             0.098  0.321
## OWS              0.477  0.870
## DWS              0.495  0.755
## WS               0.528  0.906
## WS.48            0.873  0.275
## OBPM             0.907  0.384
## DBPM             0.463  0.383
## BPM              1.000  0.502
## VORP             0.502  1.000
```

Para una mejor visualización de las relaciones entre las variables ploteo la matriz de correlaciones.

```r
corrplot(round(cor(mData),10), type = 'upper', order = "hclust", tl.col = "black", tl.cex = 0.65, tl.sr
```



Atendiendo a las relaciones entre las variables del modelo, crearé un modelo auxiliar en el cual únicamente se encuentren aquellas variables que guarden una relación significtiva entre sí.

```r
mDataAux <- select(mData,Salary,VORP,OWS,WS,Age,NBA_DraftNumber,AST.,USG.,BPM,OBPM,PER,DWS,G,MP)
```

PRIMER MODELO, TODAS LAS VARIABLES

Training y test set

Dividimos la muestra en un 70/30, el 70% de las observaciones se asignarán al training y el 30% restante al test.

```
set.seed(123)
NBA_split <- initial_split(mData, prop = .7, strata = "Salary")
NBA_train <- training(NBA_split)
NBA_test  <- testing(NBA_split)
```

Generamos las matrices.

```
NBA_train_x <- model.matrix(Salary ~ ., NBA_train)[,-1]
NBA_train_y <- NBA_train$Salary

NBA_test_x <- model.matrix(Salary~., NBA_test)[, -1]
NBA_test_y <- NBA_test$Salary
```

Caret

Aplico elastic net mediante le función del paquete CARET.

```
train_control <- trainControl(method = "cv", number = 10)

caret_mod <- train(
  x = NBA_train_x,
  y = NBA_train_y,
  method = "glmnet",
  preProc = c("center", "scale", "zv", "nzv"),
  trControl = train_control,
  tuneLength = 5
)

caret_mod
```

```
## glmnet
##
## 340 samples
##  24 predictor
##
## Pre-processing: centered (24), scaled (24)
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 306, 307, 306, 305, 307, 306, ...
## Resampling results across tuning parameters:
##
##   alpha  lambda       RMSE     Rsquared   MAE
##   0.100    4469.568   5117794  0.5454603  3839897
##   0.100   20745.899   5079590  0.5508850  3819828
##   0.100   96293.936   5000021  0.5623911  3769504
##   0.100  446956.880   4932709  0.5693543  3683790
```

```
##   0.100  2074590.159  5043122  0.5525840  3730574
##   0.325     4469.568  5117283  0.5456655  3841035
##   0.325    20745.899  5054122  0.5549962  3810859
##   0.325    96293.936  4944724  0.5706326  3731543
##   0.325   446956.880  4882917  0.5768097  3616611
##   0.325  2074590.159  5240957  0.5305437  3966956
##   0.550     4469.568  5100692  0.5479150  3832454
##   0.550    20745.899  5023569  0.5601039  3797750
##   0.550    96293.936  4907059  0.5763730  3700029
##   0.550   446956.880  4928475  0.5677952  3617557
##   0.550  2074590.159  5472321  0.5100224  4251782
##   0.775     4469.568  5098757  0.5482196  3833957
##   0.775    20745.899  4997415  0.5644787  3784084
##   0.775    96293.936  4879025  0.5807179  3674551
##   0.775   446956.880  5022291  0.5501621  3685629
##   0.775  2074590.159  5770409  0.4753793  4568528
##   1.000     4469.568  5090967  0.5495134  3831027
##   1.000    20745.899  4979432  0.5670945  3774168
##   1.000    96293.936  4860685  0.5835927  3652786
##   1.000   446956.880  5123059  0.5307428  3772799
##   1.000  2074590.159  6067466  0.4300466  4834855
##
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were alpha = 1 and lambda = 96293.94.
```

Dados los valores obtenidos a la hora de realizar elastic net mediante el paquete caret, alfa igual a 1, el mejor modelo se ha de calcular por el método Lasso. ### Lasso

```r
modelo_lasso <- cv.glmnet(NBA_train_x, NBA_train_y, alpha = 1)
min(modelo_lasso$cvm)
```

```
## [1] 2.490474e+13
```

Obtengo la media de MSE en la muestra correspondiente a test.

```r
prediction <- predict(modelo_lasso, s = modelo_lasso$lambda.min, NBA_test_x)
media_MSE <- mean((NBA_test_y - prediction)^2)
media_MSE
```

```
## [1] 3.338689e+13
```

Guardo la media del error del primer modelo.

```r
errormData <- media_MSE
```

SEGUNDO MODELO, SOLO LAS VARIABLES CORRELADAS

De nuevo divido las observaciones de la muestra en una proporción 70/30. ### Training y test set

```r
set.seed(123)
NBA_split <- initial_split(mDataAux, prop = .7, strata = "Salary")
NBA_train <- training(NBA_split)
NBA_test  <- testing(NBA_split)
```

Generamos las matrices.

```r
NBA_train_x <- model.matrix(Salary ~ ., NBA_train)[,-1]
NBA_train_y <- NBA_train$Salary

NBA_test_x <- model.matrix(Salary~., NBA_test)[, -1]
NBA_test_y <- NBA_test$Salary
```

Caret

Aplico elastic net mediante le función del paquete CARET.

```r
train_control <- trainControl(method = "cv", number = 10)

caret_mod <- train(
  x = NBA_train_x,
  y = NBA_train_y,
  method = "glmnet",
  preProc = c("center", "scale", "zv", "nzv"),
  trControl = train_control,
  tuneLength = 5
)

caret_mod
```

```
## glmnet
##
## 340 samples
##  13 predictor
##
## Pre-processing: centered (13), scaled (13)
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 306, 307, 306, 305, 307, 306, ...
## Resampling results across tuning parameters:
##
##   alpha  lambda       RMSE     Rsquared   MAE
##   0.100     4469.568  4881261  0.5817552  3689769
##   0.100    20745.899  4875911  0.5821481  3683062
##   0.100    96293.936  4865204  0.5826014  3655969
##   0.100   446956.880  4867354  0.5793516  3610275
##   0.100  2074590.159  5035778  0.5540864  3725018
##   0.325     4469.568  4880494  0.5817606  3688833
##   0.325    20745.899  4874495  0.5819093  3680435
##   0.325    96293.936  4855173  0.5836724  3644987
##   0.325   446956.880  4873273  0.5775284  3590210
##   0.325  2074590.159  5237173  0.5314584  3964965
##   0.550     4469.568  4880566  0.5816895  3689393
```

```
##   0.550    20745.899  4873984  0.5816825  3678740
##   0.550    96293.936  4845762  0.5848584  3634054
##   0.550   446956.880  4920823  0.5690391  3613983
##   0.550  2074590.159  5472598  0.5099909  4251956
##   0.775     4469.568  4881050  0.5816027  3689408
##   0.775    20745.899  4872050  0.5818683  3676562
##   0.775    96293.936  4841244  0.5853594  3624244
##   0.775   446956.880  5005685  0.5535942  3682030
##   0.775  2074590.159  5770299  0.4753813  4568418
##   1.000     4469.568  4881277  0.5815386  3689288
##   1.000    20745.899  4870414  0.5820111  3675006
##   1.000    96293.936  4842907  0.5851671  3616838
##   1.000   446956.880  5110449  0.5335401  3771979
##   1.000  2074590.159  6068386  0.4297741  4835799
##
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were alpha = 0.775 and lambda
##  = 96293.94.
```

El valor de alfa obtenido, 0.775, no se corresponde ni con el método Ridge, ni con Lasso. Aunque en se aproxima más al segundo de estos. ### Lasso

```
modelo_lassoAux <- cv.glmnet(NBA_train_x, NBA_train_y, alpha = 1)
min(modelo_lassoAux$cvm)
```

```
## [1] 2.472567e+13
```

Obtengo la media de MSE en la muestra correspondiente a test.

```
prediction <- predict(modelo_lassoAux, s = modelo_lassoAux$lambda.min, NBA_test_x)
media_MSE <- mean((NBA_test_y - prediction)^2)
media_MSE
```

```
## [1] 3.333355e+13
```

Guardo la media del error del primer modelo.

```
errormDataAux <- media_MSE
```

COMPARACION DE MODELOS

Ahora para comparar qué modelo presenta un menor error realizo una simple operación, resto a la media del error cuadrático del modelo con todas las variables, la media del error cuadrático del modelo con las variables correladas.

```
errormData - errormDataAux
```

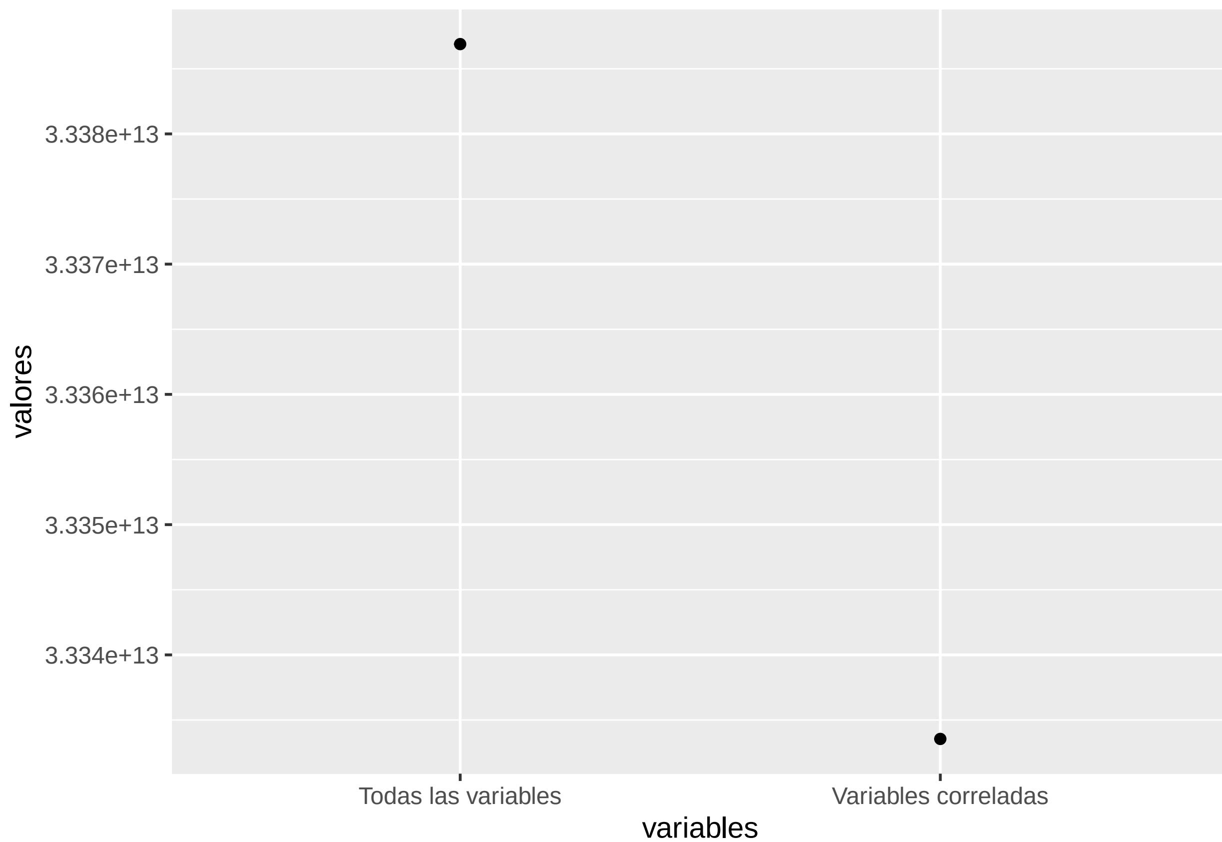
```
## [1] 53344649087
```

```r
1 - (errormDataAux / errormData)
```

```
## [1] 0.001597772
```

El valor de la diferencia es positivo, lo que indica que el MSE del segundo modelo, aquel que presenta variables correladas es menor.

```r
variables <- c('Todas las variables', 'Variables correladas')
valores <- c(errormData, errormDataAux)
df <- data.frame(variables, valores)
ggplot(df, aes( x = variables, y = valores)) + geom_point()
```



ESTIMACION DE LOS PREDICTORES DE AMBOS MODELOS

```r
predict(modelo_lasso, type = 'coefficients', s = modelo_lasso$lambda.min)
```

```
## 25 x 1 sparse Matrix of class "dgCMatrix"
##                               1
## (Intercept)     -5970909.929
## NBA_DraftNumber   -65340.797
## Age               431467.315
## G                -120231.091
## MP                  4726.962
## PER                    .
## TS.                    .
```

```
## X3PAr                  .
## FTr                    .
## ORB.          -12041.468
## DRB.           94220.445
## TRB.                   .
## AST.                   .
## STL.                   .
## BLK.         -194432.735
## TOV.                   .
## USG.           47677.515
## OWS           740743.858
## DWS                    .
## WS            190651.521
## WS.48                  .
## OBPM                   .
## DBPM                   .
## BPM                    .
## VORP          845755.186
```

```r
predict(modelo_lassoAux, type = 'coefficients', s = modelo_lassoAux$lambda.min)
```

```
## 14 x 1 sparse Matrix of class "dgCMatrix"
##                              1
## (Intercept)      -4579231.8098
## VORP               677638.4511
## OWS                513116.4188
## WS                 510894.6451
## Age                434212.7242
## NBA_DraftNumber    -69008.2979
## AST.                 -690.9363
## USG.                35547.7040
## BPM                          .
## OBPM                -3628.7406
## PER                          .
## DWS                          .
## G                 -136110.8345
## MP                   4995.8377
```