

# Postwork Sesión 5

Team 22

1/24/2021

## Objetivo

Continuar con el desarrollo de los postworks; en esta ocasión se utiliza la función `predict` para realizar predicciones de los resultados de partidos para una fecha determinada.

## Desarrollo

### Reto 1

A partir del conjunto de datos de soccer de la liga española de las temporadas 2017/2018, 2018/2019 y 2019/2020, crea el data frame `SmallData`, que contenga las columnas `date`, `home.team`, `home.score`, `away.team` y `away.score`; esto lo puede hacer con ayuda de la función `select` del paquete `dplyr`. Luego establece un directorio de trabajo y con ayuda de la función `write.csv` guarda el data frame como un archivo csv con nombre `soccer.csv`. Puedes colocar como argumento `row.names = FALSE` en `write.csv`.

**Solución:** Todos los datasets se encuentran en el directorio `../datasets`. Para leerlos y concatenarlos en un solo `data.frame` con solo aquellas columnas que se nos piden se escribió el siguiente código que fusiona los `data.frame`'s con ayuda de la función `rbind`. Posteriormente se realizan algunas modificaciones en el `data.frame` para que coincidan con las especificaciones de la librería `fbRanks`; cambio de nombre de columnas, conversión y formateo de fechas. Por último se almacena el `data.frame` resultante en una archivo `csv`.

```
library(tidyverse)
library(fbRanks)

datasets <- c('1718', '1819', '1920') # file names
colNames <-
  c('Date', 'HomeTeam', 'FTHG', 'AwayTeam', 'FTAG') # columns to be selected
smallData <-
  data.frame(matrix(ncol = length(colNames), nrow = 0)) # create empty data.frame
colnames(smallData) <- colNames # rename colnames

# merge all datasets into one
for (set in datasets) {
  path <- paste('../datasets/SP1_', set, '.csv', sep = '')
  smallData <- rbind(smallData, read.csv(path)[colNames])
}

# modify data.frame for fbFranks data.frame specs
colnames(smallData) <-
  c('date', 'home.team', 'home.score', 'away.team', 'away.score') # rename cols
smallData$date <-
  strptime(as.character(smallData$date), '%d/%m/%Y') # parse factor -> date
```

```
format(smallData$date, '%Y-%m-%d') # change date formatting

write.csv(smallData, file = 'soccer.csv') # save data.frame to CSV

print(str(smallData))

## 'data.frame':    1140 obs. of  5 variables:
## $ date      : POSIXlt, format: "17-08-18" "17-08-18" ...
## $ home.team : Factor w/ 26 levels "Alaves","Ath Bilbao",...: 13 19 6 10 17 2 4 11 14 15 ...
## $ home.score: int   1 1 2 2 1 0 2 0 1 0 ...
## $ away.team : Factor w/ 26 levels "Alaves","Ath Bilbao",...: 1 12 18 3 8 9 5 16 20 7 ...
## $ away.score: int   0 0 3 2 1 0 0 3 0 1 ...
## NULL
```

## Reto 2

Con la función `create.fbRanks.dataframes` del paquete `fbRanks` importe el archivo `soccer.csv` a R y al mismo tiempo asígnelo a una variable llamada `listasoccer`. Se creará una lista con los elementos `scores` y `teams` que son data frames listos para la función `rank.teams`. Asigna estos data frames a variables llamadas `anotaciones` y `equipos`.

**Solución:** Una vez creado el archivo `soccer.csv`, importarlo con la librería `fbRanks` es sencillo<sup>1</sup> de la siguiente manera.

```
listaSoccer <- create.fbRanks.dataframes(scores.file = 'soccer.csv')
scores <- listaSoccer$scores
teams <- listaSoccer$teams
```

## Reto 3

Con ayuda de la función `unique` crea un vector de fechas (`fecha`) que no se repitan y que correspondan a las fechas en las que se jugaron partidos. Crea una variable llamada `n` que contenga el número de fechas diferentes. Posteriormente, con la función `rank.teams` y usando como argumentos los data frames `anotaciones` y `equipos`, crea un ranking de equipos usando únicamente datos desde la fecha inicial y hasta la penúltima fecha en la que se jugaron partidos, estas fechas las deberá especificar en `max.date` y `min.date`. Guarda los resultados con el nombre `ranking`.

**Solución:** Primero se utiliza la función `unique` para encontrar las fechas que no se repiten, después se utiliza la función `length` para determinar el número de fechas sin repetir y finalmente se utiliza este dato para construir el objeto `ranking`.

```
uDates <- unique(scores$date)
n <- length(uDates)

ranking <- rank.teams(
  scores = scores,
  teams = teams,
  min.date = uDates[1],
  max.date = uDates[n - 1]
)
```

<sup>1</sup>Para importarlo sin errores, fue necesario primero especificar el nombre de las columnas y el formato de las fechas tal cual lo espera `fbRanks`. Esto se realizó previamente en el primer bloque de código.

## Reto 4

Finalmente estima las probabilidades de los eventos, el equipo de casa gana, el equipo visitante gana o el resultado es un empate para los partidos que se jugaron en la última fecha del vector de fechas `fecha`. Esto lo puedes hacer con ayuda de la función `predict` y usando como argumentos `ranking` y `fecha[n]` que deberá especificar en `date`.

**Solución:** Una vez se tiene el objeto `ranking`, se utiliza la función `predict` con el argumento `date` para realizar las predicciones solicitadas.

```
predict(ranking, date = uDates[n])
```

```
## Predicted Match Results for 1900-05-01 to 2100-06-01
## Model based on data from 17-08-18 to 2020-07-16
## -----
## 2020-07-19 Alaves vs Barcelona, HW 9%, AW 76%, T 15%, pred score 0.7-2.5 actual: AW (0-5)
## 2020-07-19 Valladolid vs Betis, HW 29%, AW 43%, T 28%, pred score 1-1.3 actual: HW (2-0)
## 2020-07-19 Villarreal vs Eibar, HW 45%, AW 30%, T 25%, pred score 1.5-1.2 actual: HW (4-0)
## 2020-07-19 Ath Madrid vs Sociedad, HW 54%, AW 20%, T 26%, pred score 1.5-0.8 actual: T (1-1)
## 2020-07-19 Espanol vs Celta, HW 32%, AW 41%, T 27%, pred score 1.2-1.4 actual: T (0-0)
## 2020-07-19 Granada vs Ath Bilbao, HW 39%, AW 31%, T 30%, pred score 1.2-1 actual: HW (4-0)
## 2020-07-19 Leganes vs Real Madrid, HW 13%, AW 66%, T 21%, pred score 0.7-1.9 actual: T (2-2)
## 2020-07-19 Levante vs Getafe, HW 25%, AW 48%, T 27%, pred score 0.9-1.4 actual: HW (1-0)
## 2020-07-19 Osasuna vs Mallorca, HW 48%, AW 27%, T 25%, pred score 1.6-1.1 actual: T (2-2)
## 2020-07-19 Sevilla vs Valencia, HW 34%, AW 40%, T 26%, pred score 1.2-1.4 actual: HW (1-0)
```