SESSION 8 SUMMARY

Victor Miguel Terrón Macias

3/2/2021

INTERFAZ GRAFICA DE USUSARIO

También conocida como GUI (del inglés graphical user interface), es un programa informático que actúa de interfaz de usuario, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz. Su principal uso consiste en proporcionar un entorno visual sencillo para permitir la comunicación con el sistema operativo de una máquina o computador.

Habitualmente las acciones se realizan mediante manipulación directa, para facilitar la interacción del usuario con la computadora. Surge como evolución de las interfaces de línea de comandos que se usaban para operar los primeros sistemas operativos y es pieza fundamental en un entorno gráfico. Como ejemplos de interfaz gráfica de usuario, cabe citar los entornos de escritorio Windows, el X-Window de GNU/LINUX o el de MAC OS, Aqua.

DASHBOARD EN R CON SHINY

Este se presenta como una alternativa a los software tradicionales proporcionando ciertas ventajas como son:

- Una alternativa a las presentaciones estáticas, logrando interactuar con la audiencia
- Es open source, esto es una gran ventaja
- Crear una mejor experiencia para que el usuario pueda interactuar con los datos en representaciones gráficas, que puedan cambiar fácilmente los parámetros, tablas, data tables, imágenes, summarys, búsquedas con dplyr, todo lo que se te ocurra y sea un objeto de R puede ser integrado, haciendo que la solución de hipótesis pueda ser intuitiva y de manera más amigable
- La facilidad de poder compartir tu información de manera remota, publicandolos en servidores gratuitos si así se considera

Para la construcción de un dashboard en R se requiere antes que todo generar información que sea útil presentar, esto lo podemos relacionar como un informe en el que se van a mostrar los mejores resultados de la investigación de algún tema o proyecto en el que estés participando, esto no es del todo cierto ya que se pueden ir construyendo al momento de tenerlo.

La manera de crear un dashboard es utilizando creando una Shiny Web Application, esta se compone de dos maneras diferentes en su creación :

- En un solo archivo que se almacena en un archivo llamado app.R
- En dos archivos ui.R / server.R

Las dos funcionan de igual manera, sin embargo si se requiere tener un mejor orden se recomienda la segunda opción.

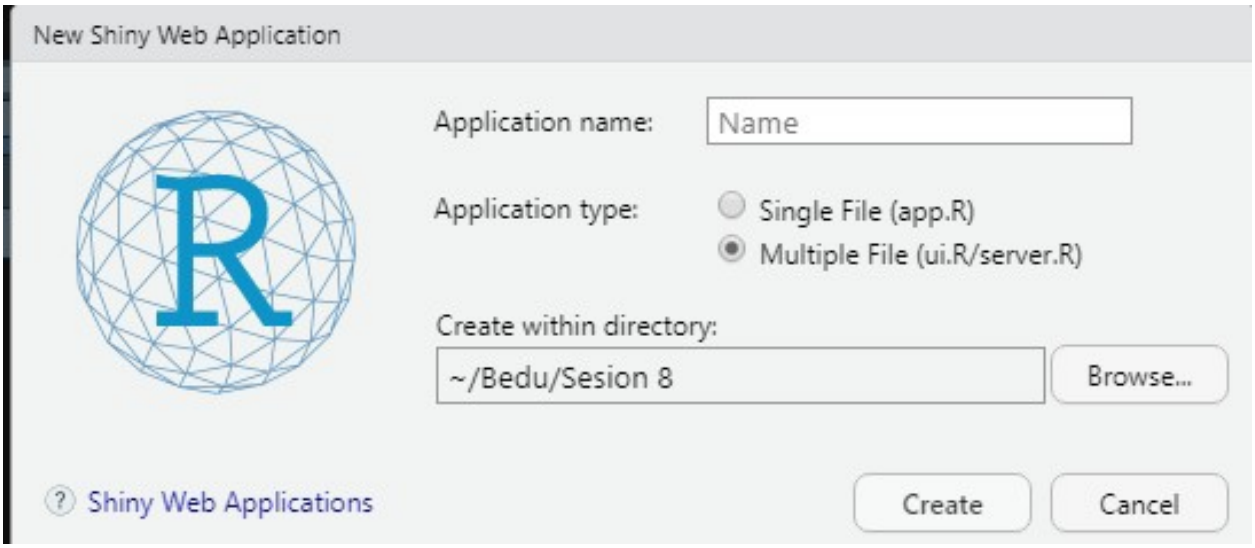


Figure 1: IMAGEN SHINY APP

Como ya se mencionó dentro de las ventajas que tiene Shiny WebApp es la publicación de los dashboard en su servidor, además de que se puede utilizar un servidor de RStudio para poder compartirlos, esto resulta útil y muy sencillo y el proceso para publicación es realmente sencillo.

Algunos sitios que son de interés para la creación de los dashboards son: * R Shiny . El sitio oficial donde se podrás encontrar diversos temas y ejemplos que te serán de gran utilidad * Bootswatch En este sitio encontraras diversos temas para poder publicar tus dashboards * Fontawesome Listas de íconos para hacer que tu dashboard quede en forma y sea amigable su visualización

¿Cómo funciona el ui.R y el server.R?

Dentro del ui.R se colocarán los comandos que permitan crear encabezados y agregar su título, paneles laterales, la parte central y demás objetos que darán a vista al dashboard, además de poder integrar pestañas, todo lo que se refiera a la interacción entre usuario y máquina para poder realizar las consultas permitidas y poder interactuar con el ratón o una pantalla touch si es el caso.

Por otro lado en el server.R se codifica la sintaxis necesaria para poder elaborar las tablas, gráficas, imágenes y demás objetos que se mostrarán en la GUI, será nuestro procesador de las entradas (inputs) que se reciban de la GUI.

Al comienzo puede parecer algo confuso el orden y el ingreso de las variables que darán forma a todos lo que se mostrará en la interfaz, sin embargo con la práctica y tomando como base algún tema su implementación se hará de manera natural.

No olvides entrar a los links de más arriba para observar el potencial que tienen las Shiny WebApp, es una forma potente y divertida de presentar tus reportes de resultados.

Durante el transcurso de esta sesión serás capaz de desarrollar las siguientes capacidades de R

- Realizar la presentación de gráficas, tablas, data tables, imágenes, queries, entre otras aplicaciones de R en dashboards, haciéndolo mediante una visualización atractiva y pudiendo compartir tu información con cualquier actor clave para su manejo.

EJEMPLO 1. AMBIENTE DE TRABAJOUI Y SERVER

OBJETIVO

- Crear un dashboard simple para mostrar gráficas de dispersión en una webApp
- Entender el entorno de trabajo con la libreria *Shiny*
- Funciones básicas de la UI y del Server

DESARROLLO

Durante esta sesión serás capaz de realizar una webApp con el uso de la librería de *Shiny*, está es útil para presentar reportes con los resultados destacados de nuestra investigación. Existen dos formas para realizar webApp, comenzaremos con dos archivos separados para tener un mayor orden:

Lo primero que se debe hacer es generar un archivo Shiny Web App en RStudio

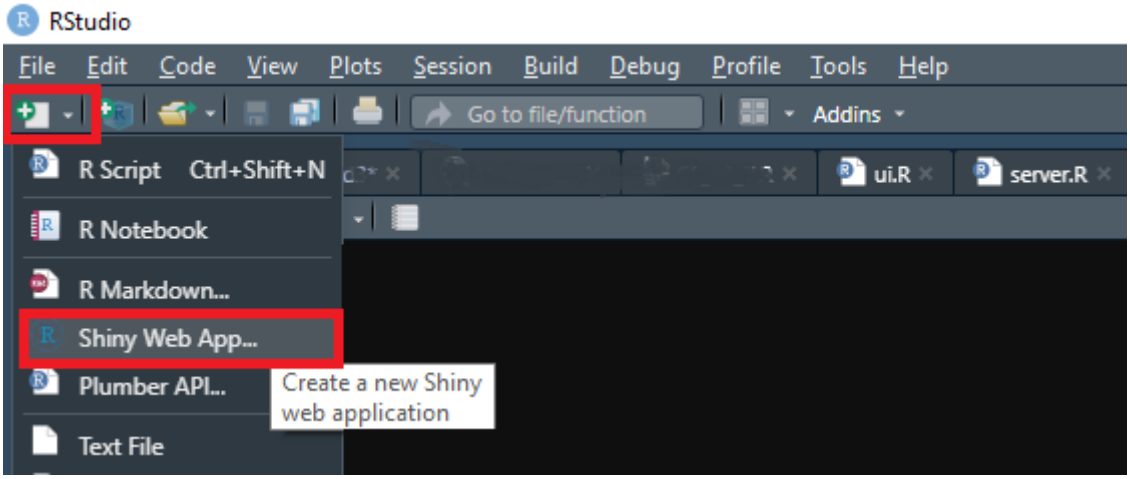


Figure 2: IMAGEN PASO 1

Hay que colocar un nombre de nuestra Web App, además de seleccionar Multiple File (esto creará dos archivos ui.R y server.R) y la ruta donde se almacenará

Una vez hecho esto, tendremos los dos archivos creados, en el UI (User interface), se establece la visualización de nuestro Dashboar o reporte, y en el Serve se establecen las variables de entrada y salida. Para ejecutar la Web App con dar clic en *RunApp* bastará. Los archivos por default tienen un ejemplo precargado el podría servir como base para ajustarlo a las necesidades de cada usuario

En la siguiente imágen podemos apreciar el ejemplo indicado anteriormente, ejecuta el ejemplo, intenta mover los parámetros para que observes el resultado tato dentro del archivo ui.R y de la webApp.

A continuación se creará una webApp desde cero, se deben borrar todos los comentarios para dejar solamente las siguientes líneas de código en el archivo *ui.R*

$$library(shiny) shinyUI()$$

Ahora en ui.R, dentro de la fución shinyUI colocaremos las siguientes instrucciones para poder visualizar las partes de nuestra webApp, posteriormente ejecuta el código para que observes el resultado y puedas ubicar donde se localiza gráficamente cada sentencia

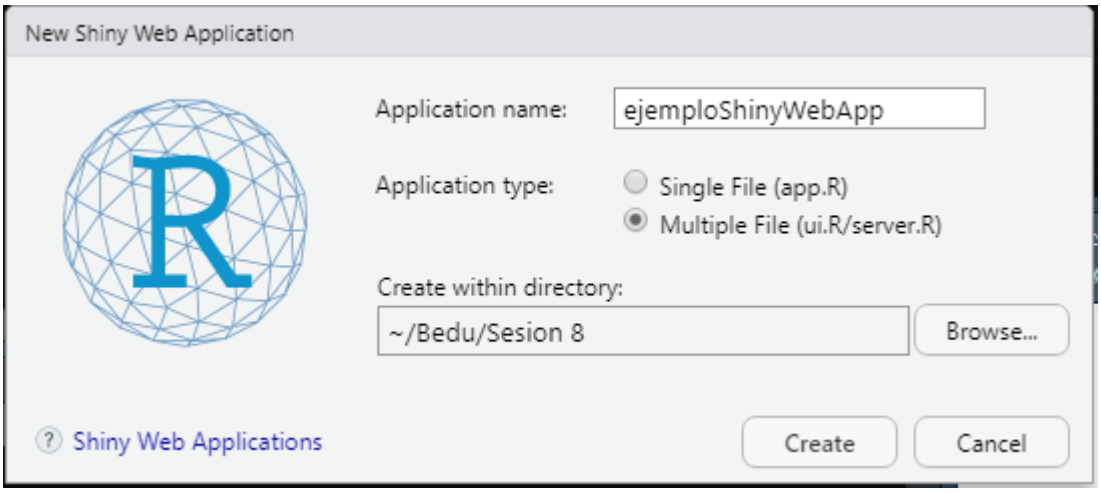


Figure 3: IMAGEN PASO 2

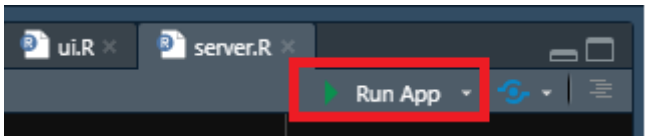


Figure 4: IMAGEN PASO 3

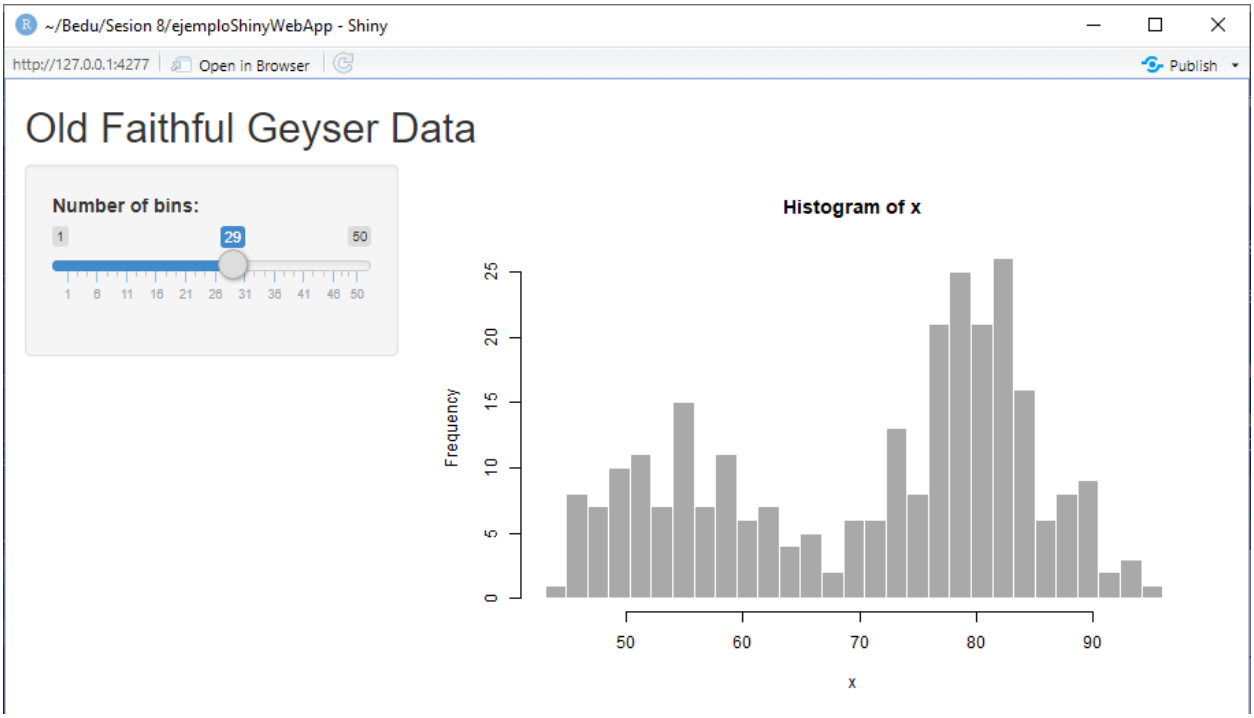


Figure 5: IMAGEN PASO 4

$library(shiny)shinyUI(pageWithSidebar(headerPanel("Esteeselheaderpanel"), sidebarPanel("Esteeselsidebarpanel"), m$

Con lo anterior ya pudiste observar la distrubución de los objetos dentro de la webApp. Ahora vamos a crear webApp una donde se puedan observar algunas gráficas de dispersión para las variables del dataset mtcars.

En el archivo ui.R realiza los siguientes cambios

```
library(shiny)
shinyUI(

    pageWithSidebar(
        headerPanel ("Aplicacion simple con Shiny"),
        sidebarPanel (
            p("Vamos a crear plots con el dataset de 'mtcars'"),
            selectInput("x", "Selecciona el eje de las X",      # Se indica que la variable "x" será la
                        choices = colnames(mtcars) )            # Sirve para desplegar las variables a
                     ),
        mainPanel (h3(textOutput("output_text")),
                   plotOutput("output_plot")
                   )
                  )
        )
```

Ahora en el archivo server.R, debes borrar todos los comentarios, de tal modo que quede de la siguiente forma, el código siguiente define los argumentos de input y output qué se visualizaran en la UI, ahora se harán los gráficos correlación entre mpg y el resto de variables de mtcars. (Con renderText y renderPlot se "renderizará" el texto y la gráfica respectivamente)

```
library(shiny)

shinyServer(function(input, output) {
    output$output_text <- renderText(paste("Grafico de mpg ~ ", input$x))   # input$x es la selección q
    output$output_plot <- renderPlot(plot(as.formula(paste("mpg~", input$x)),
                                          data = mtcars))
                                      }
            )
```

POSTWORK 8

```
# ```R
library(fbRanks)
library(dplyr)
```

```
Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag
```

```
The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```r
library(ggplot2)



# Colocar el directorio de trabajo según corresponda

# ```

# ```R
# Descarga de archivos
# https://www.football-data.co.uk/spainm.php

u1011 <- "https://www.football-data.co.uk/mmz4281/1011/SP1.csv"
u1112 <- "https://www.football-data.co.uk/mmz4281/1112/SP1.csv"
u1213 <- "https://www.football-data.co.uk/mmz4281/1213/SP1.csv"
u1314 <- "https://www.football-data.co.uk/mmz4281/1314/SP1.csv"
u1415 <- "https://www.football-data.co.uk/mmz4281/1415/SP1.csv"
u1516 <- "https://www.football-data.co.uk/mmz4281/1516/SP1.csv"
u1617 <- "https://www.football-data.co.uk/mmz4281/1617/SP1.csv"
u1718 <- "https://www.football-data.co.uk/mmz4281/1718/SP1.csv"
u1819 <- "https://www.football-data.co.uk/mmz4281/1819/SP1.csv"
u1920 <- "https://www.football-data.co.uk/mmz4281/1920/SP1.csv"

#RawData <- "C:\\\"
download.file(url = u1011, destfile ="SP1-1011.csv", mode = "wb")
download.file(url = u1112, destfile ="SP1-1112.csv", mode = "wb")
download.file(url = u1213, destfile ="SP1-1213.csv", mode = "wb")
download.file(url = u1314, destfile ="SP1-1314.csv", mode = "wb")
download.file(url = u1415, destfile ="SP1-1415.csv", mode = "wb")
download.file(url = u1516, destfile ="SP1-1516.csv", mode = "wb")
download.file(url = u1617, destfile ="SP1-1617.csv", mode = "wb")
download.file(url = u1718, destfile ="SP1-1718.csv", mode = "wb")
download.file(url = u1819, destfile ="SP1-1819.csv", mode = "wb")
download.file(url = u1920, destfile ="SP1-1920.csv", mode = "wb")
# ```
#
# ```R
# Lectura de datos

#lista <- lapply(list.files(path = RawData), read.csv)
# ```
#
# ```R
# Procesamiento de datos

#lista <- lapply(lista, select, Date:FTR)

d1011 <- read.csv("SP1-1011.csv")
d1112 <- read.csv("SP1-1112.csv")
d1213 <- read.csv("SP1-1213.csv")
```

```
d1314 <- read.csv("SP1-1314.csv")
d1415 <- read.csv("SP1-1415.csv")
d1516 <- read.csv("SP1-1516.csv")
d1617 <- read.csv("SP1-1617.csv")
d1718 <- read.csv("SP1-1718.csv")
d1819 <- read.csv("SP1-1819.csv")
d1920 <- read.csv("SP1-1920.csv")

d1011S <- select(d1011, Date:FTAG, BbMx.2.5:BbAv.2.5.1)
d1112S <- select(d1112, Date:FTAG, BbMx.2.5:BbAv.2.5.1)
d1213S <- select(d1213, Date:FTAG, BbMx.2.5:BbAv.2.5.1)
d1314S <- select(d1314, Date:FTAG, BbMx.2.5:BbAv.2.5.1)
d1415S <- select(d1415, Date:FTAG, BbMx.2.5:BbAv.2.5.1)
d1516S <- select(d1516, Date:FTAG, BbMx.2.5:BbAv.2.5.1)
d1617S <- select(d1617, Date:FTAG, BbMx.2.5:BbAv.2.5.1)
d1718S <- select(d1718, Date:FTAG, BbMx.2.5:BbAv.2.5.1)
d1819S <- select(d1819, Date:FTAG, BbMx.2.5:BbAv.2.5.1)
d1920S <- select(d1920, Date:FTAG, Max.2.5:Avg.2.5.1)
d1920S <- select(d1920S, -Time)
#colnames(d1718S); colnames(d1819S); colnames(d1920S)

# Arreglamos las fechas
d1011S <- mutate(d1011S, Date = as.Date(Date, format = "%d/%m/%y"))
d1112S <- mutate(d1112S, Date = as.Date(Date, format = "%d/%m/%y"))
d1213S <- mutate(d1213S, Date = as.Date(Date, format = "%d/%m/%y"))
d1314S <- mutate(d1314S, Date = as.Date(Date, format = "%d/%m/%y"))
d1415S <- mutate(d1415S, Date = as.Date(Date, format = "%d/%m/%y"))
d1516S <- mutate(d1516S, Date = as.Date(Date, format = "%d/%m/%y"))
d1617S <- mutate(d1617S, Date = as.Date(Date, format = "%d/%m/%y"))
d1718S <- mutate(d1718S, Date = as.Date(Date, format = "%d/%m/%y"))
d1819S <- mutate(d1819S, Date = as.Date(Date, format = "%d/%m/%Y"))
d1920S <- mutate(d1920S, Date = as.Date(Date, format = "%d/%m/%Y"))
# ```
#
# ```R
# Unimos de d1415S a d1819S

d1019S <- rbind(d1011S, d1112S, d1213S, d1314S, d1415S, d1516S, d1617S, d1718S, d1819S)
# ```
#
# ```R
# Renombrar columnas

d1019S <- rename(d1019S,  Max.2.5.O = BbMx.2.5,
                 Avg.2.5.O = BbAv.2.5,
                 Max.2.5.U = BbMx.2.5.1,
                 Avg.2.5.U = BbAv.2.5.1)

d1920S <- rename(d1920S,  Max.2.5.O = Max.2.5,
                 Avg.2.5.O = Avg.2.5,
                 Max.2.5.U = Max.2.5.1,
                 Avg.2.5.U = Avg.2.5.1)
```

```
# Ordenamos las columnas

d1019S <- select(d1019S, colnames(d1920S))

# Volvemos a unir

d1020S <- rbind(d1019S, d1920S)

# Renombramos

d1020S <- rename(d1020S, date = Date, home.team = HomeTeam, home.score = FTHG, away.team = AwayTeam, awa

# Ordenamos columnas

data <- select(d1020S, date, home.team, home.score, away.team, away.score:Avg.2.5.U) # Este data frame
# ```
#
#
# ```R
head(data, n = 2L); tail(data, n = 2L)
```

```
        date home.team home.score  away.team away.score Max.2.5.O Max.2.5.U
1 2010-08-28  Hercules          0 Ath Bilbao          1      2.26      1.73
2 2010-08-28   Levante          1    Sevilla          4      2.05      1.92
  Avg.2.5.O Avg.2.5.U
1      2.12      1.67
2      1.95      1.82
```

```
           date home.team home.score away.team away.score Max.2.5.O Max.2.5.U
3799 2020-07-19   Osasuna          2  Mallorca          2      1.97      2.03
3800 2020-07-19   Sevilla          1  Valencia          0      2.01      1.98
     Avg.2.5.O Avg.2.5.U
3799      1.88      1.91
3800      1.92      1.87
```

```
# ```

# Data frames de partidos y equipos

# ```R
md <- data %>% select(date:away.score)
write.csv(md, "match.data.csv", row.names = FALSE)
df <- create.fbRanks.dataframes(scores.file = "match.data.csv")
```

```
Alert: teams info file was not passed in.
Will construct one from the scores data frame but teams in the scores file must use a unique name.
Alert: teams resolver was not passed in.
Will construct one from the team info data frame.
```

```
teams <- df$teams; scores <- df$scores
# ```
```

```R
head(teams, n = 2L); dim(teams); head(scores, n = 2L); dim(scores)
```

```
     name
1  Alaves
2 Almeria
```

```
[1] 33  1
```

```
        date home.team home.score  away.team away.score
1 2010-08-28  Hercules          0 Ath Bilbao          1
2 2010-08-28   Levante          1    Sevilla          4
```

```
[1] 3800    5
```

```R
# ```

# Conjuntos iniciales de entrenamiento y de prueba

# ```R
f <- scores$date # Fechas de partidos
fu <- unique(f) # Fechas sin repetición
Ym <- format(fu, "%Y-%m") # Meses y años
Ym <- unique(Ym) # Meses y años sin repetir
places <- which(Ym[15]==format(scores$date, "%Y-%m")) # Consideramos partidos de 15 meses para comenzar
ffe <- scores$date[max(places)] # Fecha final conjunto de entrenamiento
# ```

# Consideraremos partidos de 15 meses para comenzar a ajustar el modelo. Así, nuestro primer conjunto d

# ```R
train <- scores %>% filter(date <= ffe)
test <- scores %>% filter(date > ffe)
# ```
#
#
# ```R
head(train, n = 1); tail(train, n = 1)
```

```
        date home.team home.score  away.team away.score
1 2010-08-28  Hercules          0 Ath Bilbao          1
```

```
          date home.team home.score away.team away.score
540 2011-12-18  Valencia          2    Malaga          0
```

```R
head(test, n = 1); tail(test, n = 1)
```

```
        date home.team home.score away.team away.score
1 2012-01-07   Levante          0  Mallorca          0
```

```
          date home.team home.score away.team away.score
3260 2020-07-19   Sevilla          1  Valencia          0
```

```
# ```

# Primer ajuste del modelo

# ```R
traindate <- unique(train$date)
testdate <- unique(test$date)
# ```


# ```R
ranks <- rank.teams(scores = scores, teams = teams,
                    min.date = traindate[1],
                    max.date = traindate[length(traindate)])
```

```
Team Rankings based on matches 2010-08-28 to 2011-12-18
   team        total attack defense n.games.Var1 n.games.Freq
1  Barcelona    2.52 2.50   2.29    Barcelona    54
2  Real Madrid  1.92 2.74   1.38    Real Madrid  54
3  Valencia     0.86 1.58   1.15    Valencia     54
4  Ath Bilbao   0.41 1.45   0.92    Ath Bilbao   54
5  Ath Madrid   0.40 1.52   0.87    Ath Madrid   54
6  Sevilla      0.35 1.42   0.90    Sevilla      54
7  Villarreal   0.29 1.17   1.04    Villarreal   54
8  Levante      0.22 1.17   0.99    Levante      54
9  Osasuna      0.15 1.17   0.95    Osasuna      54
10 Espanol      0.03 1.10   0.93    Espanol      54
11 Getafe       0.00 1.18   0.85    Getafe       54
12 Malaga       0.00 1.31   0.76    Malaga       54
13 Sp Gijon    -0.10 0.90   1.03    Sp Gijon     54
14 Betis       -0.14 1.00   0.91    Betis        16
15 Sociedad    -0.14 1.16   0.78    Sociedad     54
16 Vallecano   -0.15 1.03   0.87    Vallecano    16
17 Mallorca    -0.18 0.99   0.89    Mallorca     54
18 Santander   -0.22 0.92   0.93    Santander    54
19 La Coruna   -0.29 0.77   1.07    La Coruna    38
20 Zaragoza    -0.32 0.94   0.85    Zaragoza     54
21 Granada     -0.41 0.59   1.27    Granada      16
22 Hercules    -0.42 0.90   0.83    Hercules     38
23 Almeria     -0.62 0.91   0.71    Almeria      38
```

```
# ```

# Primera predicción


# ```R
pred <- predict(ranks, date = testdate[1])
```

```
Predicted Match Results for 1900-05-01 to 2100-06-01
Model based on data from 2010-08-28 to 2011-12-18
-----------------------------------------
2012-01-07 Levante vs Mallorca, HW 44%, AW 28%, T 28%, pred score 1.3-1  actual: T (0-0)
2012-01-07 Malaga vs Ath Madrid, HW 29%, AW 49%, T 22%, pred score 1.5-2  actual: T (0-0)
2012-01-07 Real Madrid vs Granada, HW 78%, AW 6%, T 16%, pred score 2.1-0.4  actual: HW (5-1)
2012-01-07 Santander vs Zaragoza, HW 37%, AW 33%, T 30%, pred score 1.1-1  actual: HW (1-0)
2012-01-07 Sociedad vs Osasuna, HW 31%, AW 44%, T 26%, pred score 1.2-1.5  actual: T (0-0)
```

```
# ```
```

```
# ```R
phs <- pred$scores$pred.home.score # predicted home score
pas <- pred$scores$pred.away.score # predicted away score
pht <- pred$scores$home.team # home team in predictions
pat <- pred$scores$away.team # away team in predictions
# ```
```

```
# Continuar ajustando y prediciendo
```

```
# ```R
phs <- NULL; pas <- NULL; pht <- NULL; pat <- NULL
for(i in 1:(length(unique(scores$date))-170)){
  ranks <- rank.teams(scores = scores, teams = teams,
                      min.date = unique(scores$date)[i],
                      max.date = unique(scores$date)[i+170-1],
                      silent = TRUE,
                      time.weight.eta = 0.0005)
  pred <- predict(ranks, date = unique(scores$date)[i+170],
                  silent = TRUE)

  phs <- c(phs, pred$scores$pred.home.score) # predicted home score
  pas <- c(pas, pred$scores$pred.away.score) # predicted away score
  pht <- c(pht, pred$scores$home.team) # home team in predictions
  pat <- c(pat, pred$scores$away.team) # away team in predictions
}
```

```
Warning in rpois(n * dim(newdata1)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata1)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata2)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata1)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata2)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata2)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata2)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata1)[1], exp(prate)): NAs produced
```

```
Warning in rpois(n * dim(newdata1)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata2)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata1)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata1)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata1)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata2)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata1)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata2)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata1)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata1)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata2)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata1)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata2)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata1)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata2)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata1)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata1)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata2)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata2)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata1)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata2)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata1)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata2)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata1)[1], exp(prate)): NAs produced
```

```
Warning in rpois(n * dim(newdata2)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata1)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata2)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata2)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata1)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata1)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata2)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata1)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata2)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata1)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata2)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata1)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata2)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata2)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata1)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata2)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata2)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata1)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata2)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata2)[1], exp(prate)): NAs produced

Warning in rpois(n * dim(newdata2)[1], exp(prate)): NAs produced
```

```r
# ```

# Eliminamos NA's


# ```R
buenos <- !(is.na(phs) | is.na(pas)) #
phs <- phs[buenos] # predicted home score
```

```r
pas <- pas[buenos] # predicted away score
pht <- pht[buenos] # home team in predictions
pat <- pat[buenos] # away team in predictions
momio <- data %>% filter(date >= unique(scores$date)[171]) # momios conjunto de prueba
momio <- momio[buenos,]
mean(pht == momio$home.team); mean(pat == momio$away.team)
```

```
[1] 1
```

```
[1] 1
```

```r
mean(phs + pas > 2.5 & momio$home.score + momio$away.score > 2.5)
```

```
[1] 0.3295019
```

```r
mean(phs + pas < 2.5 & momio$home.score + momio$away.score < 2.5)
```

```
[1] 0.2436143
```

```r
hs <- momio$home.score
as <- momio$away.score
# ```

# Probabilidades condicionales


# ```R
mean(phs + pas > 3) # proporción de partidos con más de tres goles según el modelo
```

```
[1] 0.2784163
```

```r
mean(phs + pas > 3 & hs + as > 2.5)/mean(phs + pas > 3)
```

```
[1] 0.6479358
```

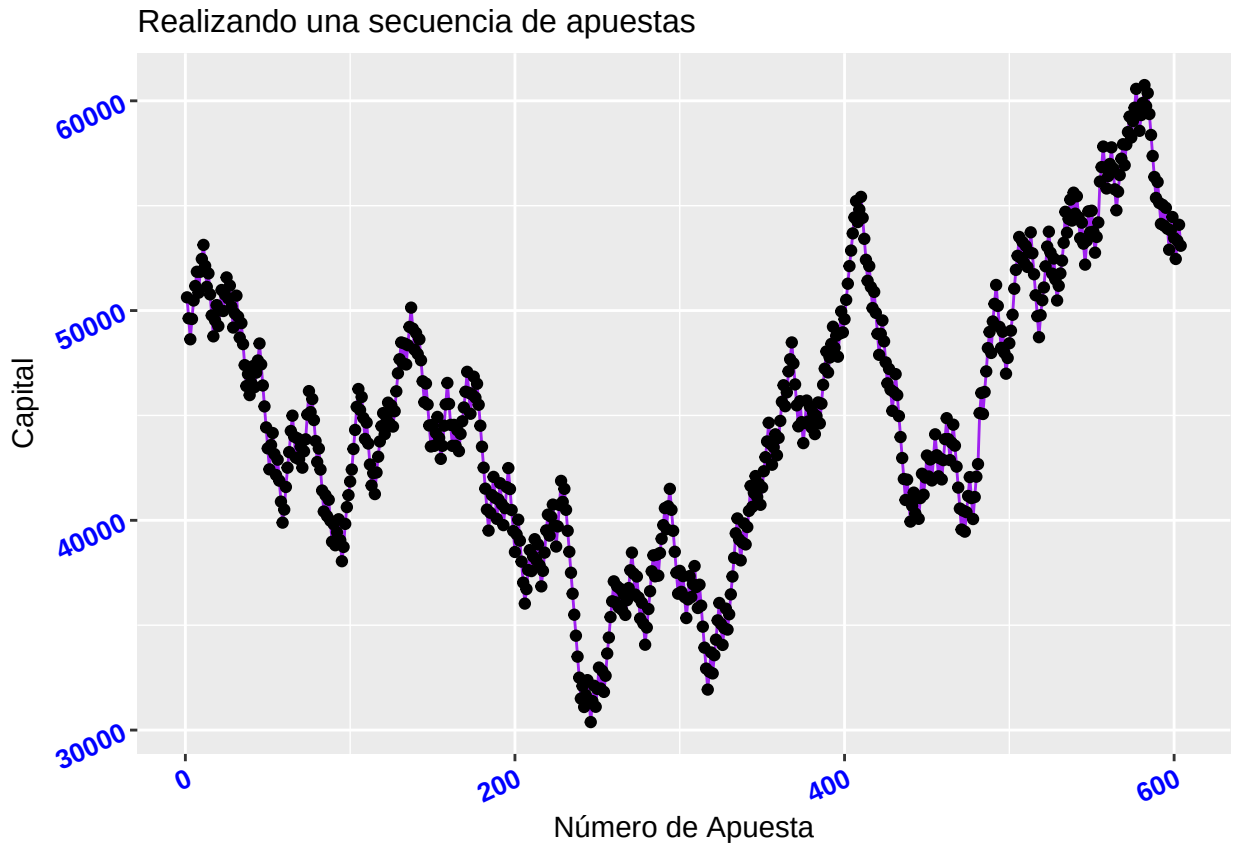
```r
# probabilidad condicional estimada de ganar en over 2.5
mean(phs + pas < 2.1) # proporción de partidos con menos de 2.1 goles según el modelo
```

```
[1] 0.1701788
```

```r
mean(phs + pas < 2.1 & hs + as < 2.5)/mean(phs + pas < 2.1)
```

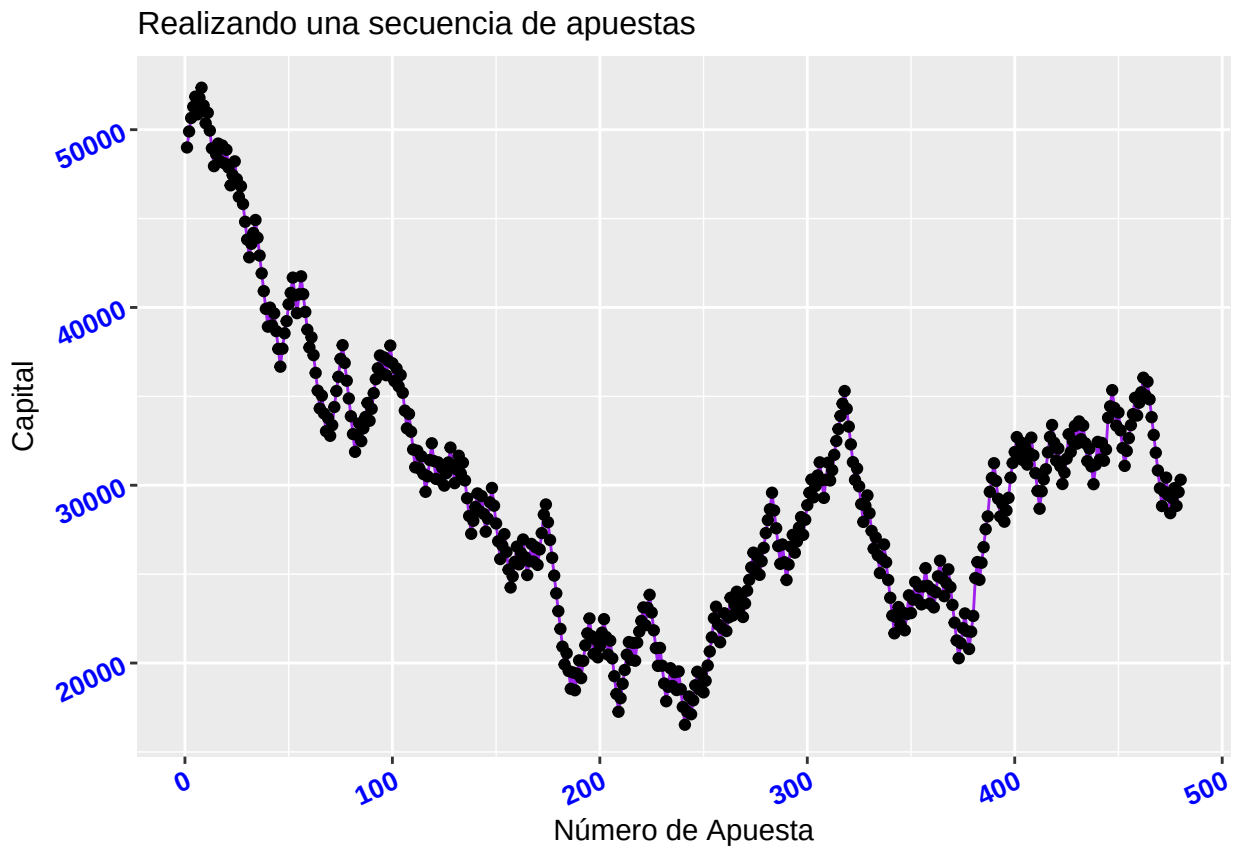
```
[1] 0.5891182
```

```
# probabilidad condicional estimada de ganar en under 2.5
# ```

# Apuestas con momios máximos


# ```R
cap <- 50000; g <- NULL

for(j in 1:length(phs)){
  if(((phs[j] + pas[j]) > 3) & (0.64/(momio$Max.2.5.O[j]^-1) > 1)){
    if((hs[j] + as[j]) > 2.5) cap <- cap + 1000*(momio$Max.2.5.O[j]-1)
    else cap <- cap - 1000
    g <- c(g, cap)
  }

  if(((phs[j] + pas[j]) < 2.1) & (0.58/(momio$Max.2.5.U[j]^-1) > 1)){
    if((hs[j] + as[j]) < 2.5) cap <- cap + 1000*(momio$Max.2.5.U[j]-1)
    else cap <- cap - 1000
    g <- c(g, cap)
  }
}
# ```

# Escenario con momios máximos

# ```R
g <- data.frame(Num_Ap = 1:length(g), Capital = g)
p <- ggplot(g, aes(x=Num_Ap, y=Capital)) + geom_line( color="purple") + geom_point() +
  labs(x = "Número de Apuesta",
       y = "Capital",
       title = "Realizando una secuencia de apuestas") +
  theme(plot.title = element_text(size=12))  +
  theme(axis.text.x = element_text(face = "bold", color="blue" , size = 10, angle = 25, hjust = 1),
        axis.text.y = element_text(face = "bold", color="blue" , size = 10, angle = 25, hjust = 1))  #
p
```




```
# ```

# Escenario con momios promedio

# ```R
cap <- 50000; g <- NULL

for(j in 1:length(phs)){
  if(((phs[j] + pas[j]) > 3) & (0.64/(momio$Avg.2.5.O[j]^-1) > 1)){
    if((hs[j] + as[j]) > 2.5) cap <- cap + 1000*(momio$Avg.2.5.O[j]-1)
    else cap <- cap - 1000
    g <- c(g, cap)
  }

  if(((phs[j] + pas[j]) < 2.1) & (0.58/(momio$Avg.2.5.U[j]^-1) > 1)){
    if((hs[j] + as[j]) < 2.5) cap <- cap + 1000*(momio$Avg.2.5.U[j]-1)
    else cap <- cap - 1000
    g <- c(g, cap)
  }
}
# ```
#
# ```R
g <- data.frame(Num_Ap = 1:length(g), Capital = g)
p <- ggplot(g, aes(x=Num_Ap, y=Capital)) + geom_line( color="purple") + geom_point() +
  labs(x = "Número de Apuesta",
```

```
       y = "Capital",
       title = "Realizando una secuencia de apuestas") +
  theme(plot.title = element_text(size=12))  +
  theme(axis.text.x = element_text(face = "bold", color="blue" , size = 10, angle = 25, hjust = 1),
        axis.text.y = element_text(face = "bold", color="blue" , size = 10, angle = 25, hjust = 1))  #
p
```




```
# ```
```