



Statistique de la première ligue anglaise 2020-2021

Yacine Bouhamidi*

Bastien Gévaudan†

Dernière compilation le 16 décembre 2021

*M1 SSD UGA, yacine.bouhamidi@gmail.com

†M1 SSD UGA, bastien@mail.com

Table des matières.

1	Introduction	3
2	Problématique	3
3	Développement	3
3.1	Ce que l'on voulait faire	3
3.2	Ce qui a été fait	3
3.2.1	Packages utilisés	3
3.2.2	Origine des données.	4
3.2.3	Transformation des données	4
3.2.4	Présentation des données	5
3.2.5	Application Shiny	6
3.2.6	Mise en ligne de l'application	9
3.3	Ce que l'on voudrait faire à l'avenir	10
4	Bibliographie	11

1 Introduction

Ce travail est un projet d'étude du M1 SSD réalisé par [Bastien Gévaudan](#) et [Yacine Bouhamidi](#). Il a été réalisé dans le cadre des cours relatifs au logiciel R, "TP logiciel spécialisé R" et "Outil de présentation et de recherche reproductible." Il est le premier projet que nous avons mené sur R, logiciel que nous avons découvert cette année avec plaisir. Ce rapport a pour but de présenter notre travail dans son développement.

2 Problématique

Le travail que nous avons mené avait pour sujet la première ligue de foot anglaise sur la saison 2020-2021. L'objectif était à partir de données collectées sur les équipes et les joueurs de cette league, de les exposer au travers d'une application Shiny.

3 Développement

Cette partie traite du développement de l'application Shiny qui a représenté le coeur de notre travail.

3.1 Ce que l'on voulait faire

Notre travail s'est découpé en plusieurs parties, le premier mois de travail a permis la découverte et une première prise en main. Par la suite nous nous sommes fixés sur l'étude des données de la ligue anglaise. Ensuite la mise en place de l'application Shiny et des graphiques. Enfin la rédaction du présent rapport ainsi que la création d'un diaporama de présentation.

3.2 Ce qui a été fait

Ce chapitre a pour but de présenter le coeur du travail qui a été mené.

3.2.1 Packages utilisés

On a importé différents packages R qui ont été nécessaires afin de réaliser l'application Shiny sur le fichier Rmarkdown nécessaire à la production de ce rapport. La liste des bibliothèques utilisées est disponible dans la bibliographie.[\[4\]](#).

```
library(shiny)
library(shinythemes)
library(shinydashboard)
library(gdata)
library(rintrojs)
library(shinyjs)
library(dplyr)
library(tidyr)
library(ggplot2)
library(leaflet)
library(DT)
```

3.2.2 Origine des données.

Les données sont originaires du site FBREF qui recense des milliers de données sur les joueurs et équipes de foot du monde entier(<https://fbref.com/fr/comps/9/10728/Stats-2020-2021-Premier-League>).

3.2.3 Transformation des données

On importe les données sur les joueurs du fichier pl.

```
pl<- read.csv2("pl.csv", sep=",")
head(pl,n=2)
```

```
##      X      X.1      X.2 X.3      X.4 X.5      X.6 Temps.de.jeu Temps.de.jeu.1
## 1 Clt Joueur Nation Pos Équipe Âge Naissance MJ Titulaire
## 2
##      Temps.de.jeu.2 Temps.de.jeu.3 Performance Performance.1 Performance.2
## 1      Min      90      Buts      PD      B-PénM
## 2
##      Performance.3 Performance.4 Performance.5 Performance.6 Par.90.minutes
## 1      PénM      PénT      CJ      CR      Buts
## 2
##      Par.90.minutes.1 Par.90.minutes.2 Par.90.minutes.3 Par.90.minutes.4 Attendu
## 1      PD      B+PD      B-PénM      B+PD-PénM      xG
## 2
##      Attendu.1 Attendu.2 Attendu.3 Par.90.minutes.5 Par.90.minutes.6
## 1      npxG      xA      npxG+xA      xG      xA
## 2
##      Par.90.minutes.7 Par.90.minutes.8 Par.90.minutes.9      X.7
## 1      xG+xA      npxG      npxG+xA      Matches
## 2
```

Les données ne sont pas directement exploitable il faut procéder à quelques modifications.

On enlève la deuxième ligne car ceux sont les totres des collones.

```
pl<-pl[-c(1,2),]
colnames(pl)<-c("ID","Joueur","Nation","Poste","Equipe","Age","Naissance","MJ","Titulaire","MinutesJouees")
```

La dernière colonne n'a aucun, il y est seulement inscrit "match." On va donc la supprimer.

```
pl<-pl[, -33]
```

On tranforme les caractères "\" en caractère ":" afin de pouvoir utiliser la fonction separate.

```
pl$Joueur<-gsub("\\", ":", pl$Joueur, fixed=TRUE)
pl<-separate(pl, Joueur, c("Joueur","A enlever"), sep = ":")
```

On enlève la colonne nom en doublon, qu'on avait renommé "A enlever."

```
pl<-pl[,-3]
pl<-separate(pl, Nation, c("A enlever","Nation"), sep = " ")
```

On enlève la collone nation en doublon, qu'on avait également renommé "A enlever."

```
pl<-pl[,-3]
pl$ID<-as.integer(pl$ID)
```

On transforme le types des valeurs des colonnes souhaitées en integer au lieu de character.

```
for (i in 1:13){
  pl[,i+5]<-as.integer(pl[,i+5])
}
```

On transforme ensuite le type des valeurs de certaines collones en double.

```
for (i in 19:32){
  pl[,i]<-as.double(pl[,i])
}
```

Enfin on importe les données du fichier Stadium, qui nous renseigne sur la position des stades des clubs de première league. On transforme le type des valeurs des coordonnées en numeric.

```
stade <- read.csv2("Stadium.csv", sep=",")
stade[,2]<-as.numeric(stade[,2])
stade[,3]<-as.numeric(stade[,3])
```

3.2.4 Présentation des données

Le résultat des transformations est donc que l'on obtient des données sous la forme suivante.

```
head(pl,n=3)
```

```
##      ID                               Joueur Nation Poste      Equipe Age
## 3  1 Patrick van Aanholt\\Patrick-van-Aanholt nl NED      DF Crystal Palace 29
## 4  2               Tammy Abraham\\Tammy-Abraham eng ENG      AT      Chelsea 22
## 5  3               Che Adams\\Che-Adams sct SCO      AT      Southampton 24
##      Naissance MJ Titulaire MinutesJouees MinJ/90 Buts PDT ButsSansPenT
## 3      1990 22      20      1777      19.7      0      1      0
## 4      1997 22      12      1040      11.6      6      1      6
## 5      1996 36      30      2667      29.6      9      5      9
##      PenaltysMarT PenaltysTen CJT CRT Buts/90 PD/90 ButsSansPen/90 PenaltysMar/90
## 3      0      0      2      0      0.00 0.05      0.05      0.00
## 4      0      0      0      0      0.52 0.09      0.61      0.52
## 5      0      0      1      0      0.30 0.17      0.47      0.30
##      CJ/90 CR/90 xGT sPenxGT xAT sPenxGT+xAT xG/90 xA/90 xG+xA/90 sPenxG/90
## 3  0.05  1.2 1.2      0.8 2.0      0.06 0.04 0.10      0.06 0.10
## 4  0.61  6.4 6.4      0.9 7.3      0.56 0.07 0.63      0.56 0.63
## 5  0.47  8.8 8.8      5.8 14.5      0.30 0.19 0.49      0.30 0.49
##      NA
## 3 Matches
## 4 Matches
## 5 Matches
```

3.2.5 Application Shiny

Une application Shiny est divisée en deux parties. Une partie UI qui va prendre en charge la gestion de l'interface utilisateur et une partie SERVER qui gère le fonctionnement de l'application.

3.2.5.1 La partie UI Au sein de la partie UI on a défini une fonction request dans laquelle on a inscrit les instructions suivantes.

dashboardPage nous permet de structurer l'allure générale de notre application en définissant les principaux éléments. Ici le dashboardHeader (Entête du tableau de bord), dont le titre est "Première League," la sidebar (barre latérale) ainsi que le body.

```
dashboardPage(  
  dashboardHeader(title = "Première League"),  
  sidebar,  
  body  
)
```

Les caractéristiques de ces différents éléments sont décrits dans les extraits de code suivant.
\newline

Ici on a défini la barre latérale, les différentes catégories que l'on va retrouver dans sur cette barre

```
sidebar<-dashboardSidebar(  
  hr(),  
  sidebarMenu(id="tabs",  
    menuItem("Accueil", tabName = "acc", icon=icon("home"), selected=TRUE),  
    menuItem("Stade", tabName="st", icon=icon("map")),  
    menuItem("Effectif", tabName="eff", icon=icon("users")),  
    menuItem("Graphiques", icon=icon("chart-line"),  
      menuSubItem("Graphique 1", tabName = "g1"),  
      menuSubItem("Graphiques 2", tabName = "g2"))  
  ))
```

Dans l'élément body on va traiter de ce qui est affiché sur l'écran principal de l'application.

Ici nous avons le code de la section effectif. Avec la possibilité d'afficher les joueurs des différentes équipes en sélectionnant une équipe grâce à selectInput. De rechercher un joueur précis grâce à une barre de recherche grâce à dataTableOutput.

```
tabItems(tabItem(tabName = "eff",  
  fluidPage(  
    h1("Effectif par équipe", style="font-family:monospace;text-align:center" ),  
    selectInput("inclub","Sélectionnez un club", choices= sort(pl$Equipe)),  
    dataTableOutput("playerdata")  
  ))
```

Ici nous avons la partie accueil. Avec les codes de mise en page. Ce permet d'afficher le logo de la première league, d'insérer la vidéo de présentation de cette première league à partir d'un lien youtube. Des notes de bas de pages sont également ajoutées grâce à "p(em(...))," qui permettent d'informer sur les créateurs de l'application ainsi que sur l'origine des données présentées.

```
tabItem(tabName = "acc",  
fluidPage( fluidRow(column(12,align="center", h1("Première League saison 2020-2021", style="font-family:  
  fluidRow(column(12, align="center", tags$img(src="pl.png",width="500px",height="200px", align ="cente
```

```

br(),
p(texte),
br(),

fluidRow(column(12, align= "center",HTML('<iframe width="560" height="315"   src="https://www.youtube

br(),
p("texte",
br() , strong("Stade"), "qui regroupe une map avec tous les stades de la league" ,
br(), strong("Effectif"),"qui regroupe les effectifs de chaque équipe",
br(), strong("Graphiques"), "oÃ¹ on peut retrouver 2 plots interactifs",
style="text-align:center;color:black;background-color:lavender;padding:15px;border-radius=20px"),
br(),
p(em("Application R shiny développée par :"),"Bouhamidi Yacine et Gévaudan Bastien",style="text-align:center"),
p(em("La base de données sur les ont été trouvé sur ce ")),
a(href="https://fbref.com/fr/comps/9/10728/Stats-2020-2021-Premier-League", "site",target="_blank")

))

```

Cette partie du code permet de coder l'onglet stade. On a précisé un titre "Map des stades." On a affiché la liste des clubs ainsi que qu'une map indiquant la position de chaque stade de première ligue.

```

tabItem(tabName = "st",
  fluidPage(
    h1("Map des stades", style="font-family:monospace;text-align:center"),
    leafletOutput("mymap"),
    dataTableOutput("stade")
  ))

```

Cette partie du code permet de programmer l'onglet du graphique numéro 1. On affiche un graphique des buts marqués en fonction des matchs joués.

```

tabItem(tabName = "g1",
  fluidPage(
    h1("Graphique numéro 1", style="font-family:monospace;text-align:center"),
    box(solidHeader= TRUE,
      plotOutput("plot1")),
    box(title="Inputs",statuts="info",solidHeader = TRUE,
      numericInput('x', 'Nombre de matchs joués', 3, min = 0, max = 38),
      numericInput('y', 'Nombre de buts', 3, min = 0, max = 38))
    )
  )

```

Cette partie du code permet de programmer l'onglet du graphique numéro 2. On a ici deux graphiques qui affichent, respectivement,

```

tabItem(tabName = "g2",
fluidPage(h1("Graphiques numéro 2", style="font-family:monospace;text-align:center"),
p("Légende : MJ = matchs joués, PDT=Passes décisives",
style="text-align:center;color:black;background-color:lavender;padding:15px;border-radius=20px"),

```

```

box(title = "Graphique général", status="primary", solidHeader = TRUE,
selectInput("xcol", "Selectionnez le x du graphique", choices = c("MJ","Titulaire", "MinutesJouees")),
selectInput("ycol", "Selectionnez le y du graphique", choices = c("Buts", "PDT", "ButsSansPenT")),
p(strong("Graphique du x en fonction du y avec toutes les équipes "),style="text-align:center"),
br(),
plotOutput("plot2", click = "plot2_click"),
p("Cliquez sur un point, voici le (ou les) joueur(s) représenté(s) par ce point :", style="text-align:center"),
verbatimTextOutput("click_info")),
box(title="Graphique selon l'équipe", status="primary", solidHeader = TRUE,height = "700px",
tabBox(height = "250px", width="450",
tabPanel("Nuage de points",p(strong("Graphique du x en fonction du y en fonction d'une équipe choisie")),
br(),
selectInput("inclub2","Selectionnez un club", choices= sort(pl$Equipe), selected= (pl$Equipe=="Arsenal")),
plotOutput("plot3")),
tabPanel("Boxplot",p(strong("Boxplot de la variable x en fonction de l'équipe choisie"), style="text-align:center"),
plotOutput("plot4"))))

)

```

3.2.5.2 La partie server L'autre partie de l'application est la partie serveur qui va se charger de gérer le fonctionnement de l'application, notamment les réactions aux interactions de l'utilisateur.

La fonction `renderDataTable` va nous permettre d'extraire les informations désirées de la base de données que l'on a traité. Ici on va sélectionner les données relatives aux nom des joueurs, à leurs postes, au nombre de matchs joué ainsi que les performances par temps de jeux.

```

output$playerdata <- renderDataTable({
  subset(pl, pl$Equipe == input$inclub, select=c("Joueur","Poste", "MJ", "Buts","PDT"))
})

```

Ici ceux sont les informations relatives aux stades qui sont extraites.

```

output$stade <- renderDataTable(
  stade %>%
  datatable(editable="row")
)

```

```

sd1 <- reactive({
  pl[, c(input$x, input$y)]
})

sd2 <- reactive({
  pl[, c(input$xcol, input$ycol, "Equipe")]
})

sd3 <- reactive({
  pl2<-pl[, c(input$xcol, input$ycol)]
  pl2<- subset(pl2, pl$Equipe == input$inclub2)
})

```


Les output suivant permettent le rendu des différents graphiques que nous avons déjà traité dans la partie UI.

```
output$plot1 <- renderPlot({
  ggplot(pl, aes(x=MJ, y=Buts))+
    geom_point()+
    geom_point(aes(x=input$x,y=input$y), colour="blue")+
    labs(x="Matches Joués", y="Buts marqués")
})
```

```
output$plot2 <- renderPlot({
  ggplot(sd2(),aes_string(x=input$xcol, y=input$ycol))+
    geom_point(aes(color=Equipe))
})
```

```
output$plot3<- renderPlot({
  ggplot(sd3(),aes_string(x=input$xcol, y=input$ycol))+
    geom_point()})
```

```
output$plot4<- renderPlot({ggplot(sd3(), aes(x=input$inclub2)) +
  geom_boxplot(aes_string(y=input$xcol))+
  labs(x="Equipe choisie")})
```

Enfin l'output de la carte des clubs.

```
output$mymap <- renderLeaflet({
  leaflet() %>%
    addTiles() %>%
    addMarkers(data = stade, ~Longitude, ~Latitude, popup = paste0("<b> Equipe : </b>"
                                                                    , stade$Equipe
                                                                    , "<br>"
                                                                    , "<b> Stade : </b>"
                                                                    , stade$Stade
                                                                    ) )
})
```

3.2.6 Mise en ligne de l'application

L'application a été réunie dans un seul fichier App.R qui a été placée dans un dépôt git.

```
library(shiny)
runGitHub(repo= "appshiny" , username="bastiengev", ref="main")
```

Le code de l'application peut être retrouvé à l'adresse suivante <https://fbref.com/fr/comps/9/10728/Stats-2020-2021-Premier-League>.

3.3 Ce que l'on voudrait faire à l'avenir

Nous avons des projets plus larges de développement de cette application Shiny. Cependant les limitations en termes de maîtrise de l'outil et du manque de temps nous ont empêché de réaliser l'ensemble du projet initial.

On pourrait poursuivre le développement d'une telle application. La partie sur les graphiques pourraient être améliorée. D'une part avec des graphiques plus complets, mais également en développant l'aspect interactif de ces graphiques. La map des stades pourrait également être améliorée, en affichant les logos des clubs et des données relatives à ces clubs directement dessus. Ici encore l'interactivité pourrait être développée.

L'aspect étude des données pourrait également être approfondie, cependant l'enjeu du cours n'étant pas l'étude des données nous avons préféré nous concentrer sur l'acquisition des savoirs relatifs à la création d'une application.

Une telle présentation des données pourrait évidemment se généraliser aux autres ligues d'Europe (voir du monde) et également prendre en compte les compétitions internationales. La mise en place d'une application traitant de l'ensemble des ligues d'Europe ainsi que des compétitions internationales, poserait des problèmes supplémentaires de présentation claire des données et de cohérence générale de l'interface.

4 Bibliographie

Pour réaliser ce travail, nous nous sommes appuyé sur les ouvrages suivants:

- Attali, Dean. 2020. *Shinyjs: Easily Improve the User Experience of Your Shiny Apps in Seconds*. <https://CRAN.R-project.org/package=shinyjs>.
- Chang, Winston. 2021. *Shinythemes: Themes for Shiny*. <https://CRAN.R-project.org/package=shinythemes>.
- Chang, Winston, and Barbara Borges Ribeiro. 2021. *Shinydashboard: Create Dashboards with 'Shiny'*. <https://CRAN.R-project.org/package=shinydashboard>.
- Chang, Winston, Joe Cheng, JJ Allaire, Carson Sievert, Barret Schloerke, Yihui Xie, Jeff Allen, Jonathan McPherson, Alan Dipert, and Barbara Borges. 2021. *Shiny: Web Application Framework for r*. <https://CRAN.R-project.org/package=shiny>.
- R Core Team. 2019. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. <https://www.R-project.org>.
- Warnes, Gregory R., Ben Bolker, Gregor Gorjanc, Gabor Grothendieck, Ales Korosec, Thomas Lumley, Don MacQueen, Arni Magnusson, Jim Rogers, and others. 2017. *Gdata: Various r Programming Tools for Data Manipulation*. <https://CRAN.R-project.org/package=gdata>.
- Wickham, Hadley. 2021. *Tidyr: Tidy Messy Data*. <https://CRAN.R-project.org/package=tidyr>.
- Xie, Yihui, J. J. Allaire, and Garrett Golemund. 2018. *R Markdown: The Definitive Guide*. Boca Raton, Florida: Chapman; Hall/CRC. <https://bookdown.org/yihui/rmarkdown>.
- Xie, Yihui, Christophe Dervieux, and Emily Riederer. 2020. *R Markdown Cookbook*. Boca Raton, Florida: Chapman; Hall/CRC. <https://bookdown.org/yihui/rmarkdown-cookbook>.