Football

Loïc Rakotoson

```
library(tidyverse)
library(DBI)
library(RSQLite)
```

```
db <- dbConnect(SQLite(), dbname = "football.sqlite")
```

1. Se connecter à la base SQLitefootball.sqlite

```
dbListTables(db)
```

2. Lister les bases

```
## [1] "Country"           "League"            "Match"
## [4] "Player"            "Player_Attributes" "Team"
## [7] "Team_Attributes"   "sqlite_sequence"
```

3. Afficher toutes les premières divisions présentes dans cette base de donnée (ligue 1 en France, sérieA en Italie etc…) La table **League** contient uniquement les premières divisions dont les `id` et les `country_id` renvoient aux `id` des pays dans **Country**.

```
query <- "SELECT Pays.Name AS Pays, Ligue.Name AS Ligue
          FROM Country AS Pays JOIN League AS Ligue ON Pays.id = Ligue.country_id"

as_tibble(
    dbGetQuery(db, query)
)
```

```
## # A tibble: 11 x 2
##    Pays        Ligue
##    <chr>       <chr>
##  1 Belgium     Belgium Jupiler League
##  2 England     England Premier League
##  3 France      France Ligue 1
##  4 Germany     Germany 1. Bundesliga
##  5 Italy       Italy Serie A
##  6 Netherlands Netherlands Eredivisie
##  7 Poland      Poland Ekstraklasa
##  8 Portugal    Portugal Liga ZON Sagres
##  9 Scotland    Scotland Premier League
## 10 Spain       Spain LIGA BBVA
## 11 Switzerland Switzerland Super League
```

```
dbListFields(db, "Match")
```

4. Afficher les champs de la table Match

```
##   [1] "id"               "country_id"       "league_id"
##   [4] "season"           "stage"            "date"
##   [7] "match_api_id"     "home_team_api_id" "away_team_api_id"
##  [10] "home_team_goal"   "away_team_goal"   "home_player_X1"
##  [13] "home_player_X2"   "home_player_X3"   "home_player_X4"
##  [16] "home_player_X5"   "home_player_X6"   "home_player_X7"
##  [19] "home_player_X8"   "home_player_X9"   "home_player_X10"
##  [22] "home_player_X11"  "away_player_X1"   "away_player_X2"
##  [25] "away_player_X3"   "away_player_X4"   "away_player_X5"
##  [28] "away_player_X6"   "away_player_X7"   "away_player_X8"
##  [31] "away_player_X9"   "away_player_X10"  "away_player_X11"
##  [34] "home_player_Y1"   "home_player_Y2"   "home_player_Y3"
##  [37] "home_player_Y4"   "home_player_Y5"   "home_player_Y6"
##  [40] "home_player_Y7"   "home_player_Y8"   "home_player_Y9"
##  [43] "home_player_Y10"  "home_player_Y11"  "away_player_Y1"
##  [46] "away_player_Y2"   "away_player_Y3"   "away_player_Y4"
##  [49] "away_player_Y5"   "away_player_Y6"   "away_player_Y7"
##  [52] "away_player_Y8"   "away_player_Y9"   "away_player_Y10"
##  [55] "away_player_Y11"  "home_player_1"    "home_player_2"
##  [58] "home_player_3"    "home_player_4"    "home_player_5"
##  [61] "home_player_6"    "home_player_7"    "home_player_8"
##  [64] "home_player_9"    "home_player_10"   "home_player_11"
##  [67] "away_player_1"    "away_player_2"    "away_player_3"
##  [70] "away_player_4"    "away_player_5"    "away_player_6"
##  [73] "away_player_7"    "away_player_8"    "away_player_9"
##  [76] "away_player_10"   "away_player_11"   "goal"
##  [79] "shoton"           "shotoff"          "foulcommit"
##  [82] "card"             "cross"            "corner"
##  [85] "possession"       "B365H"            "B365D"
##  [88] "B365A"            "BWH"              "BWD"
##  [91] "BWA"              "IWH"              "IWD"
##  [94] "IWA"              "LBH"              "LBD"
##  [97] "LBA"              "PSH"              "PSD"
## [100] "PSA"              "WHH"              "WHD"
## [103] "WHA"              "SJH"              "SJD"
## [106] "SJA"              "VCH"              "VCD"
## [109] "VCA"              "GBH"              "GBD"
## [112] "GBA"              "BSH"              "BSD"
## [115] "BSA"
```

5. Extraire dans un data-frame les résultats de la Ligue 1 française Utilisation d'une sous-requête au lieu d'un jointure pour esquiver la gestion de `.name_repair` lors de la conversion en tibble.

```
query <- "SELECT *
           FROM Match
           WHERE league_id = (
            SELECT id
            FROM League
            WHERE Name = 'France Ligue 1'
           )"

matchFR <- as_tibble(
    dbGetQuery(db, query)
)

matchFR
```

```
## # A tibble: 3,040 x 115
##       id country_id league_id season stage date  match_api_id home_team_api_id
##    <int>      <int>     <int> <chr>  <int> <chr>        <int>            <int>
##  1  4769       4769      4769 2008/~     1 2008~       483129             8583
##  2  4770       4769      4769 2008/~     1 2008~       483130             9827
##  3  4771       4769      4769 2008/~     1 2008~       483131             9746
##  4  4772       4769      4769 2008/~     1 2008~       483132             8682
##  5  4773       4769      4769 2008/~     1 2008~       483133             9748
##  6  4774       4769      4769 2008/~     1 2008~       483134             9829
##  7  4775       4769      4769 2008/~     1 2008~       483135             8481
##  8  4776       4769      4769 2008/~     1 2008~       483136             9851
##  9  4777       4769      4769 2008/~     1 2008~       483137             9874
## 10  4778       4769      4769 2008/~     1 2008~       483138             9873
## # ... with 3,030 more rows, and 107 more variables: away_team_api_id <int>,
## #   home_team_goal <int>, away_team_goal <int>, home_player_X1 <int>,
## #   home_player_X2 <int>, home_player_X3 <int>, home_player_X4 <int>,
## #   home_player_X5 <int>, home_player_X6 <int>, home_player_X7 <int>,
## #   home_player_X8 <int>, home_player_X9 <int>, home_player_X10 <int>,
## #   home_player_X11 <int>, away_player_X1 <int>, away_player_X2 <int>,
## #   away_player_X3 <int>, away_player_X4 <int>, away_player_X5 <int>,
## #   away_player_X6 <int>, away_player_X7 <int>, away_player_X8 <int>,
## #   away_player_X9 <int>, away_player_X10 <int>, away_player_X11 <int>,
## #   home_player_Y1 <int>, home_player_Y2 <int>, home_player_Y3 <int>,
## #   home_player_Y4 <int>, home_player_Y5 <int>, home_player_Y6 <int>,
## #   home_player_Y7 <int>, home_player_Y8 <int>, home_player_Y9 <int>,
## #   home_player_Y10 <int>, home_player_Y11 <int>, away_player_Y1 <int>,
## #   away_player_Y2 <int>, away_player_Y3 <int>, away_player_Y4 <int>,
## #   away_player_Y5 <int>, away_player_Y6 <int>, away_player_Y7 <int>,
## #   away_player_Y8 <int>, away_player_Y9 <int>, away_player_Y10 <int>,
## #   away_player_Y11 <int>, home_player_1 <int>, home_player_2 <int>,
## #   home_player_3 <int>, home_player_4 <int>, home_player_5 <int>,
## #   home_player_6 <int>, home_player_7 <int>, home_player_8 <int>,
## #   home_player_9 <int>, home_player_10 <int>, home_player_11 <int>,
## #   away_player_1 <int>, away_player_2 <int>, away_player_3 <int>,
## #   away_player_4 <int>, away_player_5 <int>, away_player_6 <int>,
## #   away_player_7 <int>, away_player_8 <int>, away_player_9 <int>,
## #   away_player_10 <int>, away_player_11 <int>, goal <chr>, shoton <chr>,
## #   shotoff <chr>, foulcommit <chr>, card <chr>, cross <chr>, corner <chr>,
## #   possession <chr>, B365H <dbl>, B365D <dbl>, B365A <dbl>, BWH <dbl>,
## #   BWD <dbl>, BWA <dbl>, IWH <dbl>, IWD <dbl>, IWA <dbl>, LBH <dbl>,
## #   LBD <dbl>, LBA <dbl>, PSH <dbl>, PSD <dbl>, PSA <dbl>, WHH <dbl>,
## #   WHD <dbl>, WHA <dbl>, SJH <dbl>, SJD <dbl>, SJA <dbl>, VCH <dbl>,
## #   VCD <dbl>, ...
```

6. Extraire dans un data-frame les résultats de la Ligue 1 française pour l'année 2015-2016 et ajouter le score du match et les points pour chaque match (match perdu = 0 pt, match nul = 1 pt et match gagné = 3 points) de la saison 2015-2016. On ne gardera que les colonnes qui sont utiles pour la suite.

```
matchFR <- matchFR %>%
filter(season == "2015/2016") %>%
mutate(
    score = home_team_goal - away_team_goal,
    home_points = case_when(
        score < 0 ~ 0,
        score > 0 ~ 3,
        TRUE ~ 1
    )
) %>%
select(home_team_api_id, home_team_goal, home_points,
       away_team_api_id, away_team_goal, score, season)

matchFR
```

```
## # A tibble: 380 x 7
##    home_team_api_id home_team_goal home_points away_team_api_id away_team_goal
##               <int>          <int>       <dbl>            <int>          <int>
##  1             7794              2           3             9851              1
##  2             9827              1           0             9837              2
##  3             8639              0           0             9847              1
##  4             9748              0           1             8689              0
##  5             8592              0           0             7819              1
##  6            10249              0           0             8121              2
##  7             9830              1           3             9747              0
##  8             9831              1           0             9829              2
##  9             9941              2           3             9853              1
## 10            10242              0           1             6391              0
## # ... with 370 more rows, and 2 more variables: score <int>, season <chr>
```

7. Trouver la meilleure équipe à domicile (celle qui le plus de points en ne comptant que les match à domicile) Importation de la table des équipes *(pour obtenir le nom)*.

```
team <<- as_tibble(tbl(db, "Team"))
```

Création d'une fonction qui sera utilisée dans les 3 prochaines questions.

```
best_n <- function(tb, n){
    #' Renvoie un tibble des n meilleurs équipes par points
    #' tb: tibble ayant les attributs de matchFR
```

```r
    #' n: entier

    best <- tb %>%
    group_by(home_team_api_id) %>%
    summarise(total_points = sum(home_points)) %>%
    arrange(desc(total_points)) %>%
    top_n(n, total_points) %>%
    inner_join(
        team,
        by = c(home_team_api_id = "team_api_id")
    ) %>%
    select(best_team = team_long_name, total_points)

    return(best)
}
```

La meilleure équipe à domicile.

```r
matchFR %>%
best_n(1)
```

```
## # A tibble: 1 x 2
##   best_team           total_points
##   <chr>                      <dbl>
## 1 Paris Saint-Germain           48
```

8. Trouver ensuite les trois meilleures équipes du championnat 2015-2016. Ajout des points gagnés en étant visiteur lors du championnat.

```r
matchFR <- matchFR %>%
mutate(
    away_points = case_when(
        score > 0 ~ 0,
        score < 0 ~ 3,
        TRUE ~ 1
    )
) %>%
select(home_team_api_id, home_team_goal, home_points,
       away_team_api_id, away_team_goal, away_points,
       score)
```

Création d'un vecteur de noms communs pour faciliter l'aggrégation des points gagnés à domicile et en tant que visiteur.

```r
newnames <- colnames(matchFR)[4:6]
names(newnames) <- colnames(matchFR)[1:3]
```

Les 3 meilleurs équipes du championnat en terme de points.

```r
matchFR %>%
select(all_of(newnames)) %>%
bind_rows(matchFR[,1:3]) %>%
best_n(3)
```

```
## # A tibble: 3 x 2
##   best_team           total_points
##   <chr>                      <dbl>
## 1 Paris Saint-Germain           96
## 2 Olympique Lyonnais            65
## 3 AS Monaco                     65
```

9. Trouver les trois meilleures équipes si nous ajoutons à l'attribution des points les bonus suivants :
   bonus offensif : +1 point si gain d'un match avec un ecart de 2 buts ou plus ;
   bonus spectacle : +1 point si le match est perdu mais l'équipe a quand même marqué 2 buts ou plus.

Ajout des bonus, s'il doit y en avoir, et affichage des 3 meilleurs équipes.

```
matchFR <- matchFR %>%
mutate(
    home_points = if_else(
        score >= 2 | (score < 0 & home_team_goal >= 2),
        home_points +1, home_points
    ),
    away_points = if_else(
        score <= -2 | (score > 0 & away_team_goal >= 2),
        away_points +1, away_points
    )
)

matchFR %>%
select(all_of(newnames)) %>%
bind_rows(matchFR[,1:3]) %>%
best_n(3)
```

```
## # A tibble: 3 x 2
##   best_team           total_points
##   <chr>                      <dbl>
## 1 Paris Saint-Germain          119
## 2 Olympique Lyonnais            79
## 3 OGC Nice                      73
```

Ferméture de la connexion.

```
dbDisconnect(db)
```