Foot scrapping

https://github.com/Eelai-Scheubel/scrapping

Eelai Scheubel, Laurentiu Istrati, Vasile Marcu

2025-01-14

Introduction

Ce projet utilise la librarie worldfootballR (https://cloud.r-project.org/web/packages/worldfootballR/readme/README.html) pour automatiser le processus d'extraction, filtrage et création d'une base de données. Les données utilisées proviennent du site FBref (https://fbref.com/en/)

Introduction

Le script permet en outre : - La collecte d'une série de statistiques (tirs, passes, buts) pour la ligue de football d'un pays donné - La création d'une base de données à partir de ces statistiques - L'exportation au format Excel de la base de données

Extraire des données grâce à R

```r
# Fonction pour extraire les statistiques

filter_stat <- function(team_urls, stat_type, comp) {
  stats <- fb_team_match_log_stats(team_urls = team_urls, stat_type = stat_type) %>%
    filter(Comp == comp, ForAgainst == "For") %>%
    select(-all_of(if (stat_type == "shooting") {
      c("Team_Url", "ForAgainst", "Gls_Standard")
    } else {
      c(1:13)
    }))
  Sys.sleep(3)
  return(stats)
}
```

Extraire des données grâce à R

```r
# Fonction pour fusionner les statistiques d'une équipe

combine_team_stats <- function(team_url, stat_types, comp) {
  stats_list <- lapply(stat_types, function(stat_type) {
    filter_stat(team_url, stat_type, comp)
  })
  merged_df <- do.call(cbind, stats_list)
  return(merged_df)
}
```

Fonction principale

```r
# Fonction principale

fetch_premier_league_stats <- function(url, stat_types, comp) {
  team_urls <- fb_teams_urls(url)
  final_data_frames <- lapply(team_urls, function(team_url) {
    combine_team_stats(team_url, stat_types, comp)
  })
  combined_df <- bind_rows(final_data_frames)
  return(combined_df)
}
```

Exemple d'utilisation

```
url <- "https://fbref.com/en/comps/9/2023-2024/2023-2024-Premier-League-Stats"
stat_types <- c("shooting", "passing", "keeper", "passing_types", "gca", "defense",
comp <- "Premier League"
name_xl <- "Database_PL.xlsx"
```

Temps d'exécution

```
t = Sys.time()
combined_df <- fetch_premier_league_stats(url, stat_types, comp)
Sys.time() - t
```

```
## Time difference of 22.38867 mins
```

Exemple d'application

```
head(combined_df)
```

```
##              Team       Date  Time           Comp       Round Day Venue Result
## 1 Manchester City 2023-08-11 20:00 Premier League Matchweek 1 Fri  Away      W
## 2 Manchester City 2023-08-19 20:00 Premier League Matchweek 2 Sat  Home      W
## 3 Manchester City 2023-08-27 14:00 Premier League Matchweek 3 Sun  Away      W
## 4 Manchester City 2023-09-02 15:00 Premier League Matchweek 4 Sat  Home      W
## 5 Manchester City 2023-09-16 15:00 Premier League Matchweek 5 Sat  Away      W
## 6 Manchester City 2023-09-23 15:00 Premier League Matchweek 6 Sat  Home      W
##   GF GA        Opponent Sh_Standard SoT_Standard SoT_percent_Standard
## 1  3  0         Burnley          17           8                 47.1
## 2  1  0   Newcastle Utd          14           4                 28.6
## 3  2  1   Sheffield Utd          29           9                 31.0
## 4  5  1          Fulham           6           4                 66.7
## 5  3  1       West Ham          29          13                 44.8
## 6  2  0 Nott'ham Forest           7           4                 57.1
##   G_per_Sh_Standard G_per_SoT_Standard Dist_Standard FK_Standard PK_Standard
## 1              0.18               0.38          13.9           0           0
## 2              0.07               0.25          17.9           0           0
## 3              0.07               0.22          17.3           2           0
## 4              0.67               1.00          14.8           0           1
## 5              0.10               0.23          16.4           1           0
## 6              0.29               0.50          17.2           2           0
##   PKatt_Standard xG_Expected npxG_Expected npxG_per_Sh_Expected
```

Exportation

```
write_xlsx(combined_df, name_xl)
```