cricketdata: An Open Source R package

Abstract

Open and accessible data streams are crucial for reproducible research and further development. Cricket data sources are limited and are usually not in a format ready for analysis. `cricketdata` R package allows the users to download the data as a tibble ready for analysis from two primary sources: ESPNCricinfo and Cricsheet. fetch_cricinfo() and fetch_player_data() functions allow the user to download the data from ESPNCricinfo for different formats of international cricket (tests, odis, T20), player position (batter, bowler, fielding), and whole career or innings wise. Cricsheet is another data source, primarily for ball-by-ball data. fetch_cricsheet() function downloads the ball-by-ball, match, and player data for different competitions/formats (tests, odis, T20 internationals, T20 leagues). The T20 data is further processed by adding more features (columns) using the raw data. Some other functions provide access to the individual players' playing career data and information about their playing style, country of origin, etc. The package essentially provides (almost) all publicly available cricket data ready for analysis. The package saves the user significant time in building the data pipeline, which may now be used for analysis. Here's an example of project built using `cricketdata`: https://dazzalytics.shinyapps.io/cricwar/

1 Introduction

The coverage of cricket as a sport has been limited compared to other global sports. ESPN Cricinfo is the major and one of the few online platforms dedicated to cricket coverage. It started as Cricinfo in the late 90s, and it was maintained by students and cricket fans who had immigrated to North America but were eager to keep tabs on the cricket activity around the globe. ESPN acquired Cricinfo in 2007, becoming ESPN Cricinfo. It is the most extensive repository of open cricket data with the caveat that data is not in an accessible format to be downloaded easily. You would have to copy-paste (tables) or write programming scripts to access the data in a format suitable for analysis. Recently they have added a search tool, Statsguru, that lets you parse through their database, presenting results usually in a table format.

Cricsheet is another open data source for ball-by-ball data maintained by a great fan of the game, Stephen Rushe. The cricsheet provides raw ball-by-ball data for all formats (tests, odis, T20) and both Men's and Women's games. It is an extensive project to produce ball-by-ball data, and we hugely appreciate Stephen Rushe's work. The data is available in different formats, such as JSON, YAML, and CSV.

1.1 Why `cricketdata`

The `cricketdata` (open-source) package aims to be a one-stop shop for most cricket data from all primary sources, available in an accessible form and ready for analysis. Different functions in the package allow us to download the data from Cricinfo and cricsheet as a data frame (tibble) in R. The user can access data from different formats of the game, e,g, tests, odis, international T20, league T20, etc. In particular, the

- ball-by-ball data,
- individual player play by innings data,
- player play by team wrt career or innings data,

- player id, dob, batting/bowling hand, bowling type.

cricWAR https://dazzalytics.shinyapps.io/cricwar/ is an example of sports analytic project based on `cricketdata` resources.

`cricketdata` as an open-source project is inspired primarily from the open-source work done by `Rstats` community and sports analytics projects such as `nflfastR` [1], `sportsdataverse` [2].

In the following sections, we will show how to install the package and take full advantage of the package functionality with numerous examples.

2 `cricketdata` Functionality

2.1 Installation

`cricketdata` is available on CRAN and the *stable* version can be installed.

```
install.packages("cricketdata", dependencies = TRUE)
```

You may also download the *development* version from Github

```
install.packages("devtools")
devtools::install_github("robjhyndman/cricketdata")
```

2.2 Functions

There are six main functions,

- `fetch_cricinfo()`
- `find_player_id()`
- `fetch_player_data()`
- `fetch_cricsheet()`
- `fetch_player_meta()`
- `update_player_meta()`

and a data file containing the player meta data.

- `player_meta`

We show the use of each function with examples below.

2.2.1 `fetch_cricinfo()`

Fetch team data on international cricket matches provided by ESPNCricinfo. It downloads data for international T20, ODI or Test matches, for men or women, and for batting, bowling or fielding. By default, it downloads career-level statistics for individual players.

Arguments

- matchtype: Character indicating test (default), odi, or t20.
- sex: Character indicating men (default) or women.
- activity: Character indicating batting (default), bowling or fielding.
- type: Character indicating innings-by-innings or career (default) data.
- country: Character indicating country. The default is to fetch data for all countries.

Women's T20 Bowling Data

```r
library(cricketdata)
library(tidyverse)
```

```r
# Fetch all Women's Bowling data for T20 format
wt20 <- fetch_cricinfo("T20", "Women", "Bowling")
```

```r
# Looking at data
wt20 %>%
  glimpse()
```

```
Rows: 1,798
Columns: 16
$ Player             <chr> "A Mohammed", "S Ismail", "EA Perry", "KH Brunt", "~
$ Country            <chr> "West Indies", "South Africa", "Australia", "Englan~
$ Start              <int> 2008, 2007, 2008, 2005, 2013, 2010, 2006, 2008, 201~
$ End                <int> 2021, 2022, 2021, 2022, 2022, 2022, 2022, 2020, 202~
$ Matches            <int> 117, 105, 126, 104, 84, 114, 107, 79, 72, 111, 106,~
$ Innings            <int> 113, 104, 119, 103, 83, 108, 95, 79, 72, 87, 105, 9~
$ Overs              <dbl> 395.3, 370.5, 380.5, 366.5, 278.3, 364.2, 286.3, 26~
$ Maidens            <int> 6, 15, 6, 16, 6, 9, 6, 10, 5, 4, 9, 7, 6, 6, 4, 2, ~
$ Runs               <int> 2206, 2153, 2237, 2019, 1685, 1951, 1822, 1587, 149~
$ Wickets            <int> 125, 115, 115, 108, 108, 106, 104, 102, 98, 98, 89,~
$ Average            <dbl> 17.64800, 18.72174, 19.45217, 18.69444, 15.60185, 1~
$ Economy            <dbl> 5.577750, 5.805843, 5.873961, 5.503862, 6.050269, 5~
$ StrikeRate         <dbl> 18.98400, 19.34783, 19.86957, 20.37963, 15.47222, 2~
$ BestBowlingInnings <chr> "5/10", "5/12", "4/12", "4/15", "4/18", "5/21", "4/~
$ FourWickets        <int> 4, 0, 4, 1, 3, 1, 1, 2, 3, 2, 4, 3, 1, 1, 3, 2, 2, ~
$ FiveWickets        <int> 3, 2, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, ~
```

```r
# Table showing certain features of the data
wt20 %>%
  select(Player, Country, Matches, Runs, Wickets, Economy, StrikeRate)%>%
 head() %>%
  knitr::kable(digits=2, align = "c")
```

Table 1: Women Player career profile for international T20

Player	Country	Matches	Runs	Wickets	Economy	StrikeRate
A Mohammed	West Indies	117	2206	125	5.58	18.98
S Ismail	South Africa	105	2153	115	5.81	19.35
EA Perry	Australia	126	2237	115	5.87	19.87
KH Brunt	England	104	2019	108	5.50	20.38
M Schutt	Australia	84	1685	108	6.05	15.47
Nida Dar	Pakistan	114	1951	106	5.35	20.62

```r
# Plotting Data
wt20 %>%
  filter(Wickets >= 50) %>%
  ggplot(aes(y = StrikeRate, x = Average)) +
  geom_point(alpha = 0.3, col = "blue") +
```

```
  ggtitle("Women International T20 Bowlers") +
  ylab("Balls bowled per wicket") + xlab("Runs conceded per wicket")
```

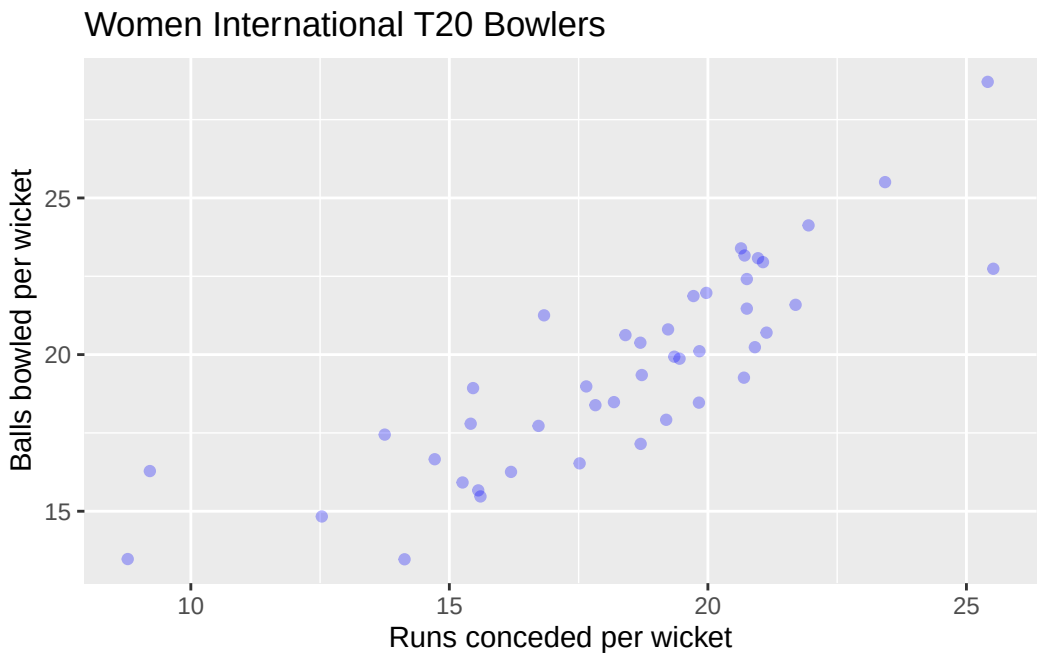

Figure 1: Strike Rate (balls bowled per wicket) Vs Average (runs conceded per wicket) for Women international T20 bowlers. Each observation represents one player, who has taken at least 50 international wickets.

USA men's ODI data by innings

```
# Fetch all USA Men's ODI data by innings
menODI <- fetch_cricinfo("ODI", "Men", "Batting", type = "innings",
                         country = "United States of America")

# Table of USA player who have scored a century
menODI %>%
  filter(Runs >= 100) %>%
  select(Player, Runs, BallsFaced, Fours, Sixes, Opposition) %>%
  knitr::kable(digits=2)
```

Table 2: Centuries, 100 runs or more in a single innings, scored by USA Batters

Player	Runs	BallsFaced	Fours	Sixes	Opposition
JS Malhotra	173	124	4	16	Papau New Guinea
MD Patel	130	101	11	6	Oman
Aaron Jones	123	87	9	6	Scotland
SR Taylor	114	123	11	3	Nepal
SJ Modani	111	133	9	0	Oman

Player	Runs	BallsFaced	Fours	Sixes	Opposition
MD Patel	100	114	9	1	Nepal

2.2.2 `fetch_player_id`

Each player has a player id on ESPNCricinfo, which is useful to access a individual player's data. This function given a string of players name or part of the name would return the name of corresponding player(s), their cricinfo id(s), and some other information.

Argument

- searchstring: string of a player's name or part of the name

```
# Fetching a player, Meg Lanning's, ID
meg_lanning <- find_player_id("Meg Lanning")
# ID
meg_lanning_id <- meg_lanning$ID
meg_lanning_id
```

```
[1] 329336
```

2.2.3 `fetch_player_data`

Fetch individual player data from all matches played. The function will scrape the data from ESPNCricinfo and return a tibble with one line per innings for all games a player has played. To identify a player, use their Cricinfo player ID. The simplest way to find this is to look up their Cricinfo Profile page. The number at the end of the URL is the ID. For example, Meg Lanning's profile page is http://www.espncricinfo.com/australia/content/player/329336.html, so her ID is 329336. Or you may use the `find_player_id` function.

Argument

- playerid
- matchtype: Character indicating test (default), odi, or t20.
- activity: Character indicating batting (default), bowling or fielding.

```
# Fetching the player Meg Lanning's playing data
MegLanning <- fetch_player_data(meg_lanning_id, "ODI") %>%
  mutate(NotOut = (Dismissal == "not out"))

dim(MegLanning)
```

```
[1] 100  14
```

```
names(MegLanning)
```

```
 [1] "Date"       "Innings"    "Opposition" "Ground"     "Runs"
 [6] "Mins"       "BF"         "X4s"        "X6s"        "SR"
[11] "Pos"        "Dismissal"  "Inns"       "NotOut"
```

```
# Compute batting average
MLave <- MegLanning %>%
  filter(!is.na(Runs)) %>%
  summarise(Average = sum(Runs) / (n() - sum(NotOut))) %>%
  pull(Average)
names(MLave) <- paste("Average =", round(MLave, 2))
```

```r
# Plot ODI scores
ggplot(MegLanning) +
  geom_hline(aes(yintercept = MLave), col="gray") +
  geom_point(aes(x = Date, y = Runs, col = NotOut)) +
  ggtitle("Meg Lanning ODI Scores") +
  scale_y_continuous(sec.axis = sec_axis(~., breaks = MLave))
```

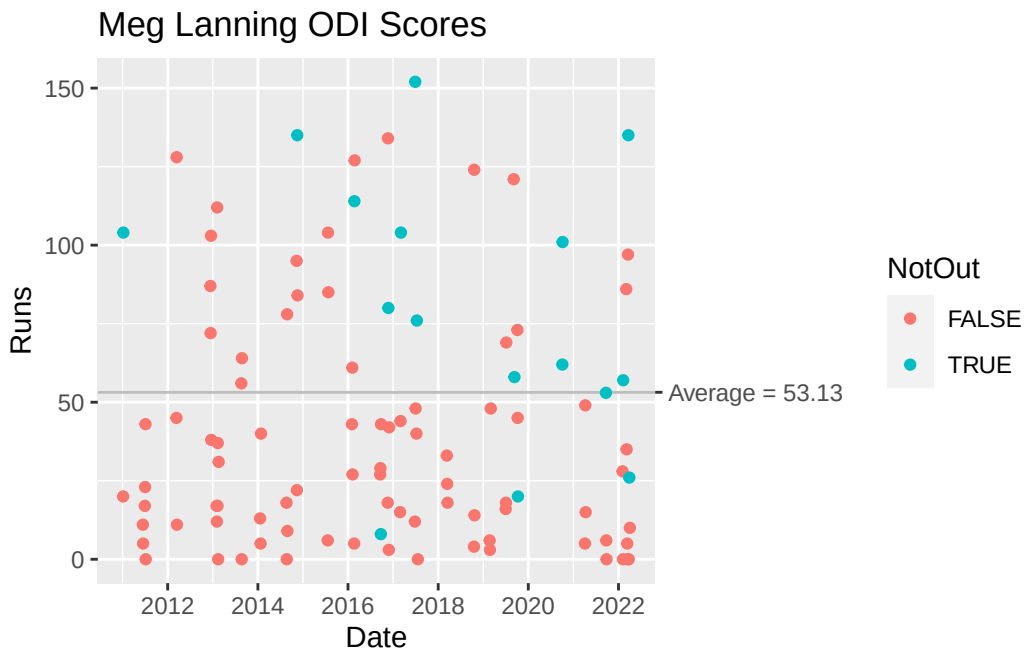

Figure 2: Meg Lanning, Australian captain, has shown amazing consistency over her career, with centuries scored in every year of her career except for 2021, when her highest score from 6 matches was 53

2.2.4 `fetch_cricsheet()`

Cricsheet is the only open accessible source for cricket ball-by-ball data. `fetch_cricsheet()` download csv data from cricsheet. Data must be specified by three factors: (a) type of data: bbb (ball-by-ball), match or player. (b) gender; (c) competition. See https://cricsheet.org/downloads/ for what the competition character codes mean.

The raw T20 data from cricsheet is further processed to add more columns (features) to facilitate analysis.

Arguments

- type: Character string giving type of data: ball-by-ball, match info or player info.
- gender: Character string giving player gender: female or male.
- competition: Character string giving name of competition. e.g. ipl for Indiana Premier League, psl for Pakistan Super League, tests for international test matches, etc.

Indian Premier League (IPL) Ball-by-Ball Data

```r
# Fetch all IPL ball-by-ball data

ipl_bbb <- fetch_cricsheet("bbb", "male", "ipl")

ipl_bbb %>%
  glimpse()
```

```
Rows: 225,954
Columns: 32
$ match_id              <int> 335982, 335982, 335982, 335982, 335982, 335982,~
$ season                <chr> "2007/08", "2007/08", "2007/08", "2007/08", "20~
$ start_date            <chr> "2008-04-18", "2008-04-18", "2008-04-18", "2008~
$ venue                 <chr> "M Chinnaswamy Stadium", "M Chinnaswamy Stadium~
$ innings               <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
$ over                  <dbl> 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 3, 3, 3,~
$ ball                  <int> 1, 2, 3, 4, 5, 6, 7, 1, 2, 3, 4, 5, 6, 1, 2, 3,~
$ batting_team          <chr> "Kolkata Knight Riders", "Kolkata Knight Riders~
$ bowling_team          <chr> "Royal Challengers Bangalore", "Royal Challenge~
$ striker               <chr> "SC Ganguly", "BB McCullum", "BB McCullum", "BB~
$ non_striker           <chr> "BB McCullum", "SC Ganguly", "SC Ganguly", "SC ~
$ bowler                <chr> "P Kumar", "P Kumar", "P Kumar", "P Kumar", "P ~
$ runs_off_bat          <int> 0, 0, 0, 0, 0, 0, 0, 0, 4, 4, 6, 4, 0, 0, 0, 0,~
$ extras                <int> 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1,~
$ ball_in_over          <int> 1, 2, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6, 1, 2, 3,~
$ extra_ball            <lgl> FALSE, FALSE, TRUE, FALSE, FALSE, FALSE, FALSE,~
$ balls_remaining       <dbl> 119, 118, 118, 117, 116, 115, 114, 113, 112, 11~
$ runs_scored_yet       <int> 1, 1, 2, 2, 2, 2, 3, 3, 7, 11, 17, 21, 21, 21, ~
$ wicket                <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE~
$ wickets_lost_yet      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
$ innings1_total        <int> 222, 222, 222, 222, 222, 222, 222, 222, 222, 22~
$ innings2_total        <int> 82, 82, 82, 82, 82, 82, 82, 82, 82, 82, 82, 82,~
$ target                <dbl> 223, 223, 223, 223, 223, 223, 223, 223, 223, 22~
$ wides                 <int> NA, NA, 1, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
$ noballs               <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
$ byes                  <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
$ legbyes               <int> 1, NA, NA, NA, NA, NA, 1, NA, NA, NA, NA, NA, N~
$ penalty               <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
$ wicket_type           <chr> "", "", "", "", "", "", "", "", "", "", "", "",~
$ player_dismissed      <chr> "", "", "", "", "", "", "", "", "", "", "", "",~
$ other_wicket_type     <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
$ other_player_dismissed <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
```

```r
# Top 20 batters wrt Boundary and Dot % in IPL 2022 season
ipl_bbb %>%
  filter(season == "2022") %>%
  group_by(striker) %>%
  summarize(Runs = sum(runs_off_bat), BallsFaced = n()-sum(!is.na(wides)),
  StrikeRate = Runs/BallsFaced, DotPercent = sum(runs_off_bat == 0)*100/BallsFaced,
  BoundaryPercent = sum(runs_off_bat %in% c(4,6))*100/BallsFaced   ) %>%
  arrange(desc(Runs)) %>%
  rename(Batter = striker) %>%
```

```
  slice(1:20) %>%
  ggplot(aes(y = BoundaryPercent, x = DotPercent, size = BallsFaced)) +
  geom_point(color = "red", alpha= 0.3) +
  geom_text(aes(label= Batter), vjust=-0.5, hjust= 0.5, color="#013369",
            position = position_dodge(0.9), size=3) +
   ylab("Boundary Percent") + xlab("Dot Percent") + ggtitle("IPL 2022: Top 20 Batters")
```

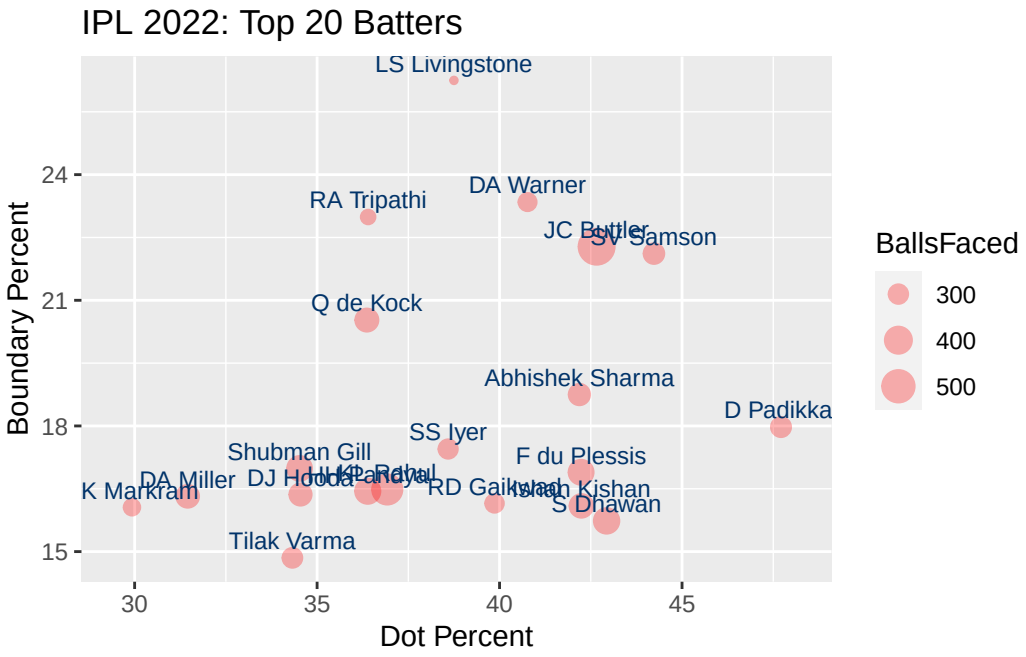

Figure 3: Top 20 prolific batters in IPL 2022. We show what percentage of balls they hit for a boundary (4 or 6) against percentage of how many balls they do not score off of (dot percent). Ideally we want to be in top left quadrant, high boundary % and low dot %.

```
# Top 10 prolific batters in IPL 2022 season.
ipl_bbb %>%
  filter(season == "2022") %>%
  group_by(striker) %>%
  summarize(Runs = sum(runs_off_bat), BallsFaced = n()-sum(!is.na(wides)),
  StrikeRate = Runs/BallsFaced,
  DotPercent = sum(runs_off_bat == 0)*100/BallsFaced,
  BoundaryPercent = sum(runs_off_bat %in% c(4,6))*100/BallsFaced   ) %>%
  arrange(desc(Runs)) %>%
  rename(Batter = striker) %>%
  slice(1:10) %>%
  knitr::kable(digits=1,align = "c")
```

Table 3: Top 10 prolific batters of IPL 2022 season. JC Butler scored the most runs in total and scored at the highest strike rate (runs per ball). His boundary percent (percentage of balls faced hit for 4s or 6s) is also the highest, while his dot percent (percentage of balls not scored of) is also among the highest.

Batter	Runs	BallsFaced	StrikeRate	DotPercent	BoundaryPercent
JC Buttler	863	579	1.5	42.7	22.3
KL Rahul	616	455	1.4	36.9	16.5
Q de Kock	508	341	1.5	36.4	20.5
HH Pandya	487	371	1.3	36.4	16.4
Shubman Gill	483	365	1.3	34.5	17.0
DA Miller	481	337	1.4	31.5	16.3
F du Plessis	468	367	1.3	42.2	16.9
S Dhawan	460	375	1.2	42.9	15.7
SV Samson	458	312	1.5	44.2	22.1
DJ Hooda	451	330	1.4	34.5	16.4

2.2.5 `player_meta`

It is a data set containing player's and cricket officials meta data such as full name, country of representation, data of birth, bowling and batting hand, bowling style, and playing role. More than 11,000 player's and officials data is available. This data was scraped from ESPNCricinfo website.

```
player_meta %>%
  filter(!is.na(playing_role)) %>%
  head() %>%
  knitr::kable(digits=1, align = "c", format = "pipe",
  col.names = c("ID", "FullName", "Country", "DOB", "BirthPlace", "BattingStyle",
                "BowlingStyle", "PlayingRole"))
```

Table 4: Player and officials meta data.

ID	FullName	Country	DOB	BirthPlace	BattingStyle	BowlingStyle	PlayingRole
1269467	Aaftab Alam Khan	Malta	1986-01-31	NA	Right hand Bat	Right arm Medium fast	Wicketkeeper Batter
1048889	Aahan Gopinath Achar	Singapore	1999-03-30	NA	Left hand Bat	Slow Left arm Orthodox	Bowler
27639	Aakash Chopra	India	1977-09-19	Agra Uttar Pradesh	Right hand Bat	Right arm Medium, Right arm Offbreak	Batter
661441	Aaliyah Alicia Alleyne	Barbados	1994-11-11	NA	Right hand Bat	Right arm Medium	Bowler
1325401	Aaliyah Williams	Barbados	1998-02-28	NA	Right hand Bat	Right arm Medium	Allrounder

ID	FullName	Country	DOB	BirthPlace	BattingStyle	BowlingStyle	PlayingRole
38965	Aamer Malik	Pakistan	1963-01-03	Mandi Bahauddin Punjab	Right hand Bat	Right arm Fast medium	Wicketkeeper

2.2.6 `fetch_player_meta()`

Fetch the player's meta data such as full name, country of representation, data of birth, bowling and batting hand, bowling style, and playing role. This meta data is useful for advance modeling, e,g, age curves, batter profile against bowling types etc.

Argument

- playerid: A vector of player IDs as given in Cricinfo profiles. Integer or character.

The cricinfo player ids can be accessed in multiple ways, e.g. use fetch_player_id() function, get the id from the player's cricinfo page or consult the `player_meta` data frame which has player meta data of more than 11,000 players.

```
# Download meta data on Meg Lanning and Ellyse Perry
aus_women <- fetch_player_meta(c(329336, 275487))

aus_women %>%
  knitr::kable(digits=1, align = "c", format = "pipe",
  col.names = c("ID", "FullName", "Country", "DOB", "BirthPlace", "BattingStyle",
                "BowlingStyle", "PlayingRole"))
```

Table 5: Australian Women player meta data.

ID	FullName	Country	DOB	BirthPlace	BattingStyle	BowlingStyle	PlayingRole
329336	Meghann Moira Lanning	Australia	1992-03-25	Singapore	Right hand Bat	Right arm Medium	Top order Batter
275487	Ellyse Alexandra Perry	Australia	1990-11-03	Wahroonga Sydney New South Wales	Right hand Bat	Right arm Fast medium	Allrounder

2.2.7 `update_player_meta()`

This function is supposed to consult the directory of all players available on cricsheet website and include the meta data of new players into the `player_meta` data frame. The data for new players will be scraped from the ESPNCricinfo.

References

[1] Sebastian Carl and Ben Baldwin. *nflfastR: Functions to Efficiently Access NFL Play by Play Data.* URL: https://cran.r-project.org/web/packages/nflfastR/index.html.

[2] Saiem Gilani. *Sports Dataverse.* URL: https://sportsdataverse.org/.