Package ‘fitzRoy’

January 12, 2021

Title  Easily Scrape and Process AFL Data
Version  0.3.3
Description  An easy package for scraping and processing Australia Rules Football (AFL) data. 'fitzRoy' provides a range of functions for accessing publicly available data from 'AFL Tables' <https://afltables.com/afl/afl_index.html>, 'Footy Wire' <https://www.footywire.com> and 'The Squiggle' <https://squiggle.com.au>. Further functions allow for easy processing, cleaning and transformation of this data into formats that can be used for analysis.
License  GPL-3
BugReports  https://github.com/jimmyday12/fitzRoy/issues
Depends  R (>= 3.5)
Imports  dplyr, httr, jsonlite, lubridate, magrittr, purrr, readr, rlang (>= 0.1.2), rvest, stringr (>= 1.3.0), tidy (>= 1.0.0), tidyselect, xml2, tibble, glue, progress
Suggests  covr, ggplot2, knitr, rmarkdown, testthat, roxygen2, elo, spelling, curl
VignetteBuilder  knitr
ByteCompile  true
Encoding  UTF-8
LazyData  true
RoxygenNote  7.1.1
Language  en-US
NeedsCompilation  no
Author  James Day [cre, aut], Robert Nguyen [aut], Matthew Erbs [ctb], Oscar Lane [aut], Jason Zivkovic [ctb]
Maintainer  James Day <jamesthomasday@gmail.com>
Repository  CRAN
Date/Publication  2021-01-12 09:40:17 UTC

R topics documented:

convert_results  .......................................................... 3
fetch_ladder  .............................................................. 3
fetch_ladder_afl  ......................................................... 4
footywire_html  ............................................................ 5
get_afltables_stats  ...................................................... 5
get_afltables_urls  ....................................................... 6
get_aflw_cookie  ........................................................... 7
get_aflw_detailed_data  ................................................ 7
get_aflw_detailed_match_data  ....................................... 8
get_aflw_match_data  .................................................... 8
get_aflw_player_stats  .................................................. 9
get_aflw_rounds  .......................................................... 10
get_aflw_round_data  ................................................... 11
get_afl_colour_palettes  ............................................... 11
get_afl_cookie  ............................................................ 12
get_afl_fixture  ........................................................... 12
get_fixture  ............................................................... 13
get_footywire_betting_odds  .......................................... 14
get_footywire_match_results  ......................................... 15
get_footywire_stats  ..................................................... 15
get_fryzigg_stats  ....................................................... 16
get_match_data  ........................................................... 17
get_match_results  ........................................................ 17
get_score_progression_raw  .......................................... 18
get_squiggle_data  ....................................................... 19
replace_teams  ............................................................ 20
replace_venues  .......................................................... 20
return_ladder  ............................................................ 21
scrape_afltables_match  ............................................... 22
update_footywire_stats  .............................................. 22
verify_year  ............................................................... 23

Index  24
**convert_results**  
*Convert AFL Men’s results into long format*

**Description**

`convert_results` returns a dataframe containing the results in long format.

**Usage**

```r
convert_results(results)
```

**Arguments**

- `results` A dataframe that has been returned from `get_match_results`

**Details**

The standard results returned by afltables.com will be in wide format. This is useful for game based analysis but less so for team based ones. This function converts the data into long format for easier analysis.

**Value**

A data frame with match results where each row is a team-match combination

**Examples**

```r
## Not run:
results <- get_match_results()
convert_results(results)
## End(Not run)
```

---

**fetch_ladder**  
*Fetch ladder*

**Description**

Returns the Ladder for the relevant Season and Round from the AFL.com.au website.

**Usage**

```r
fetch_ladder(season = NULL, round_number = NULL, comp = "AFLM", source = "AFL")
```
fetch_ladder_afl

Arguments

season  season in YYYY format
round_number  round number
comp  One of "AFLM" or "AFLW"
source  One of "AFL" (default), "Footywire", "AFLTables"

Value

returns a dataframe with the fixture that matches season, round.

Examples

## Not run:
get_ladder(2020, round = 1)
## End(Not run)
footywire_html

Helper function for get_footywire_stats

Description

Helper function for get_footywire_stats

Usage

footywire_html(x, id)

Arguments

x            URL of the match
id           Match ID number

Value

A data frame with advanced player results

get_afltables_stats

Return afltables match stats

Description

get_afltables_stats returns a data frame containing match stats for each game within the specified date range

Usage

get_afltables_stats(start_date = "1897-01-01", end_date = Sys.Date())

Arguments

start_date    character string for start date return to URLs from, in "dmy" or "ymd" format
end_date      optional, character string for end date to return URLS, in "dmy" or "ymd" format

Details

This function returns a data frame containing match stats for each game within the specified date range. The data from contains all stats on afltables match pages and returns 1 row per player.

The data for this function is hosted on github to avoid extensive scraping of historical data from afltables.com. This will be updated regularly.
get_afltables_urls

Value

a data table containing player stats for each game between start date and end date

Examples

#
## Not run:
# Gets all data
get_afltables_stats()
# Specify a date range
get_afltables_stats("01/01/2018", end_date = "01/04/2018")
## End(Not run)

get_afltables_urls Return match URLs for specified dates

Description

get_afltables_urls returns a character vector containing match URLs for the specified date range

Usage

get_afltables_urls(start_date, end_date = Sys.Date())

Arguments

start_date character string for start date return to URLs from, in "dmy" or "ymd" format

end_date optional, character string for end date to return URLs, in "dmy" or "ymd" format

Details

This function returns match URLs for the specified date range. This will typically be used to pass to to scrape_afltables_match to return player match results.

Value

a character vector of match URL’s between start_date and end_date

Examples

## Not run:
get_afltables_urls("01/01/2018", end_date = "01/04/2018")
## End(Not run)
get_aflw_cookie

Get AFL Stats cookie (internal function)

Description


Usage

get_aflw_cookie()

Value

token code

Examples

## Not run:
cookie <- get_aflw_cookie()

## End(Not run)

get_aflw_detailed_data

Get detailed AFLW data

Description

Get detailed AFLW data

Usage

get_aflw_detailed_data(matchids)

Arguments

matchids vector of match IDs, like those returned by get_aflw_match_data()

Value

Dataframe with detailed match data. Each row is a match.

Examples

## Not run:
get_aflw_detailed_data(c("CD_M20172640101", "CD_M20172640102"))

## End(Not run)
get_aflw_detailed_match_data

Get detailed womens match data (internal function)

Description

Gets detailed match data for a given match. Requires the match, round, and competition IDs, which are given in the tables produced by get_aflw_round_data()

Usage

get_aflw_detailed_match_data(matchid, roundid, competitionid, cookie)

Arguments

matchid matchid from get_match_data()
roundid roundid from get_match_data()
competitionid competitionid from get_match_data()
cookie cookie from get_womens_cookie()

Value

Dataframe with detailed match data (wide)

Examples

## Not run:
get_aflw_detailed_match_data(
  "CD_M20172640101",
  "CD_R201726401", "CD_S2017264", get_aflw_cookie()
)
## End(Not run)

get_aflw_match_data

Get AFLW match data

Description

Retrieves AFLW match data for all available matches. Sources data from https://womens.afl/

Usage

get_aflw_match_data(start_year = 2017)
get_aflw_player_stats

Arguments

start_year  optional, integer for start year to return match data onwards from

Value

a data frame of data for all available AFLW matches

Examples

## Not run:
# All data
get_aflw_match_data()

# 2018 data onward
get_aflw_match_data(start_year = 2018)

## End(Not run)

get_aflw_player_stats  Return get match stats for all current AFLW matches

Description

get_aflw_player_stats returns a data frame containing match stats for each game within the specified date range

Usage

get_aflw_player_stats(
  start = 2017,
  end = as.numeric(format(Sys.Date(), "%Y"))
)

Arguments

start  optional, character string or numeric for start year, in "YYYY"ormat
end  optional, character string or numeric for end year, in "YYYY"ormat

Details

This function returns a data frame containing match stats for each game within the specified date range. Returns 1 row per player.

The date for this function is called from an API with data stored in a PostgreSQL database on AWS. Updated at the conclusion of every game. A cached version to come.

Value

a data table containing player stats for each game between start and end years
Examples

```r
# Not run:
# Gets all data
get_aflw_player_stats()
# Specify a date range
get_aflw_player_stats(start = 2018, end = 2019)
```

## End(Not run)

---

**get_aflw_rounds**

*Get rounds (internal function)*

### Description

Returns data frame for available round data. Includes the rounds played, as well as identifiers to make further requests, importantly the roundId.

### Usage

```r
get_aflw_rounds(cookie)
```

### Arguments

- **cookie**
  - a cookie produced by `get_aflw_cookie()`

### Value

A dataframe with information about each round

### Examples

```r
# Not run:
get_aflw_rounds(get_aflw_cookie())
```

## End(Not run)
**get_aflw_round_data**  
*Get match data (internal function)*

**Description**

For a given round ID, get the data for each match played in that round. Use the column `roundId` in the dataframe created by the `get_rounds()` function to specify matches to fetch.

**Usage**

```r
get_aflw_round_data(roundid, cookie)
```

**Arguments**

- `roundid` a round ID string
- `cookie` a cookie produced by `get_womens_cookie()`

**Value**

a dataframe containing match data

**Examples**

```r
## Not run:
get_aflw_round_data("CD_R201826401", get_aflw_cookie())
## End(Not run)
```

**get_afl_colour_palettes**

*Returns a table with the colour palettes for all teams*

**Description**

`get_afl_colour_palettes` returns a dataframe containing the AFL team’s primary, secondary and tertiary colours as applicable. The data for this function is hosted on github.

**Usage**

```r
get_afl_colour_palettes()
```

**Value**

a data table containing team long name, team abbreviation, and colours
Examples

```r
## Not run:
# Gets all data
get_afl_colour_palettes()

## End(Not run)
```

---

### get_afl_cookie

**Get AFL Stats cookie (internal function)**

**Description**


**Usage**

```r
get_afl_cookie()
```

**Value**

token code

**Examples**

```r
## Not run:
cookie <- get_afl_cookie()

## End(Not run)
```

---

### get_afl_fixture

**Get AFL fixture**

**Description**

Returns the Fixture for the relevant Season and Round from the AFL.com.au website.

**Usage**

```r
get_afl_fixture(season = NULL, round_number = NULL, comp = "AFLM")
```

**Arguments**

- `season`: season in YYYY format
- `round_number`: round number
- `comp`: One of "AFLM" or "AFLW"
Value

returns a dataframe with the fixture that matches season, round.

Examples

```r
## Not run:
get_afl_fixture(2020, round = 1)

## End(Not run)
```

---

Description

`get_fixture` returns a dataframe containing upcoming AFL Men’s season fixture.

Usage

```r
get_fixture(season = lubridate::year(Sys.Date()), convert_date = FALSE)
```

Arguments

- `season`: Season to return, in yyyy format
- `convert_date`: logical, if TRUE, converts date column to date format instead of date time.

Details

The dataframe contains the home and away team as well as venue.

Value

Returns a data frame containing the date, teams and venue of each game

Examples

```r
## Not run:
get_fixture(2018)

## End(Not run)
```
get_footywire_betting_odds

Get AFL match betting odds from https://www.footywire.com

Description

get_footywire_betting_odds returns a data frame containing betting odds and basic match info for Men’s AFL matches.

Usage

get_footywire_betting_odds(
  start_season = "2010",
  end_season = lubridate::year(Sys.Date())
)

Arguments

start_season  First season to return, in yyyy format. Earliest season with data available is 2010.

end_season    Last season to return, in yyyy format

Details

The data frame contains the home and away team as well as venue.

Value

Returns a data frame containing betting odds and basic match info

Examples

## Not run:
get_footywire_betting_odds(2012, 2018)

## End(Not run)
get_footywire_match_results

Get footywire Match Results

Description

Returns the results of matches played in a particular season. You can limit how many results you return with the last_n_results parameter.

Usage

get_footywire_match_results(season, last_n_matches = NULL)

Arguments

  season  season to return results for
  last_n_matches  number of matches to return, starting from the most recent

Details

For example - you might just want to return the results from last round so you’d set last_n_results = 9.

If you want to return a large amount of results, it is more efficient to use get_match_results() however this can sometimes take some time to update the latest rounds results.

Value

Returns a data frame of match results from the year and number of results

Examples

## Not run:
get_footywire_match_results(2020, last_n_matches = 5)

## End(Not run)

get_footywire_stats  Scrape footywire player statistics.

Description

get_footywire_stats returns a dataframe containing player match stats from footywire from 2010 onwards.
**get_fryzigg_stats**

Return get match stats from fryziggafl.net/api/

---

**Description**

get_fryzigg_stats returns a data frame containing match stats for each game within the specified date range.

**Usage**

```r
get_fryzigg_stats(start = 1897, end = as.numeric(format(Sys.Date(), "%Y")))
```

**Arguments**

- `start` : optional, character string or numeric for start year, in "YYYY" format.
- `end` : optional, character string or numeric for end year, in "YYYY" format.

**Details**

This function returns a data frame containing match stats for each game within the specified date range. The data from contains all stats from the fryziggafl api and returns 1 row per player. The data for this function is called from an API with data stored in a PostgreSQL database on AWS. Updated at the conclusion of every game. A cached version to come.
**get_match_data**

**Value**

A data table containing player stats for each game between start and end years.

**Examples**

```r
# Not run:
get_fryzigg_stats()
# Specify a date range
get_fryzigg_stats(start = 2018, end = 2019)
```

---

**get_match_data**

*Helper function for get_footywire_stats*

**Description**

Helper function for get_footywire_stats.

**Usage**

```r
get_match_data(id)
```

**Arguments**

- `id` A match id from afltables.

---

**get_match_results**

*Get basic match results from afltables.com*

**Description**

get_match_results returns a dataframe containing all match results from 1897-current.

**Usage**

```r
get_match_results()
```

**Details**

The dataframe contains information about the Date, teams involved, scores and venue. It comes from afltables 'big lists' section. This is a limited dataset but is very fast to access. It generally is updated on the day after the last game.
**Value**

Returns a data frame containing a line for each match

**Examples**

```r
## Not run:
geet_match_results()

## End(Not run)
```

---

**get_score_progression_raw**

*Get raw score progression data*

---

**Description**

`get_score_progression_raw` returns a dataframe raw, unprocessed scoring progression data from afltables.

**Usage**

```r
get_score_progression_raw()
```

**Details**

The data is unprocessed and unstructured but is a starting point for analysis. It only exists for 2010 to 2017.

**Value**

Returns a data frame containing raw score progression data

**Examples**

```r
## Not run:
geet_score_progression_raw()

## End(Not run)
```
get_squiggle_data

Access Squiggle data using the squiggle API service.

Description

Use get_squiggle_data to access the Squiggle API. See instructions at api.squiggle.com.au.

Usage

get_squiggle_data(
  query = c("sources", "games", "tips", "ladder", "standings"),
  ...
)

Arguments

query       A text string. The main query to use with the API. Must be one of sources, games, tips, ladder or standings
...         (optional) An optional argument provided to the Squiggle API. See details for more info.

Details

The optional arguments to squiggle can be one of the following.

• year: an integer specifying the year to return data from, e.g. year = 2018
• round: an integer specifying the round to return data from, e.g. round = 12
• game: an integer specifying the game ID to return data from, e.g. game = 10
• source: an integer specifying the ID of the source to return data from, e.g. source = 1

For full instructions, see api.squiggle.com.au

Value

A dataframe, with the resultant data that matches the query specified in query, as well as any optional filters.

Examples

## Not run:
# Return a list of the sources, with ID's
sources <- get_squiggle_data("sources")

# Get tips for Round 1, 2018
tips <- get_squiggle_data(query = "tips", round = 1, year = 2018)
# Get tips from Squiggle 2019
squiggle <- get_squiggle_data(query = "tips", source = 1, year = 2019)

## End(Not run)

### replace_teams

**Internal function to ensure names match between different sources and also name changes. This gets applied to any web scraper**

**Description**

Internal function to ensure names match between different sources and also name changes. This gets applied to any web scraper

**Usage**

```r
replace_teams(team)
```

**Arguments**

- `team` Team name

### replace_venues

**Internal function to ensure venue names match between different sources and also name changes across time. This gets applied to any web scraper, transforming all of them to AFL Tables naming conventions.**

**Description**

Internal function to ensure venue names match between different sources and also name changes across time. This gets applied to any web scraper, transforming all of them to AFL Tables naming conventions.

**Usage**

```r
replace_venues(venue)
```

**Arguments**

- `venue` Venue name
return_ladder

Recreate the ladder for every or any given round and/or season

Description

return_ladder returns a dataframe containing the ladder for either all seasons and rounds since 1987, or individual rounds/seasons

Usage

return_ladder(match_results_df = NA, season_round = NA, season = NA)

Arguments

match_results_df
   A dataframe that has been returned from get_match_results. If empty get_match_results will execute first
season_round
   An integer of the round or vector of integers for multiple rounds. If empty, all rounds returned
season
   An integer of the season or vector of integers for multiple seasons. If empty, all seasons returned

Details

The dataframe contains information about the Round, Season, Points For/Against, Ladder Position. It can either take in a data frame created using get_match_results, or if match_results_df is unspecified, will extract all games using get_match_results. Will only allow selecting rounds of the premiership season, not finals.

Value

Returns a data frame containing a line for each team’s ladder position at each round of a season

Examples

## Not run:
return_ladder()
return_ladder(match_results_df=get_match_results_df, season_round=23, season=1990:2019)
return_ladder(season_round = 10, season = 2019)

## End(Not run)
scrape_afltables_match

*Return afltables player match stats*

**Description**

scrape_afltables_match returns a character vector containing match URLs for the specified date range.

**Usage**

```r
scrape_afltables_match(match_urls)
```

**Arguments**

- `match_urls`: A list of URL's for matches to scrape data from.

**Details**

This function returns the full afltables.com match stats for each player and each game specified in `match_urls`. It is useful to use the helper function `get_afltables_urls` to return these or simply navigate to afltables.com and find the match of interest.

**Value**

data table of afltables.com match results, with a row per player per match.

**Examples**

```r
## Not run:
scrape_afltables_match(get_afltables_urls("01/06/2018", "01/07/2018"))
## End(Not run)
```

update_footywire_stats

*Update the included footywire stats data to the specified date.*

**Description**

update_footywire_stats returns a dataframe containing player match stats from footywire.

**Usage**

```r
update_footywire_stats(check_existing = TRUE)
```
verify_year

Arguments

check_existing  A logical specifying if we should check against existing dataset. Defaults to TRUE. Making it false will download all data from all history which will take some time.

Details

The dataframe contains both basic and advanced player statistics from each match from 2010 to the specified end date.
This function utilised the included ID’s dataset to map known ID’s. It looks for any new data that isn’t already loaded and proceeds to download it.

Value

Returns a data frame containing player match stats for each match ID

Examples

## Not run:
update_footywire_stats()
## End(Not run)

---

verify_year  

*Returns year as numeric after verifying the validity of the year.*

Description

verify_year returns a numeric year, after to doing a validity check

Usage

verify_year(year)

Arguments

year  

character string or numeric for year, in "YYYY"format

Value

returns year as numeric

Examples

#  
## Not run:
verify_year(2012)
## End(Not run)
Index

convert_results, 3
fetch_ladder, 3
fetch_ladder_afl, 4
footywire_html, 5

get_afl_colour_palettes, 11
get_afl_cookie, 12
get_afl_fixture, 12
get_afltables_stats, 5
get_afltables_urls, 6
get_aflw_cookie, 7
get_aflw_detailed_data, 7
get_aflw_detailed_match_data, 8
get_aflw_match_data, 8
get_aflw_player_stats, 9
get_aflw_round_data, 11
get_aflw_rounds, 10
get_fixture, 13
get_footywire_betting_odds, 14
get_footywire_match_results, 15
get_footywire_stats, 15
get_fryzigg_stats, 16
get_match_data, 17
get_match_results, 17
get_score_progression_raw, 18
get_squiggle_data, 19

replace_teams, 20
replace_venues, 20
return_ladder, 21

scrape_afltables_match, 22
update_footywire_stats, 22

verify_year, 23