Package ‘fitzRoy’

January 10, 2022

Title Easily Scrape and Process AFL Data

Version 1.1.0

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

URL https://jimmyday12.github.io/fitzRoy/,
https://github.com/jimmyday12/fitzRoy

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), tidyr (>= 1.0.0), tidyselect, xml2, tibble, progress, glue, cli

Suggests covr, ggplot2, knitr, rmarkdown, testthat, roxygen2, spelling, curl

VignetteBuilder knitr

ByteCompile true

Encoding UTF-8

RoxygenNote 7.1.1

Language en-GB

Config/testthat/edition 3

Config/testthat/parallel true

Config/testthat/start-first fetch-player-stats, fetch-player-stats-legacy, fetch*

NeedsCompilation no

Author James Day [cre, aut],
Robert Nguyen [aut],
Matthew Erbs [ctb],
Oscar Lane [aut],
Jason Zivkovic [ctb],
Jacob Holden [ctb]

Maintainer James Day <jamesthomasday@gmail.com>

Repository CRAN

Date/Publication 2022-01-10 07:52:48 UTC

R topics documented:

calculate_coaches_vote_possibilities

Calculate Coaches Vote Possibilities

Description

`calculate_coaches_vote_possibilities` returns all possible breakdowns of coaches votes between two coaches, given a breakdown of AFLCA coaches votes

Usage

```
calculate_coaches_vote_possibilities(df, output_type)
```

Arguments

`df`	Requires the following column names: Player.Name, Coaches.Votes. These can be returned from the function `fetch_coaches_votes`.
`output_type`	One of "Coach View", "Player View". Defaults to "Coach View".

Value

For output_type "Coach View" - A list of data frames with columns: Votes, C1, C2 For output_type "Player View" - A list of data frames with columns: Player, V1, V2

Examples

```
## Not run:
# Return coaches votes for a particular match, then find the possibilities
df <- fetch_coaches_votes(comp = "AFLM", season = 2021, round = 24, team = "Western Bulldogs")
calculate_coaches_vote_possibilities(df, "Coach View")

df <- fetch_coaches_votes(comp = "AFLW", season = 2021, round = 9, team = "Western Bulldogs")
calculate_coaches_vote_possibilities(df, "Player View")

# Create a manual data frame to calculate possibilities
df <- data.frame(
  Player.Name = c(
    "Tom Liberatore", "Jack Macrae",
    "Marcus Bontempelli", "Cody Weightman",
    "Darcy Parish", "Aaron Naughton", "Jordan Ridley"
  ),
  Coaches.Votes = c(7, 6, 5, 5, 4, 2, 1)
)
calculate_coaches_vote_possibilities(df, "Player View")

## End(Not run)
```

`fetch_betting_odds_footywire` *Fetch AFL match betting odds from https://www.footywire.com*

Description

`fetch_betting_odds_footywire` returns a data frame containing betting odds and basic match info for Men's AFL matches.

Usage

```
fetch_betting_odds_footywire(
  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:
fetch_betting_odds_footywire(2012, 2018)

## End(Not run)
```

`fetch_coaches_votes` *Fetch Coaches Votes*

Description

`fetch_coaches_votes` returns all coaches votes for input season/s, round/s, and/or team's matches. The function calls a core `scrape_coaches_votes` function which scrapes the AFLCA website for coaches votes for a particular season, round and competition.

Usage

```
fetch_coaches_votes(
  season = NULL,
  round_number = NULL,
  comp = "AFLM",
  team = NULL
)
```

Arguments

`season`	Season in YYYY format. This can be an array of seasons. Defaults to null in which case the season that matches `Sys.Date()` is used.
`round_number`	Round number. For finals this is the number of H&A rounds plus the Finals week. Defaults to null in which case all rounds are used.
`comp`	One of "AFLM" (default) or "AFLW"
`team`	Team or teams whose matches should be retrieved. Defaults to null in which case all teams are used.

Value

A data frame with columns: Season, Round, Finals, Home.Team, Away.Team, Player.Name, Coaches.Votes

Examples

```
## Not run:
# Return all coaches votes across all seasons
fetch_coaches_votes(season = 2007:2021, comp = "AFLM")
fetch_coaches_votes(season = 2018:2021, comp = "AFLW")

# Return all coaches votes for a particular round
fetch_coaches_votes(season = 2021, round_number = 24, comp = "AFLM")
fetch_coaches_votes(season = 2021, round_number = 9, comp = "AFLW")

# Return all coaches votes for a particular team
fetch_coaches_votes(season = 2021, comp = "AFLM", team = "Western Bulldogs")
fetch_coaches_votes(season = 2021, comp = "AFLW", team = "Western Bulldogs")

# Return all coaches votes for a particular match
```

```
fetch_coaches_votes(season = 2021, round_number = 24, comp = "AFLM", team = "Western Bulldogs")
fetch_coaches_votes(season = 2021, round_number = 9, comp = "AFLW", team = "Western Bulldogs")

## End(Not run)
```

fetch_fixture *Return the fixture for a particular round of matches*

Description

fetch_fixture returns the Fixture for a given AFL Round. Internally, it calls a corresponding fetch_fixture_* function that depends on the source given. By default the source used will be the official AFL website.

fetch_fixture_afl(), fetch_fixture_footywire(), fetch_fixture_squiggle() can be called directly and return data from AFL website, AFL Tables and Squiggle, respectively.

Usage

```
fetch_fixture(
  season = NULL,
  round_number = NULL,
  comp = "AFLM",
  source = "AFL",
  ...
)

fetch_fixture_afl(season = NULL, round_number = NULL, comp = "AFLM")

fetch_fixture_footywire(
  season = NULL,
  round_number = NULL,
  convert_date = FALSE
)

fetch_fixture_squiggle(season = NULL, round_number = NULL)
```

Arguments

season	Season in YYYY format, defaults to NULL which returns the year corresponding the Sys.Date()
round_number	Round number, defaults to NULL which returns latest round
comp	One of "AFLM" (default) or "AFLW"
source	One of "AFL" (default), "footywire", "fryzigg", "afltables", "squiggle"
...	Optional parameters passed onto various functions depending on source.
convert_date	logical, if TRUE, converts date column to date format instead of date time.

Value

A Tibble with the fixture from the relevant season and round.

See Also

- fetch_fixture_afl for official AFL data.
- fetch_fixture_footywire for AFL Tables data.
- fetch_fixture_squiggle for Squiggle data.

Other fetch fixture functions: `fetch_player_stats()`

Examples

```
## Not run:
# Return data for whole season from AFL Website
fetch_fixture(2020)

# This is equivalent to
fetch_fixture(2020, source = "AFL")
fetch_fixture_afl(2020)

# Return AFLW data
fetch_fixture(2020, comp = "AFLW", source = "AFL")
fetch_fixture_afl(2020, comp = "AFLW")

# Not all sources have AFLW data and will return a warning
fetch_fixture(2020, comp = "AFLW", source = "footywire")
fetch_fixture(2020, comp = "AFLW", source = "squiggle")

# Different sources
fetch_fixture(2015, round = 5, source = "footywire")
fetch_fixture(2015, round = 5, source = "squiggle")

# Directly call functions for each source
fetch_fixture_afl(2018, round = 9)
fetch_fixture_footywire(2018, round = 9)
fetch_fixture_squiggle(2018, round = 9)

## End(Not run)
```

`fetch_ladder` *Fetch Ladder*

Description

`fetch_ladder` returns the Ladder for a given AFL Round. Internally, it calls a corresponding fetch_ladder_* function that depends on the source given. By default the source used will be the official AFL website.

fetch_ladder_afl(), fetch_ladder_afltables(), fetch_ladder_squiggle() can be called directly and return data from AFL website, AFL Tables and Squiggle, respectively.

Usage

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

fetch_ladder_afl(season = NULL, round_number = NULL, comp = "AFLM")

fetch_ladder_afltables(
  season = NULL,
  round_number = NULL,
  match_results_df = NULL
)

fetch_ladder_squiggle(season = NULL, round_number = NULL)
```

Arguments

season	Season in YYYY format, defaults to NULL which returns the year corresponding the Sys.Date()
round_number	Round number, defaults to NULL which returns latest round
comp	One of "AFLM" (default) or "AFLW"
source	One of "AFL" (default), "footywire", "fryzigg", "afltables", "squiggle"
...	Optional parameters passed onto various functions depending on source.
match_results_df	(optional) A dataframe from fetch_results_afltables(), provide this to prevent having to download results again.

Value

A Tibble with the ladder from the relevant season and round.

See Also

- fetch_ladder_afl for official AFL data.
- fetch_ladder_afltables for AFL Tables data.
- fetch_ladder_squiggle for Squiggle data.

Examples

```
## Not run:
# Return data from AFL Website
fetch_ladder(2020, round = 1)

# This is equivalent to
fetch_ladder(2020, round = 1, source = "AFL")
fetch_ladder_afl(2020, round = 1)

# Return AFLW data
fetch_ladder(2020, round = 1, comp = "AFLW", source = "AFL")
fetch_ladder_afl(2020, round = 1, comp = "AFLW")

# Not all sources have AFLW data and will return a warning
fetch_ladder(2020, round = 1, comp = "AFLW", source = "afltables")
fetch_ladder(2020, round = 1, comp = "AFLW", source = "squiggle")

# Different sources
fetch_ladder(2015, round = 5, source = "afltables")
fetch_ladder(2015, round = 5, source = "squiggle")

# Directly call functions for each source
fetch_ladder_afl(2018, round = 9)
fetch_ladder_afltables(2018, round = 9)
fetch_ladder_squiggle(2018, round = 9)

## End(Not run)
```

fetch_lineup *Return the selected lineup for any completed or upcoming matches*

Description

fetch_lineup returns the Lineup for matches in given AFL Round. Internally, it calls a corresponding fetch_lineup_* function that depends on the source given. By default the source used will be the official AFL website.

fetch_lineup_afl() can be called directly and return data from AFL website.

Usage

```
fetch_lineup(
  season = NULL,
  round_number = NULL,
  comp = "AFLM",
  source = "AFL",
  ...
)

fetch_lineup_afl(season = NULL, round_number = NULL, comp = "AFLM")
```

Arguments

`season`	Season in YYYY format, defaults to NULL which returns the year corresponding the `Sys.Date()`
`round_number`	Round number, defaults to NULL which returns latest round
`comp`	One of "AFLM" (default) or "AFLW"
`source`	One of "AFL" (default), "footywire", "fryzigg", "afltables", "squiggle"
`...`	Optional parameters passed onto various functions depending on source.

Value

A Tibble with the lineup from the relevant `season` and `round`.

See Also

- fetch_lineup_afl for official AFL data.

Examples

```
## Not run:
# Return data for whole season from AFL Website
fetch_lineup(2020)

# This is equivalent to
fetch_lineup(2020, source = "AFL")
fetch_lineup_afl(2020)

# Return AFLW data
fetch_lineup(2020, comp = "AFLW", source = "AFL")
fetch_lineup_afl(2020, comp = "AFLW")

# Not all sources have lineup data and will return a warning
fetch_lineup(2020, source = "footywire")
fetch_lineup(2020, source = "squiggle")


# Directly call functions for each source
fetch_lineup_afl(2018, round = 9)

## End(Not run)
```

`fetch_player_details` *Fetch Player Details*

Description

fetch_player_details returns player details such as date of birth, debut and other details. The exact details that are returned will depend on which source is provided.

By default the source used will be the official AFL website.

fetch_player_details_afl(), fetch_player_details_afltables() and fetch_player_details_footywire() can be called directly and return data from the AFL website, AFL Tables and Footywire respectively.

The function will typically be used to return the current team lists. For historical data, you can use the current argument set to FALSE. This will return all historical data for AFL.com and Footywire data. AFLTables data will always return historical data.

Usage

```
fetch_player_details(
  team = NULL,
  current = TRUE,
  comp = "AFLM",
  source = "AFL",
  ...
)

fetch_player_details_afl(season, team = NULL, comp = "AFLM")

fetch_player_details_afltables(team = NULL)

fetch_player_details_footywire(team, current = TRUE)
```

Arguments

team	team the player played for in the season for, defaults to NULL which returns all teams
current	logical, return the current team list for the current calendar year or all historical data
comp	One of "AFLM" (default) or "AFLW"
source	One of "AFL" (default), "footywire", "afltables"
...	Optional parameters passed onto various functions depending on source.
season	Season in YYYY format

Value

A Tibble with the details of the relevant players.

See Also

- fetch_player_details_afl for AFL.com data.
- fetch_player_details_footywire for Footywire data.
- fetch_player_details_footywire for AFL Tables data.

Examples

```
## Not run:
# Return data for current Hawthorn players
fetch_player_details("Hawthorn")
fetch_player_details("Adelaide", current = FALSE, comp = "AFLW")
fetch_player_details("GWS", current = TRUE, csource = "footywire")

## End(Not run)
```

fetch_player_stats *Fetch Player Stats*

Description

fetch_player_stats returns the Individual Player Statistics for AFL games. Internally, it calls a corresponding fetch_player_stats_* function that depends on the source given. By default the source used will be the official AFL website.

fetch_player_stats_footywire(), fetch_player_stats_afltables(), fetch_player_stats_fryzigg() can be called directly and return data from AFL website, AFL Tables and Squiggle, respectively.

Usage

```
fetch_player_stats(
  season = NULL,
  round_number = NULL,
  comp = "AFLM",
  source = "AFL",
  ...
)

fetch_player_stats_afl(season = NULL, round_number = NULL, comp = "AFLM")

fetch_player_stats_afltables(season = NULL, round_number = NULL)

fetch_player_stats_fryzigg(season = NULL, round_number = NULL, comp = "AFLM")

fetch_player_stats_footywire(
  season = NULL,
  round_number = NULL,
  check_existing = TRUE
)
```

Arguments

season Season in YYYY format, defaults to NULL which returns the year corresponding the Sys.Date()

`round_number` Round number, defaults to NULL which returns latest round

`comp` One of "AFLM" (default) or "AFLW"

`source` One of "AFL" (default), "footywire", "fryzigg", "afltables", "squiggle"

`...` Optional parameters passed onto various functions depending on source.

`check_existing` logical, should we check existing data. This will likely be removed in future version as it takes a long time to re-scrape data

Value

A Tibble with the player stats from the relevant `season` and `round`.

See Also

- fetch_player_stats_footywire for Footywire data.
- fetch_player_stats_afltables for AFL Tables data.
- fetch_player_stats_fryzigg for Fryzigg data.

Other fetch fixture functions: `fetch_fixture()`

Examples

```
## Not run:
# Return data for whole season from footywire
fetch_player_stats(source = "footywire")

# This is equivalent to
fetch_player_stats_footywire()

# Currently there is no AFLW data and will return a warning
fetch_player_stats(2020, comp = "AFLW", source = "footywire")

# Different sources
fetch_player_stats(2015, round = 5, source = "footywire")
fetch_player_stats(2015, round = 5, source = "fryzigg")

# Directly call functions for each source
fetch_player_stats_afltables(2020)
fetch_fixture_fryzigg(2020)
fetch_player_stats_footywire(2020)

## End(Not run)
```

fetch_results *Fetch Results*

Description

fetch_results returns the results for a given AFL Round. Internally, it calls a corresponding fetch_results_* function that depends on the source given. By default the source used will be the official AFL website.

fetch_results_afl(), fetch_results_afltables(), fetch_results_footywire(), fetch_results_squiggle() can be called directly and return data from AFL website, AFL Tables, Footywire and Squiggle, respectively.

Usage

```
fetch_results(
  season = NULL,
  round_number = NULL,
  comp = "AFLM",
  source = "AFL",
  ...
)

fetch_results_afl(season = NULL, round_number = NULL, comp = "AFLM")

fetch_results_afltables(season = NULL, round_number = NULL)

fetch_results_footywire(
  season = NULL,
  round_number = NULL,
  last_n_matches = NULL
)

fetch_results_squiggle(season = NULL, round_number = NULL)
```

Arguments

season	Season in YYYY format, defaults to NULL which returns the year corresponding the Sys.Date()
round_number	Round number, defaults to NULL which returns latest round
comp	One of "AFLM" (default) or "AFLW"
source	One of "AFL" (default), "footywire", "fryzigg", "afltables", "squiggle"
...	Optional parameters passed onto various functions depending on source.
last_n_matches	number of matches to return, starting from the most recent

Value

A Tibble with the results from the relevant season and round.

See Also

- fetch_results_afl for official AFL data.
- fetch_results_afltables for AFL Tables data.
- fetch_results_footywire for Footywire data.
- fetch_results_squiggle for Squiggle data.

Examples

```
## Not run:
# Return data for whole season from AFL Website
fetch_results(2020)

# This is equivalent to
fetch_results(2020, source = "AFL")
fetch_results_afl(2020)

# Return AFLW data
fetch_results(2020, comp = "AFLW", source = "AFL")
fetch_results_afl(2020, comp = "AFLW")

# Not all sources have AFLW data and will return a warning
fetch_results(2020, comp = "AFLW", source = "footywire")
fetch_results(2020, comp = "AFLW", source = "afltables")
fetch_results(2020, comp = "AFLW", source = "squiggle")

# Different sources
fetch_results(2015, round = 5, source = "footywire")
fetch_results(2015, round = 5, source = "afltables")
fetch_results(2015, round = 5, source = "squiggle")

# Directly call functions for each source
fetch_results_afl(2018, round = 9)
fetch_results_footywire(2018, round = 9)
fetch_results_afltables(2018, round = 9)
fetch_results_squiggle(2018, round = 9)

## End(Not run)
```

fetch_squiggle_data *Access Squiggle data using the squiggle API service.*

Description

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

Usage

```
fetch_squiggle_data(
  query = c("teams", "sources", "games", "tips", "ladder", "standings", "virtual",
     "pav"),
   ...
)
```

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)
```

`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.

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_aflw_cookie *Get AFL Stats cookie (internal function)*

Description

Gets a cookie from http://www.afl.com.au/womens/matches/stats to authenticate further requests.

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://www.womens.afl/`

Usage

```
get_aflw_match_data(start_year = 2017)
```

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" 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. 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

```
#
## 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

```
get_aflw_rounds(cookie)
```

Arguments

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

Value

A dataframe with information about each round

Examples

```
## 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

```
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

```
## 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 data frame containing the AFL team's primary, secondary and tertiary colours as applicable The data for this function is hosted on github.

Usage

```
get_afl_colour_palettes()
```

Value

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

Examples

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

## End(Not run)
```

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

Description

Gets a cookie from http://www.afl.com.au/ to authenticate further requests.

Usage

```
get_afl_cookie()
```

Value

token code

Examples

```
## 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

```
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

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

## End(Not run)
```

get_fixture *Get upcoming fixture from https://www.footywire.com*

Description

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

Usage

```
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

```
## 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.

Usage

```
get_footywire_stats(ids)
```

Arguments

`ids`	A vector containing match id's to return. Can be a single value or vector of values.

Details

The dataframe contains both basic and advanced player statistics from each match specified in the match_id input. To find match ID, find the relevant matches on https://www.footywire.com

Value

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

Examples

```
## Not run:
get_footywire_stats(ids = 5000:5100)

## End(Not run)
```

`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

```
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 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

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

## End(Not run)
```

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

```
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

```
## Not run:
get_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

```
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

```
## Not run:
get_score_progession_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("teams", "sources", "games", "tips", "ladder", "standings", "virtual",
     "pav"),
   ...
)
```

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

```
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

```
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)
```

`team_abr_afl` *Internal function to return team name abbreviation for AFL API*

Description

Internal function to return team name abbreviation for AFL API

Usage

```
team_abr_afl(team)
```

Arguments

`team`	Team name

`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

```
update_footywire_stats(check_existing = TRUE)
```

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)
```

Index