

Package ‘fitzRoy’

August 23, 2026

Title Easily Scrape and Process AFL Data

Version 1.8.0

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 MIT + file LICENSE

URL <https://jimmyday12.github.io/fitzRoy/>,
<https://github.com/jimmyday12/fitzRoy>,
<https://github.com/jimmyday12/fitzroy>

BugReports <https://github.com/jimmyday12/fitzRoy/issues>

Depends R (>= 4.1)

Imports dplyr, httr, jsonlite, lubridate, magrittr, purrr, readr,
rlang (>= 0.1.2), rvest, stringr (>= 1.3.0), tidyr (>= 1.0.0),
tidyselect, xml2, tibble, glue, cli, lifecycle, httr2, janitor,
nanoparquet

Suggests covr, elo, ggplot2, knitr, rmarkdown, testthat (>= 3.0.0),
roxygen2, spelling, curl

VignetteBuilder knitr

ByteCompile true

Encoding UTF-8

RoxygenNote 7.3.2

Language en-US

Config/testthat/edition 3

Config/testthat/parallel true

Config/testthat/start-first fetch-player-stats, fetch-results,
fetch-lineup, fetch-fixture, fetch*, helpers-footywire

LazyData true

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 2026-08-23 10:41:02 UTC

Contents

calculate_coaches_vote_possibilities	3
fetch_awards	4
fetch_awards_allaustralian	4
fetch_awards_brownlow	5
fetch_betting_odds_footywire	5
fetch_coaches_votes	6
fetch_fantasy_scores	7
fetch_fixture	8
fetch_ladder	9
fetch_lineup	11
fetch_outofcontract	13
fetch_player_details	13
fetch_player_stats	15
fetch_results	17
fetch_rising_star	19
fetch_scores	20
fetch_score_worm_data	20
fetch_squiggle_data	21
fetch_supercoach_scores	22
fetch_team_stats	22
get_aflw_detailed_data	23
get_aflw_detailed_match_data	23
get_aflw_rounds	24
get_aflw_round_data	25
get_afl_colour_palettes	25
get_afl_cookie	26
get_score_progression_raw	26
parse_team_abbr	27
plot_score_worm	27
plot_score_worm_totals	28
replace_teams	28
replace_venues	29
team_abr_afl	29

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

Data frame 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_awards	<i>Fetch AFL Awards Data</i>
--------------	------------------------------

Description

General wrapper to fetch Brownlow, All-Australian, or Rising Star awards from Footywire.

Usage

```
fetch_awards(..., award = c("brownlow", "allaustrian", "risingstar"))
```

Arguments

- ... Additional arguments passed to the specific award fetcher.
- award Character. One of "brownlow", "allaustrian", or "risingstar".

Value

A data frame containing the requested award data.

Examples

```
## Not run:
fetch_awards(2024, award = "brownlow", type = "player")
fetch_awards(2023, award = "allaustrian", type = "team")
fetch_awards(2024, award = "risingstar", type = "nominations")

## End(Not run)
```

fetch_awards_allaustralian	<i>Fetch AFL All-Australian Team or Squad</i>
----------------------------	---

Description

Fetch AFL All-Australian Team or Squad

Usage

```
fetch_awards_allaustralian(season, type = c("team", "squad"))
```

Arguments

- season A single year (e.g., 2023)
- type Either "team" (final 22) or "squad" (initial 44)

Value

A tibble with player/team info

fetch_awards_brownlow *Fetch Brownlow Medal Data*

Description

Fetch Brownlow Medal Data

Usage

```
fetch_awards_brownlow(season, type = c("player", "team"))
```

Arguments

season	A single numeric year (e.g., 2024).
type	Either "player" (default) or "team".

Value

A tibble with Brownlow vote data.

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	<i>Fetch Coaches Votes</i>
---------------------	----------------------------

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_fantasy_scores *Fetch AFL Fantasy (Dream Team) Scores*

Description

Fetch AFL Fantasy (Dream Team) Scores

Usage

```
fetch_fantasy_scores(year = 2025, rounds = 1:30)
```

Arguments

year	Integer. AFL season year.
rounds	Integer vector. Rounds to fetch (default: 1:30).

Value

A data frame of Dream Team scores.

fetch_fixture	<i>Return the fixture for a particular round of matches</i>
---------------	---

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), "AFLW", "VFL", "VFLW", "WAFL", "U18B" or "U18G." Not all data sources will have non-AFL data
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), "AFLW", "VFL", "VFLW", "WAFL", "U18B" or "U18G." Not all data sources will have non-AFL data
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), "AFLW", "VFL", "VFLW", "WAFL", "U18B" or "U18G." Not all data sources will have non-AFL data
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_outofcontract	<i>Fetch Out of Contract AFL Players</i>
---------------------	--

Description

fetch_outofcontract returns a list of AFL players out of contract for a specified year. It currently uses FootyWire as the data source.

Usage

```
fetch_outofcontract(year = 2026, source = "footywire", ...)
```

Arguments

year	Numeric. Year to fetch out of contract players (e.g. 2026).
source	Data source. Default "footywire".
...	Additional arguments passed to the source-specific function.

Value

A tibble with Player, Years_Service, Status, and Club columns.

fetch_player_details	<i>Fetch Player Details</i>
----------------------	-----------------------------

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 `current = 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,
  season = NULL,
  current = TRUE,
  comp = "AFLM",
  source = "AFL",
  player = NULL,
  player_id = NULL,
  match = c("exact", "regex", "fuzzy"),
  ...
)

fetch_player_details_afl(
  season = NULL,
  team = NULL,
  current = TRUE,
  comp = "AFLM",
  official_teams = FALSE
)

fetch_player_details_afltables(team = NULL)

fetch_player_details_footywire(team = NULL, current = TRUE)

```

Arguments

team	team the player played for in the season for, defaults to NULL which returns all teams
season	Season in YYYY format
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"
player	Character vector (optional). Filter by player name (exact/regex/fuzzy via match).
player_id	Character or numeric vector (optional). Filter by player ID (if an ID column is present).
match	One of "exact", "regex", or "fuzzy". Controls how player is matched. Default "exact".
...	Optional parameters passed onto various functions depending on source.
official_teams	boolean, defaults to FALSE. Indicates if we should match team to the official list from the API. If this is TRUE, it will use the list from the API and you can use fetch_teams_afl to see what these names should be

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_afltables](#) 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, source = "footywire")

## End(Not run)
```

fetch_player_stats	<i>Fetch Player Stats</i>
--------------------	---------------------------

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",
  player = NULL,
  player_id = NULL,
  match = c("exact", "regex", "fuzzy"),
  ...
)

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

fetch_player_stats_afltables(
  season = NULL,
  round_number = NULL,
  rescrape = FALSE,
```

```

    rescrape_start_season = 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), "AFLW", "VFL", "VFLW", "WAFL", "U18B" or "U18G." Not all data sources will have non-AFL data
source	One of "AFL" (default), "footywire", "fryzigg", "afltables", "squiggle"
player	Character vector. Filter by player name (exact/regex/fuzzy via match).
player_id	Character or numeric vector. Filter by player ID (if an ID column is present).
match	One of "exact", "regex", or "fuzzy". Controls how player is matched. Default "exact".
...	Optional parameters passed onto various functions depending on source.
rescrape	Logical, defaults to FALSE. Determines if we should re-scrape data for a given season. By default, we return cached data which is much faster. Re-scraping is slow but sometimes needed if historical data has changed.
rescrape_start_season	Numeric, if rescrape = TRUE, which season should we start scraping from. Defaults to minimum value of season
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 all rounds
comp	One of "AFLM" (default), "AFLW", "VFL", "VFLW", "WAFL", "U18B" or "U18G." Not all data sources will have non-AFL data
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_rising_star	<i>Fetch AFL Rising Star Nominations or Stats</i>
-------------------	---

Description

Fetch AFL Rising Star Nominations or Stats

Usage

```

fetch_rising_star(
  season,
  round_number = NULL,
  type = c("nominations", "stats")
)

```

Arguments

season	Integer. The year of interest (e.g. 2024).
round_number	Integer. Optional. If NULL and type = "stats", scrapes all rounds.
type	Character. Either "nominations" (default) or "stats".

Value

A tibble with Rising Star data.

Examples

```

## Not run:
fetch_rising_star(2024, type = "nominations")
fetch_rising_star(2024, round_number = 5, type = "stats")
fetch_rising_star(2024, type = "stats")

## End(Not run)

```

fetch_scores	<i>Fetch Supercoach or Dream Team Scores</i>
--------------	--

Description

Wrapper to fetch either Supercoach or AFL Fantasy (Dream Team) scores from Footywire.

Usage

```
fetch_scores(type = c("supercoach", "dream_team"), ...)
```

Arguments

type	Character. Either "supercoach" or "dream_team".
...	Additional arguments passed to the score fetchers (e.g., year, rounds).

Value

A data frame of scores.

Examples

```
## Not run:
fetch_scores(type = "supercoach", year = 2025, rounds = 1:3)
fetch_scores(type = "dream_team", year = 2025, rounds = 1:3)

## End(Not run)
```

fetch_score_worm_data	<i>Plot Score Worm</i>
-----------------------	------------------------

Description

This function plots the score difference score worms for AFL games.

Usage

```
fetch_score_worm_data(match_id)
```

Arguments

match_id	AFL match ID (providerId) can be found using <code>fetch_fixture_afl()</code>
----------	---

Value

A ggplot object showing the score worm.

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

Description

Use `fetch_squiggle_data` to access the [Squiggle API](https://api.squiggle.com.au). See instructions at api.squiggle.com.au.

Usage

```
fetch_squiggle_data(  
  query,  
  ...,  
  user_agent = "fitzRoy Package https://github.com/jimmyday12/fitzRoy"  
)
```

Arguments

<code>query</code>	A text string. The main query to use with the API. Please read the Squiggle documentation for information about valid queries
<code>...</code>	(optional) An optional argument provided to the Squiggle API . See details for more info.
<code>user_agent</code>	(optional) Use this to set something meaningful so that Squiggle admin can contact you if needed.

Details

Optional arguments can be provided to further restrict the data you are pulling.

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 <- fetch_squiggle_data("sources")  
  
# Get tips for Round 1, 2018  
tips <- fetch_squiggle_data(query = "tips", round = 1, year = 2018)  
  
# Get tips from Squiggle 2019  
squiggle <- fetch_squiggle_data(query = "tips", source = 1, year = 2019)  
  
## End(Not run)
```

fetch_supercoach_scores	<i>Fetch Supercoach Scores</i>
-------------------------	--------------------------------

Description

Fetch Supercoach Scores

Usage

```
fetch_supercoach_scores(year = 2025, rounds = 1:30)
```

Arguments

year	Integer. AFL season year.
rounds	Integer vector. Rounds to fetch (default: 1:30).

Value

A data frame of Supercoach scores.

fetch_team_stats	<i>Fetch Team Statistics</i>
------------------	------------------------------

Description

General wrapper for fetching team statistics from a specified source.

Usage

```
fetch_team_stats(  
  season,  
  summary_type = "totals",  
  source = c("afltables", "footywire", "vflstats"),  
  comp = NULL,  
  ...  
)
```

Arguments

season	Integer. The season to fetch stats for (e.g. 2024).
summary_type	Character. Either "totals" (default), "averages", or other type depending on source.
source	Character. One of "afltables" (default), "footywire", or "vflstats".
comp	Character. Competition code for vflstats. Either "VFLM" or "VFLW".
...	Additional arguments passed to the underlying data source function.

Value

A data frame of team stats for the season.

`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_rounds	<i>Get rounds (internal function)</i>
-----------------	---------------------------------------

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	<i>Get match data (internal function)</i>
---------------------	---

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	<i>Get AFL Stats cookie (internal function)</i>
----------------	---

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_score_progression_raw	<i>Get raw score progression data</i>
---------------------------	---------------------------------------

Description**[Deprecated]**

This function has been deprecated due to its inefficiency

Usage

```
get_score_progression_raw()
```

Examples

```
#  
## Not run:  
get_score_progression_raw()  
  
## End(Not run)
```

parse_team_abbr	<i>Internal function to ensure names match between different sources and also name changes. This gets applied to any web scraper</i>
-----------------	--

Description

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

Usage

```
parse_team_abbr(team_name)
```

Arguments

team_name	Team name
-----------	-----------

plot_score_worm	<i>Plot Score Worm</i>
-----------------	------------------------

Description

This function plots the score difference score worms for AFL games.

Usage

```
plot_score_worm(match_id)
```

Arguments

match_id	Champion Data match_id (providerId) of the form CD_MSSSS014RRMM where SSSS is the Season, RR is the Round and MM is the Match. e.g. 'CD_M20240142004' - can be found using <code>fetch_fixture_afl()</code>
----------	--

Value

A ggplot object showing the score worm.

plot_score_worm_totals	<i>Plot Score Worm Totals</i>
------------------------	-------------------------------

Description

This function plots the team totals score worm for AFL games.

Usage

```
plot_score_worm_totals(match_id)
```

Arguments

match_id	Champion Data match_id (providerId) of the form CD_MSSSS014RRMM where SSSS is the Season, RR is the Round and MM is the Match. e.g. 'CD_M20240142004' - can be found using fetch_fixture_afl()
----------	---

Value

A ggplot object showing the total score worm.

replace_teams	<i>Internal function to ensure names match between different sources and also name changes. This gets applied to any web scraper</i>
---------------	--

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	<i>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.</i>
----------------	--

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
-------	------------

team_abr_afl	<i>Internal function to return team name abbreviation for AFL API</i>
--------------	---

Description

Internal function to return team name abbreviation for AFL API

Usage

```
team_abr_afl(team, return_id = FALSE)
```

Arguments

team	Team name
return_id	Should we return the team ID used by the API instead of the abbreviation?

Index

- * **fetch fixture functions**
 - fetch_fixture, [8](#)
 - fetch_player_stats, [15](#)
- * **fetch ladder functions**
 - fetch_ladder, [9](#)
- * **fetch lineup functions**
 - fetch_lineup, [11](#)
- * **fetch player details functions**
 - fetch_player_details, [13](#)
- * **fetch results functions**
 - fetch_results, [17](#)
- calculate_coaches_vote_possibilities, [3](#)
- fetch_awards, [4](#)
- fetch_awards_allaustralian, [4](#)
- fetch_awards_brownlow, [5](#)
- fetch_betting_odds_footywire, [5](#)
- fetch_coaches_votes, [6](#)
- fetch_fantasy_scores, [7](#)
- fetch_fixture, [8](#), [16](#)
- fetch_fixture_afl, [9](#)
- fetch_fixture_afl(fetch_fixture), [8](#)
- fetch_fixture_afl(), [8](#)
- fetch_fixture_footywire, [9](#)
- fetch_fixture_footywire(fetch_fixture), [8](#)
- fetch_fixture_footywire(), [8](#)
- fetch_fixture_squiggle, [9](#)
- fetch_fixture_squiggle(fetch_fixture), [8](#)
- fetch_fixture_squiggle(), [8](#)
- fetch_ladder, [9](#)
- fetch_ladder_afl, [10](#)
- fetch_ladder_afl(fetch_ladder), [9](#)
- fetch_ladder_afl(), [9](#)
- fetch_ladder_afltables, [10](#)
- fetch_ladder_afltables(fetch_ladder), [9](#)
- fetch_ladder_afltables(), [9](#)
- fetch_ladder_squiggle, [10](#)
- fetch_ladder_squiggle(fetch_ladder), [9](#)
- fetch_ladder_squiggle(), [9](#)
- fetch_lineup, [11](#)
- fetch_lineup_afl, [12](#)
- fetch_lineup_afl(fetch_lineup), [11](#)
- fetch_lineup_afl(), [11](#)
- fetch_outofcontract, [13](#)
- fetch_player_details, [13](#)
- fetch_player_details_afl, [15](#)
- fetch_player_details_afl(fetch_player_details), [13](#)
- fetch_player_details_afl(), [13](#)
- fetch_player_details_afltables, [15](#)
- fetch_player_details_afltables(fetch_player_details), [13](#)
- fetch_player_details_afltables(), [13](#)
- fetch_player_details_footywire, [15](#)
- fetch_player_details_footywire(fetch_player_details), [13](#)
- fetch_player_details_footywire(), [13](#)
- fetch_player_stats, [9](#), [15](#)
- fetch_player_stats_afl(fetch_player_stats), [15](#)
- fetch_player_stats_afltables, [16](#)
- fetch_player_stats_afltables(fetch_player_stats), [15](#)
- fetch_player_stats_afltables(), [15](#)
- fetch_player_stats_footywire, [16](#)
- fetch_player_stats_footywire(fetch_player_stats), [15](#)
- fetch_player_stats_footywire(), [15](#)
- fetch_player_stats_fryzigg, [16](#)
- fetch_player_stats_fryzigg(fetch_player_stats), [15](#)
- fetch_player_stats_fryzigg(), [15](#)
- fetch_results, [17](#)
- fetch_results_afl, [18](#)
- fetch_results_afl(fetch_results), [17](#)

fetch_results_afl(), [17](#)
fetch_results_afltables, [18](#)
fetch_results_afltables
 (fetch_results), [17](#)
fetch_results_afltables(), [10](#), [17](#)
fetch_results_footywire, [18](#)
fetch_results_footywire
 (fetch_results), [17](#)
fetch_results_footywire(), [17](#)
fetch_results_squiggle, [18](#)
fetch_results_squiggle(fetch_results),
 [17](#)
fetch_results_squiggle(), [17](#)
fetch_rising_star, [19](#)
fetch_score_worm_data, [20](#)
fetch_scores, [20](#)
fetch_squiggle_data, [21](#)
fetch_supercoach_scores, [22](#)
fetch_team_stats, [22](#)

get_afl_colour_palettes, [25](#)
get_afl_cookie, [26](#)
get_aflw_detailed_data, [23](#)
get_aflw_detailed_match_data, [23](#)
get_aflw_round_data, [25](#)
get_aflw_rounds, [24](#)
get_score_progression_raw, [26](#)

parse_team_abbr, [27](#)
plot_score_worm, [27](#)
plot_score_worm_totals, [28](#)

replace_teams, [28](#)
replace_venues, [29](#)

team_abr_afl, [29](#)