Package ‘worldfootballR’

November 25, 2021

Type  Package
Title  Functions to Extract and Clean World Football (Soccer) Data
Version  0.4.5
Description  A set of functions to allow users to obtain clean and tidy
football (soccer) game, team and player data. Data is collected from a
number of popular sites, including 'FBref'<https://fbref.com/en/>,
transfer and valuations data from
'Transfermarkt'<https://www.transfermarkt.com/> and shooting location
data from 'Understat'<https://understat.com/>. It gives users the
ability to access data more efficiently, rather than having to export
data tables to files before being able to complete their analysis.
License  GPL-3
BugReports  https://github.com/JaseZiv/worldfootballR/issues
Imports  dplyr, glue, janitor, jsonlite, lubridate, magrittr, progress,
purrr, qdapRegex, readr, rlang, rvest, stats, stringi, stringr,
tidyr, utils, withr, xml2
Suggests  knitr, rmarkdown, testthat
VignetteBuilder  knitr
Encoding  UTF-8
RoxygenNote  7.1.2
NeedsCompilation  no
Author  Jason Zivkovic [aut, cre, cph],
        Tony ElHabr [ctb]
Maintainer  Jason Zivkovic <jaseziv83@gmail.com>
Repository  CRAN
Date/Publication  2021-11-25 19:50:02 UTC
R topics documented:

- .clean_advanced_stat_table .................................................. 3
- .clean_match_advanced_stats_data ........................................... 3
- .clean_player_season_stats ................................................... 4
- .clean_table_names ............................................................... 4
- .convert_value_to_numeric .................................................... 5
- .get_clean_understat_json .................................................... 5
- .get_match_report_page ........................................................ 6
- .replace_empty_na ............................................................... 6
- .tm_fix_dates ................................................................. 7
- .understat_shooting ............................................................ 7
- fb_big5_advanced_season_stats ................................................. 8
- fb_league_urls ................................................................. 8
- fb_player_match_logs ........................................................... 9
- fb_player_scouting_report ..................................................... 10
- fb_player_season_stats ........................................................ 11
- fb_player_urls ................................................................. 11
- fb_teams_urls ................................................................. 12
- get_advanced_match_stats ..................................................... 13
- get_match_lineups ............................................................. 13
- get_match_report ............................................................... 14
- get_match_results .............................................................. 15
- get_match_shooting ............................................................. 16
- get_match_summary ............................................................. 16
- get_match_urls ................................................................. 17
- get_player_market_values ..................................................... 18
- get_season_team_stats ........................................................ 18
- get_team_match_results ....................................................... 19
- player_dictionary_mapping ................................................... 20
- player_transfer_history ...................................................... 20
- tm_league_team_urls ........................................................... 21
- tm_matchday_table ............................................................ 21
- tm_player_bio ................................................................. 22
- tm_squad_stats ................................................................. 23
- tm_team_player_urls .......................................................... 23
- tm_team_transfers ............................................................. 24
- tm_team_transfer_balances .................................................... 24
- understat_league_match_results .............................................. 25
- understat_league_season_shots ................................................. 25
- understat_match_shots ......................................................... 26
- understat_player_shots ......................................................... 26
- understat_team_meta ........................................................... 27
- understat_team_players_stats ............................................... 27
- understat_team_season_shots ................................................. 28
- understat_team_stats_breakdown ............................................ 28

Index ..................................................................................... 29
Clean advanced statistic tables

**Description**

Returns cleaned dataframe for each of the team statistic tables used by `get_season_team_stats()`

**Usage**

```
clean_advanced_stat_table(input_table_element)
```

**Arguments**

- `input_table_element`
  - element of the html table on the league season page

**Value**

- a data frame for the selected league seasons advanced statistic

---

Clean each match advanced statistic tables

**Description**

Returns cleaned data frame for each of the team statistic tables for each selected match

**Usage**

```
clean_match_advanced_stats_data(df_in)
```

**Arguments**

- `df_in`
  - a raw match stats data frame

**Value**

- a cleaned data frame for the selected match advanced statistic
.clean_table_names

Clean player season statistic tables

Description

Returns cleaned dataframe for each of the player statistic tables used by fb_player_season_stats()

Usage

.clean_player_season_stats(input_table_element)

Arguments

input_table_element
    element of the html table on the player page

Value

a data frame for the selected player’s advanced statistic

---

.clean_table_names  
Clean stat table column names

Description

Returns cleaned column names for stats tables

Usage

.clean_table_names(df_in)

Arguments

df_in         a raw match stats data frame

Value

a data frame with cleaned names
**.convert_value_to_numeric**

*Convert formatted valuations to numeric*

**Description**

Returns a numeric data type for player valuations

**Usage**

`.convert_value_to_numeric(euro_value)`

**Arguments**

- `euro_value` raw valuation from transfermarkt.com

**Value**

a cleaned numeric data value for market and/or transfer valuation

---

**.get_clean_understat_json**

*Clean Understat JSON data*

**Description**

Returns a cleaned Understat data frame

**Usage**

`.get_clean_understat_json(page_url, script_name)`

**Arguments**

- `page_url` understat.com page URL
- `script_name` html JSON script name

**Value**

a cleaned Understat data frame
.get_match_report_page

*Get match report for webpage*

**Description**

Internal function to return match report details for a selected match, with input being a webpage, not url.

**Usage**

```python
.get_match_report_page(match_page)
```

**Arguments**

- `match_page`: the fbref.com URL for the required match

**Value**

returns a dataframe with the match details for a selected match

---

.replace_empty_na

*Replace Empty Values*

**Description**

Returns a NA character for empty values

**Usage**

```python
.replace_empty_na(val)
```

**Arguments**

- `val`: a value that can either be empty, or not empty

**Value**

NA_character where the extracted value is empty, or the value itself
Clean date fields

Description
Returns a date format in YYYY-MM-DD from 'mmm d, yyyy'

Usage
.tm_fix_dates(dirty_dates)

Arguments
dirty_dates  formatted date value

Value
a cleaned date

Understat shots location helper function

Description
Returns a cleaned Understat shooting location data frame

Usage
.understat_shooting(type_url)

Arguments
type_url  can be season, team, match, player URL

Value
a cleaned Understat shooting location data frame
**fb_big5_advanced_season_stats**

*Big 5 Euro League Season Stats*

**Description**

Returns data frame of selected statistics for seasons of the big 5 Euro leagues, for either whole team or individual players. Multiple seasons can be passed to the function, but only one ‘stat_type’ can be selected.

**Usage**

```r
fb_big5_advanced_season_stats(season_end_year, stat_type, team_or_player)
```

**Arguments**

- `season_end_year`  
  the year(s) the season concludes
- `stat_type`  
  the type of team statistics the user requires
- `team_or_player`  
  result either summarised for each team, or individual players

The statistic type options (stat_type) include:

- "standard",
- "shooting",
- "passing",
- "passing_types",
- "gca",
- "defense",
- "possesion",
- "playing_time",
- "misc",
- "keepers",
- "keepers_adv"

**Value**

returns a dataframe of a selected team or player statistic type for a selected season(s)

**Examples**

```r
## Not run:
fb_big5_advanced_season_stats(season_end_year=2021, stat_type="possession", team_or_player="player")
## End(Not run)
```

---

**fb_league_urls**

*Get fbref League URLs*

**Description**

Returns the URLs for season leagues of a selected country

**Usage**

```r
fb_league_urls(country, gender, season_end_year, tier = "1st")
```
fb_player_match_logs

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>country</td>
<td>the three character country code</td>
</tr>
<tr>
<td>gender</td>
<td>gender of competition, either &quot;M&quot; or &quot;F&quot;</td>
</tr>
<tr>
<td>season_end_year</td>
<td>the year the season(s) concludes (defaults to all available seasons)</td>
</tr>
<tr>
<td>tier</td>
<td>the tier of the league, ie '1st' for the EPL or '2nd' for the Championship and so on</td>
</tr>
</tbody>
</table>

Value

returns a character vector of all fbref league URLs for selected country, season, gender and tier

Examples

    fb_league_urls(country = "ENG", gender = "M", season_end_year = 2021, tier = '1st')

fb_player_match_logs (Get fbref Player Match Logs)

Description

Returns all match logs for a selected player, season and stat type

Usage

    fb_player_match_logs(player_url, season_end_year, stat_type)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>player_url</td>
<td>the URL of the player (can come from fb_player_urls())</td>
</tr>
<tr>
<td>season_end_year</td>
<td>the year the season(s) concludes</td>
</tr>
<tr>
<td>stat_type</td>
<td>the type of statistic required</td>
</tr>
</tbody>
</table>

    The statistic type options (stat_type) include: "summary", "keepers", "passing", "passing_types", "gca", "defense", "possession", "misc"

Value

returns a dataframe of a player’s match logs for a season
Examples

```r
## Not run:
season_end_year = 2021, stat_type = 'summary')

## End(Not run)
```

---

fb_player_scouting_report

*Get fbref Full Player Scouting Report*

Description

Returns the scouting report for a selected player

Usage

`fb_player_scouting_report(player_url, pos_versus)`

Arguments

- `player_url` : the URL of the player (can come from `fb_player_urls()`)
- `pos_versus` : either “primary” or “secondary” as fbref offer comparisons against multiple positions

Value

returns a dataframe of a player’s full scouting information for all seasons available on FBref

Examples

```r
## Not run:
fb_player_scouting_report(player_url = "https://fbref.com/en/players/d70ce98e/Lionel-Messi",
pos_versus = "primary")

# to filter for the last 365 days:
fb_player_scouting_report(player_url = "https://fbref.com/en/players/d70ce98e/Lionel-Messi",
pos_versus = "primary") %>% dplyr::filter(scouting_period == "Last 365 Days")

# to get secondary positions
fb_player_scouting_report(player_url = "https://fbref.com/en/players/d70ce98e/Lionel-Messi",
pos_versus = "secondary")

# for the 2020-2021 La Liga season
fb_player_scouting_report(player_url = "https://fbref.com/en/players/d70ce98e/Lionel-Messi",
pos_versus = "secondary") %>% dplyr::filter(scouting_period == "2020-2021 La Liga")

## End(Not run)
```
fb_player_season_stats

Get fbref Player Season Statistics

Description

Returns the historical season stats for a selected player(s) and stat type

Usage

fb_player_season_stats(player_url, stat_type)

Arguments

player_url  the URL(s) of the player(s) (can come from fb_player_urls())
stat_type   the type of statistic required

The statistic type options (stat_type) include:
"standard", "shooting", "passing", "passing_types", "gca", "defense", "posses-
sion", "playing_time", "misc", "keeper", "keeper_adv"

Value

returns a dataframe of a player’s historical season stats

Examples

## Not run:
                    stat_type = "standard")

multiple_playing_time <- fb_player_season_stats(
player_url = c("https://fbref.com/en/players/d70ce98e/Lionel-Messi",
stat_type = "playing_time")

## End(Not run)

fb_player_urls

Get fbref Player URLs

Description

Returns the URLs for all players for a given team

Usage

fb_player_urls(team_url)
**Arguments**

- **team_url**: the player's team URL (can be from fb_team_urls())

**Value**

returns a character vector of all fbref player URLs for a selected team

**Examples**

```r
```

---

**fb_teams_urls**

*Get fbref Team URLs*

**Description**

Returns the URLs for all teams for a given league

**Usage**

```r
fb_teams_urls(league_url)
```

**Arguments**

- **league_url**: the league URL (can be from fb_league_urls())

**Value**

returns a character vector of all fbref team URLs for a selected league

**Examples**

```r
```
get_advanced_match_stats

Get advanced match stats

Description

Returns data frame of selected statistics for each match, for either whole team or individual players. Multiple URLs can be passed to the function, but only one 'stat_type' can be selected.

Usage

get_advanced_match_stats(match_url, stat_type, team_or_player)

Arguments

- **match_url**: the three character country code for all countries
- **stat_type**: the type of team statistics the user requires
- **team_or_player**: result either summarised for each team, or individual players
  The statistic type options (stat_type) include:
  - "summary"
  - "passing"
  - "passing"_types
  - "defense"
  - "possession"
  - "misc"
  - "keeper"

Value

returns a dataframe of a selected team statistic type for a selected match(es)

Examples

```r
# Not run:
urls <- get_match_urls(country = "AUS", gender = "F", season_end_year = 2021, tier = "1st")
get_advanced_match_stats(match_url=urls,stat_type="possession",team_or_player="player")
# End(Not run)
```

get_match_lineups

Get match lineups

Description

Returns lineups for home and away teams for a selected match

Usage

get_match_lineups(match_url)
Arguments

match_url  the fbref.com URL for the required match

Value

returns a dataframe with the team lineups for a selected match

Examples

```r
## Not run:
match <- get_match_urls(country = "AUS", gender = "F", season_end_year = 2021, tier = "1st")[[1]]
get_match_lineups(match_url = match)
## End(Not run)
```

---

## get_match_report

Get match report

Description

Returns match report details for selected matches

Usage

```r
get_match_report(match_url)
```

Arguments

match_url  the fbref.com URL for the required match

Value

returns a dataframe with the match details for a selected match

Examples

```r
## Not run:
match <- get_match_urls(country = "AUS", gender = "F", season_end_year = 2021, tier = "1st")[[1]]
get_match_report(match_url = match)
## End(Not run)
```
get_match_results

Get match results

Description

Returns the game results for a given league season(s)

Usage

get_match_results(
  country,
  gender,
  season_end_year,
  tier = "1st",
  non_dom_league_url = NA
)

Arguments

country the three character country code
gender gender of competition, either "M" or "F"
season_end_year the year(s) the season concludes
tier the tier of the league, ie '1st' for the EPL or '2nd' for the Championship and so on
non_dom_league_url the URL for Cups and Competitions found at https://fbref.com/en/comps/

Value

returns a dataframe with the results of the competition, season and gender

Examples

get_match_results(country = c("ITA"), gender = "M", season_end_year = 2021)
# for results from English Championship:
get_match_results(country = "ENG", gender = "M", season_end_year = 2021, tier = "2nd")
# for international friendlies:
non_dom <- "https://fbref.com/en/comps/218/history/Friendlies-M-Seasons"
get_match_results(country = "", gender = "M", season_end_year = 2018, non_dom_league_url = non_dom)
get_match_shooting  
*Get match shooting event data*

**Description**

Returns detailed player shooting data for home and away teams for a selected match(es)

**Usage**

```r
get_match_shooting(match_url)
```

**Arguments**

- `match_url`  
  the fbref.com URL for the required match

**Value**

returns a dataframe

**Examples**

```r
## Not run:
match <- "https://fbref.com/en/matches/bf52349b/Fulham-Arsenal-September-12-2020-Premier-League"
get_match_shooting(match_url = match)
## End(Not run)
```

get_match_summary  
*Get match summary*

**Description**

Returns match summary data for selected match URLs, including goals, subs and cards

**Usage**

```r
get_match_summary(match_url)
```

**Arguments**

- `match_url`  
  the fbref.com URL for the required match

**Value**

returns a dataframe with the match events (goals, cards, subs) for selected matches
get_match_urls

Examples

## Not run:
match <- get_match_urls(country = "AUS", gender = "F", season_end_year = 2021, tier = "1st")[1]
get_match_summary(match_url = match)

## End(Not run)

---

get_match_urls  Get match URLs

Description

Returns the URL for each match played for a given league season

Usage

get_match_urls(
  country,
  gender,
  season_end_year,
  tier = "1st",
  non_dom_league_url = NA
)

Arguments

country  the three character country code

gender  gender of competition, either "M" or "F", or both

season_end_year  the year the season(s) concludes

tier  the tier of the league, ie '1st' for the EPL or '2nd' for the Championship and so on

non_dom_league_url  the URL for Cups and Competitions found at https://fbref.com/en/comps/

Value

returns a character vector of all fbref match URLs for selected competition, season and gender

Examples

get_match_urls(country = "ENG", gender = "M", season_end_year = c(2019:2021), tier = "1st")
non_dom <- "https://fbref.com/en/comps/218/history/Friendlies-M-Seasons"
get_match_urls(country = "", gender = "M", season_end_year = 2021, non_dom_league_url = non_dom)
get_player_market_values

Get player market values

Description

Returns data frame of player valuations (in Euros) from transfermarkt.com

Usage

get_player_market_values(country_name, start_year, league_url = NA)

Arguments

country_name  the country of the league’s players
start_year    the start year of the season (2020 for the 20/21 season)
league_url    league url from transfermarkt.com. To be used when country_name not available in main function

Value

returns a dataframe of player valuations for country/seasons

get_season_team_stats

Get season team stats

Description

Returns different team season statistics results for a given league season and stat type

Usage

get_season_team_stats(country, gender, season_end_year, tier, stat_type)

Arguments

country         the three character country code for all countries
gender          gender of competition, either "M", "F" or both
season_end_year the year the season(s) concludes
tier            the tier of the league, ie ‘1st’ for the EPL or ‘2nd’ for the Championship and so on
stat_type       the type of team statistics the user requires

The statistic type options (stat_type) include:
"league_table", "league_table_home_away", "standard", "keeper", "keeper_adv", "shooting", "passing", "passing_types", "goal_shot_creation", "defense", "possession", "playing_time", "misc"
**get_team_match_results**

Value

returns a dataframe of a selected team statistic type for a selected league season

Examples

```r
get_season_team_stats("ITA", "M", 2021, "1st", "defense")
```

---

**get_team_match_results**

_GET team match results_

**Description**

Returns all game results for a team in a given season

**Usage**

```r
get_team_match_results(team_url)
```

**Arguments**

- `team_url` the URL for the team season

**Value**

returns a dataframe with the results of all games played by the selected team(s)

**Examples**

```r
## Not run:
# for single teams:
man_city_url <- "https://fbref.com/en/squads/b8fd03ef/Manchester-City-Stats"
get_team_match_results(man_city_url)

## End(Not run)
```
**player_dictionary_mapping**

*Player Mapping Dictionary*

**Description**

Returns data frame of players from the top 5 Euro leagues, their player URL and their respective Transfermarkt URL. Currently only for the players who have been in the top 5 leagues since the 2017-2018 season.

**Usage**

player_dictionary_mapping()

**Value**

returns a dataframe of FBref players and respective Transfermarkt URL

**Examples**

```r
## Not run:
mapped_players <- player_dictionary_mapping()
## End(Not run)
```

---

**player_transfer_history**

*Get player transfer history*

**Description**

Returns data frame of player(s) transfer history from transfermarkt.com

**Usage**

player_transfer_history(player_urls)

**Arguments**

player_urls  the player url(s) from transfermarkt

**Value**

returns a dataframe of player transfers
tm_league_team_urls  Get transfermarkt Team URLs

Description

Returns the URLs for all teams for a given league season

Usage

tm_league_team_urls(country_name, start_year, league_url = NA)

Arguments

country_name  the country of the league’s players
start_year  the start year of the season (2020 for the 20/21 season)
league_url  league url from transfermarkt.com. To be used when country_name not available in main function

Value

returns a character vector of all transfermarkt team URLs for a selected league

---

tm_matchday_table  Get weekly league table

Description

Returns the league table for each chosen matchday from transfermarkt

Usage

tm_matchday_table(country_name, start_year, matchday, league_url = NA)

Arguments

country_name  the country of the league’s players
start_year  the start year of the season (2020 for the 20/21 season)
matchday  the matchweek number. Can be a vector of matchdays
league_url  league url from transfermarkt.com. To be used when country_name not available in main function

Value

returns a dataframe of the table for a selected league and matchday
Examples

```r
## Not run:
 tm_matchday_table(country_name="England", start_year="2020", matchday=1)
 tm_matchday_table(country_name="England", start_year="2020", matchday=c(1:5))

## End(Not run)
```

---

### tm_player_bio

Get transfermarkt player bios

#### Description

Returns data frame of player bios from transfermarkt.com

#### Usage

```r
tm_player_bio(player_urls)
```

#### Arguments

- `player_urls` - player url(s) from transfermarkt

#### Value

returns a dataframe of player bios

#### Examples

```r
## Not run:
player_url <- "https://www.transfermarkt.com/eden-hazard/profil/spieler/50202"
 tm_player_bio(player_url)
 tm_player_bio(player_urls = c("https://www.transfermarkt.com/eden-hazard/profil/spieler/50202",
 "https://www.transfermarkt.com/sergio-ramos/profil/spieler/25557",

## End(Not run)
```
tm_squad_stats

Get squad player stats

Description
Returns basic stats for players of a team season

Usage
tm_squad_stats(team_url)

Arguments

| team_url | transfermarkt.com team url for a season |

Value
returns a dataframe of all player stats for team(s)

tm_team_player_urls

Get transfermarkt Player URLs

Description
Returns the transfermarkt URLs for all players for a given team

Usage
tm_team_player_urls(team_url)

Arguments

| team_url | the player's team URL (can be from tm_league_team_urls()) |

Value
returns a character vector of all transfermarkt player URLs for a selected team
**tm_team_transfers**  
*Get team transfers*

**Description**  
Returns all transfer arrivals and departures for a given team season

**Usage**  
```
tm_team_transfers(team_url, transfer_window = "all")
```

**Arguments**
- `team_url`: transfermarkt.com team url for a season
- `transfer_window`: which window the transfer occurred - options include "all" for both, "summer" or "winter"

**Value**
returns a dataframe of all team transfers

---

**tm_team_transfer_balances**  
*Team transfer balances*

**Description**  
Returns all team’s transfer aggregated performances for a chosen league season

**Usage**  
```
tm_team_transfer_balances(country_name, start_year, league_url = NA)
```

**Arguments**
- `country_name`: the country of the league’s players
- `start_year`: the start year of the season (2020 for the 20/21 season)
- `league_url`: league url from transfermarkt.com. To be used when country_name not available in main function

**Value**
returns a dataframe of the summarised financial transfer performance of all teams for a league season
**understat_league_match_results**

*Get Understat season match results*

**Description**

Returns match results for all matches played in the selected league season from Understat.com

**Usage**

`understat_league_match_results(league, season_start_year)`

**Arguments**

- `league` the available leagues in Understat as outlined below
- `season_start_year` the year the season started

The leagues currently available for Understat are: "EPL", "La liga", "Bundesliga", "Serie A", "Ligue 1", "RFPL"

**Value**

returns a dataframe of match results for a selected league season

---

**understat_league_season_shots**

*Get Understat league season shot locations*

**Description**

Returns shooting locations for all matches played in the selected league season from Understat.com

**Usage**

`understat_league_season_shots(league, season_start_year)`

**Arguments**

- `league` the available leagues in Understat as outlined below
- `season_start_year` the year the season started

The leagues currently available for Understat are: "EPL", "La liga", "Bundesliga", "Serie A", "Ligue 1", "RFPL"

**Value**

returns a dataframe of shooting locations for a selected league season
understat_match_shots  Get Understat match shot locations

Description

Returns shooting locations for a selected match from Understat.com

Usage

understat_match_shots(match_url)

Arguments

match_url  the URL of the match played

Value

returns a dataframe of shooting locations for a selected team season

understat_player_shots  Get all Understat shot locations for a player

Description

Returns shooting locations for a selected player for all matches played from Understat.com

Usage

understat_player_shots(player_url)

Arguments

player_url  the URL of a selected player

Value

returns a dataframe of shooting locations for a selected player
understat_team_meta  Get Understat team info

Description
Retrieve Understat team metadata, including team URLs. Similar to `understrr::get_team_meta`.

Usage
```r
understat_team_meta(team_names)
```

Arguments
- `team_names` a vector of team names (can be just 1)

Value
a data.frame

Examples
```r
## Not run:
understat_team_meta(team_name = c("Liverpool", "Manchester City"))
## End(Not run)
```

understat_team_players_stats  Get Understat team player stats

Description
Retrieve Understat team player stats.

Usage
```r
understat_team_players_stats(team_url)
```

Arguments
- `team_url` the URL of the team season

Value
a dataframe of player stats for a selected team season
understat_team_season_shots

*Get Understat team season shot locations*

**Description**

Returns shooting locations for all matches played by a selected team from Understat.com

**Usage**

`understat_team_season_shots(team_url)`

**Arguments**

- `team_url` the URL of the team season

**Value**

returns a dataframe of shooting locations for a selected team season

understat_team_stats_breakdown

*Get Understat team statistics breakdowns*

**Description**

Returns a data frame for the selected team(s) with stats broken down in different ways. Breakdown groups include:

**Usage**

`understat_team_stats_breakdown(team_urls)`

**Arguments**

- `team_urls` the url(s) of the teams in question

**Details**


**Value**

returns a dataframe of all stat groups and values
Index

.clean_advanced_stat_table, 3
.clean_match_advanced_stats_data, 3
.clean_player_season_stats, 4
.clean_table_names, 4
.convert_value_to_numeric, 5
.get_clean_understat_json, 5
.get_match_report_page, 6
.replace_empty_na, 6
.tm_fix_dates, 7
.understat_shooting, 7

fb_big5_advanced_season_stats, 8
fb_league_urls, 8
fb_player_match_logs, 9
fb_player_scouting_report, 10
fb_player_season_stats, 11
fb_player_urls, 11
fb_teams_urls, 12

get_advanced_match_stats, 13
get_match_lineups, 13
get_match_report, 14
get_match_results, 15
get_match_shooting, 16
get_match_summary, 16
get_match_urls, 17
get_player_market_values, 18
get_season_team_stats, 18
get_team_match_results, 19

player_dictionary_mapping, 20
player_transfer_history, 20

tm_league_team_urls, 21
tm_matchday_table, 21
tm_player_bio, 22
tm_squad_stats, 23
tm_team_player_urls, 23
tm_team_transfer_balances, 24
tm_team_transfers, 24

understat_league_match_results, 25
understat_league_season_shots, 25
understat_match_shots, 25
understat_player_shots, 26
understat_team_meta, 27
understat_team_players_stats, 27
understat_team_season_shots, 28
understat_team_stats_breakdown, 28