# Package: worldfootballR (via r-universe)

September 2, 2024

**Type** Package

**Title** Extract and Clean World Football (Soccer) Data

**Version** 0.6.6.0000

**Description** 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', transfer and valuations data from 'Transfermarkt'<https://www.transfermarkt.com/> and shooting location and other match stats 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

**URL** https://github.com/JaseZiv/worldfootballR

**BugReports** https://github.com/JaseZiv/worldfootballR/issues

**Depends** R (>= 4.0.0)

**Imports** dplyr, glue, httr, janitor, jsonlite, lubridate, magrittr, progress, purrr, qdapRegex, readr, rlang, rstudioapi, rvest (>= 1.0.4), stats, stringi, stringr, tidyr (>= 1.2.0), tidyselect, utils, withr, xml2, tibble, cli

**Suggests** chromote, R6, knitr, rmarkdown, testthat

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**Repository** https://jaseziv.r-universe.dev

**RemoteUrl** https://github.com/jaseziv/worldfootballr

**RemoteRef** HEAD

**RemoteSha** e1093e0995cd79283fb1d59a27a5a8f0dc7b485f

## Contents

---

`fb_advanced_match_stats` *Get FBref 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. Replaces the deprecated function get_advanced_match_stats()

## Usage

```
fb_advanced_match_stats(match_url, stat_type, team_or_player, time_pause = 3)
```

## 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 |

| | |
|---|---|
| `time_pause` | the wait time (in seconds) between page loads<br>The statistic type options (stat_type) include:<br>*"summary"*, *"passing"*, *"passing"_types*, *"defense"* , *"possession"*, *"misc"*, *"keeper"* |

## Value

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

## Examples

```
## Not run:
try({
urls <- fb_match_urls(country = "AUS", gender = "F", season_end_year = 2021, tier = "1st")

df <- fb_advanced_match_stats(match_url=urls,stat_type="possession",team_or_player="player")
})

## End(Not run)
```

---

`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

```
fb_big5_advanced_season_stats(
  season_end_year,
  stat_type,
  team_or_player,
  time_pause = 3
)
```

## 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 |
| `time_pause` | the wait time (in seconds) between page loads<br>The statistic type options (stat_type) include:<br>*"standard"*, *"shooting"*, *"passing"*, *"passing_types"*, *"gca"*, *"defense"*, *"possession"*, *"playing_time"*, *"misc"*, *"keepers"*, *"keepers_adv"* |

## Value

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

## Examples

```
## Not run:
try({
fb_big5_advanced_season_stats(season_end_year=2021,stat_type="possession",team_or_player="player")
})

## End(Not run)
```

---

`fb_league_stats` *Get FBref Team or Player Season Statistics for an Entire League*

---

## Description

Get the season stats for all teams / players in a selected league

## Usage

```
fb_league_stats(
  country,
  gender,
  season_end_year,
  tier = "1st",
  non_dom_league_url = NA,
  stat_type,
  team_or_player,
  time_pause = 3,
  rate = purrr::rate_backoff(max_times = 3)
)
```

## 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/ |
| `stat_type` | the type of statistic required. Must be one of the following:<br>• standard<br>• shooting |

- passing
- passing_types
- gca
- defense
- possession
- playing_time
- misc
- keepers
- keepers_adv

| | |
|---|---|
| `team_or_player` | result either summarised for each team, or individual players |
| `time_pause` | the wait time (in seconds) between page loads |
| `rate` | passed to ‘purrr::insistently‘. The function can be brittle, and works only on a second or third try. This parameter controls how much the function will retry to get a result. |

## Details

**[Experimental]**

## Value

a dataframe of season stats for all teams / players in a league

## Examples

```
## Not run:
try({
fb_league_stats(
  country = "ENG",
  gender = "M",
  season_end_year = 2022,
  tier = "1st",
  stat_type = "shooting",
  team_or_player = "player"
)
## Non-domestic league
##   Note that this is more likely to fail due to the volume of players
fb_league_stats(
  country = NA_character_,
  gender = "M",
  season_end_year = 2023,
  tier = NA_character_,
  non_dom_league_url = "https://fbref.com/en/comps/8/history/Champions-League-Seasons",
  stat_type = "standard",
  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

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

### Arguments

| | |
|---|---|
| `country` | the three character country code |
| `gender` | gender of competition, either "M" or "F" |
| `season_end_year` | the year the season(s) concludes (defaults to all available seasons) |
| `tier` | the tier of the league, ie '1st' for the EPL or '2nd' for the Championship and so on |

### Value

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

### Examples

```
## Not run:
try({
fb_league_urls(country = "ENG", gender = "M", season_end_year = 2021, tier = '1st')
})

## End(Not run)
```

## `fb_match_lineups` *Get FBref match lineups*

### Description

Returns lineups for home and away teams for a selected match Replaces the deprecated function get_match_lineups

### Usage

```
fb_match_lineups(match_url, time_pause = 3)
```

### Arguments

| | |
|---|---|
| `match_url` | the fbref.com URL for the required match |
| `time_pause` | the wait time (in seconds) between page loads |

### Value

returns a dataframe with the team lineups for a selected match

### Examples

```
## Not run:
try({
match <- fb_match_urls(country = "AUS", gender = "F", season_end_year = 2021, tier = "1st")[1]
df <- fb_match_lineups(match_url = match)
})

## End(Not run)
```

---

## `fb_match_report` *Get FBref match report*

---

### Description

Returns match report details for selected matches. Replaces the deprecated function get_match_report

### Usage

```
fb_match_report(match_url, time_pause = 3)
```

### Arguments

| | |
|---|---|
| `match_url` | the fbref.com URL for the required match |
| `time_pause` | the wait time (in seconds) between page loads |

### Value

returns a dataframe with the match details for a selected match

### Examples

```
## Not run:
try({
match <- fb_match_urls(country = "AUS", gender = "F", season_end_year = 2021, tier = "1st")[1]
df <- fb_match_report(match_url = match)
})

## End(Not run)
```

## `fb_match_results` *Get FBref match results*

### Description

Returns the game results for a given league season(s) Replaces the deprecated function get_match_results

### Usage

```
fb_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

```
## Not run:
try({
df <- fb_match_results(country = c("ITA"), gender = "M", season_end_year = 2021)
# for results from English Championship:
df <- fb_match_results(country = "ENG", gender = "M", season_end_year = 2021, tier = "2nd")
# for international friendlies:

})

## End(Not run)
```

## fb_match_shooting *Get FBref match shooting event data*

### Description

Returns detailed player shooting data for home and away teams for a selected match(es) Replaces the deprecated function get_match_shooting

### Usage

```
fb_match_shooting(match_url, time_pause = 3)
```

### Arguments

| | |
|---|---|
| match_url | the fbref.com URL for the required match |
| time_pause | the wait time (in seconds) between page loads |

### Value

returns a dataframe

### Examples

```
## Not run:
try({
match <- "https://fbref.com/en/matches/bf52349b/Fulham-Arsenal-September-12-2020-Premier-League"
df <- fb_match_shooting(match_url = match)
})

## End(Not run)
```

## fb_match_summary *Get FBref match summary*

### Description

Returns match summary data for selected match URLs, including goals, subs and cards Replaces the deprecated function get_match_summary

### Usage

```
fb_match_summary(match_url, time_pause = 3)
```

### Arguments

| | |
|---|---|
| `match_url` | the fbref.com URL for the required match |
| `time_pause` | the wait time (in seconds) between page loads |

### Value

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

### Examples

```
## Not run:
try({
match <- fb_match_urls(country = "AUS", gender = "F", season_end_year = 2021, tier = "1st")[1]
df <- fb_match_summary(match_url = match)
})

## End(Not run)
```

---

## `fb_match_urls` *Get FBref match URLs*

---

### Description

Returns the URL for each match played for a given league season Replaces the deprecated get_match_urls

### Usage

```
fb_match_urls(
  country,
  gender,
  season_end_year,
  tier = "1st",
  non_dom_league_url = NA,
  time_pause = 3
)
```

### 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/ |
| `time_pause` | the wait time (in seconds) between page loads |

### Value

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

### Examples

```
## Not run:
try({
fb_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"
fb_match_urls(country = "", gender = "M", season_end_year = 2021, non_dom_league_url = non_dom)
})

## End(Not run)
```

## `fb_player_goal_logs` *Get player goal logs*

### Description

Returns the player's career goal and assist logs

### Usage

```
fb_player_goal_logs(player_urls, time_pause = 3, goals_or_assists = "goals")
```

### Arguments

| | |
|---|---|
| `player_urls` | the URL(s) of the player(s) |
| `time_pause` | the wait time (in seconds) between page loads |
| `goals_or_assists` | select whether to return data of "goals" (the default), "assists", or "both" |

### Value

returns a dataframe of the player's goals and assists

### Examples

```
## Not run:
try({
# for single players:
jwp_url <- "https://fbref.com/en/players/3515d404/"
fb_player_goal_logs(player_urls = jwp_url, goals_or_assists = "goals")
})

## End(Not run)
```

## `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, time_pause = 3)
```

### Arguments

| | |
|---|---|
| `player_url` | the URL of the player (can come from fb_player_urls()) |
| `season_end_year` | the year the season(s) concludes |
| `stat_type` | the type of statistic required |
| `time_pause` | the wait time (in seconds) between page loads<br>The statistic type options (stat_type) include:<br>*"summary"*, *"keepers"*, *"passing"*, *"passing_types"*, *"gca"*, *"defense"*, *"possession"*, *"misc"* |

### Value

returns a dataframe of a player's match logs for a season

### Examples

```
try({
fb_player_match_logs("https://fbref.com/en/players/3bb7b8b4/Ederson",
season_end_year = 2021, stat_type = 'summary')
})
```

## `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,
  league_comp_name = NULL,
  time_pause = 3
)
```

## 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 |
| `league_comp_name` | the league or competition name(s) you want the scouting report for. Defaults to all |
| `time_pause` | the wait time (in seconds) between page loads |

## Value

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

## Examples

```
## Not run:
try({
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", league_comp_name = "Last 365 Days Men's Big 5 Leagues, UCL, UEL")

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

# for the last 365 days and also the 2022 WC
fb_player_scouting_report(player_url = "https://fbref.com/en/players/d70ce98e/Lionel-Messi",
pos_versus = "secondary",
league_comp_name = c("Last 365 Days Men's Big 5 Leagues, UCL, UEL", "2022 World Cup"))
})

## 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, national = FALSE, time_pause = 3)
```

## Arguments

| | |
|---|---|
| `player_url` | the URL(s) of the player(s) (can come from fb_player_urls()) |
| `stat_type` | the type of statistic required |
| `national` | the category of the required stats, with FALSE being Club Stats and TRUE being National Team Stats |
| `time_pause` | the wait time (in seconds) between page loads<br>The statistic type options (stat_type) include:<br>*"standard"*, *"shooting"*, *"passing"*, *"passing_types"*, *"gca"*, *"defense"*, *"possession" "playing_time"*, *"misc"*, *"keeper"*, *"keeper_adv"* |

## Value

returns a dataframe of a player's historical season stats

## Examples

```
## Not run:
try({
  standard_stats <- fb_player_season_stats(
    "https://fbref.com/en/players/3bb7b8b4/Ederson",
    stat_type = "standard"
  )

  multiple_playing_time <- fb_player_season_stats(
    player_url = c(
      "https://fbref.com/en/players/d70ce98e/Lionel-Messi",
      "https://fbref.com/en/players/dea698d9/Cristiano-Ronaldo"
    ),
    stat_type = "playing_time"
  )

  national_standard_stats <- fb_player_season_stats(
    "https://fbref.com/en/players/3bb7b8b4/Ederson",
    stat_type = "standard",
    national = TRUE
  )

  multiple_national_playing_time <- fb_player_season_stats(
    player_url = c(
      "https://fbref.com/en/players/d70ce98e/Lionel-Messi",
      "https://fbref.com/en/players/dea698d9/Cristiano-Ronaldo"
    ),
    stat_type = "playing_time",
    national = TRUE
  )
})

## 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, time_pause = 3)
```

## Arguments

| | |
|---|---|
| `team_url` | the player's team URL (can be from fb_team_urls()) |
| `time_pause` | the wait time (in seconds) between page loads |

## Value

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

## Examples

```
## Not run:
try({
fb_player_urls("https://fbref.com/en/squads/fd962109/Fulham-Stats")
})

## End(Not run)
```

## `fb_season_team_stats` *Get FBref season team stats*

### Description

Returns different team season statistics results for a given league season and stat type Replaces the deprecated function get_season_team_stats

### Usage

```
fb_season_team_stats(
  country,
  gender,
  season_end_year,
  tier,
  stat_type,
  time_pause = 3
)
```

### 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 |
| `time_pause` | the wait time (in seconds) between page loads<br>The statistic type options (stat_type) include:<br>*"league_table"*, *"league_table_home_away"*, *"standard"*, *"keeper"*, *"keeper_adv"*, *"shooting"*, *"passing"*, *"passing_types"*, *"goal_shot_creation"*, *"defense"* , *"possession"*, *"playing_time"*, *"misc"* |

### Value

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

### Examples

```
## Not run:
try({
fb_season_team_stats("ITA", "M", 2021, "1st", "defense")
})

## End(Not run)
```

## fb_squad_wages *Get team player wages*

### Description

Returns all player wages from FBref via Capology

### Usage

```
fb_squad_wages(team_urls, time_pause = 3)
```

### Arguments

| | |
|---|---|
| team_urls | the URL(s) of the teams(s) (can come from fb_teams_urls()) |
| time_pause | the wait time (in seconds) between page loads |

### Value

returns a dataframe with all available estimated player wages for the selected team(s)

### Examples

```
## Not run:
try({
# for single teams:
man_city_url <- "https://fbref.com/en/squads/b8fd03ef/Manchester-City-Stats"
fb_squad_wages(team_urls = man_city_url)
})

## End(Not run)
```

## fb_teams_urls *Get fbref Team URLs*

### Description

Returns the URLs for all teams for a given league

### Usage

```
fb_teams_urls(league_url, time_pause = 3)
```

### Arguments

| | |
|---|---|
| league_url | the league URL (can be from fb_league_urls()) |
| time_pause | the wait time (in seconds) between page loads |

## Value

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

## Examples

```
## Not run:
try({
fb_teams_urls("https://fbref.com/en/comps/9/Premier-League-Stats")
})

## End(Not run)
```

---

## `fb_team_goal_logs` *Get team goal logs*

---

## Description

Returns the team's season goal logs

## Usage

```
fb_team_goal_logs(team_urls, time_pause = 3, for_or_against = "for")
```

## Arguments

| | |
|---|---|
| `team_urls` | the URL(s) of the team(s) (can come from fb_teams_urls()) |
| `time_pause` | the wait time (in seconds) between page loads |
| `for_or_against` | select whether to return data of goals "for" (the default), goals "against", or "both" |

## Value

returns a dataframe of the team's goals scored and conceded in the season

## Examples

```
## Not run:
try({
# for single teams:
man_city_url <- "https://fbref.com/en/squads/b8fd03ef/Manchester-City-Stats"
fb_team_goal_logs(team_urls = man_city_url, for_or_against = "for")
})

## End(Not run)
```

## fb_team_match_log_stats *Get team match log stats*

### Description

Returns all match statistics for a team(s) in a given season

### Usage

```
fb_team_match_log_stats(team_urls, stat_type, time_pause = 3)
```

### Arguments

| | |
|---|---|
| `team_urls` | the URL(s) of the teams(s) (can come from fb_teams_urls()) |
| `stat_type` | the type of statistic required |
| `time_pause` | the wait time (in seconds) between page loads<br>The statistic type options (stat_type) include:<br>*"shooting"*, *"keeper"*, *"passing"*, *"passing_types"*, *"gca"*, *"defense"*, *"misc"* |

### Value

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

### Examples

```
## Not run:
try({
# for single teams:
man_city_url <- "https://fbref.com/en/squads/b8fd03ef/Manchester-City-Stats"
fb_team_match_log_stats(team_urls = man_city_url, stat_type = "passing")
})

## End(Not run)
```

## fb_team_match_results *Get FBref team match results*

### Description

Returns all game results for a team in a given season Replaces the deprecated function get_team_match_results

### Usage

```
fb_team_match_results(team_url, time_pause = 3)
```

## Arguments

| | |
|---|---|
| `team_url` | the URL for the team season |
| `time_pause` | the wait time (in seconds) between page loads |

## Value

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

## Examples

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

## End(Not run)
```

---

`fb_team_match_stats` *Get FBref match team stats*

---

## Description

Returns match team stats for selected matches.

## Usage

```
fb_team_match_stats(match_url, time_pause = 3)
```

## Arguments

| | |
|---|---|
| `match_url` | the fbref.com URL for the required match |
| `time_pause` | the wait time (in seconds) between page loads |

## Value

returns a dataframe with the match team stats for a selected match

## Examples

```
## Not run:
try({
match <- fb_match_urls(country = "AUS", gender = "F", season_end_year = 2021, tier = "1st")[1]
df <- fb_team_match_stats(match_url = match)
})

## End(Not run)
```

## `fb_team_player_stats` *Get fbref Team's Player Season Statistics*

### Description

Returns the team's players season stats for a selected team(s) and stat type

### Usage

```
fb_team_player_stats(team_urls, stat_type, time_pause = 3)
```

### Arguments

| | |
|---|---|
| `team_urls` | the URL(s) of the teams(s) (can come from fb_teams_urls()) |
| `stat_type` | the type of statistic required |
| `time_pause` | the wait time (in seconds) between page loads<br>The statistic type options (stat_type) include:<br>*"standard"*, *"shooting"*, *"passing"*, *"passing_types"*, *"gca"*, *"defense"*, *"possession" "playing_time"*, *"misc"*, *"keeper"*, *"keeper_adv"* |

### Value

returns a dataframe of all players of a team's season stats

### Examples

```
## Not run:
try({
fb_team_player_stats("https://fbref.com/en/squads/d6a369a2/Fleetwood-Town-Stats",
                        stat_type = 'standard')

league_url <- fb_league_urls(country = "ENG", gender = "M",
                                             season_end_year = 2022, tier = "3rd")
team_urls <- fb_teams_urls(league_url)
multiple_playing_time <- fb_team_player_stats(team_urls,
                          stat_type = "playing_time")
})

## End(Not run)
```

## 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, time_pause = 3)
```

### 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 |
| `time_pause` | the wait time (in seconds) between page loads<br>The statistic type options (stat_type) include:<br>*"summary"*, *"passing"*, *"passing"_types*, *"defense"* , *"possession"*, *"misc"*, *"keeper"* |

### Value

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

### Examples

```
## Not run:
try({
urls <- get_match_urls(country = "AUS", gender = "F", season_end_year = 2021, tier = "1st")

df <- 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, time_pause = 3)
```

### Arguments

| | |
|---|---|
| `match_url` | the fbref.com URL for the required match |
| `time_pause` | the wait time (in seconds) between page loads |

### Value

returns a dataframe with the team lineups for a selected match

### Examples

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

## End(Not run)
```

## get_match_report *Get match report*

### Description

Returns match report details for selected matches

### Usage

```
get_match_report(match_url, time_pause = 3)
```

### Arguments

| | |
|---|---|
| `match_url` | the fbref.com URL for the required match |
| `time_pause` | the wait time (in seconds) between page loads |

### Value

returns a dataframe with the match details for a selected match

### Examples

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

```
## Not run:
try({
df <- get_match_results(country = c("ITA"), gender = "M", season_end_year = 2021)
# for results from English Championship:
df <- get_match_results(country = "ENG", gender = "M", season_end_year = 2021, tier = "2nd")
# for international friendlies:

})

## End(Not run)
```

## `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

```
get_match_shooting(match_url, time_pause = 3)
```

### Arguments

| | |
|---|---|
| `match_url` | the fbref.com URL for the required match |
| `time_pause` | the wait time (in seconds) between page loads |

### Value

returns a dataframe

### Examples

```
## Not run:
try({
match <- "https://fbref.com/en/matches/bf52349b/Fulham-Arsenal-September-12-2020-Premier-League"
df <- 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

```
get_match_summary(match_url, time_pause = 3)
```

### Arguments

| | |
|---|---|
| `match_url` | the fbref.com URL for the required match |
| `time_pause` | the wait time (in seconds) between page loads |

### Value

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

### Examples

```
## Not run:
try({
match <- get_match_urls(country = "AUS", gender = "F", season_end_year = 2021, tier = "1st")[1]
df <- 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,
  time_pause = 3
)
```

### 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/ |
| `time_pause` | the wait time (in seconds) between page loads |

### Value

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

### Examples

```
## Not run:
try({
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)
})

## End(Not run)
```

---

## `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,
  time_pause = 3
)
```

### 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 |
| `time_pause` | the wait time (in seconds) between page loads<br>The statistic type options (stat_type) include:<br>*"league_table"*, *"league_table_home_away*", *"standard"*, *"keeper"*, *"keeper_adv"*, *"shooting"*, *"passing"*, *"passing_types"*, *"goal_shot_creation"*, *"defense"* , *"possession"*, *"playing_time"*, *"misc"* |

### Value

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

### Examples

```
## Not run:
try({
get_season_team_stats("ITA", "M", 2021, "1st", "defense")
})

## End(Not run)
```

## get_team_match_results

*Get team match results*

### Description

Returns all game results for a team in a given season

### Usage

```
get_team_match_results(team_url, time_pause = 3)
```

### Arguments

| | |
|---|---|
| `team_url` | the URL for the team season |
| `time_pause` | the wait time (in seconds) between page loads |

### Value

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

### Examples

```
## Not run:
try({
# 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)
```

## load_fb_advanced_match_stats

*Load pre-saved FBref match advanced stats*

### Description

Loading version of `fb_advanced_match_stats`. Only some leagues available.

## Usage

```
load_fb_advanced_match_stats(
  country,
  gender,
  tier,
  stat_type,
  team_or_player,
  season_end_year = NA
)
```

## Arguments

| | |
|---|---|
| `country` | the three character country code |
| `gender` | gender of competition, either "M" or "F" |
| `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 |
| `team_or_player` | result either summarised for each team, or individual players |
| `season_end_year` | the year(s) the season concludes |

## Value

returns a dataframe

## Examples

```
try({
load_fb_advanced_match_stats(
  country = "ENG",
  gender = "M",
  tier = "1st",
  stat_type = "summary",
  team_or_player = "player"
)

load_fb_advanced_match_stats(
  country = c("ITA", "ESP"),
  gender = "M",
  tier = "1st",
  season_end_year = 2023,
  stat_type = "defense",
  team_or_player = "player"
)
})
```

## load_fb_big5_advanced_season_stats

*Load Big 5 Euro League Season Stats*

### Description

Loading version of `fb_big5_advanced_season_stats` 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

```
load_fb_big5_advanced_season_stats(
  season_end_year = NA,
  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<br>The statistic type options (stat_type) include:<br>*"standard"*, *"shooting"*, *"passing"*, *"passing_types"*, *"gca"*, *"defense"*, *"possession"*, *"playing_time"*, *"misc"*, *"keepers"*, *"keepers_adv"* |

### Value

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

### Examples

```
try({
df <- load_fb_big5_advanced_season_stats(
season_end_year = c(2018:2022), stat_type = "defense", team_or_player = "player"
)

df <- load_fb_big5_advanced_season_stats(
season_end_year = 2022, stat_type = "defense", team_or_player = "player"
)
})
```

## `load_fb_match_shooting` *Load pre-saved FBref match shooting event data*

### Description

Loading version of `fb_match_shooting`. Only some leagues available.

### Usage

```
load_fb_match_shooting(country, gender, tier, season_end_year = NA)
```

### Arguments

| | |
|---|---|
| `country` | the three character country code |
| `gender` | gender of competition, either "M" or "F" |
| `tier` | the tier of the league, ie ’1st’ for the EPL or ’2nd’ for the Championship and so on |
| `season_end_year` | the year(s) the season concludes |

### Value

returns a dataframe

### Examples

```
try({
load_fb_match_shooting(
  country = "ENG",
  gender = "M",
  tier = "1st"
)

load_fb_match_shooting(
  country = c("ITA", "ESP"),
  gender = "M",
  tier = "1st",
  season_end_year = 2019
)
})
```

## `load_fb_match_summary` *Load pre-saved FBref match summary data*

### Description

Loading version of `fb_match_summary`. Only some leagues available.

### Usage

```
load_fb_match_summary(country, gender, tier, season_end_year = NA)
```

### Arguments

| | |
|---|---|
| `country` | the three character country code |
| `gender` | gender of competition, either "M" or "F" |
| `tier` | the tier of the league, ie '1st' for the EPL or '2nd' for the Championship and so on |
| `season_end_year` | the year(s) the season concludes |

### Value

returns a dataframe

### Examples

```
try({
load_fb_match_summary(
  country = "ENG",
  gender = "M",
  tier = "1st"
)

load_fb_match_summary(
  country = c("ITA", "ESP"),
  gender = "M",
  tier = "1st",
  season_end_year = 2019
)
})
```

---

## `load_match_comp_results` *Load match competition results*

---

### Description

Returns the game results for a competition(s), ie League cups or international competitions from FBref. comp_name comes from https://github.com/JaseZiv/worldfootballR_data/tree/master/data/match_results_cups#readm

### Usage

```
load_match_comp_results(comp_name)
```

### Arguments

| | |
|---|---|
| `comp_name` | the three character country code |

### Value

returns a dataframe with the results of the competition name

### Examples

```
try({
df <- load_match_comp_results(
comp_name = "Coppa Italia"
)
# for multiple competitions:
cups <- c("FIFA Women's World Cup",
          "FIFA World Cup")
df <- load_match_comp_results(
comp_name = cups
)
})
```

---

## `load_match_results` *Load match results*

---

### Description

Loading version of `get_match_results` Returns the game results for a given league season(s) from FBref

### Usage

```
load_match_results(country, gender, season_end_year, tier)
```

### 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 |

### Value

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

### Examples

```
try({
df <- load_match_results(
country = c("ITA"), gender = "M", season_end_year = 2021, tier = "1st"
)
# for results from English 1st div for men and women:
df <- load_match_results(
country = "ENG", gender = c("M", "F"), season_end_year = 2021, tier = "1st"
)
})
```

---

## `load_understat_league_shots` *Load Understat league shot locations*

---

### Description

Loading version of `understat_league_season_shots`, but for all seasons Returns shooting locations for all matches played in the selected league

### Usage

```
load_understat_league_shots(league)
```

### Arguments

| | |
|---|---|
| `league` | the available leagues in Understat as outlined below<br>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

### Examples

```
## Not run:
try({
  df <- load_understat_league_shots(league = "Serie A")
})

## 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

```
try({
mapped_players <- player_dictionary_mapping()
})
```

---

## `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_expiring_contracts` *Get expiring contracts*

### Description

Returns a data frame of players with expiring contracts for a selected league and time period

### Usage

```
tm_expiring_contracts(country_name, contract_end_year, league_url = NA)
```

### Arguments

| | |
|---|---|
| `country_name` | the country of the league's players |
| `contract_end_year` | the year the contract is due to expire |
| `league_url` | league url from transfermarkt.com. To be used when country_name not available in main function |

### Value

returns a dataframe of expiring contracts in the selected league

## `tm_get_player_absence` *Get Player Absences*

### Description

Returns data frame of a player's absences from suspension from transfermarkt.com

### Usage

```
tm_get_player_absence(player_urls)
```

### Arguments

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

## Value

returns a dataframe

## Examples

```
## Not run:
try({
player_urls <- c("https://www.transfermarkt.com/cristian-romero/profil/spieler/355915",
"https://www.transfermarkt.com/micky-van-de-ven/profil/spieler/557459")

df <- tm_get_player_absence(player_urls)
})

## End(Not run)
```

---

## `tm_league_debutants` *Get league debutants*

---

### Description

Returns a data frame of debutants for a selected league

### Usage

```
tm_league_debutants(
  country_name,
  league_url = NA,
  debut_type,
  debut_start_year,
  debut_end_year
)
```

### Arguments

| | |
|---|---|
| `country_name` | the country of the league's players |
| `league_url` | league url from transfermarkt.com. To be used when country_name not available in main function |
| `debut_type` | whether you want 'league' debut or 'pro' debut |
| `debut_start_year` | the season start year of the beginning of the period you want results for |
| `debut_end_year` | the season start year of the end of the period you want results for |

### Value

returns a dataframe of players who debuted in the selected league

## `tm_league_injuries` *Get league injuries*

### Description

Returns a data frame of all currently injured players players for a selected league

### Usage

```
tm_league_injuries(country_name, league_url = NA)
```

### Arguments

| | |
|---|---|
| `country_name` | the country of the league's players |
| `league_url` | league url from transfermarkt.com. To be used when country_name not available in main function |

### Value

returns a dataframe of injured players in the selected league

## `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

```
## Not run:
try({
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

```
tm_player_bio(player_urls)
```

### Arguments

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

### Value

returns a dataframe of player bios

### Examples

```
## Not run:
try({
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",
                       "https://www.transfermarkt.com/ivo-grbic/profil/spieler/226073"))
})

## End(Not run)
```

---

## `tm_player_injury_history` *Get player injury history*

---

### Description

Returns data frame of a player's injury history transfermarkt.com

### Usage

```
tm_player_injury_history(player_urls)
```

### Arguments

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

### Value

returns a dataframe of player injury history

## `tm_player_market_values` *Get Transfermarkt player market values*

### Description

Returns data frame of player valuations (in Euros) from transfermarkt.com Replaces the deprecated function get_player_market_values

### Usage

```
tm_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

## `tm_player_transfer_history` *Get Transfermarkt player transfer history*

### Description

Returns data frame of player(s) transfer history from transfermarkt.com Replaces the deprecated function player_transfer_history

### Usage

```
tm_player_transfer_history(player_urls, get_extra_info = TRUE)
```

### Arguments

| | |
|---|---|
| `player_urls` | the player url(s) from transfermarkt |
| `get_extra_info` | allows users to decide if they want to scrape extra info (contract length, countries involved) or not |

### Value

returns a dataframe of player transfers

## `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_staff_job_history` *Get Staff Member's job history*

### Description

Returns all roles a selected staff member(s) has held and performance data

### Usage

```
tm_staff_job_history(staff_urls)
```

### Arguments

|  |  |
|---|---|
| `staff_urls` | transfermarkt.com staff(s) url (can use tm_league_staff_urls() to get) |

### Value

returns a data frame of all roles a selected staff member(s) has held and performance data

## `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_staff_history` *Get team staff history*

### Description

Returns all people who have held the selected role in a team's history

### Usage

```
tm_team_staff_history(team_urls, staff_role = "Manager")
```

### Arguments

| | |
|---|---|
| `team_urls` | transfermarkt.com team(s) url for a season |
| `staff_role` | the role description which can be found here: https://github.com/JaseZiv/worldfootballR_data/blob/maste data/transfermarkt_staff/tm_staff_types.csv |

### Value

returns a data frame of all selected staff roles for a team(s) history

## tm_team_staff_urls *Get transfermarkt Club Staff URLs*

### Description

Returns the transfermarkt URLs for all staff of selected roles for a given team

### Usage

```
tm_team_staff_urls(team_urls, staff_role)
```

### Arguments

| | |
|---|---|
| `team_urls` | the staff member's team URL (can be from tm_league_team_urls()) |
| `staff_role` | role of the staff member URLs required for with options including: *"Manager"*, *"Assistant Manager"*, *"Goalkeeping Coach"*, *"Fitness Coach"*, *"Conditioning Coach"* |

### Value

returns a character vector of all transfermarkt staff URLs for a selected team(s)

## 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_available_teams` *Get Understat available teams*

### Description

Returns all available team names for the selected leagues

### Usage

```
understat_available_teams(leagues)
```

### Arguments

| | |
|---|---|
| `leagues` | the available leagues in Understat as outlined below #' The leagues currently available for Understat are: *"EPL"*, *"La liga"*, *"Bundesliga"*, *"Serie A"*, *"Ligue 1"*, *"RFPL"*<br>#' @return a character vector teams names |

### Examples

```
## Not run:
try({
understat_available_teams(leagues = c('EPL', 'La liga'))
})

## End(Not run)
```

## `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<br>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<br>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_players` *Get Understat match player data*

---

### Description

Returns player values for a selected match from Understat.com.

### Usage

```
understat_match_players(match_url)
```

### Arguments

| | |
|---|---|
| `match_url` | A ‘character‘ string with the URL of the match played. |

### Value

returns a ‘data.frame‘ with data for all players for the match.

---

## `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_match_stats` *Get Understat match stats table data*

### Description

Returns the Stats values for a selected match from Understat.com.

### Usage

```
understat_match_stats(match_url)
```

### Arguments

| | |
|---|---|
| `match_url` | A ‘character‘ string with the URL of the match played. |

### Details

For ‘draw_chances‘, ‘home_chances‘ and ‘away_chances‘, values below 10

### Value

returns a ‘data.frame‘ with data from the stats table for the match.

## `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 'understatr::get_team_meta'.

### Usage

```
understat_team_meta(team_names)
```

### Arguments

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

### Value

a data.frame

### Examples

```
## Not run:
try({
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

```
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

*"Situation"*, *"Formation"*, *"Game state"*, *"Timing"*, *"Shot zones"*, *"Attack speed"*, *"Result"*

### Value

returns a dataframe of all stat groups and values

# Index