Package ‘cfbfastR’

June 14, 2022

Title Access College Football Play by Play Data
Version 1.9.0
Description A utility to quickly obtain clean and tidy college football
data. Serves as a wrapper around the
<https://collegefootballdata.com/> API and provides functions to
access live play by play and box score data from ESPN
<https://www.espn.com> when available. It provides users the
capability to access a plethora of endpoints, and supplement that data
with additional information (Expected Points Added/Win Probability
added).
License MIT + file LICENSE
URL https://cfbfastR.sportsdataverse.org/,
https://github.com/sportsdataverse/cfbfastR
BugReports https://github.com/sportsdataverse/cfbfastR/issues
Depends R (>= 4.0.0)
Imports cli (>= 3.0.0), data.table (>= 1.14.0), dplyr, glue, httr,
janitor, jsonlite, magrittr, mgcv (>= 1.8-33), nnet, progressr
(>= 0.6.0), purrr (>= 0.3.0), Rcpp (>= 1.0.7), RcppParallel (>= 5.1.4), rlang, stringr (>= 1.3.0), tibble (>= 3.0), tidyr (>= 1.0.0)
Suggests crayon (>= 1.3.4), curl, DBI, ggplot2, ggrepel, qs (>= 0.25.1), rmarkdown, RSQLite, stats, stringi, testthat, usethis
(>= 1.6.0), xgboost (>= 1.1)
Encoding UTF-8
LazyData true
RoxygenNote 7.2.0
Config/testthat/edition 3
NeedsCompilation no
Author Saiem Gilani [cre, aut],
Akshay Easwaran [aut],
Jared Lee [aut],
1
Eric Hess [aut],
Michael Egle [ctb],
Nate Manzo [ctb],
Jason DeLoach [ctb],
Tej Seth [ctb],
Conor McQuiston [ctb],
Tan Ho [ctb],
Keegan Abdoo [ctb],
Matt Spencer [ctb],
Sebastian Carl [ctb],
John Edwards [ctb]

Maintainer Saiem Gilani <saiem.gilani@gmail.com>
Repository CRAN
Date/Publication 2022-06-13 22:10:02 UTC

R topics documented:

cfbdd_betting_lines .............................................. 4

cfbdd_calendar .................................................. 5

cfbdd_coaches .................................................... 6

cfbdd_conferences ............................................... 7

cfbdd_draft ....................................................... 8

cfbdd_draft_picks ............................................... 8

cfbdd_draft_positions ........................................... 10

cfbdd_draft_teams ............................................... 10

cfbdd_drives ..................................................... 11

cfbdd_games ....................................................... 13

cfbdd_game_box_advanced ....................................... 14

cfbdd_game_info ................................................ 17

cfbdd_game_media ................................................ 19

cfbdd_game_player_stats ....................................... 20

cfbdd_game_records ............................................. 22

cfbdd_game_team_stats ......................................... 23

cfbdd_game_weather ............................................. 26

cfbdd_metrics .................................................... 27

cfbdd_metrics_ppa_games ....................................... 28

cfbdd_metrics_ppa_players_games .............................. 30

cfbdd_metrics_ppa_players_season ............................. 31

cfbdd_metrics_ppa_predicted ................................... 33

cfbdd_metrics_ppa_teams ....................................... 33

cfbdd_metrics_wp ................................................ 35

cfbdd_metrics_wp_pregame ...................................... 36

cfbdd_pbp_data ................................................... 37

cfbdd_play ......................................................... 47

cfbdd_players ..................................................... 48

cfbdd_player_info ............................................... 49

cfbdd_player_returning ......................................... 50
R topics documented:

cfbd_player_usage ........................................ 51

cfbd_plays ........................................ 52

cfbd_play_stats_player ................................ 54

cfbd_play_stats_types ................................ 57

cfbd_play_types ...................................... 57

cfbd_rankings ........................................ 58

cfbd_ratings ........................................ 59

cfbd_ratings_elo ...................................... 60

cfbd_ratings_sp ...................................... 61

cfbd_ratings_sp_conference .......................... 62

cfbd_ratings_srs ...................................... 64

cfbd_recruiting ...................................... 65

cfbd_recruiting_player .............................. 66

cfbd_recruiting_position ............................ 68

cfbd_recruiting_team ................................ 69

cfbd_recruiting_transfer_portal .................... 70

cfbd_stats .......................................... 71

cfbd_stats_categories .............................. 72

cfbd_stats_game_advanced ........................... 72

cfbd_stats_season_advanced ....................... 75

cfbd_stats_season_player .......................... 78

cfbd_stats_season_team ............................. 81

cfbd_teams .......................................... 83

cfbd_team_info ...................................... 84

cfbd_team_matchup ................................... 85

cfbd_team_matchup_records ......................... 86

cfbd_team_roster .................................... 87

cfbd_team_talent .................................... 88

cfbd_venues ........................................ 89

csv_from_url ........................................ 90

data .................................................. 90

espn_cfb_calendar ................................... 91

espn_cfb_pbp ........................................ 92

espn_cfb_scoreboard ................................ 92

espn_cfb_team_stats ................................ 95

espn_metrics ........................................ 101

espn_ratings_fpi .................................... 102

load_cfb_pbp ........................................ 103

load_cfb_rosters .................................... 104

load_cfb_schedules .................................. 104

load_cfb_teams ...................................... 105

rds_from_url ........................................ 106

register_cfbd ........................................ 106

update_cfb_db ........................................ 107

Index 109
CFBD Betting Lines Endpoint Overview

Description

Get betting lines information for games

Usage

cfbd_betting_lines(
    game_id = NULL,
    year = NULL,
    week = NULL,
    season_type = "regular",
    team = NULL,
    home_team = NULL,
    away_team = NULL,
    conference = NULL,
    line_provider = NULL
)

Arguments

game_id  (Integer optional): Game ID filter for querying a single game
Can be found using the cfbd_game_info() function

year     (Integer required): Year, 4 digit format(YYYY)

week    (Integer optional): Week - values from 1-15, 1-14 for seasons pre-playoff (i.e. 2013 or earlier)

season_type (String default regular): Select Season Type: regular or postseason

team       (String optional): D-I Team

home_team   (String optional): Home D-I Team

away_team   (String optional): Away D-I Team

conference  (String optional): Conference abbreviation - Select a valid FBS conference
Conference abbreviations P5: ACC, B12, B1G, SEC, PAC
Conference abbreviations G5 and FBS Independents: CUSA, MAC, MWC, Ind, SBC, AAC

line_provider  (String optional): Select Line Provider - Caesars, consensus, numberfire, or teamrankings

Value

Betting information for games with the following columns:

game_id:integer. Unique game identifier - game_id.
season:integer. Season parameter.

season_type:character) Season Type (regular, postseason, both

week:integer. Week, values from 1-15, 1-14 for seasons pre-playoff (i.e. 2013 or earlier).

start_date:character. Start Date

home_team:character. Home D-I Team.


home_score:integer. Home Score.

away_team:character. Away D-I Team.

away_conference:character. Away D-I Conference.

away_score:integer. Away Score.

provider:character. Line provider.

spread:character. Spread for the game.

formatted_spread:character. Formatted spread for the game.

spread_open:character. Opening spread for the game.

over_under:character. Over/Under for the game.

over_under_open:character. Opening over/under for the game.

home_moneyline:character. Home team moneyline.

away_moneyline:character. Away team moneyline.

Examples

try(cfbd_betting_lines(year = 2018, week = 12, team = "Florida State"))

<table>
<thead>
<tr>
<th>cfbd_calendar</th>
<th>Get calendar of weeks by season.</th>
</tr>
</thead>
</table>

Description

Get calendar of weeks by season.

Usage

cfbd_calendar(year)

Arguments

year (Integer required): Year, 4 digit format (YYYY)
Value

cfbd_calendar() - A data frame with 5 variables:

season: character. Calendar season.
week: integer. Calendar game week.
season_type: character. Season type of calendar week.
first_game_start: character. First game start time of the calendar week.
last_game_start: character. Last game start time of the calendar week.

Examples

try(cfbd_calendar(2019))

cfbd_coaches       CFBD Coaches Endpoint Overview

Description

Coach information search A coach search function which provides coaching records and school
history for a given coach

Usage

cfbd_coaches(
  first = NULL,
  last = NULL,
  team = NULL,
  year = NULL,
  min_year = NULL,
  max_year = NULL
)

Arguments

first (String optional): First name for the coach you are trying to look up
last (String optional): Last name for the coach you are trying to look up
team (String optional): Team - Select a valid team, D1 football
year (Integer optional): Year, 4 digit format (YYYY).
min_year (Integer optional): Minimum Year filter (inclusive), 4 digit format (YYYY).
max_year (Integer optional): Maximum Year filter (inclusive), 4 digit format (YYYY).
Value

- **first_name**:character. First name of coach.
- **last_name**:character. Last name of coach.
- **hire_date**:character. Hire date of coach.
- **school**:character. School of coach.
- **year**:integer. Season of record.
- **games**:integer. Games as coach.
- **wins**:integer. Wins for the season.
- **losses**:integer. Losses for the season.
- **ties**:integer. Ties for the season.
- **preseason_rank**:integer. Preseason rank for the school of coach.
- **postseason_rank**:integer. Postseason rank for the school of coach.
- **srs**:character. Simple Rating System adjustment for team.
- **sp_overall**:character. Bill Connelly’s SP+ overall for team.
- **sp_offense**:character. Bill Connelly’s SP+ offense for team.
- **sp_defense**:character. Bill Connelly’s SP+ defense for team.

Examples

```r
try(cfbd_coaches(first = "Nick", last = "Saban", team = "alabama"))
```

---

### CFBD Conferences Endpoint Overview

#### Description

**Get college football conference information** Pulls all college football conferences and returns as data frame

#### Usage

```r
cfbd_conferences()
```

#### Value

- **cfbd_conferences()** - A data frame with 94 rows and 5 variables:
  - conference_id: Referencing conference id.
  - name: Conference name.
  - long_name: Long name for Conference.
  - abbreviation: Conference abbreviation.
  - classification: Conference classification (fbs,fcs,ii,iii)
Examples

```r
try(cfbd_conferences())
```

cfbd_draft

CFBD NFL Draft Endpoint Overview

Description

cfbd_draft_teams(): Get list of NFL teams.
cfbd_draft_positions(): Get list of NFL positions for mapping to college positions.
cfbd_draft_picks(): Get list of NFL Draft picks.

details

Get list of NFL teams:
cfbd_draft_teams()

Get list of NFL positions for mapping to collegiate:
cfbd_draft_positions()

Get list of NFL Draft picks:
cfbd_draft_picks(year = 2020, college = "Texas")
cfbd_draft_picks(nfl_team = "Cincinatti")

Get list of NFL draft picks

Description

Get list of NFL draft picks

Usage

```r
cfbd_draft_picks(
    year = NULL,
    nfl_team = NULL,
    college = NULL,
    conference = NULL,
    position = NULL
)
```
**cfbd_draft_picks**

**Arguments**

- **year** *(Integer required): NFL draft class, 4 digit format (YYYY)*
- **nfl_team** *(String): NFL drafting team, see cfbd_draft_teams() for valid selections.*
- **college** *(String): NFL draftee college team, see cfbd_team_info() for valid selections.*
- **conference** *(String): NFL draftee college team conference, see cfbd_conferences() for valid selections.*
- **position** *(String): NFL position abbreviation, see cfbd_draft_positions() for valid selections.*

**Value**

- **cfbd_draft_picks()** - A data frame with 23 variables:
  - college_athlete_id: integer. College athlete referencing id.
  - nfl_athlete_id: integer NFL athlete referencing id.
  - college_id: integer College team referencing id.
  - college_team: character College team name.
  - college_conference: character Conference of college team.
  - nfl_team: character NFL team name of drafted player.
  - year: integer. NFL draft class year.
  - overall: integer. Overall draft pick number.
  - round: integer. Round of NFL draft the draftee was picked in.
  - pick: integer. Pick number of the NFL draftee within the round they were picked in.
  - name: character. NFL draftee name.
  - position: character. NFL draftee position.
  - height: double. NFL draftee height.
  - weight: integer. NFL draftee weight.
  - pre_draft_ranking: integer Pre-draft ranking (ESPN).
  - pre_draft_position_ranking: integer. Pre-draft position ranking (ESPN).
  - pre_draft_grade: double. Pre-draft scouts grade (ESPN).
  - hometown_info_city: character. Hometown of the NFL draftee.
  - hometown_info_state_province: character. Hometown state of the NFL draftee.
  - hometown_info_country: character. Hometown country of the NFL draftee.
  - hometown_info_latitude: character. Hometown latitude of the NFL draftee.
  - hometown_info_longitude: character. Hometown longitude of the NFL draftee.
  - hometown_info_county_fips: character. Hometown FIPS code of the NFL draftee.

**Examples**

```r
try(cfbd_draft_picks(year = 2020))

try(cfbd_draft_picks(year = 2016, position = "PK"))
```
cfbd_draft_positions  Get list of NFL positions

Description
Get list of NFL positions

Usage
   cfbd_draft_positions()

Value
   cfbd_draft_positions() - A data frame with 2 variables:
   position_name: character. NFL Position group name.
   position_abbreviation: integer  NFL position group abbreviation.

Examples
   try(cfbd_draft_positions())

---

cfbd_draft_teams  Get list of NFL teams

Description
Get list of NFL teams

Usage
   cfbd_draft_teams()

Value
   cfbd_draft_teams() - A data frame with 4 variables:
   nfl_location: character. NFL team location (city).
   nfl_nickname: integer  NFL team nickname (mascot).
   nfl_display_name: integer  NFL team display name (usually more neat/complete).
   nfl_logo: character  URL for NFL team logo.
Examples

try(cfbd_draft_teams())

cfbd_drives  CFBD Drives Endpoint

Description

Get college football game drives

Usage

cfbd_drives(
  year,
  season_type = "regular",
  week = NULL,
  team = NULL,
  offense_team = NULL,
  defense_team = NULL,
  conference = NULL,
  offense_conference = NULL,
  defense_conference = NULL
)

Arguments

year  (Integer required): Year, 4 digit format (YYYY)
season_type  (String default regular): Select Season Type: regular, postseason, or both
week  (Integer optional): Week - values from 1-15, 1-14 for seasons pre-playoff, i.e. 2013 or earlier
team  (String optional): D-I Team
offense_team  (String optional): Offense D-I Team
defense_team  (String optional): Defense D-I Team
conference  (String optional): DI Conference abbreviation - Select a valid FBS conference Conference abbreviations P5: ACC, B12, B1G, SEC, PAC Conference abbreviations G5 and FBS Independents: CUSA, MAC, MWC, Ind, SBC, AAC
offense_conference  (String optional): Offense DI Conference abbreviation - Select a valid FBS conference Conference abbreviations P5: ACC, B12, B1G, SEC, PAC Conference abbreviations G5 and FBS Independents: CUSA, MAC, MWC, Ind, SBC, AAC
defense_conference  (String optional): Defense DI Conference abbreviation - Select a valid FBS conference Conference abbreviations P5: ACC, B12, B1G, SEC, PAC Conference abbreviations G5 and FBS Independents: CUSA, MAC, MWC, Ind, SBC, AAC
Value

cfbd_drives() - A data frame with 23 variables as follows:

- **offense**: character. Drive offense.
- **offense_conference**: character. Drive offense’s conference.
- **defense**: character. Drive defense.
- **defense_conference**: character. Drive defense’s conference.
- **game_id**: integer. Unique game identifier - game_id.
- **drive_id**: character. Unique drive identifier - drive_id.
- **drive_number**: integer. Drive number in game.
- **scoring**: logical. Drive ends in a score.
- **start_period**: integer. Period (or Quarter) in which the drive starts.
- **start_yardline**: integer. Yard line at the drive start.
- **start_yards_to_goal**: integer. Yards-to-Goal at the drive start.
- **end_period**: integer. Period (or Quarter) in which the drive ends.
- **end_yardline**: integer. Yard line at drive end.
- **end_yards_to_goal**: integer. Yards-to-Goal at drive end.
- **plays**: integer. Number of drive plays.
- **yards**: integer. Total drive yards.
- **drive_result**: character. Result of the drive description.
- **is_home_offense**: logical. Flag for if the offense on the field is the home offense.
- **start_offense_score**: numeric. Offense score at the start of the drive.
- **start_defense_score**: numeric. Defense score at the start of the drive.
- **end_offense_score**: numeric. Offense score at the end of the drive.
- **end_defense_score**: numeric. Defense score at the end of the drive.
- **time_minutes_start**: integer. Minutes at drive start.
- **time_seconds_start**: integer. Seconds at drive start.
- **time_minutes_end**: integer. Minutes at drive end.
- **time_seconds_end**: integer. Seconds at drive end.
- **time_minutes_elapsed**: double. Minutes elapsed during drive.
- **time_seconds_elapsed**: integer. Seconds elapsed during drive.

Examples

try(cfbd_drives(2018, week = 1, team = "TCU"))

try(cfbd_drives(2018, team = "Texas A&M", defense_conference = "SEC"))
CFBD Games Endpoint Overview

Description
Get results, statistics and information for games

cfbd_game_box_advanced(): Get game advanced box score information.
cfbd_game_player_stats(): Get results information from games.
cfbd_game_team_stats(): Get team statistics by game.
cfbd_game_info(): Get results information from games.
cfbd_game_weather(): Get weather from games.
cfbd_game_records(): Get team records by year.
cfbd_calendar(): Get calendar of weeks by season.
cfbd_game_media(): Get game media information (TV, radio, etc).

Details
Get game advanced box score information.:  
cfbd_game_box_advanced(game_id = 40114233)

Get player statistics by game:
cfbd_game_player_stats(2018, week = 15, conference = "Ind")
cfbd_game_player_stats(2013, week = 1, team = "Florida State", category = "passing")

Get team records by year:
cfbd_game_records(2018, team = "Notre Dame")
cfbd_game_records(2013, team = "Florida State")

Get team statistics by game:
cfbd_game_team_stats(2019, team = "LSU")
cfbd_game_team_stats(2013, team = "Florida State")

Get results information from games.:  
cfbd_game_info(2018, week = 1)
cfbd_game_info(2018, week = 7, conference = "Ind")

# 7 OTs LSU @ TAMU  
cfbd_game_info(2018, week = 13, team = "Texas A&M", quarter_scores = TRUE)
### Get weather from games.
```
cfbd_game_weather(2018, week = 1)
cfbd_game_info(2018, week = 7, conference = "Ind")
```

Get calendar of weeks by season.:
```
cfbd_calendar(2019)
```

Get game media information (TV, radio, etc.).:
```
cfbd_game_media(2019, week = 4, conference = "ACC")
```

---

**cfbd_game_box_advanced**

Get game advanced box score information.

---

Description

Get game advanced box score information.

Usage
```
cfbd_game_box_advanced(game_id, long = FALSE)
```

Arguments

- **game_id** *(Integer required)*: Game ID filter for querying a single game. Can be found using the `cfbd_game_info()` function.
- **long** *(Logical default FALSE)*: Return the data in a long format.

Value

- `cfbd_game_box_advanced()`: A data frame with 2 rows and 69 variables:
  - **team**: character. Team name.
  - **plays**: double. Number of plays.
  - **ppa_overall_total**: double. Predicted points added (PPA) overall total.
  - **ppa_overall_quarter1**: double. Predicted points added (PPA) overall Q1.
  - **ppa_overall_quarter2**: double. Predicted points added (PPA) overall Q2.
  - **ppa_overall_quarter3**: double. Predicted points added (PPA) overall Q3.
  - **ppa_overall_quarter4**: double. Predicted points added (PPA) overall Q4.
  - **ppa_passing_total**: double. Passing predicted points added (PPA) total.
  - **ppa_passing_quarter1**: double. Passing predicted points added (PPA) Q1.
  - **ppa_passing_quarter2**: double. Passing predicted points added (PPA) Q2.
ppa_passing_quarter3: double. Passing predicted points added (PPA) Q3.
ppa_passing_quarter4: double. Passing predicted points added (PPA) Q4.
ppa_rushing_total: double. Rushing predicted points added (PPA) total.
ppa_rushing_quarter1: double. Rushing predicted points added (PPA) Q1.
ppa_rushing_quarter2: double. Rushing predicted points added (PPA) Q2.
ppa_rushing_quarter3: double. Rushing predicted points added (PPA) Q3.
ppa_rushing_quarter4: double. Rushing predicted points added (PPA) Q4.
cumulative_ppa_plays: double. Cumulative predicted points added (PPA) added total.
cumulative_ppa_overall_total: double. Cumulative predicted points added (PPA) total.
cumulative_ppa_overall_quarter1: double. Cumulative predicted points added (PPA) Q1.
cumulative_ppa_overall_quarter2: double. Cumulative predicted points added (PPA) Q2.
cumulative_ppa_overall_quarter3: double. Cumulative predicted points added (PPA) Q3.
cumulative_ppa_overall_quarter4: double. Cumulative predicted points added (PPA) Q4.
cumulative_ppa_passing_total: double. Cumulative passing predicted points added (PPA) total.
cumulative_ppa_passing_quarter1: double. Cumulative passing predicted points added (PPA) Q1.
cumulative_ppa_passing_quarter2: double. Cumulative passing predicted points added (PPA) Q2.
cumulative_ppa_passing_quarter3: double. Cumulative passing predicted points added (PPA) Q3.
cumulative_ppa_passing_quarter4: double. Cumulative passing predicted points added (PPA) Q4.
cumulative_ppa_rushing_total: double. Cumulative rushing predicted points added (PPA) total.
cumulative_ppa_rushing_quarter1: double. Cumulative rushing predicted points added (PPA) Q1.
cumulative_ppa_rushing_quarter2: double. Cumulative rushing predicted points added (PPA) Q2.
cumulative_ppa_rushing_quarter3: double. Cumulative rushing predicted points added (PPA) Q3.
cumulative_ppa_rushing_quarter4: double. Cumulative rushing predicted points added (PPA) Q4.
success_rates_overall_total: double. Success rates overall total.
success_rates_overall_quarter1: double. Success rates overall Q1.
success_rates_overall_quarter3: double. Success rates overall Q3.
success_rates_standard_downs_total: double. Success rates standard downs total.
success_rates_passing_downs_total: double. Success rates passing downs total.
success_rates_passing_downs_quarter1: double. Success rates passing downs Q1.
success_rates_passing_downs_quarter2: double. Success rates passing downs Q2.
success_rates_passing_downs_quarter3: double. Success rates passing downs Q3.
explosiveness_overall_total: double. Explosiveness rates overall total.
explosiveness_overall_quarter1: double. Explosiveness rates overall Q1.
explosiveness_overall_quarter2: double. Explosiveness rates overall Q2.
explosiveness_overall_quarter3: double. Explosiveness rates overall Q3.
rushing_power_success: double. Rushing power success rate.
rushing_stuff_rate: double. Rushing stuff rate.
rushing_line_yds: double. Rushing offensive line yards.
rushing_line_yds_avg: double. Rushing line yards average.
rushing_second_lvl_yds: double. Rushing second-level yards.
rushing_second_lvl_yds_avg: double. Average second level yards per rush.
rushing_open_field_yds: double. Rushing open field yards.
rushing_open_field_yds_avg: double. Average rushing open field yards average.
havoc_total: double. Total havoc rate.
havoc_front_seven: double. Front-7 players havoc rate.
havoc_db: double. Defensive back players havoc rate.
scoring_opps_opportunities: double. Number of scoring opportunities.
scoring_opps_points: double. Points on scoring opportunity drives.
scoring_opps_pts_per_opp: double. Points per scoring opportunity drives.
field_pos_avg_start: double. Average starting field position.
field_pos_avg_starting_predicted_pts: double. Average starting predicted points (PP) for the average starting field position.

Examples

try(cfbd_game_box_advanced(game_id = 40114233))
cfbd_game_info

Get results information from games.

Description

Get results information from games.

Usage

    cfbd_game_info(
        year,  
        week = NULL,  
        season_type = "regular",  
        team = NULL,  
        home_team = NULL,  
        away_team = NULL,  
        conference = NULL,  
        game_id = NULL, 
        quarter_scores = FALSE
    )

Arguments

- **year**: (Integer required) Year, 4 digit format (YYYY)
- **week**: (Integer optional) Week - values from 1-15, 1-14 for seasons pre-playoff (i.e. 2013 or earlier)
- **season_type**: (String default regular) Select Season Type: regular, postseason, or both
- **team**: (String optional) D-I Team
- **home_team**: (String optional) Home D-I Team
- **away_team**: (String optional) Away D-I Team
- **conference**: (String optional) Conference abbreviation - Select a valid FBS conference Conference abbreviations P5: ACC, B12, B1G, SEC, PAC Conference abbreviations G5 and FBS Independents: CUSA, MAC, MWC, Ind, SBC, AAC
- **game_id**: (Integer optional) Game ID filter for querying a single game
- **quarter_scores**: (Logical default FALSE) This is a parameter to return the list columns that give the score at each quarter: home_line_scores and away_line_scores. I have defaulted the parameter to false so that you will not have to go to the trouble of dropping it.

Value

    cfbd_game_info() - A data frame with 22 variables:

    - **game_id**: integer. Referencing game id.
    - **season**: integer. Season of the game.
week: integer. Game week.

season_type: character. Season type of the game.

start_date: character. Game date.

start_time_tbd: logical. TRUE/FALSE flag for if the game’s start time is to be determined.

neutral_site: logical. TRUE/FALSE flag for the game taking place at a neutral site.

conference_game: logical. TRUE/FALSE flag for this game qualifying as a conference game.

attendance: integer. Reported attendance at the game.

venue_id: integer. Referencing venue id.

venue: character. Venue name.

home_id: integer. Home team referencing id.

home_team: character. Home team name.


home_points: integer. Home team points.

home_post_win_prob: character. Home team post-game win probability.

home_pregame_elo: character. Home team pre-game ELO rating.

home_postgame_elo: character. Home team post-game ELO rating.

away_id: integer. Away team referencing id.

away_team: character. Away team name.

away_conference: character. Away team conference.

away_points: integer. Away team points.

away_post_win_prob: character. Away team post-game win probability.

away_pregame_elo: character. Away team pre-game ELO rating.

away_postgame_elo: character. Away team post-game ELO rating.

excitement_index: character. Game excitement index.

highlights: character. Game highlight urls.

notes: character. Game notes.

Examples

try(cfbd_game_info(2018, week = 7, conference = "Ind"))
cfbd_game_media

Get game media information (TV, radio, etc).

Description

Get game media information (TV, radio, etc).

Usage

```r
cfbd_game_media(
    year,
    week = NULL,
    season_type = "both",
    team = NULL,
    conference = NULL,
    media_type = NULL
)
```

Arguments

- **year** *(Integer required)*: Year, 4 digit format (YYYY)
- **week** *(Integer optional)*: Week, values from 1-15, 1-14 for seasons pre-playoff (i.e. 2013 or earlier)
- **season_type** *(String default both)*: Select Season Type, regular, postseason, or both
- **team** *(String optional)*: D-I Team
- **conference** *(String optional)*: Conference abbreviation - Select a valid FBS conference Conference abbreviations P5: ACC, B12, B1G, SEC, PAC Conference abbreviations G5 and FBS Independents: CUSA, MAC, MWC, Ind, SBC, AAC
- **media_type** *(String optional)*: Media type filter: tv, radio, web, ppv, or mobile

Value

- **cfbd_game_media()** - A data frame with 13 variables:
  - game_id: integer. Referencing game id.
  - season: integer. Season of the game.
  - week: integer. Game week.
  - season_type: character. Season type of the game.
  - start_time: character. Game start time.
  - is_start_time_tbd: logical. TRUE/FALSE flag for if the start time is still to be determined.
  - home_team: character. Home team of the game.
  - away_team: character. Away team of the game.
away_conference: character. Conference of the away team.
tv: list. TV broadcast networks.
web: list. Web viewing platforms carrying the game.

Examples

```r
try(cfbd_game_media(2019, week = 4, conference = "ACC"))
```

---

cfbd_game_player_stats

**Get player statistics by game**

Description

**Get player statistics by game**

Usage

```r
cfbd_game_player_stats(
  year,
  week = NULL,
  season_type = "regular",
  team = NULL,
  conference = NULL,
  category = NULL,
  game_id = NULL
)
```

Arguments

- **year** *(Integer required): Year, 4 digit format(YYYY)*
- **week** *(Integer optional): Week - values from 1-15, 1-14 for seasons pre-playoff (i.e. 2013 or earlier)*
- **season_type** *(String default regular): Select Season Type: regular or postseason*
- **team** *(String optional): D-I Team*
- **conference** *(String optional): Conference abbreviation - Select a valid FBS conference Conference abbreviations P5: ACC, B12, B1G, SEC, PAC Conference abbreviations G5 and FBS Independents: CUSA, MAC, MWC, Ind, SBC, AAC*
- **category** *(String optional): Category filter (e.g defensive) Offense: passing, receiving, rushing Defense: defensive, fumbles, interceptions Special Teams: punting, puntReturns, kicking, kickReturns*
- **game_id** *(Integer optional): Game ID filter for querying a single game Can be found using the cfbd_game_info() function*
Value

cfbd_game_player_stats() - A data frame with 32 variables:

game_id: integer. Referencing game id.
team: character. Team name.
conference: character. Conference of the team.
home_away: character. Flag for if the team was the home or away team.
points: integer. Team points.
category: character. Statistic category.
athlete_id: character. Athlete referencing id.
name: character. Player name.
c_att: character. Completions - Pass attempts.
yds: double. Statistic yardage.
avg: double. Average per statistic.
td: double. Touchdowns scored.
int: double. Interceptions thrown.
qbr: double. Quarterback rating.
car: double. Number of rushing carries.
long: double. Longest carry/reception of the game.
rec: double. Number of pass receptions.
no: double. Player number.
fg: character. Field goal attempts in the game.
pct: double. Field goal percentage in the game.
xp: character. Extra points kicked in the game.
pts: double. Total kicking points in the game.
tb: double. Touchbacks (for kicking) in the game.
in_20: double. Punts inside the 20 yardline in the game.
fum: double. Player fumbles in the game.
lost: double. Player fumbles lost in the game.
tot: double. Total tackles in the game.
solo: double. Solo tackles in the game.
sacks: double. Total sacks in the game.
tfl: double. Total tackles for loss in the game.
pd: double. Total passes defensed in the game.
qb_hur: double. Total quarterback hurries in the game.

Examples

try(cfbd_game_player_stats(year = 2020, week = 15, team = "Alabama"))

try(cfbd_game_player_stats(2013, week = 1, team = "Florida State", category = "passing"))
**Description**

Get team records by year

**Usage**

```r
cfbd_game_records(year, team = NULL, conference = NULL)
```

**Arguments**

- `year` *(Integer optional): Year, 4 digit format (YYYY)*
- `team` *(String optional): Team - Select a valid team, D1 football*
- `conference` *(String optional): DI Conference abbreviation - Select a valid FBS conference Conference abbreviations P5: ACC, B12, B1G, SEC, PAC Conference abbreviations G5 and FBS Independents: CUSA, MAC, MWC, Ind, SBC, AAC*

**Value**

```r
cfbd_game_records() - A data frame with 21 variables:
```

- `year`: integer. Season of the games.
- `team`: character. Team name.
- `conference`: character. Conference of the team.
- `division`: character. Division in the conference of the team.
- `expected_wins`: numeric. Expected number of wins based on post-game win probability.
- `total_games`: integer. Total number of games played.
- `total_wins`: integer. Total wins.
- `total_losses`: integer. Total losses.
- `total_ties`: integer. Total ties.
- `home_wins`: integer. Total home wins.
- `home_losses`: integer. Total home losses.
- `home_ties`: integer. Total home ties.
- `away_wins`: integer. Total away wins.
- `away_losses`: integer. Total away losses.
- `away_ties`: integer. Total away ties.
Examples

```
try(cfbd_game_records(2018, team = "Notre Dame"))
try(cfbd_game_records(2013, team = "Florida State"))
```

**cfbd_game_team_stats**  Get team statistics by game

**Description**

Get team statistics by game

**Usage**

```r
cfbd_game_team_stats(
  year,
  week = NULL,
  season_type = "regular",
  team = NULL,
  conference = NULL,
  game_id = NULL,
  rows_per_team = 1
)
```

**Arguments**

- **year**  (*Integer required): Year, 4 digit format (YYYY)
- **week**  (*Integer optional): Week - values range from 1-15, 1-14 for seasons pre-playoff, i.e. 2013 or earlier
- **season_type**  (*String default: regular): Select Season Type - regular, postseason, or both
- **team**  (*String optional): D-I Team
- **game_id**  (*Integer optional): Game ID filter for querying a single game Can be found using the `cfbd_game_info()` function
- **rows_per_team**  (*Integer default 1): Both Teams for each game on one or two row(s), Options: 1 or 2
Value

cfbd_game_team_stats() - A data frame with 78 variables:

game_id: integer. Referencing game id.
school: character. Team name.
conference: character. Conference of the team.
home_away: character. Home/Away Flag.
opponent: character. Opponent team name.
opponent_conference: character. Conference of the opponent team.
points: integer. Team points.
total_yards: character. Team total yards.
net_passing_yards: character. Team net passing yards.
completion_attempts: character. Team completion attempts.
passing_tds: character. Team passing touchdowns.
yards_per_pass: character. Team game yards per pass.
passes_intercepted: character. Team passes intercepted.
interception_yards: character. Interception yards.
interception_tds: character. Interceptions returned for a touchdown.
rushing_attempts: character. Team rushing attempts. see also: ESTABLISH IT.
rushing_yards: character. Team rushing yards.
rush_tds: character. Team rushing touchdowns.
yards_per_rush_attempt: character. Team yards per rush attempt.
first_downs: character. First downs earned by the team.
third_down_eff: character. Third down efficiency.
fourth_down_eff: character. Fourth down efficiency.
punt_returns: character. Team punt returns.
punt_return_yards: character. Team punt return yards.
punt_return_tds: character. Team punt return touchdowns.
kick_return_yards: character. Team kick return yards.
kick_return_tds: character. Team kick return touchdowns.
kick_returns: character. Team kick returns.
kicking_points: character. Team points from kicking the ball.
fumbles_recovered: character. Team fumbles recovered.
fumbles_lost: character. Team fumbles lost.
total_fumbles: character. Team total fumbles.
tackles: character. Team tackles.
tackles_for_loss: character. Team tackles for a loss.
sacks: character. Team sacks.
qb_hurries: character. Team QB hurries.
interceptions: character. Team interceptions.
passes_deflected: character. Team passes deflected.
turnovers: character. Team turnovers.
defensive_tds: character. Team defensive touchdowns.
total_penalties_yards: character. Team total penalty yards.
possession_time: character. Team time of possession.
points_allowed: integer. Points for the opponent.
total_yards_allowed: character. Opponent total yards.
net_passing_yards_allowed: character. Opponent net passing yards.
completion_attempts_allowed: character. Opponent completion attempts.
passing_tds_allowed: character. Opponent passing TDs.
yards_per_pass_allowed: character. Opponent yards per pass allowed.
passes_intercepted_allowed: character. Opponent passes intercepted.
interception_yards_allowed: character. Opponent interception yards.
interception_tds_allowed: character. Opponent interception TDs.
rushing_attempts_allowed: character. Opponent rushing attempts.
rushing_yards_allowed: character. Opponent rushing yards.
rush_tds_allowed: character. Opponent rushing touchdowns.
yards_per_rush_attempt_allowed: character. Opponent rushing yards per attempt.
first_downs_allowed: character. Opponent first downs.
third_down_eff_allowed: character. Opponent third down efficiency.
fourth_down_eff_allowed: character. Opponent fourth down efficiency.
punt_returns_allowed: character. Opponent punt returns.
punt_return_yards_allowed: character. Opponent punt return yards.
punt_return_tds_allowed: character. Opponent punt return touchdowns.
kick_return_yards_allowed: character. Opponent kick return yards.
kick_return_tds_allowed: character. Opponent kick return touchdowns.
kick_returns_allowed: character. Opponent kick returns.
kicking_points_allowed: character. Opponent points from kicking.
fumblesRecovered_allowed: character. Opponent fumbles recovered.
fumbles_lost_allowed: character. Opponent fumbles lost.
total_fumbles_allowed: character. Opponent total number of fumbles.
tackles_allowed: character. Opponent tackles.
tackles_for_loss_allowed: character. Opponent tackles for loss.
sacks_allowed: character. Opponent sacks.
qb_hurries_allowed: character. Opponent quarterback hurries.
interceptions_allowed: character. Opponent interceptions.
passes_deflected_allowed: character. Opponent passes deflected.
turnovers_allowed: character. Opponent turnovers.
defensive_tds_allowed: character. Opponent defensive touchdowns.
total_penalties_yards_allowed: character. Opponent total penalty yards.
possession_time_allowed: character. Opponent time of possession.

Examples

try(cfbd_game_team_stats(2019, team = "LSU"))
try(cfbd_game_team_stats(2013, team = "Florida State"))

cfbd_game_weather      Get weather from games.

Description

Get weather from games.

Usage

cfbd_game_weather(
  year,
  week = NULL,
  season_type = "regular",
  team = NULL,
  conference = NULL
)

Arguments

year (Integer required): Year, 4 digit format(YYYY)
week (Integer optional): Week - values from 1-15, 1-14 for seasons pre-playoff (i.e. 2013 or earlier)
season_type (String default regular): Select Season Type: regular, postseason, or both
team (String optional): D-I Team
conference (String optional): Conference abbreviation - Select a valid FBS conference Conference abbreviations P5: ACC, B12, B1G, SEC, PAC Conference abbreviations G5 and FBS Independents: CUSA, MAC, MWC, Ind, SBC, AAC
**Value**

`cfbd_game_weather()` - A data frame with 23 variables:

- `game_id`: integer. Referencing game id.
- `season`: integer. Season of the game.
- `week`: integer. Game week.
- `season_type`: character. Season type of the game.
- `start_date`: character. Game date.
- `start_time_tbd`: logical. TRUE/FALSE flag for if the game’s start time is to be determined.
- `game_indoors`: logical. TRUE/FALSE flag for if the game is indoors
- `home_team`: character. Home team name.
- `away_team`: character. Away team name.
- `venue_id`: integer. Referencing venue id.
- `venue`: character. Venue name.
- `temperature`: integer. Temperature.
- `dew_point`: integer. Dew Point.
- `wind_direction`: integer. Wind direction.
- `wind_speed`: integer. Wind Speed.
- `pressure`: integer. Pressure.
- `weather_condition`: character. Weather condition.

---

**CFBD Metrics Endpoint Overview**

**Description**

- `cfbd_metrics_ppa_games()`: Get team game averages for predicted points added (PPA).
- `cfbd_metrics_ppa_players_games()`: Get player game averages for predicted points added (PPA).
- `cfbd_metrics_ppa_players_season()`: Get player season averages for predicted points added (PPA).
- `cfbd_metrics_ppa_predicted()`: Calculate predicted points using Down and Distance.
- `cfbd_metrics_ppa_teams()`: Get team averages for predicted points added (PPA).
- `cfbd_metrics_wp_pregame()`: Get pre-game win probability data from CFBD API.
- `cfbd_metrics_wp()`: Get win probability chart data from CFBD API.
Get team game averages for predicted points added (PPA):

cfbd_metrics_ppa_games(year = 2019, team = "TCU")

Get player game averages for predicted points added (PPA):

cfbd_metrics_ppa_players_games(year = 2019, week = 3, team = "TCU")

Get player season averages for predicted points added (PPA):

cfbd_metrics_ppa_players_season(year = 2019, team = "TCU")

Get team averages for predicted points added (PPA):

cfbd_metrics_ppa_teams(year = 2019, team = "TCU")

Get pre-game and post-game win probability data from CFBD API:

cfbd_metrics_wp_pregame(year = 2019, week = 9, team = "Texas A&M")
cfbd_metrics_wp(game_id = 401012356)

Calculate predicted points using down and distance:

cfbd_metrics_ppa_predicted(down = 1, distance = 10)
**Arguments**

- **year** *(Integer required): Year, 4 digit format (YYYY)*
- **week** *(Integer optional): Week - values range from 1-15, 1-14 for seasons pre-playoff, i.e. 2013 or earlier*
- **team** *(String optional): D-I Team*
- **conference** *(String optional): Conference abbreviation - Select a valid FBS conference Conference abbreviations P5: ACC, B12, B1G, SEC, PAC Conference abbreviations G5 and FBS Independents: CUSA, MAC, MWC, Ind, SBC, AAC*
- **excl_garbage_time** *(Logical default FALSE): Select whether to exclude Garbage Time (TRUE or FALSE)*

**Value**

cfbd_metrics_ppa_games() - A data frame with 18 variables:

- **game_id**: integer. Referencing game id.
- **season**: integer. Season of the game.
- **week**: integer. Game week of the season.
- **conference**: character. Conference of the team.
- **team**: character. Team name.
- **opponent**: character. Team Opponent.
- **off_overall**: character. Offense overall predicted points added (PPA).
- **off_passing**: character. Offense passing predicted points added (PPA).
- **off_rushing**: character. Offense rushing predicted points added (PPA).
- **off_first_down**: character. Offense 1st down predicted points added (PPA).
- **off_second_down**: character. Offense 2nd down predicted points added (PPA).
- **off_third_down**: character. Offense 3rd down predicted points added (PPA).
- **def_overall**: character. Defense overall predicted points added (PPA).
- **def_passing**: character. Defense passing predicted points added (PPA).
- **def_rushing**: character. Defense rushing predicted points added (PPA).
- **def_first_down**: character. Defense 1st down predicted points added (PPA).
- **def_second_down**: character. Defense 2nd down predicted points added (PPA).
- **def_third_down**: character. Defense 3rd down predicted points added (PPA).

**Examples**

```r
try(cfbd_metrics_ppa_games(year = 2019, team = "TCU"))
```
cfbd_metrics_ppa_players_games

Get player game averages for predicted points added (PPA)

Description

Get player game averages for predicted points added (PPA)

Usage

```r
cfbd_metrics_ppa_players_games(
  year = NULL,
  week = NULL,
  team = NULL,
  position = NULL,
  athlete_id = NULL,
  threshold = NULL,
  excl_garbage_time = FALSE
)
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>year</td>
<td>(Integer required): Year, 4 digit format (YYYY)</td>
</tr>
<tr>
<td>week</td>
<td>(Integer optional): Week - values range from 1-15, 1-14 for seasons pre-playoff, i.e. 2013 or earlier</td>
</tr>
<tr>
<td>team</td>
<td>(String optional): D-I Team. Required if year not provided.</td>
</tr>
<tr>
<td>position</td>
<td>(string optional): Position abbreviation of the player you are searching for. Position Group - options include:</td>
</tr>
<tr>
<td>athlete_id</td>
<td>(Integer optional): Athlete ID filter for querying a single athlete Can be found using the <code>cfbd_player_info()</code> function.</td>
</tr>
<tr>
<td>excl_garbage_time</td>
<td>(Logical default FALSE): Select whether to exclude Garbage Time (TRUE or FALSE)</td>
</tr>
</tbody>
</table>

Value

`cfbd_metrics_ppa_players_games()` - A data frame with 9 variables:

- **season**: integer. Season of the game.
- **week**: integer. Game week of the season.
name: **character**. Athlete name.
position: **character**. Athlete position.
team: **character**. Team name.
opponent: **character**. Team Opponent name.
avg_PPA_all: **double**. Average overall predicted points added (PPA).
avg_PPA_pass: **double**. Average passing predicted points added (PPA).
avg_PPA_rush: **double**. Average rushing predicted points added (PPA).

Examples

```r
try(cfbd_metrics_ppa_players_games(year = 2019, week = 3, team = "TCU"))
```

---

cfbd_metrics_ppa_players_season

*Get player season averages for predicted points added (PPA)*

Description

*Get player season averages for predicted points added (PPA)*

Usage

```r
cfbd_metrics_ppa_players_season(
  year = NULL,
  team = NULL,
  conference = NULL,
  position = NULL,
  athlete_id = NULL,
  threshold = NULL,
  excl_garbage_time = FALSE
)
```

Arguments

- **year**: *(Integer required): Year, 4 digit format (YYYY)*
- **team**: *(String optional): D-I Team*
- **position**: *(string optional): Position abbreviation of the player you are searching for. Position Group - options include:
  - Offense: QB, RB, FB, TE, OL, G, OT, C, WR*
• Defense: DB, CB, S, LB, DE, DT, NT, DL
• Special Teams: K, P, LS, PK

athlete_id  (Integer optional): Athlete ID filter for querying a single athlete Can be found using the \texttt{cfbd\_player\_info()} function.


excl\_garbage\_time  (Logical default FALSE): Select whether to exclude Garbage Time (TRUE or FALSE)

\textbf{Value}

\texttt{cfbd\_metrics\_ppa\_players\_season()} - A data frame with 23 variables:

\begin{itemize}
  \item \texttt{season}: integer. Season.
  \item \texttt{athlete\_id}: character. Athlete referencing id.
  \item \texttt{name}: character. Athlete name.
  \item \texttt{position}: character. Athlete Position.
  \item \texttt{team}: character. Team name.
  \item \texttt{conference}: character. Team conference.
  \item \texttt{countable\_plays}: integer. Number of plays which can be counted.
  \item \texttt{avg\_PPA\_all}: double. Average overall predicted points added (PPA).
  \item \texttt{avg\_PPA\_pass}: double. Average passing predicted points added (PPA).
  \item \texttt{avg\_PPA\_rush}: double. Average rushing predicted points added (PPA).
  \item \texttt{avg\_PPA\_first\_down}: double. Average 1st down predicted points added (PPA).
  \item \texttt{avg\_PPA\_second\_down}: double. Average 2nd down predicted points added (PPA).
  \item \texttt{avg\_PPA\_third\_down}: double. Average 3rd down predicted points added (PPA).
  \item \texttt{avg\_PPA\_standard\_downs}: double. Average standard down predicted points added (PPA).
  \item \texttt{avg\_PPA\_passing\_downs}: double. Average passing down predicted points added (PPA).
  \item \texttt{total\_PPA\_all}: double. Total overall predicted points added (PPA).
  \item \texttt{total\_PPA\_pass}: double. Total passing predicted points added (PPA).
  \item \texttt{total\_PPA\_rush}: double. Total rushing predicted points added (PPA).
  \item \texttt{total\_PPA\_first\_down}: double. Total 1st down predicted points added (PPA).
  \item \texttt{total\_PPA\_second\_down}: double. Total 2nd down predicted points added (PPA).
  \item \texttt{total\_PPA\_third\_down}: double. Total 3rd down predicted points added (PPA).
  \item \texttt{total\_PPA\_standard\_downs}: double. Total standard down predicted points added (PPA).
  \item \texttt{total\_PPA\_passing\_downs}: double. Total passing down predicted points added (PPA).
\end{itemize}

\textbf{Examples}

\begin{verbatim}
try(cfbd_metrics_ppa_players_season(year = 2019, team = "TCU"))
\end{verbatim}
cfbd_metrics_ppa_predicted

*Calculate predicted points using down and distance*

**Description**

Calculate predicted points using down and distance

**Usage**

```r
cfbd_metrics_ppa_predicted(down, distance)
```

**Arguments**

- `down` *(Integer required): Down filter*
- `distance` *(Integer required): Distance filter*

**Value**

- `cfbd_metrics_ppa_predicted()` - A data frame with 2 variables:
  - `yard_line`: *integer*. Yards to goal
  - `predicted_points`: *character*. Predicted points at in that down-distance-yardline scenario

**Examples**

```r
try(cfbd_metrics_ppa_predicted(down = 1, distance = 10))
try(cfbd_metrics_ppa_predicted(down = 3, distance = 10))
```

----------

cfbd_metrics_ppa_teams

*Get team averages for predicted points added (PPA)*

**Description**

Get team averages for predicted points added (PPA)

**Usage**

```r
cfbd_metrics_ppa_teams(
  year = NULL,
  team = NULL,
  conference = NULL,
  excl_garbage_time = FALSE
)
```
Arguments

- **year** *(Integer optional): Year, 4 digit format (YYYY)*
- **team** *(String optional): D-I Team*
- **conference** *(String optional): Conference name - select a valid FBS conference Conference names P5: ACC, Big 12, Big Ten, SEC, Pac-12 Conference names G5 and FBS Independents: Conference USA, Mid-American, Mountain West, FBS Independents, American Athletic*
- **excl_garbage_time** *(Logical default FALSE): Select whether to exclude Garbage Time (TRUE or FALSE)*

Value

`cfbd_metrics_ppa_teams()` - A data frame with 21 variables:

- `season`: integer.
- `conference`: character.
- `team`: character.
- `off_overall`: character. Offense overall predicted points added (PPA).
- `off_passing`: character. Offense passing predicted points added (PPA).
- `off_rushing`: character. Offense rushing predicted points added (PPA).
- `off_first_down`: character. Offense 1st down predicted points added (PPA).
- `off_second_down`: character. Offense 2nd down predicted points added (PPA).
- `off_third_down`: character. Offense 3rd down predicted points added (PPA).
- `off_cumulative_total`: character. Offense cumulative total predicted points added (PPA).
- `off_cumulative_passing`: character. Offense cumulative total passing predicted points added (PPA).
- `off_cumulative_rushing`: character. Offense cumulative total rushing predicted points added (PPA).
- `def_overall`: character. Defense overall predicted points added (PPA).
- `def_passing`: character. Defense passing predicted points added (PPA).
- `def_rushing`: character. Defense rushing predicted points added (PPA).
- `def_first_down`: character. Defense 1st down predicted points added (PPA).
- `def_second_down`: character. Defense 2nd down predicted points added (PPA).
- `def_cumulative_total`: character. Defense cumulative total predicted points added (PPA).
- `def_cumulative_passing`: character. Defense cumulative total passing predicted points added (PPA).
- `def_cumulative_rushing`: character. Defense cumulative total rushing predicted points added (PPA).
Examples

```
try(cfbd_metrics_ppa_teams(year = 2019, team = "TCU"))
```

---

**cfbd_metrics_wp**

Get win probability chart data from API

**Description**

Get win probability chart data from API

**Usage**

```
cfbd_metrics_wp(game_id)
```

**Arguments**

- **game_id** *(Integer required)*: Game ID filter for querying a single game. Can be found using the `cfbd_game_info()` function.

**Value**

- `cfbd_metrics_wp()` - A data frame with 16 variables:
  - `play_id`: character. Play referencing id.
  - `play_text`: character. A text description of the play.
  - `home_id`: integer. Home team referencing id.
  - `home`: character. Home team name.
  - `away_id`: integer. Away team referencing id.
  - `away`: character. Away team name.
  - `home_ball`: logical. Home team has the ball.
  - `home_score`: integer. Home team score.
  - `away_score`: integer. Away team score.
  - `down`: integer. Down of the play.
  - `distance`: integer. Distance to the sticks (to 1st down marker of goal-line in goal-to-go situations).
  - `away_win_prob`: double. Away team win probability.
  - `play_number`: integer. Game play number.
  - `yard_line`: integer. Yard line of the play (0-100 yards).

**Examples**

```
try(cfbd_metrics_wp(game_id = 401012356))
```
cfbd_metrics_wp_pregame

Get pre-game win probability data from API

Description

Get pre-game win probability data from API

Usage

```r
cfbd_metrics_wp_pregame(
  year = NULL,
  week = NULL,
  team = NULL,
  season_type = "regular"
)
```

Arguments

- `year` 
  `(Integer optional): Year, 4 digit format (YYYY)`
- `week` 
  `(Integer optional): Week - values from 1-15, 1-14 for seasons pre-playoff, i.e. 2013 or earlier`
- `team` 
  `(String optional): D-I Team`
- `season_type` 
  `(String default regular): Select Season Type: regular or postseason`

Value

`cfbd_metrics_wp_pregame()` - A data frame with 9 variables:

- `season`: `integer`. Season of game.
- `season_type`: `character`. Season type of game.
- `week`: `integer`. Game week of the season.
- `game_id`: `integer`. Referencing game id.
- `home_team`: `character`. Home team name.
- `away_team`: `character`. Away team name.
- `home_win_prob`: `double`. Home win probability - pre-game prediction.
- `away_win_prob`: `double`. Away win probability - pre-game prediction.

Examples

```r
try(cfbd_metrics_wp_pregame(year = 2019, week = 9, team = "Texas A&M"))
```
Description

Extract college football (D-I) play by play Data - for plays

Usage

cfbd_pbp_data(
    year,  # Select year, (example: 2018)
    season_type = "regular",  # Select Season Type (regular, postseason, both)
    week = 1,  # Select week, this is optional (also numeric)
    team = NULL,  # Select team name (example: Texas, Texas A&M, Clemson)
    play_type = NULL,  # Select play type (example: see the cfbd_play_type_df)
    epa_wpa = FALSE,  # Logical parameter (TRUE/FALSE) to return the Expected Points Added/Win
    ...  # Probability Added variables
)

Arguments

- year: Select year, (example: 2018)
- season_type: Select Season Type (regular, postseason, both)
- week: Select week, this is optional (also numeric)
- team: Select team name (example: Texas, Texas A&M, Clemson)
- play_type: Select play type (example: see the cfbd_play_type_df)
- epa_wpa: Logical parameter (TRUE/FALSE) to return the Expected Points Added/Win Probability Added variables
- ...: Additional arguments passed to an underlying function.

Value

A data frame with 351 variables:

- season: double.
- wk: double.
- id_play: character.
- game_id: integer.
- game_play_number: double.
- half_play_number: double.
- pos_team: character.
- def_pos_team: character.
half: integer.
period: integer.
clock.minutes: integer.
clock.seconds: integer.
play_type: character.
play_text: character.
down: double.
distance: double.
yards_to_goal: double.
yards_gained: double.
penalty_1st_conv: logical.
new_series: double.
change_of_pos_team: double.
lag_change_of_pos_team: double.
orig_play_type: character.
lead_play_type: character.
lag_play_type: character.
lag_play_type2: character.
lag_play_type3: character.
row: integer.
drive_play_number: double.
drive_event_number: double.
lag_play_text: character.
lag_play_text2: character.
lead_play_text: character.
firstD_by_kickoff: double.
firstD_by_poss: double.
firstD_by_penalty: double.
firstD_by_yards: double.
lag_first_by_penalty: double.
lag_first_by_penalty2: double.
lag_first_by_yards: double.
lag_first_by_yards2: double.
first_by_penalty: double.
first_by_yards: double.
play_after_turnover: double.
lag_change_of_poss: double.
lag_kickoff_play: double.
lag_punt: double.
lag_scoring_play: double.
lag_turnover_vec: double.
lag_downs_turnover: double.
lag_defense_score_play: double.
EPA: double.
ep_before: double.
ep_after: double.
wp: double.
wp_before: double.
wp_after: double.
def_wp_before: double.
def_wp_after: double.
penalty_detail: character.
yds_penalty: double.
downs_turnover: double.
turnover: double.
pos_score_diff_start: double.
pos_score_pts: double.
pos_team_score: integer.
def_pos_team_score: integer.
log_ydstogo: double.
ExpScoreDiff: double.
ExpScoreDiff_Time_Ratio: double.
half_clock.minutes: double.
TimeSecsRem: double.
adj_TimeSecsRem: double.
Goal_To_Go: logical.
Under_two: logical.
home: character.
away: character.
home_wp_before: double.
away_wp_before: double.
home_wp_after: double.
away_wp_after: double.
end_of_half: double.
pos_team_receives_2H_kickoff: double.
lead_pos_team: character.
lag_pos_team: character.
Under_three: logical.
down_end: integer.
distance_end: double.
log_ydstogo_end: double.
yards_to_goal_end: double.
TimeSecsRem_end: double.
Goal_To_Go_end: logical.
Under_two_end: logical.
def_EPA: double.
offense_score_play: double.
defense_score_play: double.
ppa: character.
yard_line: integer.
scoring: logical.
pos_team_timeouts_rem_before: double.
def_pos_team_timeouts_rem_before: double.
pos_team_timeouts: integer.
def_pos_team_timeouts: integer.
pos_score_diff: integer.
pos_score_diff_start_end: double.
offense_play: character.
defense_play: character.
offense_receives_2H_kickoff: double.
change_of_poss: double.
score_pts: double.
score_diff_start: double.
score_diff: integer.
offense_score: integer.
defense_score: integer.
offense_conference: character.
defense_conference: character.
off_timeout_called: character.
def_timeout_called: character.
offense_timeouts: integer.
defense_timeouts: integer.
off_timeouts_rem_before: double.
def_timeouts_rem_before: double.
rusher_player_name: character.
yds_rushed: double.
passer_player_name: character.
receiver_player_name: character.
yds_receiving: double.
yds_sacked: double.
sack_players: character.
sack_player_name: character.
sack_player_name2: character.
pass_breakup_player_name: character.
interception_player_name: character.
yds_int_return: double.
fumble_player_name: character.
fumble_forced_player_name: character.
fumbleRecovered_player_name: character.
yds_fumble_return: double.
punter_player_name: character.
yds_punted: double.
punt_returner_player_name: character.
yds_punt_return: double.
yds_punt_gained: double.
punt_block_player_name: character.
punt_block_return_player_name: character.
fg_kicker_player_name: character.
yds_fg: double.
fg_block_player_name: character.
fg_return_player_name: character.
kickoff_player_name: character.
yds_kickoff: double.
kickoff_returner_player_name: character.
yds_kickoff_return: double.
new_id: double.
orig_drive_number: integer.
drive_number: integer.
drive_result_detailed: character.
new_drive_pts: double.
drive_id: double.
drive_result: character.
drive_start_yards_to_goal: double.
drive_end_yards_to_goal: integer.
drive_yards: integer.
drive_scoring: double.
drive_pts: double.
drive_start_period: integer.
drive_end_period: integer.
drive_time_minutes_start: integer.
drive_time_seconds_start: integer.
drive_time_minutes_end: integer.
drive_time_seconds_end: integer.
drive_time_minutes_elapsed: integer.
drive_time_seconds_elapsed: integer.
drive_numbers: double.
number_of_drives: double.
pts_scored: double.
drive_result_detailed_flag: character.
drive_result2: character.
drive_num: double.
lag_drive_result_detailed: character.
lead_drive_result_detailed: character.
lag_new_drive_pts: double.
id_drive: character.
rush: double.
rush_td: double.
pass: double.
pass_td: double.
completion: double.
pass_attempt: double.
target: double.
sack_vec: double.
sack: double.
int: double.
int_td: double .
turnover_vec: double .
turnover_vec_lag: double .
turnover_indicator: double .
kickoff_play: double .
receives_2H_kickoff: double .
missing_yard_flag: logical .
scoring_play: double .
td_play: double .
touchdown: double .
safety: double .
fumble_vec: double .
kickoff_tb: double .
kickoff_onside: double .
kickoff_oob: double .
kickoff_fair_catch: double .
kickoff_downed: double .
kickoff_safety: double .
kick_play: double .
punt: double .
punt_play: double .
punt_tb: double .
punt_oob: double .
punt_fair_catch: double .
punt_downed: double .
punt_safety: double .
punt_blocked: double .
penalty_safety: double .
fg_inds: double .
fg_made: logical .
fg_make_prob: double .
home_EPA: double .
away_EPA: double .
home_EPA_rush: double .
away_EPA_rush: double .
home_EPA_pass: double .
away_EPA_pass: double .
total_home_EPA: double.
total_away_EPA: double.
total_home_EPA_rush: double.
total_away_EPA_rush: double.
total_home_EPA_pass: double.
total_away_EPA_pass: double.
net_home_EPA: double.
net_away_EPA: double.
net_home_EPA_rush: double.
net_away_EPA_rush: double.
net_home_EPA_pass: double.
net_away_EPA_pass: double.
success: double.
epa_success: double.
rz_play: double.
scoring_opp: double.
middle_8: logical.
stuffed_run: double.
spread: double.
formatted_spread: character.
No_Score_before: double.
FG_before: double.
Opp_FG_before: double.
Opp_Safety_before: double.
Opp_TD_before: double.
Safety_before: double.
TD_before: double.
No_Score_after: double.
FG_after: double.
Opp_FG_after: double.
Opp_Safety_after: double.
Opp_TD_after: double.
Safety_after: double.
TD_after: double.
penalty_flag: logical.
penalty_declined: logical.
penalty_no_play: logical.
penalty_offset: logical.
penalty_text: logical.
penalty_play_text: character.
lead_wp_before2: double.
wpa_half_end: double.
wpa_base: double.
wpa_base_nxt: double.
wpa_change: double.
wpa_change_nxt: double.
wpa_base_ind: double.
wpa_base_nxt_ind: double.
wpa_change_ind: double.
wpa_change_nxt_ind: double.
lead_wp_before: double.
lead_pos_team2: character.
lag_change_of_pos_team2: double.
lag_punt2: double.
lag_score_diff: double.
lag_offense_play: character.
lead_offense_play: character.
lead_offense_play2: character.
lag_pos_score_diff: double.
lag_off_timeouts: double.
lag_def_timeouts: double.
lag_TimeSecsRem2: double.
lag_TimeSecsRem: double.
lead_TimeSecsRem: double.
lead_TimeSecsRem2: double.
lag_yards_to_goal2: integer.
lag_yards_to_goal: integer.
lead_yards_to_goal: integer.
lead_yards_to_goal2: integer.
lag_down2: double.
lag_down: double.
lead_down: double.
lead_down2: double.
lead_distance: double.
lead_distance2: integer.
lead_play_type2: character.
lead_play_type3: character.
lag_ep_before3: double.
lag_ep_before2: double.
lag_ep_before: double.
lead_ep_before: double.
lead_ep_before2: double.
lag_ep_after: double.
lag_ep_after2: double.
lag_ep_after3: double.
lead_ep_after: double.
lead_ep_after2: double.
play_number: integer.
over_under: double.
event: double.
game_event_number: double.
game_row_number: integer.
half_event: double.
half_event_number: double.
half_row_number: integer.
lag_distance3: integer.
lag_distance2: integer.
lag_distance: integer.
lag_yards_gained3: integer.
lag_yards_gained2: integer.
lag_yards_gained: integer.
lead_yards_gained: integer.
lead_yards_gained2: integer.
lag_play_text3: character.
lead_play_text2: character.
lead_play_text3: character.
lag_change_of_poss2: double.
lag_change_of_poss3: double.
lag_change_of_pos_team3: double.
lag_kickoff_play2: double.
lag_kickoff_play3: double.
Cfbd_play

Cfbd_play

Description

College football plays data

cfbd_plays(): CFBD’s college football play-by-play.
cfbd_play_stats_player(): Gets player info associated by play.
cfbd_play_stats_types(): Gets CFBD play stat types.
cfbd_play_types(): Gets CFBD play types.

Details

Pull first 3 weeks of 2020 season using cfbd_plays():

year_vector <- 2020
week_vector <- 1:3
weekly_year_df <- expand.grid(year = year_vector, week = week_vector)
tictoc::tic()
year_split <- split(weekly_year_df, weekly_year_df$year)
for (i in 1:length(year_split)) {
  i <- 1
  progressr::with_progress({
    year_split[i] <- year_split[i] %>%
      dplyr::mutate(
        pbp = purrr::map2(
          .x = year, .y = week, cfbd_plays, season_type = "both"
        )
      )
  })
}
Gets player info associated by play:
cfbd_play_stats_player(game_id = 401110722)

Gets CFBD play stat types:
cfbd_play_stats_types()

Gets CFBD play types:
cfbd_play_types()

cfb_players

<table>
<thead>
<tr>
<th>Description</th>
<th>CFBD Players Endpoint Overview</th>
</tr>
</thead>
<tbody>
<tr>
<td>Player information lookup:</td>
<td></td>
</tr>
<tr>
<td>cfbd_player_info(): Player information search.</td>
<td></td>
</tr>
<tr>
<td>cfbd_player_returning(): Player returning production.</td>
<td></td>
</tr>
<tr>
<td>cfbd_player_usage(): Player usage.</td>
<td></td>
</tr>
</tbody>
</table>

Player information lookup:

| cfbd_player_info(search_term = "James", position = "DB", team = "Florida State", year = 2017) |
| cfbd_player_info(search_term = "Lawrence", team = "Clemson") |

Get player returning production:

| cfbd_player_returning(year = 2019, team = "Florida State") |

Get player usage metrics:

| cfbd_player_usage(year = 2019, position = "WR", team = "Florida State") |
cfbd_player_info

**Player information lookup**

**Description**

Player information lookup

**Usage**

```r
cfbd_player_info(search_term, position = NULL, team = NULL, year = NULL)
```

**Arguments**

- `search_term` *(String required):* Search term for the player you are trying to look up
- `position` *(string optional):* Position of the player you are searching for. Position Group - options include:
  - Offense: QB, RB, FB, TE, OL, G, OT, C, WR
  - Defense: DB, CB, S, LB, DE, DT, NT, DL
  - Special Teams: K, P, LS, PK
- `team` *(String optional):* Team - Select a valid team, D1 football
- `year` *(Integer optional):* Year, 4 digit format (YYYY). If left NULL, API default will only provide results for most recent year of final rosters: 2020

**Value**

```r
cfbd_player_info() - A data frame with 12 variables:

- `athlete_id`: character. Unique player identifier athlete_id.
- `team`: character. Team of the player.
- `name`: character. Player name.
- `first_name`: character. Player first name.
- `last_name`: character. Player last name.
- `weight`: integer. Player weight.
- `jersey`: integer. Player jersey number.
- `position`: character. Player position.
- `home_town`: character. Player home town.
- `team_color`: character. Player team color.
- `team_color_second`: character. Player team secondary color.
```
Examples

```r
try(cfbd_player_info(search_term = "James", position = "DB", team = "Florida State", year = 2017))
try(cfbd_player_info(search_term = "Lawrence", team = "Clemson"))
```

---

**cfbd_player_returning**  
Get player returning production

### Description

Get player returning production

### Usage

```r
cfbd_player_returning(year = 2019, team = NULL, conference = NULL)
```

### Arguments

- **year**: (Integer required, default 2019): Year, 4 digit format (YYYY).
- **team**: (String optional): Team - Select a valid team, D1 football
- **conference**: (String optional): Conference abbreviation - Select a valid FBS conference Conference abbreviations P5: ACC, B12, B1G, SEC, PAC Conference abbreviations G5 and FBS Independents: CUSA, MAC, MWC, Ind, SBC, AAC

### Value

```r
cfbd_player_returning()  
- A data frame with 15 variables:
  - season:integer. Returning player season.
  - team:character. Team name.
  - total_ppa:double. Total predicted points added returning.
  - total_passing_ppa:double. Total passing predicted points added returning.
  - total_receiving_ppa:double. Total receiving predicted points added returning.
  - total_rushing_ppa:double. Total rushing predicted points added returning.
  - percent_ppa:double. Percentage of prior year’s predicted points added returning.
  - percent_passing_ppa:double. Percentage of prior year’s passing predicted points added returning.
  - percent_receiving_ppa:double. Percentage of prior year’s receiving predicted points added returning.
  - percent_rushing_ppa:double. Percentage of prior year’s rushing predicted points added returning.
```
**usage:** double.

**passing_usage:** double.

**receiving_usage:** double.

**rushing_usage:** double.

**Examples**

```
try(cfbd_player_returning(year = 2019, team = "Florida State"))
```

---

**cfbd_player_usage**

Get player usage metrics

**Description**

Get player usage metrics

**Usage**

```
cfbd_player_usage(
    year = 2019,
    team = NULL,
    conference = NULL,
    position = NULL,
    athlete_id = NULL,
    excl_garbage_time = FALSE
)
```

**Arguments**

- **year** *(Integer required, default 2019): Year, 4 digit format (YYYY).*
- **team** *(String optional): Team - Select a valid team, D1 football.*
- **conference** *(String optional): Conference abbreviation - Select a valid FBS conference Conference abbreviations P5: ACC, B12, B1G, SEC, PAC Conference abbreviations G5 and FBS Independents: CUSA, MAC, MWC, Ind, SBC, AAC.*
- **position** *(string optional): Position of the player you are searching for. Position Group - options include:
  - Offense: QB, RB, FB, TE, OL, G, OT, C, WR
  - Defense: DB, CB, S, LB, DE, DT, NT, DL
  - Special Teams: K, P, LS, PK
- **athlete_id** *(Integer optional): Athlete ID filter for querying a single athlete Can be found using the cfbd_player_info() function.*
- **excl_garbage_time** *(Logical default FALSE): Select whether to exclude Garbage Time (TRUE/FALSE)*
Value

cfbd_player_usage() - A data frame with 14 variables:

- **season**: integer. Player usage season.
- **athlete_id**: character. Referencing athlete id.
- **name**: character. Athlete name.
- **position**: character. Athlete position.
- **team**: character. Team name.
- **conference**: character. Conference of team.
- **usg_overall**: double. Player usage of overall offense.
- **usg_pass**: double. Player passing usage percentage.
- **usg_rush**: double. Player rushing usage percentage.
- **usg_1st_down**: double. Player first down usage percentage.
- **usg_2nd_down**: double. Player second down usage percentage.
- **usg_3rd_down**: double. Player third down usage percentage.
- **usg_standard_downs**: double. Player standard down usage percentage.
- **usg_passing_downs**: double. Player passing down usage percentage.

Examples

```
try(cfbd_player_usage(year = 2019, position = "WR", team = "Florida State"))
```

---

cfbd_plays Get college football play-by-play data.

Description

Get college football play-by-play data.

Usage

```
cfbd_plays(
  year = 2020,
  season_type = "regular",
  week = 1,
  team = NULL,
  offense = NULL,
  defense = NULL,
  conference = NULL,
  offense_conference = NULL,
  defense_conference = NULL,
  play_type = NULL
)
```
Arguments

- **year**: Select year, (example: 2018)
- **season_type**: Select Season Type (regular, postseason, both)
- **week**: Select week, this is optional (also numeric)
- **team**: Select team name (example: Texas, Texas A&M, Clemson)
- **offense**: Select offense name (example: Texas, Texas A&M, Clemson)
- **defense**: Select defense name (example: Texas, Texas A&M, Clemson)
- **conference**: Select conference name (example: ACC, B1G, B12, SEC, PAC, MAC, MWC, CUSA, Ind, SBC, AAC, Western, MVIAA, SWC, PCC, Big 6, etc.)
- **offense_conference**: Select conference name (example: ACC, B1G, B12, SEC, PAC, MAC, MWC, CUSA, Ind, SBC, AAC, Western, MVIAA, SWC, PCC, Big 6, etc.)
- **defense_conference**: Select conference name (example: ACC, B1G, B12, SEC, PAC, MAC, MWC, CUSA, Ind, SBC, AAC, Western, MVIAA, SWC, PCC, Big 6, etc.)
- **play_type**: Select play type (example: see the cfbd_play_type_df)

Value

cfbd_plays() - A data frame with 29 columns:

- **play_id**: character. Referencing play id.
- **offense**: character. Offense on the field.
- **offense_conference**: character. Conference of the offense on the field.
- **defense**: character. Defense on the field.
- **defense_conference**: character. Conference of the defense on the field.
- **home**: character. Home team.
- **away**: character. Away team.
- **offense_score**: integer. Offense's post-play score.
- **defense_score**: integer. Defense's post-play score.
- **game_id**: integer. Referencing game id.
- **drive_id**: character. Referencing drive id.
- **drive_number**: integer. Drive number in the game.
- **play_number**: integer. Play number in the game.
- **period**: integer. Game period (quarter).
- **offense_timeouts**: integer. Timeouts for the offense at the end of the play.
- **defense_timeouts**: integer. Timeouts for the defense at the end of the play.
- **yard_line**: integer. Yard line (~0-50) of the play.
- **yards_to_goal**: integer. Yards to the goal line (~0-100).
- **down**: integer. Down of the play.
distance: integer. Distance to the sticks, i.e. 1st down or goal-line in goal-to-go situations.
scoiring: logical. Scoring play flag.
yards_gained: integer. Yards net gained by the offense on the play.
play_type: character. Categorical label of the type of the play.
play_text: character. A text description of the play.
ppa: character. Predicted Points Added (calculated by CFBD).
clock.minutes: integer. Minutes left on the clock.
clock.seconds: integer. Seconds left on the clock.

Examples

```r
## Not run:
try(cfbd_plays(year = 2021, week = 1))
## End(Not run)
```

cfbd_play_stats_player

Gets player info associated by play

Description

Gets player info associated by play

Usage

```r
cfbd_play_stats_player(
    year = NULL,
    week = NULL,
    team = NULL,
    game_id = NULL,
    athlete_id = NULL,
    stat_type_id = NULL,
    season_type = "regular"
)
```

Arguments

- **year** *(Integer optional): Year, 4 digit format (YYYY)*
- **week** *(Integer optional): Week - values from 1-15, 1-14 for seasons pre-playoff, i.e. 2013 or earlier*
- **team** *(String optional): D-I Team*
- **game_id** *(Integer optional): Game ID filter for querying a single game Can be found using the cfbd_game_info() function*
athlete_id (Integer optional): Athlete ID filter for querying a single athlete. Can be found using the `cfbd_player_info()` function.

stat_type_id (Integer optional): Stat Type ID filter for querying a single stat type. Can be found using the `cfbd_play_stats_types()` function.

season_type (String default regular): Select Season Type: regular, postseason, or both.

Value

cfbd_play_stats_player() - A data frame with 54 variables:

- play_id: character. Referencing play id.
- game_id: integer. Referencing game id.
- season: integer. Season of the play.
- week: integer. Week of the play.
- opponent: character. Opponent of the offense on the play.
- team_score: integer. Offense team score.
- opponent_score: integer. Defense team score.
- drive_id: character. Referencing drive id.
- period: integer. Game period (quarter) of the play.
- yards_to_goal: integer. Yards to the goal line (~0-100).
- down: integer. Down of the play.
- distance: integer. Distance to the sticks, i.e. 1st down or goal-line in goal-to-go situations.
- reception_player: character. Pass receiver player name.
- reception_yds: integer. Reception yards.
- completion_player_id: character. Passing player reference id.
- completion_player: character. Passing player name.
- completion_yds: integer. Passing yards.
- rush_player: character. Rushing player name.
- interception_player_id: character. Intercepting player reference id.
- interception_player: character. Intercepting player name.
- interception_thrown_player_id: character. Interception passing player reference id.
- interception_thrown_player: character. Interception passing player name.
- touchdown_player_id: character. Touchdown scoring player reference id.
- touchdown_player: character. Touchdown scoring player name.
touchdown_stat: integer. Touchdown scoring stat.
incompletion_player_id: character. Incomplete receiver player reference id.
incompletion_player: character. Incomplete receiver player name.
target_player_id: character. Targeted receiver player reference id.
target_player: character. Targeted receiver player name.
target_stat: integer. Target stat.
fumble_recovered_player_id: logical. Fumble recovering player reference id.
fumble_recovered_player: logical. Fumble recovering player name.
fumble_recovered_stat: logical. Fumble recovered stat.
fumble_forced_player_id: logical. Fumble forcing player reference id.
fumble_forced_player: logical. Fumble forcing player name.
fumble_forced_stat: logical. Fumble forced stat.
fumble_player_id: logical. Fumbling player reference id.
fumble_player: logical. Fumbling player name.
fumble_stat: logical. Fumble stat.
sack_player_id: character. Sacking player(s) reference id.
sack_player: character. Sacking player(s) name.
sack_taken_player: character. Sack taking player name.
pass_breakup_player: logical. Pass breakup player name.
pass_breakup_stat: logical. Pass breakup (PBU) stat.

Examples

try(cfbd_play_stats_player(game_id = 401110722))
### cfbd_play_stats_types

**Get college football mapping for play stats types**

**Description**

Get college football mapping for play stats types

**Usage**

```r
cfbd_play_stats_types()
```

**Value**

The `cfbd_play_stats_types()` function returns a data frame with 22 rows and 2 variables:

- `play_stat_type_id`: integer
  - Referencing play stat type ID.
- `name`: character
  - Type of play stats.

**Examples**

```r
try(cfbd_play_stats_types())
```

### cfbd_play_types

**Get college football mapping for play types**

**Description**

Get college football mapping for play types

**Usage**

```r
cfbd_play_types()
```

**Value**

The `cfbd_play_types()` function returns a data frame with 48 rows and 3 variables:

- `play_type_id`: integer
  - Referencing play type id.
- `text`: character
  - Play type description.
- `abbreviation`: character
  - Play type abbreviation used for function call

**Examples**

```r
try(cfbd_play_types())
```
cfbd_rankings  Get historical Coaches and AP poll data

Description

Get historical Coaches and AP poll data

Usage

cfbd_rankings(year, week = NULL, season_type = "regular")

Arguments

year (Integer required): Year, 4 digit format (YYYY)
week (Integer optional): Week, values from 1-15, 1-14 for seasons pre-playoff (i.e. 2013 or earlier)
season_type (String default regular): Season type - regular or postseason

Value

cfbd_rankings() - A data frame with 9 variables:

season: integer. Rankings season.
season_type: character. Season type of rankings.
week: integer. Week of rankings.
poll: character. Name of the poll.
rank: integer. Rank in the poll.
school: character. Team name.
conference: character. Conference of the team.
first_place_votes: integer. Number of first place votes.
points: integer. Total poll points.

Examples

try(cfbd_rankings(year = 2019, week = 12))
try(cfbd_rankings(year = 2018, week = 14))
try(cfbd_rankings(year = 2013, season_type = "postseason"))
CFBD Ratings and Rankings Endpoints Overview

Description

cfbd_rankings(): Gets Historical CFB poll rankings at a specific week.
cfbd_rankings_sp(): Get SP historical rating data.
cfbd_ratings_sp_conference(): Get SP conference-level historical rating data.
cfbd_ratings_srs(): Get SRS historical rating data.

Get historical Coaches and AP poll data:

cfbd_rankings(year = 2019, week = 12)
cfbd_rankings(year = 2018, week = 14)
cfbd_rankings(year = 2013, season_type = "postseason")

Get SP historical rating data:
At least one of year or team must be specified for the function to run

cfbd_rankings_sp(year = 2018)
cfbd_rankings_sp(team = "Texas A&M")
cfbd_rankings_sp(year = 2019, team = "Texas")

Get conference level SP historical rating data:

cfbd_ratings_sp_conference(year = 2019)
cfbd_ratings_sp_conference(year = 2012, conference = "SEC")
cfbd_ratings_sp_conference(year = 2016, conference = "ACC")

Get SRS historical rating data:
At least one of year or team must be specified for the function to run

cfbd_ratings_srs(year = 2019, team = "Texas")
cfbd_ratings_srs(year = 2018, conference = "SEC")

Get Elo historical rating data:
Acquire the CFBD calculated elo ratings data by team, year, week, and conference

cfbd_ratings_elo(year = 2019, team = "Texas")
cfbd_ratings_elo(year = 2018, conference = "SEC")
cfbd_ratings_elo  Get Elo historical rating data

Description
Acquire the CFBD calculated elo ratings data by team, year, week, and conference

Usage

cfbd_ratings_elo(year = NULL, week = NULL, team = NULL, conference = NULL)

Arguments

- **year** *(Integer optional):* Year, 4 digit format (YYYY)
- **week** *(Integer optional):* Maximum Week of ratings.
- **team** *(String optional):* D-I Team
- **conference** *(String optional):* Conference abbreviation - SRS information by conference
  Conference abbreviations P5: ACC, B12, B1G, SEC, PAC Conference abbreviations G5 and FBS Independents: CUSA, MAC, MWC, Ind, SBC, AAC

Value

*cfbd_ratings_elo()* - A data frame with 6 variables:

- **year**: integer. Season of the SRS rating.
- **team**: character. Team name.
- **conference**: character. Conference of the team.
- **division**: logical. Division in the conference for the team.
- **rating**: double. Simple Rating System (SRS) rating.
- **ranking**: integer. Simple Rating System ranking within the group returned.

Examples

try(cfbd_ratings_elo(year = 2019, team = "Texas"))

try(cfbd_ratings_elo(year = 2018, conference = "SEC"))
**cfbd_ratings_sp**  
Get SP historical rating data

**Description**
At least one of year or team must be specified for the function to run

**Usage**
```r
cfbd_ratings_sp(year = NULL, team = NULL)
```

**Arguments**
- `year` *(Integer optional): Year, 4 digit format (YYYY)*
- `team` *(String optional): D-I Team*

**Value**
- `cfbd_ratings_sp()` - A data frame with 26 variables:
  - **year**: integer. Season of the ratings.
  - **team**: character. Team name.
  - **conference**: character. Conference of the team.
  - **rating**: double. SP+ rating.
  - **ranking**: integer. Ranking in the SP+ ratings.
  - **second_order_wins**: logical. Total second-order wins - Not available for recent seasons.
  - **sos**: logical. Strength of schedule - Not available for recent seasons.
  - **offense_raking**: integer. Overall offense ranking.
  - **offense_rating**: double. Overall offense rating.
  - **offense_success**: logical. Offense success rating - Not available for recent seasons.
  - **offense_explosiveness**: logical. Offense explosiveness rating - Not available for recent seasons.
  - **offense_rushing**: logical. Offense rushing rating - Not available for recent seasons.
  - **offense_passing**: logical. Offense passing rating - Not available for recent seasons.
  - **offense_standard_downs**: logical. Offense standard downs rating - Not available for recent seasons.
  - **offense_passing_downs**: logical. Offensive passing downs rating - Not available for recent seasons.
  - **offense_run_rate**: logical. Offense rushing rate - Not available for recent seasons.
  - **offense_pace**: logical. Offense pace factor - Not available for recent seasons.
  - **defense_raking**: integer. Overall defense ranking.
  - **defense_rating**: double. Overall defense rating.
defense_explosiveness: logical. Defense explosiveness rating - Not available for recent seasons.
defense_rushing: logical. Defense rushing rating - Not available for recent seasons.
defense_passing: logical. Defense passing rating - Not available for recent seasons.
defense_passing_downs: logical. Defensive passing downs rating - Not available for recent seasons.
defense_havoc_total: logical. Total defensive havoc rate - Not available for recent seasons.
defense_havoc_front_seven: logical. Defense havoc rate from front 7 players - Not available for recent seasons.
defense_havoc_db: logical. Defense havoc rate from defensive backs - Not available for recent seasons.
special_teams_rating: double. Special teams rating.

Examples

```
try(cfbd_ratings_sp(year = 2018))
try(cfbd_ratings_sp(team = "Texas A&M"))
try(cfbd_ratings_sp(year = 2019, team = "Texas"))
```

cfbd_ratings_sp_conference

Get conference level SP historical rating data

Description

Get conference level SP historical rating data

Usage

```
cfbd_ratings_sp_conference(year = NULL, conference = NULL)
```

Arguments

- **year**: (Integer optional): Year, 4 digit format (YYYY)
- **conference**: (String optional): Conference abbreviation - S&P+ information by conference
  Conference abbreviations P5: ACC, B12, B1G, SEC, PAC Conference abbreviations G5 and FBS Independents: CUSA, MAC, MWC, Ind, SBC, AAC
Value

cfbd_ratings_sp_conference() - A data frame with 25 variables:

- year: integer. Season of the conference rating.
- rating: double. Conference SP+ rating.
- second_order_wins: logical. Second-order wins for the conference - Not available for recent seasons.
- sos: logical. Strength of schedule for the conference - Not available for recent seasons.
- offense_rating: double. Overall offense rating for the conference.
- offense_success: logical. Offense success rating for the conference - Not available for recent seasons.
- offense_explosiveness: logical. Offense explosiveness rating for the conference - Not available for recent seasons.
- offense_rushing: logical. Offense rushing rating for the conference - Not available for recent seasons.
- offense_passing: logical. Offense passing rating for the conference - Not available for recent seasons.
- offense_standard_downs: logical. Offense standard downs rating for the conference - Not available for recent seasons.
- offense_passing_downs: logical. Offensive passing downs rating for the conference - Not available for recent seasons.
- offense_run_rate: logical. Offense rushing rate for the conference - Not available for recent seasons.
- offense_pace: logical. Offense pace factor for the conference - Not available for recent seasons.
- defense_ranking: integer. Overall defense ranking for the conference.
- defense_rating: double. Overall defense rating for the conference.
- defense_explosiveness: logical. Defense explosiveness rating for the conference - Not available for recent seasons.
- defense_rushing: logical. Defense rushing rating for the conference - Not available for recent seasons.
- defense_passing: logical. Defense passing rating for the conference - Not available for recent seasons.
- defense_passing_downs: logical. Defensive passing downs rating for the conference - Not available for recent seasons.
- defense_havoc_total: logical. Total defensive havoc rate for the conference - Not available for recent seasons.
defense_havoc_front_seven: logical. Defense havoc rate from front 7 players for the conference - Not available for recent seasons.
defense_havoc_db: logical. Defense havoc rate from defensive backs for the conference - Not available for recent seasons.
special_teams_rating: double. Special teams rating for the conference.

Examples

try(cfbd_ratings_sp_conference(year = 2019))
try(cfbd_ratings_sp_conference(year = 2012, conference = "SEC"))
try(cfbd_ratings_sp_conference(year = 2016, conference = "ACC"))

cfbd_ratings_srs - Get SRS historical rating data

Description

At least one of year or team must be specified for the function to run

Usage

cfbd_ratings_srs(year = NULL, team = NULL, conference = NULL)

Arguments

year (Integer optional): Year, 4 digit format (YYYY)
team (String optional): D-I Team
conference (String optional): Conference abbreviation - SRS information by conference
Conference abbreviations P5: ACC, B12, B1G, SEC, PAC Conference abbreviations G5 and FBS Independents: CUSA, MAC, MWC, Ind, SBC, AAC

Value

cfbd_ratings_srs() - A data frame with 6 variables:

year: integer. Season of the SRS rating.
team: character. Team name.
conference: character. Conference of the team.
division: logical. Division in the conference for the team.
rating: double. Simple Rating System (SRS) rating.
ranking: integer. Simple Rating System ranking within the group returned.
Examples

```r
try(cfbd_ratings_srs(year = 2019, team = "Texas"))
try(cfbd_ratings_srs(year = 2018, conference = "SEC"))
```

---

**cfbd_recruiting**

**CFB Recruiting Endpoint Overview**

**Description**

- `cfbd_recruiting_player()`: Get college football player recruiting information for a single year with filters available for team, recruit type, state and position.
- `cfbd_recruiting_position()`: Get college football position group recruiting information.
- `cfbd_recruiting_team()`: Get college football recruiting team rankings information.

**Get player recruiting rankings:**
Get college football player recruiting information for a single year with filters available for team, recruit type, state and position.

```r
cfbd_recruiting_player(2018, team = "Texas")
cfbd_recruiting_player(2016, recruit_type = "JUCO")
cfbd_recruiting_player(2020, recruit_type = "HighSchool", position = "OT", state = "FL")
```

**Get college football position group recruiting information:**

```r
cfbd_recruiting_position(2018, team = "Texas")
cfbd_recruiting_position(2016, 2020, team = "Virginia")
cfbd_recruiting_position(2015, 2020, conference = "SEC")
```

**Get college football recruiting team rankings information:**

```r
cfbd_recruiting_team(2018, team = "Texas")
cfbd_recruiting_team(2016, team = "Virginia")
cfbd_recruiting_team(2016, team = "Texas A&M")
cfbd_recruiting_team(2011)
```
Details

Gets CFB team recruiting ranks with filters available for year and team. At least one of `year` or `team` must be specified for the function to run.

If you would like CFB recruiting information for players, please see the `cfbd_recruiting_player()` function.

If you would like to get CFB recruiting information based on position groups during a time period for all FBS teams, please see the `cfbd_recruiting_position()` function.

`cfbd_recruiting_player()` - At least one of `year` or `team` must be specified for the function to run.
`cfbd_recruiting_position()` - If only `start_year` is provided, function will get CFB recruiting information based on position groups during that year for all FBS teams.

---

`cfbd_recruiting_player`

Get player recruiting rankings

---

Description

Get player recruiting rankings

Usage

```r
cfdb_recruiting_player(
  year = NULL,
  team = NULL,
  recruit_type = "HighSchool",
  state = NULL,
  position = NULL
)
```

Arguments

- `year` *(Integer optional):* Year, 4 digit format (YYYY) - Minimum: 2000, Maximum: 2020 currently
- `team` *(String optional):* D-I Team
- `recruit_type` *(String optional):* default API return is 'HighSchool', other options include 'JUCO' or 'PrepSchool' - For position group information
- `state` *(String optional):* Two letter State abbreviation
- `position` *(String optional):* Position Group - options include:
  - Offense: 'PRO', 'DUAL', 'RB', 'FB', 'TE', 'OT', 'OG', 'OC', 'WR'
  - Defense: 'CB', 'S', 'OLB', 'ILB', 'WDE', 'SDE', 'DT'
  - Special Teams: 'K', 'P'
**Value**

`cfbd_recruiting_player()` - A data frame with 14 variables:

- **id**: integer. Referencing id - 247Sports.
- **athlete_id**: athlete referencing id.
- **recruit_type**: character. High School, Prep School, or Junior College.
- **year**: integer. Recruit class year.
- **ranking**: integer. Recruit Ranking.
- **name**: character. Recruit Name.
- **school**: character. School recruit attended.
- **committed_to**: character. School the recruit is committed to.
- **position**: character. Recruit position.
- **height**: double. Recruit height.
- **weight**: integer. Recruit weight.
- **stars**: integer. Recruit stars.
- **rating**: double. 247 composite rating.
- **city**: character. Hometown of the recruit.
- **state_province**: character. Hometown state of the recruit.
- **country**: character. Hometown country of the recruit.
- **hometown_info_latitude**: character. Hometown latitude.
- **hometown_info_longitude**: character. Hometown longitude.
- **hometown_info_fips_code**: character. Hometown FIPS code.

**Examples**

```r
try(cfbd_recruiting_player(2018, team = "Texas"))

try(cfbd_recruiting_player(2016, recruit_type = "JUCO"))

try(cfbd_recruiting_player(2020, recruit_type = "HighSchool", position = "OT", state = "FL"))
```
cfbd_recruiting_position

Get college football position group recruiting information.

Description

Get college football position group recruiting information.

Usage

cfbd_recruiting_position(
    start_year = NULL,
    end_year = NULL,
    team = NULL,
    conference = NULL
)

Arguments

start_year (Integer optional): Start Year, 4 digit format (YYYY). Note: 2000 is the minimum value
end_year (Integer optional): End Year, 4 digit format (YYYY). Note: 2020 is the maximum value currently
team (String optional): Team - Select a valid team, D-I football
conference (String optional): Conference abbreviation - Select a valid FBS conference - Conference abbreviations P5: ACC, B12, B1G, SEC, PAC Conference abbreviations G5 and FBS Independents: CUSA, MAC, MWC, Ind, SBC, AAC

Value

cfbd_recruiting_position() - A data frame with 7 variables:

team: character. Recruiting team.
conference: character. Recruiting team conference.
position_group: character. Position group of the recruits.
avg_rating: double. Average rating of the recruits in the position group.
total_rating: double. Sum of the ratings of the recruits in the position group.
commits: integer. Number of commits in the position group.
avg_stars: double. Average stars of the recruits in the position group.
Examples

try(cfbd_recruiting_position(2018, team = "Texas"))

try(cfbd_recruiting_position(2016, 2020, team = "Virginia"))

try(cfbd_recruiting_position(2015, 2020, conference = "SEC"))

---

cfbd_recruiting_team  Get college football recruiting team rankings information.

Description

Get college football recruiting team rankings information.

Usage

cfbd_recruiting_team(year = NULL, team = NULL)

Arguments

year  (Integer optional): Recruiting Class Year, 4 digit format (YYYY). Note: 2000 is the minimum value

team  (String optional): Team - Select a valid team, D1 football

Value

cfbd_recruiting_team() - A data frame with 4 variables:

year: integer. Recruiting class year.
rank: integer. Team Recruiting rank.
team: character. Recruiting Team.
points: character. Team talent points.

Examples

try(cfbd_recruiting_team(2018, team = "Texas"))

try(cfbd_recruiting_team(2016, team = "Virginia"))

try(cfbd_recruiting_team(2016, team = "Texas A&M"))

try(cfbd_recruiting_team(2011))
Get Transfer Portal Data

Usage

cfbd_recruiting_transfer_portal(year)

Arguments

year (Integer required): Year of the offseason (2021 would return transfer portal data starting from the end of the 2020 season), 4 digit format (YYYY).

Value

cfbd_recruiting_transfer_portal() - A data frame with 11 variables:

  season:integer  Season of transfer.
  first_name:character  Player's first name.
  last_name:character  Player's last name.
  position:character  Player position.
  origin:character  original team.
  destination:character  new team.
  transfer_date:character  Date of transfer.
  rating:character  Player's 247 transfer rating.
  stars:integer  Player's star rating.
  eligibility:character  Player's eligibility status.

Examples

try(cfbd_recruiting_transfer_portal(year = 2021))
CFBD Stats Endpoint Overview

Description

cfb.stats_categories(): Get college football mapping for stats categories.
cfb.stats_season_team(): Get season statistics by team.
cfb.stats_season_advanced(): Get season advanced statistics by team.
cfb.stats_game_advanced(): Get game advanced stats.
cfb.stats_season_player(): Get season statistics by player.

Details

Get game advanced stats:

cfb.stats_game_advanced(year = 2018, week = 12, team = "Texas A&M")
cfb.stats_game_advanced(2019, team = "LSU")
cfb.stats_game_advanced(2013, team = "Florida State")

Get season advanced statistics by team:

cfb.stats_season_advanced(2019, team = "LSU")

Get season statistics by player:

cfb.stats_season_player(year = 2018, conference = "B12", start_week = 1, end_week = 7)
cfb.stats_season_player(2019, team = "LSU", category = "passing")
cfb.stats_season_player(2013, team = "Florida State", category = "passing")

Get season statistics by team:

cfb.stats_season_team(year = 2018, conference = "B12", start_week = 1, end_week = 8)
cfb.stats_season_team(2019, team = "LSU")
cfb.stats_season_team(2013, team = "Florida State")

Get stats categories:
This function identifies all Stats Categories identified in the regular stats endpoint.
cfb.stats_categories()
**cfbd_stats_categories**  Get stats categories

**Description**
This function identifies all Stats Categories identified in the regular stats endpoint.

**Usage**
```r
cfbd_stats_categories()
```

**Value**
```r
cfbd_stats_categories()  
A data frame with 38 values:

name  Statistics Categories
```

**Examples**
```r
try(cfbd_stats_categories())
```

---

**cfbd_stats_game_advanced**  Get game advanced stats

**Description**
Get game advanced stats

**Usage**
```r
cfbd_stats_game_advanced(
  year,
  week = NULL,
  team = NULL,
  opponent = NULL,
  excl_garbage_time = FALSE,
  season_type = "both"
)
```
Arguments

- **year**  
  (Integer required): Year, 4 digit format (YYYY)

- **week**  
  (Integer optional): Week - values from 1-15, 1-14 for seasons pre-playoff (i.e. 2013 or earlier)

- **team**  
  (String optional): D-I Team

- **opponent**  
  (String optional): Opponent D-I Team

- **excl_garbage_time**  
  (Logical default FALSE): Select whether to exclude Garbage Time (TRUE/FALSE)

- **season_type**  
  (String default both): Select Season Type: regular, postseason, or both.

Value

**cfbd_stats_game_advanced()** - A data frame with 60 variables:

- **game_id**: integer. Referencing game id.
- **week**: integer. Game week of the season.
- **team**: character. Team name.
- **opponent**: character. Opponent team name.
- **off_plays**: integer. Offense plays in the game.
- **off_drives**: integer. Offense drives in the game.
- **off_ppa**: double. Offense predicted points added (PPA).
- **off_total_ppa**: double. Offense total predicted points added (PPA).
- **off_success_rate**: double. Offense success rate.
- **off_explosiveness**: double. Offense explosiveness rate.
- **off_power_success**: double. Offense power success rate.
- **off_stuff_rate**: double. Opponent stuff rate.
- **off_line_yds**: double. Offensive line yards.
- **off_line_yds_total**: integer. Offensive line yards total.
- **off_second_lvl_yds**: double. Offense second-level yards.
- **off_second_lvl_yds_total**: integer. Offense second-level yards total.
- **off_open_field_yds**: integer. Offense open field yards.
- **off_open_field_yds_total**: integer. Offense open field yards total.
- **off_standard_downs_ppa**: double. Offense standard downs predicted points added (PPA).
- **off_standard_downs_success_rate**: double. Offense standard downs success rate.
- **off_standard_downs_explosiveness**: double. Offense standard downs explosiveness rate.
- **off_passing_downs_ppa**: double. Offense passing downs predicted points added (PPA).
- **off_passing_downs_success_rate**: double. Offense passing downs success rate.
- **off_passing_downs_explosiveness**: double. Offense passing downs explosiveness rate.
- **off_rushing_plays_ppa**: double. Offense rushing plays predicted points added (PPA).
off_rushing_plays_total_ppa: double. Offense rushing plays total predicted points added (PPA).
off_rushing_plays_explosiveness: double. Offense rushing plays explosiveness rate.
off_passing_plays_ppa: double. Offense passing plays predicted points added (PPA).
off_passing_plays_total_ppa: double. Offense passing plays total predicted points added (PPA).
off_passing_plays_explosiveness: double. Offense passing plays explosiveness rate.
def_plays: integer. Defense plays in the game.
def_ppa: double. Defense predicted points added (PPA).
def_total_ppa: double. Defense total predicted points added (PPA).
def_explosiveness: double. Defense explosiveness rate.
def_stuff_rate: double. Opponent stuff rate.
def_line_yds: double. Offensive line yards.
def_line_yds_total: integer. Offensive line yards total.
def_second_lvl_yds_total: integer. Defense second-level yards total.
def_open_field_yds: integer. Defense open field yards.
def_open_field_yds_total: integer. Defense open field yards total.
def_passing_downs_ppa: double. Defense passing downs predicted points added (PPA).
def_passing_downs_explosiveness: double. Defense passing downs explosiveness rate.
def_rushing_plays_ppa: double. Defense rushing plays predicted points added (PPA).
def_rushing_plays_total_ppa: double. Defense rushing plays total predicted points added (PPA).
def_rushing_plays_explosiveness: double. Defense rushing plays explosiveness rate.
def_passing_plays_ppa: double. Defense passing plays predicted points added (PPA).
def_passing_plays_total_ppa: double. Defense passing plays total predicted points added (PPA).
def_passing_plays_explosiveness: double. Defense passing plays explosiveness rate.
Examples

try(cfbd_stats_game_advanced(year = 2018, week = 12, team = "Texas A&M"))

try(cfbd_stats_game_advanced(2019, team = "LSU"))

try(cfbd_stats_game_advanced(2013, team = "Florida State"))

description

Get season advanced statistics by team

Usage

cfbd_stats_season_advanced(
  year,                            # (Integer required): Year, 4 digit format (YYYY)
  team = NULL,                    # (String optional): D-I Team
  excl_garbage_time = FALSE,      # (Logical default FALSE): Select whether to exclude Garbage Time (TRUE/FALSE)
  start_week = NULL,              # (Integer optional): Starting Week - values range from 1-15, 1-14 for seasons pre-playoff, i.e. 2013 or earlier
  end_week = NULL                 # (Integer optional): Ending Week - values range from 1-15, 1-14 for seasons pre-playoff, i.e. 2013 or earlier
)

Arguments

year

(team)
excl_garbage_time

(start_week)

Value

cfbd_stats_season_advanced() - A data frame with 81 variables:

season: integer. Season of the statistics.

team: character. Team name.

conference: character. Conference of the team.
off_plays: integer. Offense plays in the game.
off_drives: integer. Offense drives in the game.
off_ppa: double. Offense predicted points added (PPA).
off_total_ppa: double. Offense total predicted points added (PPA).
off_explosiveness: double. Offense explosiveness rate.
off_power_success: double. Offense power success rate.
off_stuff_rate: double. Offense rushing stuff rate.
off_line_yds: double. Offensive line yards.
off_line_yds_total: integer. Offensive line yards total.
off_second_lvl_yds: double. Offense second-level yards.
off_second_lvl_yds_total: integer. Offense second-level yards total.
off_open_field_yds: integer. Offense open field yards.
off_open_field_yds_total: integer. Offense open field yards total.
off_total_opportunities: integer. Offense opportunities.
off_pts_per_opp: double. Offense points per scoring opportunity.
off_field_pos_avg_start: double. Offense starting average field position.
off_field_pos_avg_predicted_points: double. Offense starting average field position predicted points (PP).
off_havoc_total: double. Offense havoc rate total.
off_havoc_front_seven: double. Offense front-7 havoc rate.
off_havoc_db: double. Offense defensive back havoc rate.
off_standard_downs_rate: double. Offense standard downs rate.
off_standard_downs_ppa: double. Offense standard downs predicted points added (PPA).
off_standard_downs_explosiveness: double. Offense standard downs explosiveness rate.
off_passing_downs_rate: double. Offense passing downs rate.
off_passing_downs_ppa: double. Offense passing downs predicted points added (PPA).
off_passing_downs_explosiveness: double. Offense passing downs explosiveness rate.
off_rushing_plays_rate: double. Offense rushing plays rate.
off_rushing_plays_ppa: double. Offense rushing plays predicted points added (PPA).
off_rushing_plays_total_ppa: double. Offense rushing plays total predicted points added (PPA).
off_rushing_plays_explosiveness: double. Offense rushing plays explosiveness rate.
off_passing_plays_rate: double. Offense passing plays rate.
off_passing_plays_ppa: double. Offense passing plays predicted points added (PPA).
off_passing_plays_total_ppa: double. Offense passing plays total predicted points added (PPA).
off_passing_plays_explosiveness: double. Offense passing plays explosiveness rate.
def_plays: integer. Defense plays in the game.
def_ppa: double. Defense predicted points added (PPA).
def_total_ppa: double. Defense total predicted points added (PPA).
def_explosiveness: double. Defense explosiveness rate.
def_stuff_rate: double. Defense rushing stuff rate.
def_line_yds: double. Defense Offensive line yards allowed.
def_line_yds_total: integer. Defense Offensive line yards total allowed.
def_second_lvl_yds_total: integer. Defense second-level yards total.
def_open_field_yds: integer. Defense open field yards.
def_open_field_yds_total: integer. Defense open field yards total.
def_total_opportunities: integer. Defense opportunities.
def_pts_per_opp: double. Defense points per scoring opportunity.
def_field_pos_avg_predicted_points: double. Defense starting average field position predicted points (PP).
def_havoc_total: double. Defense havoc rate total.
def_havoc_frontSeven: double. Defense front-7 havoc rate.
def_passing_downs_ppa: double. Defense passing downs predicted points added (PPA).
def_passing_downs_explosiveness: double. Defense passing downs explosiveness rate.
def_rushing_downs_ppa: double. Defense rushing plays predicted points added (PPA).
def_rushing_downs_total_ppa: double. Defense rushing plays total predicted points added (PPA).
def_rushing_plays_explosiveness: double. Defense rushing plays explosiveness rate.
def_passing_plays_ppa: double. Defense passing plays predicted points added (PPA).
def_passing_plays_total_ppa: double. Defense passing plays total predicted points added (PPA).
def_passing_plays_explosiveness: double. Defense passing plays explosiveness rate.

Examples

try(cfbd_stats_season_advanced(2019, team = "LSU"))

cfbd_stats_season_player

Get season statistics by player

Description

Get season statistics by player

Usage

cfbd_stats_season_player(
    year,
    season_type = "regular",
    team = NULL,
    conference = NULL,
    start_week = NULL,
    end_week = NULL,
    category = NULL
)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>year</td>
<td>(Integer required): Year, 4 digit format (YYYY)</td>
</tr>
<tr>
<td>season_type</td>
<td>(String default: regular): Select Season Type - regular, postseason, or both</td>
</tr>
<tr>
<td>team</td>
<td>(String optional): D-I Team</td>
</tr>
<tr>
<td>conference</td>
<td>(String optional): Conference abbreviation - Select a valid FBS conference Conference abbreviations P5: ACC, B12, B1G, SEC, PAC Conference abbreviations G5 and FBS Independents: CUSA, MAC, MWC, Ind, SBC, AAC</td>
</tr>
<tr>
<td>start_week</td>
<td>(Integer optional): Starting Week - values range from 1-15, 1-14 for seasons pre-playoff, i.e. 2013 or earlier</td>
</tr>
<tr>
<td>end_week</td>
<td>(Integer optional): Ending Week - values range from 1-15, 1-14 for seasons pre-playoff, i.e. 2013 or earlier</td>
</tr>
</tbody>
</table>
category (String optional): Category filter (e.g defensive) Offense: passing, receiving, rushing Defense: defensive, fumbles, interceptions Special Teams: punting, puntReturns, kicking, kickReturns

Value

cfbd_stats_season_player() - A data frame with 59 variables:

team: character. Team name.
conference: character. Conference of the team.
athlete_id: character. Athlete referencing id.
player: character. Player name.
category: character. Statistic category.
passing_completions: double. Passing completions.
passing_att: double. Passing attempts.
passing_pct: double. Passing completion percentage.
passing_yds: double. Passing yardage.
passing_td: double. Passing touchdowns.
passing_int: double. Passing interceptions.
passing_ypa: double. Passing yards per attempt.
rushing_car: double. Rushing yards per carry.
rushing_yds: double. Rushing yards total.
rushing_td: double. Rushing touchdowns.
rushing_ypc: double. Rushing yards per carry.
rushing_long: double. Rushing longest yardage attempt.
receiving_rec: double. Receiving - pass receptions.
receiving_yds: double. Receiving - pass reception yards.
receiving_td: double. Receiving - passing reception touchdowns.
receiving_ypr: double. Receiving - passing yards per reception.
receiving_long: double. Receiving - longest pass reception yardage.
fumbles_fum: double. Fumbles.
fumbles_rec: double. Fumbles recovered.
fumbles_lost: double. Fumbles lost.
defensive_solo: double. Defensive solo tackles.
defensive_tot: double. Defensive total tackles.
defensive_tfl: double. Defensive tackles for loss.
defensive_sacks: double. Defensive sacks.
defensive_qb_hur: double. Defensive quarterback hurries.
interceptions_int: double. Interceptions total.
interceptions_yds: double. Interception return yards.
interceptions_avg: double. Interception return yards average.
interceptions_td: double. Interception return touchdowns.
kicking_fga: double. Kicking - field goals attempted.
kicking_pct: double. Kicking - field goal percentage.
kicking_xpa: double. Kicking - extra points attempted.
kicking_xpm: double. Kicking - extra points made.
kicking_pts: double. Kicking - total points.
kicking_long: double. Kicking - longest successful field goal attempt.
kick_returns_no: double. Kick Returns - number of kick returns.
kick_returns_avg: double. Kick Returns - kick return average yards per return.
punting_no: double. Punting - number of punts.
punting_ypp: double. Punting - yards per punt.
punting_long: double. Punting - longest punt yardage.
punting_in_20: double. Punting - punt downed inside the 20 yard line.
punt_returns_no: double. Punt Returns - number of punt returns.
punt_returns_avg: double. Punt Returns - punt return average yards per return.

Examples

try(cfbd_stats_season_player(year = 2018, conference = "B12", start_week = 1, end_week = 7))

try(cfbd_stats_season_player(2019, team = "LSU", category = "passing"))

try(cfbd_stats_season_player(2013, team = "Florida State", category = "passing"))
cfbd_stats_season_team

Get season statistics by team

Description

Get season statistics by team

Usage

```r
cfbd_stats_season_team(
  year,
  season_type = "regular",
  team = NULL,
  conference = NULL,
  start_week = NULL,
  end_week = NULL
)
```

Arguments

- **year** (*Integer required*): Year, 4 digit format (YYYY)
- **season_type** (*String default: regular*): Select Season Type - regular, postseason, or both
- **team** (*String optional*): D-I Team
- **conference** (*String optional*): Conference abbreviation - Select a valid FBS conference Conference abbreviations P5: ACC, B12, B1G, SEC, PAC Conference abbreviations G5 and FBS Independents: CUSA, MAC, MWC, Ind, SBC, AAC
- **start_week** (*Integer optional*): Starting Week - values range from 1-15, 1-14 for seasons pre-playoff, i.e. 2013 or earlier
- **end_week** (*Integer optional*): Ending Week - values range from 1-15, 1-14 for seasons pre-playoff, i.e. 2013 or earlier

Value

`cfbd_stats_season_team()` - A data frame with 32 variables:

- **season**: `integer`. Season for stats.
- **team**: `character`. Team name.
- **conference**: `character`. Conference of team.
- **games**: `integer`. Number of games.
- **time_of_poss_total**: `integer`. Time of possession total.
- **time_of_poss_pg**: `double`. Time of possession per game.
- **pass_comps**: `integer`. Total number of pass completions.
- **pass_atts**: `integer`. Total number of pass attempts.
completion_pct: double. Passing completion percentage.
pass_ypa: double. Passing yards per attempt.
pass_ypr: double. Passing yards per reception.
pass_TDs: integer. Passing touchdowns.
interceptions: integer. Passing interceptions.
int_pct: double. Interception percentage (of attempts).
rush_atts: integer. Rushing attempts.
rush_yds: integer. Rushing yards.
rush_TDs: integer. Rushing touchdowns.
rush_ypc: double. Rushing yards per carry.
total_yds: integer. Rushing total yards.
fumbles_lost: integer. Fumbles lost.
turnovers: integer. Turnovers total.
turnovers_pg: double. Turnovers per game.
first_downs: integer. Number of first downs.
third_downs: integer. Number of third downs.
third_down_convs: integer. Number of third down conversions.
third_conv_rate: double. Third down conversion rate.
fourth_down_convs: integer. Fourth down conversions.
fourth_downs: integer. Fourth downs.
fourth_conv_rate: double. Fourth down conversion rate.
penalties: integer. Total number of penalties.
penalty_yds: integer. Penalty yards total.
penalties_pg: double. Penalties per game.
penalty_yds_pg: double. Penalty yardage per game.
yards_per_penyalty: double. Average yards per penalty.
ikick_returns: integer. Number of kick returns.
kick_return_yds: integer. Total kick return yards.
kick_return_TDs: integer. Total kick return touchdowns.
punt_returns: integer. Number of punt returns.
punt_return_yds: integer. Punt return total yards.
punt_return_TDs: integer. Punt return total touchdowns.
passes_intercepted: integer. Passes intercepted.
passes_intercepted_TDs: integer. Pass interception return touchdowns.
Examples

try(cfbd_stats_season_team(year = 2018, conference = "B12", start_week = 1, end_week = 8))
try(cfbd_stats_season_team(2019, team = "LSU"))
try(cfbd_stats_season_team(2013, team = "Florida State"))

---

**Description**

cfbd_team_info(): Team Info Lookup.
cfbd_team_roster(): Get a team’s full roster by year.
cfbd_team_talent(): Get composite team talent rankings for all teams in a given year.
cfbd_team_matchup_records(): Get matchup history records between two teams.
cfbd_team_matchup(): Get matchup history between two teams.

**Team info lookup:**
Lists all teams in conference or all D-I teams if conference is left NULL Currently, support is only provided for D-I

cfbd_team_info(conference = "SEC")
cfbd_team_info(conference = "Ind")
cfbd_team_info(year = 2019)

**Get team rosters:**

It is now possible to access yearly rosters:
cfbd_team_roster(year = 2020)

Get a team’s full roster by year. If team is not selected, API returns rosters for every team from the selected year:
cfbd_team_roster(year = 2013, team = "Florida State")

Get composite team talent rankings:
Extracts team talent composite for all teams in a given year as sourced from 247 rankings
cfbd_team_talent()

cfbd_team_talent(year = 2018)

**Get matchup history between two teams:**
cfbd_team_matchup("Texas A&M", "TCU")

cfbd_team_matchup("Texas A&M", "TCU", min_year = 1975)

cfbd_team_matchup("Florida State", "Florida", min_year = 1975)

Get matchup history records between two teams:

    cfbd_team_matchup_records("Texas", "Oklahoma")

    cfbd_team_matchup_records("Texas A&M", "TCU", min_year = 1975)

---

**cfbd_team_info**

Team info lookup

### Description

Team info lookup

### Usage

    cfbd_team_info(conference = NULL, only_fbs = TRUE, year = NULL)

### Arguments

- **conference** *(String optional)*: Conference abbreviation - Select a valid FBS conference Conference abbreviations P5: ACC, B12, B1G, SEC, PAC, Conference abbreviations G5 and FBS Independents: CUSA, MAC, MWC, Ind, SBC, AAC
- **only_fbs** *(Logical default TRUE)*: Filter for only returning FBS teams for a given year. If year is left blank while only_fbs is TRUE, then will return values for most current year
- **year** *(Integer optional)*: Year, 4 digit format (YYYY). Filter for getting a list of major division team for a given year

### Value

    cfbd_team_info() - A data frame with 12 variables:

- **team_id**: integer. Referencing team id.
- **school**: character. Team name.
- **mascot**: character. Team mascot.
- **abbreviation**: character. Team abbreviations.
- **alt_name1**: character. Team alternate name 1 (as it appears in play_text).
- **alt_name2**: character. Team alternate name 2 (as it appears in play_text).
- **alt_name3**: character. Team alternate name 3 (as it appears in play_text).
- **conference**: character. Conference of team.
division: character. Division of team within the conference.
color: character. Team color (primary).
alt_color: character. Team color (alternate).
logos: character. Team logos.
venue_id: character. Referencing venue id.
venue_name: character. Stadium name.
city: character. Team/venue city.
state: character. Team/venue state.
zip: character. Team/venue zip code (someone double check Miami (FL) on if they’re in the same zip code).
country_code: character. Team/venue country code.
timezone: character. Team/venue timezone.
latitude: character. Venue latitude.
longitude: character. Venue longitude.
elevation: character. Venue elevation.
capacity: character. Venue capacity.
year_constructed: character. Year the venue was constructed.
grass: character. TRUE/FALSE response on whether the field is grass or not (oh, and there are so many others).
dome: character. TRUE/FALSE flag for if the venue is a domed stadium.

Examples

try(cfbd_team_info(conference = "SEC"))
try(cfbd_team_info(conference = "Ind"))
try(cfbd_team_info(year = 2019))

---

cfbd_team_matchup Get matchup history between two teams.

Description

Get matchup history between two teams.

Usage

cfbd_team_matchup(team1, team2, min_year = NULL, max_year = NULL)
Arguments

- **team1**: (String required): D-I Team 1
- **team2**: (String required): D-I Team 2
- **min_year**: (Integer optional): Minimum of year range, 4 digit format (YYYY)
- **max_year**: (Integer optional): Maximum of year range, 4 digit format (YYYY)

Value

`cfbd_team_matchup` - A data frame with 11 variables:

- **season**: integer. Season the game took place.
- **week**: integer. Game week of the season.
- **season_type**: character. Season type of the game.
- **date**: character. Game date.
- **neutral_site**: logical. TRUE/FALSE flag for if the game took place at a neutral site.
- **venue**: character. Stadium name.
- **home_team**: character. Home team of the game.
- **home_score**: integer. Home score in the game.
- **away_team**: character. Away team of the game.
- **away_score**: integer. Away score in the game.
- **winner**: character. Winner of the matchup.

Examples

```r
try(cfbd_team_matchup("Texas", "Oklahoma"))
try(cfbd_team_matchup("Texas A&M", "TCU"))
try(cfbd_team_matchup("Texas A&M", "TCU", min_year = 1975))
try(cfbd_team_matchup("Florida State", "Florida", min_year = 1975))
```
Usage

```r
cfb_dp_team_matchup_records(team1, team2, min_year = NULL, max_year = NULL)
```

Arguments

- `team1` *(String required)*: D-I Team 1
- `team2` *(String required)*: D-I Team 2
- `min_year` *(Integer optional)*: Minimum of year range, 4 digit format (YYYY)
- `max_year` *(Integer optional)*: Maximum of year range, 4 digit format (YYYY)

Value

`cfbd_team_matchup_records()` - A data frame with 7 variables:

- `start_year`: character. Span starting year.
- `end_year`: character. Span ending year.
- `team1`: character. First team selected in query.
- `team1_wins`: character. First team wins in series against `team2`.
- `team2`: character. Second team selected in query.
- `team2_wins`: character. Second team wins in series against `team1`.
- `ties`: character. Number of ties in the series.

Examples

```r
try(cfbd_team_matchup_records("Texas", "Oklahoma"))

try(cfbd_team_matchup_records("Texas A&M", "TCU", min_year = 1975))
```

---

### cfb_dp_team_roster

Get team rosters

**Description**

Get a teams full roster by year. If team is not selected, API returns rosters for every team from the selected year.

**Usage**

```r
cfb_dp_team_roster(year, team = NULL)
```
Arguments

- **year**: (Integer required): Year, 4 digit format (YYYY)
- **team**: (String optional): Team, select a valid team in D-I football

Value

cfbd_team_roster() - A data frame with 12 variables:

- **athlete_id**: character. Referencing athlete id.
- **first_name**: character. Athlete first name.
- **last_name**: character. Athlete last name.
- **team**: character. Team name.
- **weight**: integer. Athlete weight.
- **height**: integer. Athlete height.
- **jersey**: integer. Athlete jersey number.
- **year**: integer. Athlete year.
- **position**: character. Athlete position.
- **home_city**: character. Hometown of the athlete.
- **home_state**: character. Hometown state of the athlete.
- **home_country**: character. Hometown country of the athlete.
- **home_latitude**: numeric. Hometown latitude.
- **home_longitude**: number. Hometown longitude.
- **home_county_fips**: integer. Hometown FIPS code.
- **headshot_url**: character. Player ESPN headshot url.

Examples

```r
try(cfbd_team_roster(year = 2013, team = "Florida State"))
```

---

**cfbd_team_talent**  
Get composite team talent rankings for all teams in a given year

Description

Extracts team talent composite as sourced from 247 rankings

Usage

cfbd_team_talent(year = NULL)
Arguments

year (Integer optional): Year 4 digit format (YYYY)

Value

cfbd_team_talent() - A data frame with 3 variables:

year: integer. Season for the talent rating.
school: character. Team name.
talent: double. Overall roster talent points (as determined by 247Sports).

Examples

try(cfbd_team_talent())

try(cfbd_team_talent(year = 2018))

---

cfbd_venues CFBD Venues Endpoint Overview

Description

Pulls all college football venues and data on capacity, grass, city/state, location, elevation, dome, timezone and construction year.

Usage

cfbd_venues()

Details

CFB Venue Information

cfbd_venues()

Value

A data frame with 337 rows and 13 variables:

venue_id: integer. Referencing venue ID.
name: character. Venue name.
capacity: integer. Stadium capacity.
grass: logical. TRUE/FALSE response on whether the field is grass or not (oh, and there are so many others).
city:character. Venue city.
state:character. Venue state.
zip:character. Venue zip.
country_code:character. Venue country code.
location:list. Venue location.
elevation:character. Venue elevation.
year_construced:integer. Year in which the venue was constructed.
done:logical. TRUE/FALSE response to whether the venue has a dome or not.
timezone:character. Time zone in which the venue resides (i.e. Eastern Time -> "America/New York").

Examples

try(cfbd_venues())

csv_from_url Load .csv / .csv.gz file from a remote connection

Description

This is a thin wrapper on data.table::fread

Usage

csv_from_url(...)

Arguments

... passed to data.table::fread

data Data in the package for reference

Description

Data in the package for reference

College Football Mapping for Play Types
College Football Conferences
Usage

cfbd_play_type_df
cfbd_conf_types_df

Format

A data frame with 45 rows and 3 variables:

- **id**: Referencing play id
- **text**: play type description
- **abbreviation**: play type abbreviation used for function call

A data frame with 11 rows and 4 variables:

- **id**: Referencing conference id
- **name**: Conference name
- **short_name**: Short name for Conference
- **abbreviation**: Conference abbreviation

---

Descripción

look up the men’s college football calendar for a given season

Usage

```
espn_cfb_calendar(year = NULL, groups = NULL)
```

Arguments

- **year**: (int): Used to define different seasons. 2002 is the earliest available season.
- **groups**: (string): Used to define different divisions. FBS or FCS.

Value

```
espn_cfb_calendar() - A data frame with 8 variables:
season: character.
season_type: character.
label: character.
alternate_label: character.
detail: character.
week: character.
start_date: character.
end_date: character.
```
Examples

try(espn_cfb_calendar(2021))

espn_cfb_pbp  

*Get ESPN college football PBP data*

Description

Get ESPN college football PBP data

Usage

espn_cfb_pbp(game_id, epa_wpa = FALSE)

Arguments

- **game_id**: Game ID
- **epa_wpa**: Logical parameter (TRUE/FALSE) to return the Expected Points Added/Win Probability Added variables

Author(s)

Saiem Gilani

Examples

try(espn_cfb_pbp(game_id = 401282614))

espn_cfb_scoreboard  

*ESPN Scoreboard*

Description

ESPN Scoreboard

Get live scoreboard data from ESPN or look up the college football schedule for a given season

ESPN Schedule
Usage

espn_cfb_scoreboard(date = NULL)

espn_cfb_schedule(
  year = NULL,
  week = NULL,
  season_type = NULL,
  groups = NULL,
  limit = 500
)

Arguments

date (Integer required - YYYYMMDD): Date to pull
year (int): Used to define different seasons. 2002 is the earliest available season.
week (int): Week of the schedule.
season_type (string): "regular", "postseason", "off-season", or "both".
groups (string): Used to define different divisions. FBS or FCS.
limit (int): number of records to return, default: 500.

Value

espn_cfb_scoreboard() & espn_cfb_schedule() - A data frame with 33 or 54 variables depending on if there are completed games: shared variables

matchup: character. Long matchup description with full team names (Utah Utes at UCLA Bruins).
matchup_short: character. Short matchup description with team abbreviations (UTAH @ UCLA).
season: integer. Season of the game.
type: character. Season type of the game in CFBD format.
slug: character. Season type of the game in ESPN format.
game_id: character. Referencing game ID.
game_uid: character.
game_date: character. Game date.
attendance: integer. Reported attendance at the game.
home_team_name: character. Home team mascot name (Sun Devils).
home_team_logo: character. Home team logo url.
home_team_abb: character. Home team abbreviation (ASU).
home_team_id: character. Home team ID.
home_team_location: character. Home team name (Arizona State).
home_team_full: character. Home team full name (Arizona State Sun Devils).
home_team_color: character. Home team color.
home_score: integer. Home team points.
home_win: integer. 1 if home team won, 0 if home team lost, NA if game is unfinished
home_record: character Home team record.
away_team_name: character. Away team mascot name (Sun Devils).
away_team_logo: character. Away team logo url.
away_team_abb: character. Away team abbreviation (ASU).
away_team_id: character. Away team ID.
away_team_location: character. Away team name (Arizona State).
away_team_full: character. Away team full name (Arizona State Sun Devils).
away_team_color: character. Away team color.
away_score: integer. Away team points.
away_win: integer. 1 if away team won, 0 if home team lost, NA if game is unfinished
away_record: character. Away team record.
status_name: character. Status of the game
start_date: character. Game date.

Unique variables when there are completed games
broadcast_market: character. Broadcast market (typically "national" or NA)
broadcast_name: character. Broadcast channel i.e. ESPN, ABC, FOX
passing_leader_yards: numeric. Passing yards of game’s passing leader
passing_leader_stat: character. Stat line of game’s passing leader
passing_leader_name: character. Name of game’s passing leader
passing_leader_shortname: character. First initial and last name of game’s passing leader
passing_leader_headshot: character. Headshot url of game’s passing leader
passing_leader_team_id: character. Team ID of game’s passing leader
passing_leader_pos: character. Position of game’s passing leader
rushing_leader_yards: numeric. Passing yards of game’s rushing leader
rushing_leader_stat: character. Stat line of game’s rushing leader
rushing_leader_name: character. Name of game’s rushing leader
rushing_leader_shortname: character. First initial and last name of game’s rushing leader
rushing_leader_headshot: character. Headshot url of game’s rushing leader
rushing_leader_team_id: character. Team ID of game’s rushing leader
rushing_leader_pos: character. Position of game’s rushing leader
receiving_leader_yards: numeric. Passing yards of game’s receiving leader
receiving_leader_stat: character. Stat line of game’s receiving leader
receiving_leader_name: character. Name of game’s receiving leader
receiving_leader_shortname: character. First initial and last name of game’s receiving leader
receiving_leader_headshot: character. Headshot url of game’s receiving leader
receiving_leader_team_id: character. Team ID of game’s receiving leader
receiving_leader_pos: character. Position of game’s receiving leader
Examples

try(espn_cfb_scoreboard())

try(espn_cfb_schedule(2021, week = 8))

espn_cfb_team_stats  Get ESPN college football team stats data

Description

Get ESPN college football team stats data

Usage

espn_cfb_team_stats(team_id, year, season_type = "regular", total = FALSE)

Arguments

<table>
<thead>
<tr>
<th>team_id</th>
<th>Team ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>year</td>
<td>Year</td>
</tr>
<tr>
<td>season_type</td>
<td>(character, default: regular): Season type - regular or postseason</td>
</tr>
<tr>
<td>total</td>
<td>(boolean, default: FALSE): Totals</td>
</tr>
</tbody>
</table>

Value

Returns a tibble with the following columns:

col_name types
general_fumbles numeric
general_fumbles_lost numeric
general_fumbles_forced numeric
general_fumblesRecovered numeric
general_fumbles_touchdowns numeric
general_games_played numeric
general_offensive_two_pt_returns numeric
general_offensive_fumbles_touchdowns numeric
general_defensive_fumbles_touchdowns numeric
passing_avg_gain numeric
passing_completion_pct numeric
passing_completions numeric
passing_espnqb_rating numeric
passing_interception_pct numeric
passing_interceptions numeric
passing_long_passing numeric
passing_misc_yards numeric
passing_net_passing_yards numeric
passing_net_passing_yards_per_game numeric
passing_net_total_yards numeric
passing_net_yards_per_game numeric
passing_passing_attempts numeric
passing_passing_big_plays numeric
passing_passing_first_downs numeric
passing_passing_fumbles numeric
passing_passing_fumbles_lost numeric
passing_passing_touchdown_pct numeric
passing_passing_touchdowns numeric
passing_passing_yards numeric
passing_passing_yards_after_catch numeric
passing_passing_yards_at_catch numeric
passing_passing_yards_per_game numeric
passing_qb_rating numeric
passing_sacks numeric
passing_sack_yards_lost numeric
passing_team_games_played numeric
passing_total_offensive_plays numeric
passing_total_points numeric
passing_total_points_per_game numeric
passing_total_touchdowns numeric
passing_total_yards numeric
passing_total_yards_from_scrimmage numeric
passing_two_point_pass_convs numeric
passing_two_pt_pass numeric
passing_two_pt_pass_attempts numeric
passing_yards_from_scrimmage_per_game numeric
passing_yards_per_completion numeric
passing_yards_per_game numeric
passing_yards_per_pass_attempt numeric
passing_net_yards_per_pass_attempt numeric
passing_quarterback_rating numeric
rushing_avg_gain numeric
rushing_espnrb_rating numeric
rushing_long_rushing numeric
rushing_misc_yards numeric
rushing_net_total_yards numeric
rushing_net_yards_per_game numeric
rushing_rushing_attempts numeric
rushing_rushing_big_plays numeric
rushing_rushing_first_downs numeric
rushing_rushing_fumbles numeric
rushing_rushing_fumbles_lost numeric
rushing_rushing_touchdowns numeric
<table>
<thead>
<tr>
<th>Stat</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>rushing_rushing_yards</td>
<td>numeric</td>
</tr>
<tr>
<td>rushing_rushing_yards_per_game</td>
<td>numeric</td>
</tr>
<tr>
<td>rushing_stuffs</td>
<td>numeric</td>
</tr>
<tr>
<td>rushing_stuff_yards_lost</td>
<td>numeric</td>
</tr>
<tr>
<td>rushing_team_games_played</td>
<td>numeric</td>
</tr>
<tr>
<td>rushing_total_offensive_plays</td>
<td>numeric</td>
</tr>
<tr>
<td>rushing_total_points</td>
<td>numeric</td>
</tr>
<tr>
<td>rushing_total_points_per_game</td>
<td>numeric</td>
</tr>
<tr>
<td>rushing_total_touchdowns</td>
<td>numeric</td>
</tr>
<tr>
<td>rushing_total_yards</td>
<td>numeric</td>
</tr>
<tr>
<td>rushing_total_yards_from_scrimmage</td>
<td>numeric</td>
</tr>
<tr>
<td>rushing_two_point_rush_convs</td>
<td>numeric</td>
</tr>
<tr>
<td>rushing_two_pt_rush</td>
<td>numeric</td>
</tr>
<tr>
<td>rushing_two_pt_rush_attempts</td>
<td>numeric</td>
</tr>
<tr>
<td>rushing_yards_from_scrimmage_per_game</td>
<td>numeric</td>
</tr>
<tr>
<td>rushing_yards_per_game</td>
<td>numeric</td>
</tr>
<tr>
<td>rushing_yards_per_rush_attempt</td>
<td>numeric</td>
</tr>
<tr>
<td>receiving_avg_gain</td>
<td>numeric</td>
</tr>
<tr>
<td>receiving_espnwr_rating</td>
<td>numeric</td>
</tr>
<tr>
<td>receiving_long_reception</td>
<td>numeric</td>
</tr>
<tr>
<td>receiving_misc_yards</td>
<td>numeric</td>
</tr>
<tr>
<td>receiving_net_total_yards</td>
<td>numeric</td>
</tr>
<tr>
<td>receiving_net_yards_per_game</td>
<td>numeric</td>
</tr>
<tr>
<td>receiving_receiving_big_plays</td>
<td>numeric</td>
</tr>
<tr>
<td>receiving_receiving_first_downs</td>
<td>numeric</td>
</tr>
<tr>
<td>receiving_receiving_fumbles</td>
<td>numeric</td>
</tr>
<tr>
<td>receiving_receiving_fumbles_lost</td>
<td>numeric</td>
</tr>
<tr>
<td>receiving_receiving_targets</td>
<td>numeric</td>
</tr>
<tr>
<td>receiving_receiving_touchdowns</td>
<td>numeric</td>
</tr>
<tr>
<td>receiving_receiving_yards</td>
<td>numeric</td>
</tr>
<tr>
<td>receiving_receiving_yards_after_catch</td>
<td>numeric</td>
</tr>
<tr>
<td>receiving_receiving_yards_at_catch</td>
<td>numeric</td>
</tr>
<tr>
<td>receiving_receiving_yards_per_game</td>
<td>numeric</td>
</tr>
<tr>
<td>receiving_receptions</td>
<td>numeric</td>
</tr>
<tr>
<td>receiving_team_games_played</td>
<td>numeric</td>
</tr>
<tr>
<td>receiving_total_offensive_plays</td>
<td>numeric</td>
</tr>
<tr>
<td>receiving_total_points</td>
<td>numeric</td>
</tr>
<tr>
<td>receiving_total_points_per_game</td>
<td>numeric</td>
</tr>
<tr>
<td>receiving_total_touchdowns</td>
<td>numeric</td>
</tr>
<tr>
<td>receiving_total_yards</td>
<td>numeric</td>
</tr>
<tr>
<td>receiving_total_yards_from_scrimmage</td>
<td>numeric</td>
</tr>
<tr>
<td>receiving_two_point_rec_convs</td>
<td>numeric</td>
</tr>
<tr>
<td>receiving_two_pt_reception</td>
<td>numeric</td>
</tr>
<tr>
<td>receiving_two_pt_reception_attempts</td>
<td>numeric</td>
</tr>
<tr>
<td>receiving_yards_from_scrimmage_per_game</td>
<td>numeric</td>
</tr>
<tr>
<td>receiving_yards_per_game</td>
<td>numeric</td>
</tr>
<tr>
<td>receiving_yards_per_reception</td>
<td>numeric</td>
</tr>
<tr>
<td>defensive_assist_tackles</td>
<td>numeric</td>
</tr>
<tr>
<td>Metric</td>
<td>Type</td>
</tr>
<tr>
<td>-----------------------------------------------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>defensive_avg_interception_yards</td>
<td>numeric</td>
</tr>
<tr>
<td>defensive_avg_sack_yards</td>
<td>numeric</td>
</tr>
<tr>
<td>defensive_avg_stuff_yards</td>
<td>numeric</td>
</tr>
<tr>
<td>defensive_blocked_field_goal_touchdowns</td>
<td>numeric</td>
</tr>
<tr>
<td>defensive_blocked_punt_touchdowns</td>
<td>numeric</td>
</tr>
<tr>
<td>defensive_defensive_touchdowns</td>
<td>numeric</td>
</tr>
<tr>
<td>defensive_hurries</td>
<td>numeric</td>
</tr>
<tr>
<td>defensive_kicks_blocked</td>
<td>numeric</td>
</tr>
<tr>
<td>defensive_long_interception</td>
<td>numeric</td>
</tr>
<tr>
<td>defensive_misc_touchdowns</td>
<td>numeric</td>
</tr>
<tr>
<td>defensive_passes_batted_down</td>
<td>numeric</td>
</tr>
<tr>
<td>defensive_passes_defended</td>
<td>numeric</td>
</tr>
<tr>
<td>defensive_two_pt_returns</td>
<td>numeric</td>
</tr>
<tr>
<td>defensive_sacks</td>
<td>numeric</td>
</tr>
<tr>
<td>defensive_sack_yards</td>
<td>numeric</td>
</tr>
<tr>
<td>defensive_safeties</td>
<td>numeric</td>
</tr>
<tr>
<td>defensive_solo_tackles</td>
<td>numeric</td>
</tr>
<tr>
<td>defensive_stuffs</td>
<td>numeric</td>
</tr>
<tr>
<td>defensive_stuff_yards</td>
<td>numeric</td>
</tr>
<tr>
<td>defensive_tackles_for_loss</td>
<td>numeric</td>
</tr>
<tr>
<td>defensive_team_games_played</td>
<td>numeric</td>
</tr>
<tr>
<td>defensive_total_tackles</td>
<td>numeric</td>
</tr>
<tr>
<td>defensive_yards_allowed</td>
<td>numeric</td>
</tr>
<tr>
<td>defensive_points_allowed</td>
<td>numeric</td>
</tr>
<tr>
<td>defensive_one_pt_safeties_made</td>
<td>numeric</td>
</tr>
<tr>
<td>defensive_interceptions_interceptions</td>
<td>numeric</td>
</tr>
<tr>
<td>defensive_interceptions_interception_touchdowns</td>
<td>numeric</td>
</tr>
<tr>
<td>kicking_avg_kickoff_return_yards</td>
<td>numeric</td>
</tr>
<tr>
<td>kicking_avg_kickoff_yards</td>
<td>numeric</td>
</tr>
<tr>
<td>kicking_extra_point_attempts</td>
<td>numeric</td>
</tr>
<tr>
<td>kicking_extra_point_pct</td>
<td>numeric</td>
</tr>
<tr>
<td>kicking_extra_points_blocked</td>
<td>numeric</td>
</tr>
<tr>
<td>kicking_extra_points_blocked_pct</td>
<td>numeric</td>
</tr>
<tr>
<td>kicking_extra_points_made</td>
<td>numeric</td>
</tr>
<tr>
<td>kicking_fair_catches</td>
<td>numeric</td>
</tr>
<tr>
<td>kicking_fair_catch_pct</td>
<td>numeric</td>
</tr>
<tr>
<td>kicking_field_goal_attempts</td>
<td>numeric</td>
</tr>
<tr>
<td>kicking_field_goal_attempts1_19</td>
<td>numeric</td>
</tr>
<tr>
<td>kicking_field_goal_attempts20_29</td>
<td>numeric</td>
</tr>
<tr>
<td>kicking_field_goal_attempts30_39</td>
<td>numeric</td>
</tr>
<tr>
<td>kicking_field_goal_attempts40_49</td>
<td>numeric</td>
</tr>
<tr>
<td>kicking_field_goal_attempts50_59</td>
<td>numeric</td>
</tr>
<tr>
<td>kicking_field_goal_attempts60_99</td>
<td>numeric</td>
</tr>
<tr>
<td>kicking_field_goal_attempts50</td>
<td>numeric</td>
</tr>
<tr>
<td>kicking_field_goal_attempts60_99</td>
<td>numeric</td>
</tr>
<tr>
<td>kicking_field_goal_attempts_yards</td>
<td>numeric</td>
</tr>
<tr>
<td>kicking_field_goal_pct</td>
<td>numeric</td>
</tr>
<tr>
<td>kicking_field_goalsBlocked</td>
<td>numeric</td>
</tr>
<tr>
<td>Metric</td>
<td>Type</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>kicking_field_goals_blocked_pct</td>
<td>numeric</td>
</tr>
<tr>
<td>kicking_field_goals_made</td>
<td>numeric</td>
</tr>
<tr>
<td>kicking_field_goals_made1_19</td>
<td>numeric</td>
</tr>
<tr>
<td>kicking_field_goals_made20_29</td>
<td>numeric</td>
</tr>
<tr>
<td>kicking_field_goals_made30_39</td>
<td>numeric</td>
</tr>
<tr>
<td>kicking_field_goals_made40_49</td>
<td>numeric</td>
</tr>
<tr>
<td>kicking_field_goals_made50_59</td>
<td>numeric</td>
</tr>
<tr>
<td>kicking_field_goals_made60_99</td>
<td>numeric</td>
</tr>
<tr>
<td>kicking_field_goals_made50</td>
<td>numeric</td>
</tr>
<tr>
<td>kicking_field_goals_made_yards</td>
<td>numeric</td>
</tr>
<tr>
<td>kicking_field_goals_missed_yards</td>
<td>numeric</td>
</tr>
<tr>
<td>kicking_kickoff_returns</td>
<td>numeric</td>
</tr>
<tr>
<td>kicking_kickoff_return_touchdowns</td>
<td>numeric</td>
</tr>
<tr>
<td>kicking_kickoff_return_yards</td>
<td>numeric</td>
</tr>
<tr>
<td>kicking_kickoffs</td>
<td>numeric</td>
</tr>
<tr>
<td>kicking_kickoff_yards</td>
<td>numeric</td>
</tr>
<tr>
<td>kicking_long_field_goal_attempt</td>
<td>numeric</td>
</tr>
<tr>
<td>kicking_long_field_goal_made</td>
<td>numeric</td>
</tr>
<tr>
<td>kicking_long_kickoff</td>
<td>numeric</td>
</tr>
<tr>
<td>kicking_team_games_played</td>
<td>numeric</td>
</tr>
<tr>
<td>kicking_total_kicking_points</td>
<td>numeric</td>
</tr>
<tr>
<td>kicking_touchback_pct</td>
<td>numeric</td>
</tr>
<tr>
<td>kicking_touchbacks</td>
<td>numeric</td>
</tr>
<tr>
<td>returning_def_fumble_returns</td>
<td>numeric</td>
</tr>
<tr>
<td>returning_def_fumble_return_yards</td>
<td>numeric</td>
</tr>
<tr>
<td>returning_fumble_recoveries</td>
<td>numeric</td>
</tr>
<tr>
<td>returning_fumble_recovery_yards</td>
<td>numeric</td>
</tr>
<tr>
<td>returning_kick_return_fair_catches</td>
<td>numeric</td>
</tr>
<tr>
<td>returning_kick_return_fair_catch_pct</td>
<td>numeric</td>
</tr>
<tr>
<td>returning_kick_return_fumbles</td>
<td>numeric</td>
</tr>
<tr>
<td>returning_kick_return_fumbles_lost</td>
<td>numeric</td>
</tr>
<tr>
<td>returning_kick_returns</td>
<td>numeric</td>
</tr>
<tr>
<td>returning_kick_return_touchdowns</td>
<td>numeric</td>
</tr>
<tr>
<td>returning_kick_return_yards</td>
<td>numeric</td>
</tr>
<tr>
<td>returning_long_kick_return</td>
<td>numeric</td>
</tr>
<tr>
<td>returning_long_punt_return</td>
<td>numeric</td>
</tr>
<tr>
<td>returning_misc_fumble_returns</td>
<td>numeric</td>
</tr>
<tr>
<td>returning_misc_fumble_return_yards</td>
<td>numeric</td>
</tr>
<tr>
<td>returning_opf_fumble_recoveries</td>
<td>numeric</td>
</tr>
<tr>
<td>returning_opf_fumble_recovery_yards</td>
<td>numeric</td>
</tr>
<tr>
<td>returning_opf_special_team_fumble_returns</td>
<td>numeric</td>
</tr>
<tr>
<td>returning_opf_special_team_fumble_return_yards</td>
<td>numeric</td>
</tr>
<tr>
<td>returning_punt_return_fair_catches</td>
<td>numeric</td>
</tr>
<tr>
<td>returning_punt_return_fair_catch_pct</td>
<td>numeric</td>
</tr>
<tr>
<td>returning_punt_fumbles</td>
<td>numeric</td>
</tr>
<tr>
<td>returning_punt_fumbles_lost</td>
<td>numeric</td>
</tr>
<tr>
<td>returning_punt_returns</td>
<td>numeric</td>
</tr>
<tr>
<td>returning_punt_returns_started_inside_the10</td>
<td>numeric</td>
</tr>
<tr>
<td>Metric</td>
<td>Type</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>returning_punt_returns_started_inside_the20</td>
<td>numeric</td>
</tr>
<tr>
<td>returning_punt_return_touchdowns</td>
<td>numeric</td>
</tr>
<tr>
<td>returning_punt_return_yards</td>
<td>numeric</td>
</tr>
<tr>
<td>returning_special_team_fumble_returns</td>
<td>numeric</td>
</tr>
<tr>
<td>returning_special_team_fumble_return_yards</td>
<td>numeric</td>
</tr>
<tr>
<td>returning_team_games_played</td>
<td>numeric</td>
</tr>
<tr>
<td>returning_yards_per_kick_return</td>
<td>numeric</td>
</tr>
<tr>
<td>returning_yards_per_punt_return</td>
<td>numeric</td>
</tr>
<tr>
<td>returning_yards_per_return</td>
<td>numeric</td>
</tr>
<tr>
<td>puntng_avg_punt_return_yards</td>
<td>numeric</td>
</tr>
<tr>
<td>puntng_fair_catches</td>
<td>numeric</td>
</tr>
<tr>
<td>puntng_gross_avg_punt_yards</td>
<td>numeric</td>
</tr>
<tr>
<td>puntng_long_punt</td>
<td>numeric</td>
</tr>
<tr>
<td>puntng_net_avg_punt_yards</td>
<td>numeric</td>
</tr>
<tr>
<td>puntng_punt_returns</td>
<td>numeric</td>
</tr>
<tr>
<td>puntng_punt_return_yards</td>
<td>numeric</td>
</tr>
<tr>
<td>puntng_punts</td>
<td>numeric</td>
</tr>
<tr>
<td>puntng_punts_blocked</td>
<td>numeric</td>
</tr>
<tr>
<td>puntng_punts_blocked_pct</td>
<td>numeric</td>
</tr>
<tr>
<td>puntng_punts_inside10</td>
<td>numeric</td>
</tr>
<tr>
<td>puntng_punts_inside10pct</td>
<td>numeric</td>
</tr>
<tr>
<td>puntng_punts_inside20</td>
<td>numeric</td>
</tr>
<tr>
<td>puntng_punts_inside20pct</td>
<td>numeric</td>
</tr>
<tr>
<td>puntng_punt_yards</td>
<td>numeric</td>
</tr>
<tr>
<td>puntng_team_games_played</td>
<td>numeric</td>
</tr>
<tr>
<td>puntng_touchback_pct</td>
<td>numeric</td>
</tr>
<tr>
<td>puntng_touchbacks</td>
<td>numeric</td>
</tr>
<tr>
<td>scoring_defensive_points</td>
<td>numeric</td>
</tr>
<tr>
<td>scoring_field_goals</td>
<td>numeric</td>
</tr>
<tr>
<td>scoring_kick_extra_points</td>
<td>numeric</td>
</tr>
<tr>
<td>scoring_misc_points</td>
<td>numeric</td>
</tr>
<tr>
<td>scoring_passing_touchdowns</td>
<td>numeric</td>
</tr>
<tr>
<td>scoring_receiving_touchdowns</td>
<td>numeric</td>
</tr>
<tr>
<td>scoring_return_touchdowns</td>
<td>numeric</td>
</tr>
<tr>
<td>scoring_rushing_touchdowns</td>
<td>numeric</td>
</tr>
<tr>
<td>scoring_total_points</td>
<td>numeric</td>
</tr>
<tr>
<td>scoring_total_points_per_game</td>
<td>numeric</td>
</tr>
<tr>
<td>scoring_total_toucinvalidate_dfations</td>
<td>numeric</td>
</tr>
<tr>
<td>scoring_total_two_point_convs</td>
<td>numeric</td>
</tr>
<tr>
<td>scoring_two_point_pass_convs</td>
<td>numeric</td>
</tr>
<tr>
<td>scoring_two_point_rec_convs</td>
<td>numeric</td>
</tr>
<tr>
<td>scoring_two_point_rush_convs</td>
<td>numeric</td>
</tr>
<tr>
<td>scoring_one_pt_safeties_made</td>
<td>numeric</td>
</tr>
<tr>
<td>miscellaneous_first_downs</td>
<td>numeric</td>
</tr>
<tr>
<td>miscellaneous_first_downs_passing</td>
<td>numeric</td>
</tr>
<tr>
<td>miscellaneous_first_downs_penalty</td>
<td>numeric</td>
</tr>
<tr>
<td>miscellaneous_first_downs_per_game</td>
<td>numeric</td>
</tr>
<tr>
<td>miscellaneous_first_downs_rushing</td>
<td>numeric</td>
</tr>
</tbody>
</table>
espn_metrics

miscellaneous_fourth_down_attempts numeric
miscellaneous_fourth_down_conv_pct numeric
miscellaneous_fourth_down_convs numeric
miscellaneous_fumbles_lost numeric
miscellaneous_possession_time_seconds numeric
miscellaneous_redzone_efficiency_pct numeric
miscellaneous_redzone_field_goal_pct numeric
miscellaneous_redzone_scoring_pct numeric
miscellaneous_redzone_touchdown_pct numeric
miscellaneous_third_down_attempts numeric
miscellaneous_third_down_conv_pct numeric
miscellaneous_third_down_convs numeric
miscellaneous_total_giveaways numeric
miscellaneous_total_penalties numeric
miscellaneous_total_penalty_yards numeric
miscellaneous_total_takeaways numeric
miscellaneous_total_drives numeric
miscellaneous_turn_over_differential numeric

Author(s)

Saiem Gilani

Examples

try(espn_cfb_team_stats(team_id = 52, year = 2020))

espn_metrics

<table>
<thead>
<tr>
<th>ESPN Metrics</th>
</tr>
</thead>
</table>

Description

ESPN Metrics

Get win probability chart data from ESPN Graciously contributed by MrCaseB:

Usage

espn_metrics_wp(game_id)

Arguments

game_id (Integer required): Game ID filter for querying a single game Can be found using the cfbd_game_info() function
Value

`espn_metrics_wp()` - A data frame with 5 variables:

game_id: character. Referencing game ID (should be same as game_id from other functions).
play_id: character. Referencing play ID.
seconds_left: integer. Seconds left in the game.
home_win_percentage: double. The probability of the home team winning the game.
away_win_percentage: double. The probability of the away team winning the game (calculated as 1 - home_win_percentage - tie_percentage).
tie_percentage: double. The probability of the game ending the final period in a tie.

Examples

```r
try(espn_metrics_wp(game_id = 401114164))
```

<table>
<thead>
<tr>
<th>espn_ratings_fpi</th>
<th>ESPN FPI Ratings</th>
</tr>
</thead>
</table>

Description

Get FPI historical rating data (most recent of each year)

Usage

`espn_ratings_fpi(year = 2019)`

Arguments

year: Year

Details

Adapted from sabinanalytic’s fork of the cfbfastR repo

Value

A data frame with 20 variables:

year: double. Season of the Football Power Index (FPI) Rating.
team_id: character. Unique ESPN team ID - team_id.
name: character. Team Name.
abbr: character. Team abbreviation.
load_cfb_pbp

trend: character. Football Power Index (FPI) ranking trend.
proj_w: character. Projected Win total for the season.
proj_l: character. Projected Loss total for the season.
win_out: double. Probability the team wins out.
win_6: double. Probability the team wins at least six games.
win_div: double. Probability the team wins at their division.
playoff: double. Probability the team reaches the playoff.
nc_game: double. Probability the team reaches the national championship game.
nc_win: double. Probability the team wins the national championship game.
win_conf: double. Probability the team wins their conference game.
w: character. Wins on the season.
l: character. Losses on the season.
t: character. Ties on the season.

Examples

try(espn_ratings_fpi(year=2019))

load_cfb_pbp

Load cleaned play-by-play from the data repo

Description

helper that loads multiple seasons from the data repo either into memory or writes it into a db using some forwarded arguments in the dots

Usage

load_cfb_pbp(
    seasons = most_recent_cfb_season(),
    ..., 
    dbConnection = NULL,
    tablename = NULL
)

Arguments

seasons A vector of 4-digit years associated with given College Football seasons.
... Additional arguments passed to an underlying function that writes the season data into a database (used by update_cfb_db())
dbConnection A DBIConnection object, as returned by DBI::dbConnect()
tablename The name of the play by play data table within the database
### load_cfb_rosters  
**Load College Football Rosters**

**Description**

Load College Football Rosters

**Usage**

```r
load_cfb_rosters(seasons = most_recent_cfb_season())
```

**Arguments**

- **seasons**  
  A numeric vector of seasons to return, defaults to returning this year's data if it is September or later. If set to `TRUE`, will return all available data.

**Value**

A tibble of season-level roster data.

**See Also**

- `cfbd_team_roster()`
- Issues with this data should be filed here: [https://github.com/sportsdataverse/cfbfastR-data](https://github.com/sportsdataverse/cfbfastR-data)

**Examples**

```r
try(load_cfb_rosters(2020))
```

---

### load_cfb_schedules  
**Load CFB Game/Schedule Data from data repo**

**Description**

This returns game/schedule information

**Usage**

```r
load_cfb_schedules(seasons = most_recent_cfb_season())
```

**Arguments**

- **seasons**  
  A numeric vector of seasons to return, default `TRUE` returns all available data.
Value

A tibble of game information for past and/or future games.

See Also

cfbd_game_info()

Issues with this data should be filed here: https://github.com/sportsdataverse/cfbfastR-data

Examples

try(load_cfb_schedules(2020))

load_cfb_teams

Load CFB team info from the data repo

Description

Loads team information including colors and logos - useful for plots!

Usage

load_cfb_teams(fbs_only = TRUE)

Arguments

fbs_only if TRUE, returns only FBS teams, otherwise returns all teams in the dataset

Value

A tibble of team-level image URLs and hex color codes.

See Also

cfbd_team_info()

Issues with this data should be filed here: https://github.com/sportsdataverse/cfbfastR-data

Examples

try(load_cfb_teams())
rds_from_url  Load .rds file from a remote connection

Description
Load .rds file from a remote connection

Usage
rds_from_url(url)

Arguments
url a character url

Value
a dataframe as created by readRDS()

register_cfbd  CFBD API Key Registration

Description
Save your API Key as a system environment variable CFBD_API_KEY

Usage
cfbd_key()

Details
To get access to an API key, follow the instructions at https://collegefootballdata.com/key

Using the key: You can save the key for consistent usage by adding CFBD_API_KEY=XXXX~YOUR~API~KEY~HERE~XXXX to your .Renviron file (easily accessed via usethis::edit_r_environ()). Run usethis::edit_r_environ(), a new script will pop open named .Renviron, THEN paste the following in the new script that pops up (without quotations)

CFBD_API_KEY = XXXX~YOUR~API~KEY~HERE~XXXX
Save the script and restart your RStudio session, by clicking Session (in between Plots and Build) and click Restart R (there also exists the shortcut Ctrl + Shift + F10 to restart your session).

If set correctly, from then on you should be able to use any of the cfbd_ functions without any other changes.

**For less consistent usage:** At the beginning of every session or within an R environment, save your API key as the environment variable CFBD_API_KEY (with quotations) using a command like the following.

```r
Sys.setenv(CFBD_API_KEY = "XXXX-YOUR-API-KEY-HERE-XXXXX")
```

---

**update_cfb_db**

**Update or create a cfbfastR play-by-play database**

**Description**

`update_cfb_db()` updates or creates a database with cfbfastR play by play data of all completed games since 2014.

**Usage**

```r
update_cfb_db(
  dbdir = getOption("cfbfastR.dbdirectory", default = "."),
  dbname = "cfb_pbp_db",
  tblname = "cfbfastR_pbp",
  force_rebuild = FALSE,
  db_connection = NULL
)
```

**Arguments**

- **dbdir**
  - Directory in which the database is or shall be located. Can also be set globally with options(cfbfastR.dbdirectory).
- **dbname**
  - File name of an existing or desired SQLite database within dbdir
- **tblname**
  - The name of the play by play data table within the database
- **force_rebuild**
  - Hybrid parameter (logical or numeric) to rebuild parts of or the complete play by play data table within the database (please see details for further information)
- **db_connection**
  - A DBIConnection object, as returned by DBI::dbConnect() (please see details for further information)
Details

This function creates and updates a data table with the name tblname within a SQLite database (other drivers via db_connection) located in dbdir and named dbname. The data table combines all play by play data for every available game back to the 2014 season and adds the most recent completed games as soon as they are available for cfbfastR.

The argument force_rebuild is of hybrid type. It can rebuild the play by play data table either for the whole cfbfastR era (with force_rebuild = TRUE) or just for specified seasons (e.g. force_rebuild = c(2019, 2020)). Please note the following behavior:

- force_rebuild = TRUE: The data table with the name tblname will be removed completely and rebuilt from scratch. This is helpful when new columns are added during the Off-Season.
- force_rebuild = c(2019, 2020): The data table with the name tblname will be preserved and only rows from the 2019 and 2020 seasons will be deleted and re-added. This is intended to be used for ongoing seasons because ESPN’s data provider can make changes to the underlying data during the week.

The parameter db_connection is intended for advanced users who want to use other DBI drivers, such as MariaDB, Postgres or odbc. Please note that the arguments dbdir and dbname are dropped in case a db_connection is provided but the argument tblname will still be used to write the data table into the database.
cfbd_stats_season_team(), 81
cfbd_team_info, 84
cfbd_team_info(), 9, 84, 105
cfbd_team_matchup, 85, 86
cfbd_team_matchup_records, 86
cfbd_team_matchup_records(), 87
cfbd_team_roster, 87
cfbd_team_roster(), 88, 104
cfbd_team_talent, 88
cfbd_team_talent(), 89
cfbd_teams, 83
cfbd_venues, 89
csv_from_url, 90
data, 90
DBI::dbConnect(), 103, 107
draft (cfbd_draft), 8
espn_cfb_calendar, 91
espn_cfb_calendar(), 91
espn_cfb_pbp, 92
espn_cfb_schedule
espNet_cfb_schedule(), 92
espn_cfb_schedule(), 93
espn_cfb_scoreboard, 92
espn_cfb_scoreboard(), 93
espn_cfb_team_stats, 95
espn_metrics, 101
espn_metrics_wp(espn_metrics), 101
espn_metrics_wp(), 102
espn_ratings_fpi, 102
has_cfbd_key (register_cfbd), 106
load_cfb_pbp, 103
load_cfb_rosters, 104
load_cfb_schedules, 104
load_cfb_teams, 105
nfl (cfbd_draft), 8
nfl_draft (cfbd_draft), 8
nfl_teams (cfbd_draft), 8
rds_from_url, 106
readRDS(), 106
recruiting (cfbd_recruiting), 65
register_cfbd, 106
update_cfb_db, 107
update_cfb_db(), 103