

Package ‘nhlscraper’

November 30, 2025

Title Scraper for National Hockey League Data

Version 0.4.0

Description

Scrapes and cleans data from the 'NHL' and 'ESPN' APIs into data.frames and lists. Wraps 125+ endpoints documented in <<https://github.com/RentoSaijo/nhlscraper/wiki>> from high-level multi-season summaries and award winners to low-level decisecond replays and bookmakers' odds, making them more accessible. Features cleaning and visualization tools, primarily for play-by-plays.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (>= 3.5.0)

Imports httr2 (>= 1.2.0), jsonlite (>= 2.0.0)

Suggests testthat (>= 3.0.0), knitr (>= 1.50.0), rmarkdown (>= 2.29.0)

Config/testthat/edition 3

VignetteBuilder knitr

URL <https://rentosaijo.github.io/nhlscraper/>,
<https://github.com/RentoSaijo/nhlscraper>

BugReports <https://github.com/RentoSaijo/nhlscraper/issues>

Copyright Copyright: NHL and the NHL Shield are registered trademarks of the National Hockey League. NHL and NHL team marks are the property of the NHL and its teams.

NeedsCompilation no

Author Rento Saijo [aut, cre, cph] (ORCID:
<<https://orcid.org/0009-0008-4919-7349>>),
Lars Skytte [ctb]

Maintainer Rento Saijo <rentosaijo0527@gmail.com>

Repository CRAN

Date/Publication 2025-11-30 16:30:14 UTC

Contents

attendance	6
awards	7
award_winners	7
boxscore	8
bracket	8
calculate_angle	9
calculate_distance	9
calculate_expected_goals_v1	10
calculate_expected_goals_v2	11
calculate_expected_goals_v3	12
coaches	12
coach_career_statistics	13
coach_franchise_statistics	13
combine_reports	14
countries	14
count_goals_shots	15
drafts	15
draft_picks	16
draft_prospects	16
draft_rankings	17
draft_tracker	17
draw_NHL_rink	18
espn_futures	18
espn_games	19
espn_game_odds	19
espn_game_summary	20
espn_injuries	20
espn_players	21
espn_player_summary	21
espn_play_by_play	22
espn_teams	22
espn_team_summary	23
espn_transactions	23
expansion_drafts	24
expansion_draft_picks	24
flag_is_home	25
flag_is_rebound	25
flag_is_rush	26
franchises	27
franchise_playoff_situational_results	27
franchise_season_statistics	28
franchise_statistics	28
franchise_team_statistics	29
franchise_versus_franchise	29
games	30
game_odds	30

game_rosters	31
game_type_now	31
gc_play_by_play	32
gc_play_by_plays	32
gc_summary	33
general_managers	33
get_attendance	34
get_awards	34
get_award_winners	34
get_bracket	35
get_configuration	35
get_countries	35
get_drafts	36
get_draft_picks	36
get_draft_rankings	36
get_draft_tracker	37
get_espn_athlete	37
get_espn_athletes	38
get_espn_coach	38
get_espn_coaches	38
get_espn_coach_career	39
get_espn_event	39
get_espn_events	39
get_espn_event_odds	40
get_espn_event_officials	40
get_espn_event_play_by_play	40
get_espn_event_stars	41
get_espn_futures	41
get_espn_injuries	41
get_espn_team	42
get_espn_teams	42
get_espn_transactions	42
get_franchises	43
get_franchise_season_by_season	43
get_franchise_team_totals	44
get_franchise_vs_franchise	44
get_games	45
get_game_boxscore	45
get_game_landing	46
get_game_story	46
get_gc_play_by_play	47
get_glossary	47
get_goalies	47
get_goalie_leaders	48
get_goalie_milestones	48
get_goalie_statistics	49
get_officials	49
get_partner_odds	49

get_players	50
get_player_game_log	50
get_player_landing	51
get_schedule	51
get_scoreboards	52
get_scores	52
get_seasons	53
get_season_now	53
get_series	53
get_series_schedule	54
get_shift_charts	54
get_skaters	55
get_skater_leaders	55
get_skater_milestones	56
get_skater_statistics	56
get_spotlight_players	56
get_standings	57
get_standings_information	57
get_streams	58
get_teams	58
get_team_prospects	58
get_team_roster	59
get_team_roster_statistics	59
get_team_schedule	60
get_team_scoreboard	60
get_team_seasons	61
get_team_statistics	61
get_tv_schedule	61
get_venues	62
get_wsc_play_by_play	62
glossary	63
goalie_edge_five_versus_five	63
goalie_edge_leaders	64
goalie_edge_save_percentage	65
goalie_edge_seasons	66
goalie_edge_shot_location	66
goalie_edge_summary	67
goalie_game_report	68
goalie_game_scoring	69
goalie_game_statistics	69
goalie_leaders	70
goalie_milestones	70
goalie_regular_statistics	71
goalie_report_configurations	71
goalie_scoring	72
goalie_season_report	72
goalie_season_statistics	73
goalie_series_statistics	74

goalie_statistics	74
ig_game_cumulative_expected_goals	75
ig_game_shot_locations	75
location	76
lottery_odds	77
normalize_coordinates	77
officials	78
penalty_shots	78
ping	79
players	79
player_game_log	80
player_seasons	80
player_summary	81
playoff_season_statistics	81
replay	82
roster	82
roster_statistics	83
schedule	84
scores	85
seasons	85
season_now	86
series	86
series_schedule	87
shifts	87
skater_edge_leaders	88
skater_edge_seasons	88
skater_edge_shot_location	89
skater_edge_shot_speed	90
skater_edge_skating_distance	91
skater_edge_skating_speed	92
skater_edge_summary	93
skater_edge_zone_time	93
skater_game_report	94
skater_leaders	95
skater_milestones	96
skater_playoff_statistics	96
skater_regular_statistics	97
skater_report_configurations	97
skater_season_report	98
skater_season_statistics	99
skater_series_statistics	99
skater_statistics	100
spotlight_players	100
standings	101
standings_rules	101
streams	102
strip_game_id	102
strip_situation_code	103

strip_time_period	103
teams	104
team_edge_leaders	105
team_edge_seasons	105
team_edge_shot_location	106
team_edge_shot_speed	107
team_edge_skating_distance	108
team_edge_skating_speed	109
team_edge_summary	110
team_edge_zone_time	110
team_game_report	111
team_logos	112
team_month_schedule	113
team_prospects	113
team_report_configurations	114
team_seasons	114
team_season_report	115
team_season_schedule	116
team_season_statistics	117
team_week_schedule	117
tv_schedule	118
venues	118
wsc_play_by_play	119
wsc_play_by_plays	119
wsc_summary	120
x_game_cumulative_expected_goals	120
x_game_shot_locations	121
Index	123

attendance	<i>Access the attendance by season and game type</i>
------------	--

Description

attendance() scrapes the attendance by season and game type.

Usage

attendance()

Value

data.frame with one row per season

Examples

all_attendance <- attendance()

awards	<i>Access all the awards</i>
--------	------------------------------

Description

awards() scrapes all the awards.

Usage

```
awards()
```

Value

data.frame with one row per award

Examples

```
all_awards <- awards()
```

award_winners	<i>Access all the award winners/finalists</i>
---------------	---

Description

award_winners() scrapes all the award winners/finalists.

Usage

```
award_winners()
```

Value

data.frame with one row per winner/finalist

Examples

```
all_award_winners <- award_winners()
```

boxscore	<i>Access the boxscore for a game, team, and position</i>
----------	---

Description

boxscore() scrapes the boxscore for a given set of game, team, and position.

Usage

```
boxscore(game = 2023030417, team = "home", position = "forwards")
```

Arguments

- game integer ID (e.g., 2025020275); see [games\(\)](#) for reference
- team character of 'h'/'home' or 'a'/'away'
- position character of 'f'/'forwards', 'd'/'defensemen', or 'g'/'goalies'

Value

data.frame with one row per player

Examples

```
boxscore_COL_forwards_Martin_Necas_legacy_game <- boxscore(  
  game = 2025020275,  
  team = 'H',  
  position = 'F'  
)
```

bracket	<i>Access the playoff bracket for a season</i>
---------	--

Description

bracket() scrapes the playoff bracket for a given season.

Usage

```
bracket(season = season_now())
```

Arguments

- season integer in YYYYYYYYY (e.g., 20242025); see [seasons\(\)](#) for reference

Value

data.frame with one row per series

Examples

```
bracket_20242025 <- bracket(season = 20242025)
```

calculate_angle	<i>Calculate the Euclidean angle from the attacking net for all the events (plays) in a play-by-play</i>
-----------------	--

Description

calculate_angle() calculates the Euclidean angle from the attacking net for all the events (plays) in a play-by-play.

Usage

```
calculate_angle(play_by_play)
```

Arguments

play_by_play data.frame of play-by-play(s); see [gc_play_by_play\(\)](#) and/or [wsc_play_by_play\(\)](#) for reference; must be untouched by non-nhlscraper functions; saves time if [normalize_coordinates\(\)](#) has already been called

Value

data.frame with one row per event (play) and added angle column

Examples

```
# May take >5s, so skip.

test <- gc_play_by_play()
test_is_home_flagged <- flag_is_home(test)
test_coords_normalized <- normalize_coordinates(test_is_home_flagged)
test_angle_calculated <- calculate_angle(test_coords_normalized)
```

calculate_distance	<i>Calculate the Euclidean distance from the attacking net for all the events (plays) in a play-by-play</i>
--------------------	---

Description

calculate_distance() calculates the Euclidean distance from the attacking net for all the events (plays) in a play-by-play.

Usage

```
calculate_distance(play_by_play)
```

Arguments

`play_by_play` data.frame of play-by-play(s); see [gc_play_by_play\(\)](#) and/or [wsc_play_by_play\(\)](#) for reference; must be untouched by non-nhlscraper functions; saves time if [normalize_coordinates\(\)](#) has already been called

Value

data.frame with one row per event (play) and added distance column

Examples

```
# May take >5s, so skip.

test          <- gc_play_by_play()
test_is_home_flagged <- flag_is_home(test)
test_coords_normalized <- normalize_coordinates(test_is_home_flagged)
test_distance_calculated <- calculate_distance(test_coords_normalized)
```

```
calculate_expected_goals_v1
```

Calculate version 1 of the expected goals for all the events (plays) in a play-by-play

Description

`calculate_expected_goals_v1()` calculates version 1 of the expected goals for all the events (plays) in a play-by-play using a pre-estimated logistic regression model of goal probability on distance, angle, empty net, and strength state.

Usage

```
calculate_expected_goals_v1(play_by_play)
```

```
calculate_xG_v1(play_by_play)
```

Arguments

`play_by_play` data.frame of play-by-play(s); see [gc_play_by_play\(\)](#) and/or [wsc_play_by_play\(\)](#) for reference; must be untouched by non-nhlscraper functions; saves time if [calculate_distance\(\)](#), [calculate_angle\(\)](#), and/or [strip_situation_code\(\)](#) have already been called

Value

data.frame with one row per event (play) and added xG_v1 column

Examples

```
# May take >5s, so skip.

test <- gc_play_by_play()
test <- calculate_expected_goals_v1(test)
```

calculate_expected_goals_v2

Calculate version 2 of the expected goals for all the events (plays) in a play-by-play

Description

calculate_expected_goals_v2() calculates version 2 of the expected goals for all the events (plays) in a play-by-play using a pre-estimated logistic regression model of goal probability on distance, angle, empty net, strength state, rebound, and rush indicators.

Usage

```
calculate_expected_goals_v2(play_by_play)

calculate_xG_v2(play_by_play)
```

Arguments

play_by_play data.frame of play-by-play(s); see [gc_play_by_play\(\)](#) and/or [wsc_play_by_play\(\)](#) for reference; must be untouched by non-nhlscraper functions; saves time if [calculate_distance\(\)](#), [calculate_angle\(\)](#), [strip_situation_code\(\)](#), [flag_is_rebound\(\)](#), and/or [flag_is_rush\(\)](#) have already been called

Value

data.frame with one row per event (play) and added xG_v2 column

Examples

```
# May take >5s, so skip.

test <- gc_play_by_play()
test <- calculate_expected_goals_v2(test)
```

```
calculate_expected_goals_v3
```

Calculate version 3 of the expected goals for all the events (plays) in a play-by-play

Description

`calculate_expected_goals_v3()` calculates version 3 of the expected goals for all the events (plays) in a play-by-play using a pre-estimated logistic regression model of goal probability on distance, angle, empty net, strength state, rebound, rush, and goal differential.

Usage

```
calculate_expected_goals_v3(play_by_play)
```

```
calculate_xG_v3(play_by_play)
```

Arguments

`play_by_play` data.frame of play-by-play(s); see [gc_play_by_play\(\)](#) and/or [wsc_play_by_play\(\)](#) for reference; must be untouched by non-nhlscraper functions; saves time if [calculate_distance\(\)](#), [calculate_angle\(\)](#), [strip_situation_code\(\)](#), [flag_is_rebound\(\)](#), [flag_is_rush\(\)](#), and/or [count_goals_shots\(\)](#) have already been called

Value

data.frame with one row per event (play) and an added `xG_v3` column containing expected goals for applicable shot attempts.

Examples

```
# May take >5s, so skip.

test <- gc_play_by_play()
test <- calculate_expected_goals_v3(test)
```

```
coaches
```

Access all the coaches

Description

`coaches()` scrapes all the coaches.

Usage

```
coaches()
```

Value

data.frame with one row per coach

Examples

```
all_coaches <- coaches()
```

```
coach_career_statistics
```

Access the career statistics for all the coaches

Description

coach_career_statistics() scrapes the career results for all the coaches.

Usage

```
coach_career_statistics()
```

```
coach_career_stats()
```

Value

data.frame with one row per coach

Examples

```
coach_career_stats <- coach_career_statistics()
```

```
coach_franchise_statistics
```

Access the statistics for all the coaches by franchise and game type

Description

coach_franchise_statistics() scrapes the statistics for all the coaches by franchise and game type.

Usage

```
coach_franchise_statistics()
```

```
coach_franchise_stats()
```

Value

data.frame with one row per franchise per coach per game type

Examples

```
coach_franchise_stats <- coach_franchise_statistics()
```

combine_reports	<i>Access the draft combine reports</i>
-----------------	---

Description

combine_reports() scrapes the draft combine reports.

Usage

```
combine_reports()
```

Value

data.frame with one row per player

Examples

```
combine_reports <- combine_reports()
```

countries	<i>Access all the countries</i>
-----------	---------------------------------

Description

countries scrapes all the countries.

Usage

```
countries()
```

Value

data.frame with one row per country

Examples

```
all_countries <- countries()
```

count_goals_shots	<i>Count the as-of-event goal, shots on goal, Fenwick, and Corsi attempts and differentials for all the events (plays) in a play-by-play by perspective</i>
-------------------	---

Description

count_goals_shots() counts the as-of-event goal, shots on goal, Fenwick, and Corsi attempts and differentials for all the events (plays) in a play-by-play by perspective.

Usage

```
count_goals_shots(play_by_play)
```

Arguments

play_by_play data.frame of play-by-play(s); see [gc_play_by_play\(\)](#) and/or [wsc_play_by_play\(\)](#) for reference; must be untouched by non-nhlscraper functions; saves time if [flag_is_home\(\)](#) has already been called

Value

data.frame with one row per event (play) and added columns: homeGoals, awayGoals, homeSOG, awaySOG, homeFenwick, awayFenwick, homeCorsi, awayCorsi, goalsFor, goalsAgainst, SOGFor, SOGAgainst, fenwickFor, fenwickAgainst, corсиFor, corсиAgainst, goalDifferential, SOGDifferential, fenwickDifferential, and corсиDifferential

Examples

```
# May take >5s, so skip.

test          <- gc_play_by_play()
test_is_home_flagged <- flag_is_home(test)
test_goals_shots_counted <- count_goals_shots(test_is_home_flagged)
```

drafts	<i>Access all the drafts</i>
--------	------------------------------

Description

drafts() scrapes all the drafts.

Usage

```
drafts()
```

Value

data.frame with one row per draft

Examples

```
all_drafts <- drafts()
```

draft_picks	<i>Access all the draft picks</i>
-------------	-----------------------------------

Description

draft_picks() scrapes all the draft picks.

Usage

```
draft_picks()
```

Value

data.frame with one row per pick

Examples

```
# May take >5s, so skip.  
all_draft_picks <- draft_picks()
```

draft_prospects	<i>Access all the draft prospects</i>
-----------------	---------------------------------------

Description

draft_prospects() scrapes all the draft prospects.

Usage

```
draft_prospects()
```

Value

data.frame with one row per player

Examples

```
# May take >5s, so skip.  
all_prospects <- draft_prospects()
```

draft_rankings	<i>Access the draft rankings for a class and category</i>
----------------	---

Description

draft_rankings() scrapes the draft rankings for a given set of class and category.

Usage

```
draft_rankings(class = season_now()%%10000, category = 1)
```

Arguments

class	integer in YYYY (e.g., 2017); see drafts() for reference
category	integer in 1:4 (where 1 = North American Skaters, 2 = International Skaters, 3 = North American Goalies, and 4 = International Goalies) OR character of 'NAS'/'NA Skaters'/'North American Skaters', 'INTLS'/'INTL Skaters'/'International Skaters', 'NAG'/'NA Goalies'/'North American Goalies', 'INTLG'/'INTL Goalies'/'International Goalies'

Value

data.frame with one row per player

Examples

```
draft_rankings_INTL_Skaters_2017 <- draft_rankings(
  class = 2017,
  category = 2
)
```

draft_tracker	<i>Access the real-time draft tracker</i>
---------------	---

Description

draft_tracker() scrapes the real-time draft tracker.

Usage

```
draft_tracker()
```

Value

data.frame with one row per player

Examples

```
draft_tracker <- draft_tracker()
```

draw_NHL_rink	<i>Draw a full NHL rink</i>
---------------	-----------------------------

Description

draw_NHL_rink() draws a full NHL rink such that the x and y coordinates span -100 to 100 and -43 to +43, respectively. Use [graphics::points\(\)](#) to create custom graphs; check out an example on the online documentation!

Usage

```
draw_NHL_rink()
```

Value

NULL

Examples

```
draw_NHL_rink()
```

espn_futures	<i>Access the ESPN futures for a season</i>
--------------	---

Description

espn_futures() scrapes the ESPN futures for a given season.

Usage

```
espn_futures(season = season_now())
```

Arguments

season integer in YYYYYYYYY (e.g., 20242025); see [seasons\(\)](#) for reference

Value

nested data.frame with one row per type (outer) and book (inner)

Examples

```
ESPN_futures_20252026 <- espn_futures(20252026)
```

espn_games	<i>Access the ESPN games for a season</i>
------------	---

Description

espn_games() scrapes the ESPN games for a given season.

Usage

```
espn_games(season = season_now())
```

Arguments

season integer in YYYYYYYY (e.g., 20242025); see [seasons\(\)](#) for reference

Value

data.frame with one row per ESPN game

Examples

```
ESPN_games_20242025 <- espn_games(season = 20242025)
```

espn_game_odds	<i>Access the ESPN odds for a game</i>
----------------	--

Description

espn_game_odds() scrapes the ESPN odds for a given game.

Usage

```
espn_game_odds(game = 401777460)
```

Arguments

game integer ID (e.g., 401777460); see [espn_games\(\)](#) for reference

Value

data.frame with one row per provider

Examples

```
ESPN_odds_SCF_20242025 <- espn_game_odds(game = 401777460)
```

espn_game_summary	<i>Access the ESPN summary for a game</i>
-------------------	---

Description

espn_game_summary() scrapes the ESPN summary for a game.

Usage

```
espn_game_summary(game = 401777460)
```

Arguments

game integer ID (e.g., 401777460); see [espn_games\(\)](#) for reference

Value

list with various items

Examples

```
ESPN_summary_SCF_20242025 <- espn_game_summary(game = 401777460)
```

espn_injuries	<i>Access the real-time ESPN injury reports</i>
---------------	---

Description

espn_injuries() scrapes the real-time ESPN injury reports for all the teams.

Usage

```
espn_injuries()
```

Value

nested data.frame with one row per team (outer) and player (inner)

Examples

```
ESPN_injuries_now <- espn_injuries()
```

espn_players	<i>Access all the ESPN players</i>
--------------	------------------------------------

Description

espn_players() scrapes all the ESPN players.

Usage

```
espn_players()
```

Value

data.frame with one row per ESPN player

Examples

```
all_ESPN_players <- espn_players()
```

espn_player_summary	<i>Access the ESPN summary for a player</i>
---------------------	---

Description

espn_player_summary() scrapes the ESPN summary for a player.

Usage

```
espn_player_summary(player = 3988803)
```

Arguments

player integer ID (e.g., 3988803); see [espn_players\(\)](#) for reference

Value

data.frame with one row

Examples

```
ESPN_summary_Charlie_McAvoy <- espn_player_summary(player = 3988803)
```

espn_play_by_play	<i>Access the ESPN play-by-play for a game</i>
-------------------	--

Description

`espn_play_by_play()` scrapes the ESPN play-by-play for a given game.

Usage

```
espn_play_by_play(game = 401777460)
```

```
espn_pbp(game = 401777460)
```

Arguments

`game` integer ID (e.g., 401777460); see [espn_games\(\)](#) for reference

Value

data.frame with one row per event (play)

Examples

```
ESPN_pbp_SCF_20242025 <- espn_play_by_play(game = 401777460)
```

espn_teams	<i>Access all the ESPN teams</i>
------------	----------------------------------

Description

`espn_teams()` scrapes all the ESPN teams.

Usage

```
espn_teams()
```

Value

data.frame with one row per ESPN team

Examples

```
all_ESPN_teams <- espn_teams()
```

espn_team_summary	<i>Access the ESPN summary for a team</i>
-------------------	---

Description

espn_team_summary() scrapes the ESPN summary for a team.

Usage

```
espn_team_summary(team = 3988803)
```

Arguments

team	integer ID (e.g., 1); see espn_teams() for reference
------	--

Value

data.frame with one row

Examples

```
ESPN_summary_Boston_Bruins <- espn_team_summary(team = 1)
```

espn_transactions	<i>Access the ESPN transactions for a season</i>
-------------------	--

Description

espn_transactions() scrapes the ESPN transactions for a given season.

Usage

```
espn_transactions(season = season_now())
```

Arguments

season	integer in YYYYYYYY (e.g., 20242025); the summer of the latter year is included
--------	---

Value

data.frame with one row per transaction

Examples

```
ESPN_transactions_20242025 <- espn_transactions(season = 20242025)
```

expansion_drafts	<i>Access all the expansion drafts</i>
------------------	--

Description

expansion_drafts() scrapes all the expansion drafts.

Usage

```
expansion_drafts()
```

Value

data.frame with one row per expansion draft

Examples

```
all_expansion_drafts <- expansion_drafts()
```

expansion_draft_picks	<i>Access all the expansion draft picks</i>
-----------------------	---

Description

expansion_draft_picks() scrapes all the expansion draft picks.

Usage

```
expansion_draft_picks()
```

Value

data.frame with one row per pick

Examples

```
all_expansion_draft_picks <- expansion_draft_picks()
```

flag_is_home	<i>Flag if the event belongs to the home team or not for all the events (plays) in a play-by-play</i>
--------------	---

Description

flag_is_home() flags if the event belongs to the home team or not for all the events (plays) in a play-by-play.

Usage

```
flag_is_home(play_by_play)
```

Arguments

play_by_play data.frame of play-by-play(s); see [gc_play_by_play\(\)](#) and/or [wsc_play_by_play\(\)](#) for reference; must be untouched by non-nhlscraper functions

Value

data.frame with one row per event (play) and added isHome column

Examples

```
# May take >5s, so skip.

test          <- gc_play_by_play()
test_is_home_flagged <- flag_is_home(test)
```

flag_is_rebound	<i>Flag if the shot attempt is a rebound attempt or not for all the shots in a play-by-play</i>
-----------------	---

Description

flag_is_rebound() flags if the shot attempt is a rebound attempt or not for all the shots in a play-by-play.

Usage

```
flag_is_rebound(play_by_play)
```

Arguments

play_by_play data.frame of play-by-play(s); see [gc_play_by_play\(\)](#) and/or [wsc_play_by_play\(\)](#) for reference; must be untouched by non-nhlscraper functions; saves time if [flag_is_home\(\)](#) and/or

Value

data.frame with one row per event (play) and added isRebound column

Examples

```
# May take >5s, so skip.
```

```
test          <- gc_play_by_play()
test_is_home_flagged <- flag_is_home(test)
test_game_id_stripped <- strip_game_id(test_is_home_flagged)
test_time_period_stripped <- strip_time_period(test_game_id_stripped)
test_is_rebound_flagged <- flag_is_rebound(test_time_period_stripped)
```

flag_is_rush	<i>Flag if the shot attempt is a rush attempt or not for all the shots in a play-by-play</i>
--------------	--

Description

flag_is_rush() flags if the shot attempt is a rush attempt or not for all the shots in a play-by-play.

Usage

```
flag_is_rush(play_by_play)
```

Arguments

play_by_play data.frame of play-by-play(s); see [gc_play_by_play\(\)](#) and/or [wsc_play_by_play\(\)](#) for reference; must be untouched by non-nhlscraper functions; saves time if [flag_is_home\(\)](#) and [strip_time_period\(\)](#) had already been called

Value

data.frame with one row per event (play) and added isRush column

Examples

```
# May take >5s, so skip.
```

```
test          <- gc_play_by_play()
test_is_home_flagged <- flag_is_home(test)
test_game_id_stripped <- strip_game_id(test_is_home_flagged)
test_time_period_stripped <- strip_time_period(test_game_id_stripped)
test_is_rush_flagged <- flag_is_rush(test_time_period_stripped)
```

franchises	<i>Access all the franchises</i>
------------	----------------------------------

Description

franchises() scrapes all the franchises.

Usage

```
franchises()
```

Value

data.frame with one row per franchise

Examples

```
all_franchises <- franchises()
```

franchise_playoff_situational_results	<i>Access the playoff series results for all the franchises by situation</i>
---------------------------------------	--

Description

franchise_playoff_situational_results() scrapes the playoff series results for all the franchises by situation.

Usage

```
franchise_playoff_situational_results()
```

Value

data.frame with one row per franchise per situation

Examples

```
franchise_playoff_situational_results <-  
franchise_playoff_situational_results()
```

`franchise_season_statistics`*Access the statistics for all the franchises by season and game type*

Description

`franchise_season_statistics()` scrapes the statistics for all the franchises by season and game type.

Usage

```
franchise_season_statistics()
```

```
franchise_season_stats()
```

Value

data.frame with one row per franchise per season per game type

Examples

```
# May take >5s, so skip.  
franchise_season_stats <- franchise_season_statistics()
```

`franchise_statistics` *Access the all-time statistics for all the franchises by game type*

Description

`franchise_statistics()` scrapes the all-time statistics for all the franchises by game type.

Usage

```
franchise_statistics()
```

```
franchise_stats()
```

Value

data.frame with one row per franchise per game type

Examples

```
franchise_stats <- franchise_statistics()
```

```
franchise_team_statistics
```

Access the all-time statistics for all the franchises by team and game type

Description

`franchise_team_statistics()` scrapes the all-time statistics for all the franchises by team and game type.

Usage

```
franchise_team_statistics()
```

```
franchise_team_stats()
```

Value

data.frame with one row per team per franchise per game type

Examples

```
franchise_team_stats <- franchise_team_statistics()
```

```
franchise_versus_franchise
```

Access the all-time statistics versus other franchises for all the franchises by game type

Description

`franchise_versus_franchise()` scrapes the all-time statistics versus other franchises for all the franchises by game type.

Usage

```
franchise_versus_franchise()
```

```
franchise_vs_franchise()
```

Value

data.frame with one row per franchise per franchise per game type

Examples

```
# May take >5s, so skip.  
franchise_vs_franchise <- franchise_versus_franchise()
```

games	<i>Access all the games</i>
-------	-----------------------------

Description

games() scrapes all the games.

Usage

```
games()
```

Value

data.frame with one row per game

Examples

```
# May take >5s, so skip.  
all_games <- games()
```

game_odds	<i>Access the real-time game odds for a country by partnered bookmaker</i>
-----------	--

Description

game_odds() scrapes the real-time game odds for a given country by partnered bookmaker.

Usage

```
game_odds(country = "US")
```

Arguments

country two-letter code (e.g., 'CA'); see [countries\(\)](#) for reference

Value

data.frame with one row per game

Examples

```
game_odds_CA <- game_odds(country = 'CA')
```

game_rosters	<i>Access the rosters for a game</i>
--------------	--------------------------------------

Description

game_rosters() scrapes the rosters for a given game.

Usage

```
game_rosters(game = 2023030417)
```

Arguments

game integer ID (e.g., 2025020275); see [games\(\)](#) for reference

Value

data.frame with one row per player

Examples

```
rosters_Martin_Necas_legacy_game <- game_rosters(game = 2025020275)
```

game_type_now	<i>Access the game type as of now</i>
---------------	---------------------------------------

Description

game_type_now() scrapes the current game type.

Usage

```
game_type_now()
```

Value

integer in 1:3 (where 1 = pre-season, 2 = regular season, 3 = playoff/post-season)

Examples

```
game_type_now <- game_type_now()
```

gc_play_by_play	<i>Access the GameCenter (GC) play-by-play for a game</i>
-----------------	---

Description

gc_play_by_play() scrapes the GC play-by-play for a given game.

Usage

```
gc_play_by_play(game = 2023030417)
```

```
gc_pbp(game = 2023030417)
```

Arguments

game integer ID (e.g., 2025020275); see [games\(\)](#) for reference

Value

data.frame with one row per event (play)

Examples

```
gc_pbp_Martin_Necas_legacy_game <- gc_play_by_play(game = 2025020275)
```

gc_play_by_plays	<i>Access the GameCenter (GC) play-by-plays for a season</i>
------------------	--

Description

gc_play_by_plays() loads the GC play-by-plays for a given season.

Usage

```
gc_play_by_plays(season = 20242025)
```

```
gc_pbps(season = 20242025)
```

Arguments

season integer in YYYYYYYY (e.g., 20242025); see [seasons\(\)](#) for reference

Value

data.frame with one row per event (play) per game

Examples

```
# May take >5s, so skip.
gc_pbps_20212022 <- gc_play_by_plays(season = 20212022)
```

gc_summary	<i>Access the GameCenter (GC) summary for a game</i>
------------	--

Description

gc_summary() scrapes the GC summary for a given game.

Usage

```
gc_summary(game = 2023030417)
```

Arguments

game integer ID (e.g., 2025020275); see [games\(\)](#) for reference

Value

list of various items

Examples

```
gc_summary_Martin_Necas_legacy_game <- gc_summary(game = 2025020275)
```

general_managers	<i>Access all the general managers</i>
------------------	--

Description

general_managers() scrapes all the general managers.

Usage

```
general_managers()
```

```
gms()
```

Value

data.frame with one row per general manager

Examples

```
all_GMs <- general_managers()
```

get_attendance	<i>Access the attendance by season and game type</i>
----------------	--

Description

get_attendance() is deprecated. Use [attendance\(\)](#) instead.

Usage

```
get_attendance()
```

Value

data.frame with one row per season

get_awards	<i>Access all the awards</i>
------------	------------------------------

Description

get_awards() is deprecated. Use [awards\(\)](#) instead.

Usage

```
get_awards()
```

Value

data.frame with one row per award

get_award_winners	<i>Access all the award winners/finalists</i>
-------------------	---

Description

get_award_winners() is deprecated. Use [award_winners\(\)](#) instead.

Usage

```
get_award_winners()
```

Value

data.frame with one row per winner/finalist

get_bracket	<i>Access the playoff bracket for a season</i>
-------------	--

Description

get_bracket() is deprecated. Use [bracket\(\)](#) instead.

Usage

```
get_bracket(season = season_now())
```

Arguments

season integer in YYYYYYYYY (e.g., 20242025); see [seasons\(\)](#) for reference

Value

data.frame with one row per series

get_configuration	<i>Access the configurations for team, skater, and goalie reports</i>
-------------------	---

Description

get_configuration() is defunct. Use [team_report_configurations\(\)](#), [skater_report_configurations\(\)](#), and/or [goalie_report_configurations\(\)](#) instead.

Usage

```
get_configuration()
```

get_countries	<i>Access all the countries</i>
---------------	---------------------------------

Description

get_countries() is deprecated. Use [countries\(\)](#) instead.

Usage

```
get_countries()
```

Value

data.frame with one row per country

get_drafts	<i>Access all the drafts</i>
------------	------------------------------

Description

get_drafts() is deprecated. Use [drafts\(\)](#) instead.

Usage

```
get_drafts()
```

Value

data.frame with one row per draft

get_draft_picks	<i>Access all the draft picks</i>
-----------------	-----------------------------------

Description

get_draft_picks() is deprecated. Use [draft_picks\(\)](#) instead.

Usage

```
get_draft_picks()
```

Value

data.frame with one row per pick

get_draft_rankings	<i>Access the draft rankings for a year and player type</i>
--------------------	---

Description

get_draft_rankings() is deprecated. Use [draft_rankings\(\)](#) instead.

Usage

```
get_draft_rankings(year = season_now()%%10000, player_type = 1)
```

Arguments

year	integer in YYYY (e.g., 2017); see drafts() for reference
player_type	integer in 1:4 (where 1 = North American Skaters, 2 = International Skaters, 3 = North American Goalies, and 4 = International Goalies)

Value

data.frame with one row per player

get_draft_tracker	<i>Access the real-time draft tracker</i>
-------------------	---

Description

get_draft_tracker() is deprecated. Use [draft_tracker\(\)](#) instead.

Usage

```
get_draft_tracker()
```

Value

data.frame with one row per player

get_espn_athlete	<i>Access the ESPN summary for an athlete (player) and season</i>
------------------	---

Description

get_espn_athlete() is defunct. Use [espn_player_summary\(\)](#) instead.

Usage

```
get_espn_athlete()
```

get_espn_athletes	<i>Access all the ESPN athletes (players)</i>
-------------------	---

Description

get_espn_athletes() is deprecated. Use [espn_players\(\)](#) instead.

Usage

```
get_espn_athletes()
```

Value

data.frame with one row per ESPN athlete (player)

get_espn_coach	<i>Access the ESPN statistics for a coach and (multiple) season(s)</i>
----------------	--

Description

get_espn_coach() is defunct. Use [coach_career_statistics\(\)](#) instead.

Usage

```
get_espn_coach()
```

get_espn_coaches	<i>Access the ESPN coaches for a season</i>
------------------	---

Description

get_espn_coaches() is defunct. Use [coaches\(\)](#) instead.

Usage

```
get_espn_coaches()
```

get_espn_coach_career *Access the career ESPN statistics for a coach*

Description

get_espn_coach_career() is defunct. Use [coach_career_statistics\(\)](#) instead.

Usage

```
get_espn_coach_career()
```

get_espn_event *Access the ESPN summary for an event (game)*

Description

get_espn_event() is deprecated. Use [espn_game_summary\(\)](#) instead.

Usage

```
get_espn_event(event = 401777460)
```

Arguments

event integer ID (e.g., 401777460); see [espn_games\(\)](#) for reference

Value

data.frame with one row per event (game)

get_espn_events *Access the ESPN events (games) by start and end dates*

Description

get_espn_events() is defunct. Use [espn_games\(\)](#) instead.

Usage

```
get_espn_events()
```

`get_espn_event_odds` *Access the ESPN odds for an event (game)*

Description

`get_espn_event_odds()` is deprecated. Use [espn_game_odds\(\)](#) instead.

Usage

```
get_espn_event_odds(event = 401777460)
```

Arguments

`event` integer ID (e.g., 401777460); see [espn_games\(\)](#) for reference

Value

data.frame with one row per provider

`get_espn_event_officials`
Access the officials for an ESPN event (game)

Description

`get_espn_event_officials()` is defunct. Use [gc_summary\(\)](#) and/or [wsc_summary\(\)](#) instead.

Usage

```
get_espn_event_officials()
```

`get_espn_event_play_by_play`
Access the ESPN play-by-play for an event (game)

Description

`get_espn_event_play_by_play()` is deprecated. Use [espn_play_by_play\(\)](#) instead.

Usage

```
get_espn_event_play_by_play(event = 401777460)
```


Arguments

event integer ID (e.g., 401777460); see [espn_games\(\)](#) for reference

Value

data.frame with one row per play

get_espn_event_stars *Access the three stars for an ESPN event (game)*

Description

get_espn_event_stars() is defunct. Use [gc_summary\(\)](#) and/or [wsc_summary\(\)](#) instead.

Usage

get_espn_event_stars()

get_espn_futures *Access the ESPN futures for a season*

Description

get_espn_futures() is defunct. Use [espn_futures\(\)](#) instead.

Usage

get_espn_futures()

get_espn_injuries *Access the real-time ESPN injury reports*

Description

get_espn_injuries() is deprecated. Use [espn_injuries\(\)](#) instead.

Usage

get_espn_injuries()

Value

nested data.frame with one row per team (outer) and player (inner)

get_espn_team	<i>Access the ESPN summary for a team and season</i>
---------------	--

Description

get_espn_team() is defunct. Use [espn_team_summary\(\)](#) instead.

Usage

```
get_espn_team()
```

get_espn_teams	<i>Access all the ESPN teams for a season</i>
----------------	---

Description

get_espn_teams() is defunct. Use [espn_teams\(\)](#) instead.

Usage

```
get_espn_teams()
```

get_espn_transactions	<i>Access the ESPN transactions by start and end dates</i>
-----------------------	--

Description

get_espn_transactions() is defunct. Use [espn_transactions\(\)](#) instead.

Usage

```
get_espn_transactions()
```

get_franchises	<i>Access all the franchises</i>
----------------	----------------------------------

Description

get_franchises() is deprecated. Use [franchises\(\)](#) instead.

Usage

```
get_franchises()
```

Value

data.frame with one row per franchise

get_franchise_season_by_season	<i>Access the statistics for all the franchises by season and game type</i>
--------------------------------	---

Description

get_franchise_season_by_season() is deprecated. Use [franchise_season_statistics\(\)](#) instead.

Usage

```
get_franchise_season_by_season()
```

Value

data.frame with one row per franchise per season per game type

`get_franchise_team_totals`

Access the all-time statistics for all the franchises by team and game type

Description

`get_franchise_team_totals()` is deprecated. Use [franchise_team_statistics\(\)](#) instead.

Usage

```
get_franchise_team_totals()
```

Value

data.frame with one row per team per franchise per game type

`get_franchise_vs_franchise`

Access the all-time statistics versus other franchises for all the franchises by game type

Description

`get_franchise_vs_franchise()` is deprecated. Use [franchise_versus_franchise\(\)](#) instead.

Usage

```
get_franchise_vs_franchise()
```

Value

data.frame with one row per franchise per franchise per game type

get_games	<i>Access all the games</i>
-----------	-----------------------------

Description

get_games() is deprecated. Use [games\(\)](#) instead.

Usage

```
get_games()
```

Value

data.frame with one row per game

get_game_boxscore	<i>Access the boxscore for a game, team, and player type</i>
-------------------	--

Description

get_game_boxscore() is deprecated. Use [boxscore\(\)](#) instead.

Usage

```
get_game_boxscore(game = 2023030417, team = "home", player_type = "forwards")
```

Arguments

game	integer ID (e.g., 2025020275); see games() for reference
team	character of 'home' or 'away'
player_type	character of 'forwards', 'defense', or 'goalies'

Value

data.frame with one row per player

get_game_landing	<i>Access the GameCenter (GC) summary for a game</i>
------------------	--

Description

get_game_landing() is deprecated. Use [gc_summary\(\)](#) instead.

Usage

```
get_game_landing(game = 2023030417)
```

Arguments

game	integer ID (e.g., 2025020275); see games() for reference
------	--

Value

list of various items

get_game_story	<i>Access the World Showcase (WSC) summary for a game</i>
----------------	---

Description

get_game_story() is deprecated. Use [wsc_summary\(\)](#) instead.

Usage

```
get_game_story(game = 2023030417)
```

Arguments

game	integer ID (e.g., 2025020275); see games() for reference
------	--

Value

list of various items

get_gc_play_by_play	<i>Access the GameCenter (GC) play-by-play for a game</i>
---------------------	---

Description

get_gc_play_by_play() is deprecated. Use [gc_play_by_play\(\)](#) instead.

Usage

```
get_gc_play_by_play(game = 2023030417)
```

Arguments

game	integer ID (e.g., 2025020275); see games() for reference
------	--

Value

data.frame with one row per event (play)

get_glossary	<i>Access the glossary</i>
--------------	----------------------------

Description

get_glossary() is deprecated. Use [glossary\(\)](#) instead.

Usage

```
get_glossary()
```

Value

data.frame with one row per terminology

get_goalies	<i>Access all the goalies for a range of seasons</i>
-------------	--

Description

get_goalies() is defunct. Use [players\(\)](#) instead.

Usage

```
get_goalies()
```

get_goalie_leaders	<i>Access the goalie statistics leaders for a season, game type, and category</i>
--------------------	---

Description

get_goalie_leaders() is deprecated. Use [goalie_leaders\(\)](#) instead.

Usage

```
get_goalie_leaders(season = "current", game_type = "", category = "wins")
```

Arguments

- | | |
|-----------|---|
| season | integer in YYYYYYYYY (e.g., 20242025); see seasons() for reference |
| game_type | integer in 1:3 (where 1 = pre-season, 2 = regular season, 3 = playoff/post-season)
OR character of 'pre', 'regular', or playoff/'post'; see seasons() for reference;
most functions will NOT support pre-season |
| category | character of 'wins', 'shutouts', 'savePctg', or 'goalsAgainstAverage' |

Value

data.frame with one row per player

get_goalie_milestones	<i>Access the goalies on milestone watch</i>
-----------------------	--

Description

get_goalie_milestones() is deprecated. Use [goalie_milestones\(\)](#) instead.

Usage

```
get_goalie_milestones()
```

Value

data.frame with one row per player

`get_goalie_statistics` *Access various reports for all the goalies by season or game*

Description

`get_goalie_statistics()` is defunct. Use [goalie_season_report\(\)](#) or [goalie_game_report\(\)](#) instead.

Usage

```
get_goalie_statistics()
```

`get_officials` *Access all the officials*

Description

`get_officials()` is deprecated. Use [officials\(\)](#) instead.

Usage

```
get_officials()
```

Value

data.frame with one row per official

`get_partner_odds` *Access the real-time game odds for a country by partnered bookmaker*

Description

`get_partner_odds()` is deprecated. Use [game_odds\(\)](#) instead.

Usage

```
get_partner_odds(country = "US")
```

Arguments

country two-letter code (e.g., 'CA'); see [countries\(\)](#) for reference

Value

data.frame with one row per game

get_players	<i>Access all the players</i>
-------------	-------------------------------

Description

get_players() is deprecated. Use [players\(\)](#) instead.

Usage

```
get_players()
```

Value

data.frame with one row per player

get_player_game_log	<i>Access the game log for a player, season, and game type</i>
---------------------	--

Description

get_player_game_log() is deprecated. Use [player_game_log\(\)](#) instead.

Usage

```
get_player_game_log(player = 8478402, season = "now", game_type = "")
```

Arguments

- | | |
|-----------|--|
| player | integer ID (e.g., 8480039); see players() for reference |
| season | integer in YYYYYYYY (e.g., 20242025); see seasons() for reference |
| game_type | integer in 1:3 (where 1 = pre-season, 2 = regular season, 3 = playoff/post-season)
OR character of 'pre', 'regular', or playoff'/'post'; see seasons() for reference;
most functions will NOT support pre-season |

Value

data.frame with one row per game

get_player_landing	<i>Access the summary for a player</i>
--------------------	--

Description

get_player_landing() is deprecated. Use [player_summary\(\)](#) instead.

Usage

```
get_player_landing(player = 8478402)
```

Arguments

player integer ID (e.g., 8480039); see [players\(\)](#) for reference

Value

list with various items

get_schedule	<i>Access the schedule for a date</i>
--------------	---------------------------------------

Description

get_schedule() is deprecated. Use [schedule\(\)](#) instead.

Usage

```
get_schedule(date = "2025-01-01")
```

Arguments

date character in 'YYYY-MM-DD' (e.g., '2025-01-01'); see [seasons\(\)](#) for reference

Value

data.frame with one row per game

get_scoreboards	<i>Access the scoreboards for a date</i>
-----------------	--

Description

get_scoreboards() is deprecated. Use [scores\(\)](#) instead.

Usage

```
get_scoreboards(date = "now")
```

Arguments

date	character in 'YYYY-MM-DD' (e.g., '2025-01-01'); see seasons() for reference
------	---

Value

data.frame with one row per game

get_scores	<i>Access the scores for a date</i>
------------	-------------------------------------

Description

get_scores() is deprecated. Use [scores\(\)](#) instead.

Usage

```
get_scores(date = "now")
```

Arguments

date	character in 'YYYY-MM-DD' (e.g., '2025-01-01'); see seasons() for reference
------	---

Value

data.frame with one row per game

get_seasons	<i>Access all the seasons</i>
-------------	-------------------------------

Description

get_seasons() is deprecated. Use [seasons\(\)](#) instead.

Usage

```
get_seasons()
```

Value

data.frame with one row per season

get_season_now	<i>Access the season and game type as of now</i>
----------------	--

Description

get_season_now() is defunct. Use [season_now\(\)](#) and/or [game_type_now\(\)](#) instead.

Usage

```
get_season_now()
```

get_series	<i>Access the playoff series for a season and round</i>
------------	---

Description

get_series() is defunct.

Usage

```
get_series()
```

get_series_schedule	<i>Access the playoff schedule for a season and series</i>
---------------------	--

Description

get_series_schedule() is deprecated. Use [series_schedule\(\)](#) instead.

Usage

```
get_series_schedule(season = season_now(), series = "a")
```

Arguments

season	integer in YYYYYYYYY (e.g., 20242025); see seasons() for reference
series	one-letter code (e.g., 'O'); see series() and/or bracket() for reference

Value

data.frame with one row per game

get_shift_charts	<i>Access the shift charts for a game</i>
------------------	---

Description

get_shift_charts() is deprecated. Use [shifts\(\)](#) instead.

Usage

```
get_shift_charts(game = 2023030417)
```

Arguments

game	integer ID (e.g., 2025020275); see games() for reference
------	--

Value

data.frame with one row per shift

get_skaters	<i>Access all the skaters for a range of seasons</i>
-------------	--

Description

get_skaters() is defunct. Use [players\(\)](#) instead.

Usage

get_skaters()

get_skater_leaders	<i>Access the skater statistics leaders for a season, game type, and category</i>
--------------------	---

Description

get_skater_leaders() is deprecated. Use [skater_leaders\(\)](#) instead.

Usage

get_skater_leaders(season = "current", game_type = "", category = "points")

Arguments

- | | |
|-----------|---|
| season | integer in YYYYYYYY (e.g., 20242025); see seasons() for reference |
| game_type | integer in 1:3 (where 1 = pre-season, 2 = regular season, 3 = playoff/post-season) OR character of 'pre', 'regular', or 'playoff'/'post'; see seasons() for reference; most functions will NOT support pre-season |
| category | character of 'assists', 'goals', 'goalsSh', 'goalsPp', 'points', 'penaltyMins', 'toi', 'plusMinus', or 'faceoffLeaders' |

Value

data.frame with one row per player

`get_skater_milestones` *Access the skaters on milestone watch*

Description

`get_skater_milestones()` is deprecated. Use `skater_milestones()` instead.

Usage

```
get_skater_milestones()
```

Value

data.frame with one row per player

`get_skater_statistics` *Access various reports for all the skaters by season or game*

Description

`get_skater_statistics()` is defunct. Use `skater_season_report()` or `skater_game_report()` instead.

Usage

```
get_skater_statistics()
```

`get_spotlight_players` *Access the spotlight players*

Description

`get_spotlight_players()` is deprecated. Use `spotlight_players()` instead.

Usage

```
get_spotlight_players()
```

Value

data.frame with one row per player

get_standings	<i>Access the standings for a date</i>
---------------	--

Description

get_standings() is deprecated. Use [standings\(\)](#) instead.

Usage

```
get_standings(date = "2025-01-01")
```

Arguments

date	character in 'YYYY-MM-DD' (e.g., '2025-01-01'); see seasons() for reference
------	---

Value

data.frame with one row per team

get_standings_information	<i>Access the standings rules by season</i>
---------------------------	---

Description

get_standings_information() is deprecated. Use [standings_rules\(\)](#) instead.

Usage

```
get_standings_information()
```

Value

data.frame with one row per season

get_streams	<i>Access all the streams</i>
-------------	-------------------------------

Description

get_streams() is deprecated. Use [streams\(\)](#) instead.

Usage

```
get_streams()
```

Value

data.frame with one row per stream

get_teams	<i>Access all the teams</i>
-----------	-----------------------------

Description

get_teams() is deprecated. Use [teams\(\)](#) instead.

Usage

```
get_teams()
```

Value

data.frame with one row per team

get_team_prospects	<i>Access the prospects for a team and position</i>
--------------------	---

Description

get_team_prospects() is deprecated. Use [team_prospects\(\)](#) instead.

Usage

```
get_team_prospects(team = "NJD", player_type = "forwards")
```

Arguments

team	three-letter code (e.g., 'COL'); see teams() for reference
player_type	character of 'forwards', 'defensemen', or 'goalies'

Value

data.frame with one row per player

get_team_roster	<i>Access the roster for a team, season, and player type</i>
-----------------	--

Description

get_team_roster() is deprecated. Use [roster\(\)](#) instead.

Usage

```
get_team_roster(team = "NJD", season = "current", player_type = "forwards")
```

Arguments

- team three-letter code (e.g., 'COL'); see [teams\(\)](#) for reference
- season integer in YYYYYYYY (e.g., 20242025); see [seasons\(\)](#) for reference
- player_type character of 'forwards', 'defensemen', or 'goalies'

Value

data.frame with one row per player

get_team_roster_statistics	<i>Access the roster statistics for a team, season, game type, and player type</i>
----------------------------	--

Description

get_team_roster_statistics() is deprecated. Use [roster_statistics\(\)](#) instead.

Usage

```
get_team_roster_statistics(  
  team = "NJD",  
  season = "now",  
  game_type = 2,  
  player_type = "skaters"  
)
```

Arguments

team	three-letter code (e.g., 'COL'); see teams() for reference
season	integer in YYYYYYYY (e.g., 20242025); see seasons() for reference
game_type	integer in 1:3 (where 1 = pre-season, 2 = regular season, 3 = playoff/post-season) OR character of 'pre', 'regular', or playoff/'post'; see seasons() for reference; most functions will NOT support pre-season
player_type	character of 'skaters' or 'goalies'

Value

data.frame with one row per player

get_team_schedule	<i>Access the schedule for a team and season</i>
-------------------	--

Description

get_team_schedule() is deprecated. Use [team_season_schedule\(\)](#) instead.

Usage

```
get_team_schedule(team = "NJD", season = "now")
```

Arguments

team	three-letter code (e.g., 'COL'); see teams() for reference
season	integer in YYYYYYYY (e.g., 20242025); see seasons() for reference

Value

data.frame with one row per game

get_team_scoreboard	<i>Access the team scoreboard as of now</i>
---------------------	---

Description

get_team_scoreboard() is defunct.

Usage

```
get_team_scoreboard()
```

get_team_seasons	<i>Access the season(s) and game type(s) in which a team played</i>
------------------	---

Description

get_team_seasons() is deprecated. Use [team_seasons\(\)](#) instead.

Usage

```
get_team_seasons(team = "NJD")
```

Arguments

team three-letter code (e.g., 'COL'); see [teams\(\)](#) for reference

Value

data.frame with one row per season

get_team_statistics	<i>Access various reports for all the teams by season or game</i>
---------------------	---

Description

get_team_statistics() is defunct. Use [team_season_report\(\)](#) and/or [team_game_report\(\)](#) instead.

Usage

```
get_team_statistics()
```

get_tv_schedule	<i>Access the NHL Network TV schedule for a date</i>
-----------------	--

Description

get_tv_schedule() is deprecated. Use [tv_schedule\(\)](#) instead.

Usage

```
get_tv_schedule(date = "now")
```

Arguments

date character in 'YYYY-MM-DD' (e.g., '2025-01-01'); see [seasons\(\)](#) for reference

Value

data.frame with one row per program

get_venues	<i>Access all the venues</i>
------------	------------------------------

Description

get_venues() is deprecated. Use [venues\(\)](#) instead.

Usage

get_venues()

Value

data.frame with one row per venue

get_wsc_play_by_play	<i>Access the World Showcase (WSC) play-by-play for a game</i>
----------------------	--

Description

get_wsc_play_by_play() is deprecated. Use [wsc_play_by_play\(\)](#) instead.

Usage

get_wsc_play_by_play(game = 2023030417)

Arguments

game integer ID (e.g., 2025020275); see [games\(\)](#) for reference

Value

data.frame with one row per event (play)

glossary	<i>Access the glossary</i>
----------	----------------------------

Description

glossary() scrapes the glossary.

Usage

```
glossary()
```

Value

data.frame with one row per terminology

Examples

```
glossary <- glossary()
```

goalie_edge_five_versus_five	<i>Access the EDGE 5 vs. 5 statistics for a goalie, season, game type, and category</i>
------------------------------	---

Description

goalie_edge_five_versus_five() scrapes the EDGE 5 vs. 5 statistics for a given set of goalie, season, game_type, and category.

Usage

```
goalie_edge_five_versus_five(  
  player = 8476945,  
  season = "now",  
  game_type = "",  
  category = "details"  
)
```

```
goalie_edge_5_vs_5(  
  player = 8476945,  
  season = "now",  
  game_type = "",  
  category = "details"  
)
```

Arguments

player	integer ID (e.g., 8478406)
season	integer in YYYYYYYY (e.g., 20242025); see goalie_edge_seasons() for reference
game_type	integer in 1:3 (where 1 = pre-season, 2 = regular season, 3 = playoff/post-season) OR character of 'pre', 'regular', or 'playoff'/'post'; see goalie_edge_seasons() for reference; most functions will NOT support pre-season
category	character of 'd'/'details' or 'l'/'l10'/'last 10'

Value

list with four items (category = 'details') or data.frame with one row per game (category = 'last 10')

Examples

```
Mackenzie_Blackwood_L10_5_vs_5_regular_20242025 <- goalie_edge_five_versus_five(
  player    = 8478406,
  season    = 20242025,
  game_type = 2,
  category  = 'L'
)
```

goalie_edge_leaders	<i>Access the goalie EDGE statistics leaders for a season and game type</i>
---------------------	---

Description

goalie_edge_leaders() scrapes the goalie EDGE statistics leaders for a given set of season and game_type.

Usage

```
goalie_edge_leaders(season = "now", game_type = "")
```

Arguments

season	integer in YYYYYYYY (e.g., 20242025); see goalie_edge_seasons() for reference
game_type	integer in 1:3 (where 1 = pre-season, 2 = regular season, 3 = playoff/post-season) OR character of 'pre', 'regular', or 'playoff'/'post'; see goalie_edge_seasons() for reference; most functions will NOT support pre-season

Value

list of various items

Examples

```
goalie_EDGE_leaders_regular_20242025 <- goalie_edge_leaders(
  season    = 20242025,
  game_type = 2
)
```

goalie_edge_save_percentage

Access the EDGE save percentage statistics for a goalie, season, game type, and category

Description

goalie_edge_save_percentage() scrapes the EDGE save percentage statistics for a given set of goalie, season, game_type, and category.

Usage

```
goalie_edge_save_percentage(
  player = 8476945,
  season = "now",
  game_type = "",
  category = "details"
)
```

Arguments

player	integer ID (e.g., 8478406)
season	integer in YYYYYYYY (e.g., 20242025); see goalie_edge_seasons() for reference
game_type	integer in 1:3 (where 1 = pre-season, 2 = regular season, 3 = playoff/post-season) OR character of 'pre', 'regular', or 'playoff'/'post'; see goalie_edge_seasons() for reference; most functions will NOT support pre-season
category	character of 'd'/'details' or 'l'/'l10'/'last 10'

Value

list with two items (category = 'details') or data.frame with one row per game (category = 'last 10')

Examples

```
Mackenzie_Blackwood_L10_sP_regular_20242025 <-
  goalie_edge_save_percentage(
    player    = 8478406,
    season    = 20242025,
    game_type = 2,
    category  = 'L'
  )
```

goalie_edge_seasons	<i>Access the season(s) and game type(s) in which there exists goalie EDGE statistics</i>
---------------------	---

Description

goalie_edge_seasons scrapes the season(s) and game type(s) in which the NHL recorded goalie EDGE statistics.

Usage

```
goalie_edge_seasons()
```

Value

data.frame with one row per season

Examples

```
goalie_EDGE_seasons <- goalie_edge_seasons()
```

goalie_edge_shot_location	<i>Access the EDGE shot location statistics for a goalie, season, game type, and category</i>
---------------------------	---

Description

goalie_edge_shot_location() scrapes the EDGE shot location statistics for a given set of goalie, season, game_type, and category.

Usage

```
goalie_edge_shot_location(  
  player = 8476945,  
  season = "now",  
  game_type = "",  
  category = "details"  
)
```

Arguments

player	integer ID (e.g., 8478406)
season	integer in YYYYYYYY (e.g., 20242025); see goalie_edge_seasons() for reference
game_type	integer in 1:3 (where 1 = pre-season, 2 = regular season, 3 = playoff/post-season) OR character of 'pre', 'regular', or 'playoff'/'post'; see goalie_edge_seasons() for reference; most functions will NOT support pre-season
category	character of 'd'/'details' or 't'/'totals'

Value

data.frame with one row per shot location

Examples

```
Mackenzie_Blackwood_shot_location_totals_regular_20242025 <-
  goalie_edge_shot_location(
    player    = 8478406,
    season    = 20242025,
    game_type = 2,
    category  = 'T'
  )
```

goalie_edge_summary	<i>Access the EDGE summary for a goalie, season, and game type</i>
---------------------	--

Description

goalie_edge_summary() scrapes the EDGE summary for a given set of goalie, season, and game_type.

Usage

```
goalie_edge_summary(player = 8476945, season = "now", game_type = "")
```

Arguments

player	integer ID (e.g., 8478406)
season	integer in YYYYYYYY (e.g., 20242025); see goalie_edge_seasons() for reference
game_type	integer in 1:3 (where 1 = pre-season, 2 = regular season, 3 = playoff/post-season) OR character of 'pre', 'regular', or 'playoff'/'post'; see goalie_edge_seasons() for reference; most functions will NOT support pre-season

Value

list of various items

Examples

```
Mackenzie_Blackwood_EDGE_summary_regular_20242025 <- goalie_edge_summary(
  player    = 8478406,
  season    = 20242025,
  game_type = 2
)
```

goalie_game_report	<i>Access various reports for a season, game type, and category for all the goalies by game</i>
--------------------	---

Description

goalie_game_report() scrapes various reports for a given set of season, game_type, and category for all the goalies by game.

Usage

```
goalie_game_report(
  season = season_now(),
  game_type = game_type_now(),
  category = "summary"
)
```

Arguments

season	integer in YYYYYYYY (e.g., 20242025); see seasons() for reference
game_type	integer in 1:3 (where 1 = pre-season, 2 = regular season, 3 = playoff/post-season) OR character of 'pre', 'regular', or 'playoff'/'post'; see seasons() for reference; most functions will NOT support pre-season
category	character (e.g., 'advanced'); see goalie_report_configurations() for reference

Value

data.frame with one row per game per goalie

Examples

```
# May take >5s, so skip.
advanced_goalie_game_report_playoffs_20212022 <-
  goalie_game_report(
    season    = 20212022,
    game_type = 3,
    category  = 'advanced'
  )
```

goalie_game_scoring	<i>Access the scoring statistics for all the goalies by game</i>
---------------------	--

Description

goalie_game_scoring() scrapes the scoring statistics for all the goalies by game.

Usage

```
goalie_game_scoring()
```

Value

data.frame with one row per player per game

Examples

```
goalie_game_scoring <- goalie_game_scoring()
```

goalie_game_statistics	<i>Access the statistics for all the goalies by game</i>
------------------------	--

Description

goalie_game_statistics() scrapes the statistics for all the goalies by game.

Usage

```
goalie_game_statistics()
```

```
goalie_game_stats()
```

Value

data.frame with one row per goalie per game

Examples

```
goalie_game_stats <- goalie_game_statistics()
```

goalie_leaders	<i>Access the goalie statistics leaders for a season, game type, and category</i>
----------------	---

Description

goalie_leaders() scrapes the goalie statistics leaders for a given set of season, game_type, and category.

Usage

```
goalie_leaders(season = "current", game_type = "", category = "wins")
```

Arguments

season	integer in YYYYYYYY (e.g., 20242025); see seasons() for reference
game_type	integer in 1:3 (where 1 = pre-season, 2 = regular season, 3 = playoff/post-season) OR character of 'pre', 'regular', or playoff'/'post'; see seasons() for reference; most functions will NOT support pre-season
category	character of 'w'/'wins', 's'/'shutouts', 's%'/'sP'/'save %'/'save percentage', or 'gaa'/'goals against average'

Value

data.frame with one row per player

Examples

```
GAA_leaders_regular_20242025 <- goalie_leaders(
  season    = 20242025,
  game_type = 2,
  category  = 'GAA'
)
```

goalie_milestones	<i>Access the goalies on milestone watch</i>
-------------------	--

Description

goalie_milestones() scrapes the goalies on milestone watch.

Usage

```
goalie_milestones()
```

Value

data.frame with one row per player

Examples

```
goalie_milestones <- goalie_milestones()
```

```
goalie_regular_statistics
```

Access the career regular season statistics for all the goalies

Description

goalie_regular_statistics() scrapes the career regular season statistics for all the goalies.

Usage

```
goalie_regular_statistics()
```

```
goalie_regular_stats()
```

Value

data.frame with one row per goalie

Examples

```
goalie_career_regular_statistics <- goalie_regular_statistics()
```

```
goalie_report_configurations
```

Access the configurations for goalie reports

Description

goalie_report_configurations() scrapes the configurations for [goalie_season_report\(\)](#) and [goalie_game_report\(\)](#).

Usage

```
goalie_report_configurations()
```

```
goalie_report_configs()
```

Value

list with various items

Examples

```
goalie_report_configs <- goalie_report_configurations()
```

goalie_scoring	<i>Access the career scoring statistics for all the goalies</i>
----------------	---

Description

goalie_scoring() scrapes the career scoring statistics for all the goalies.

Usage

```
goalie_scoring()
```

Value

data.frame with one row per player

Examples

```
goalie_scoring <- goalie_scoring()
```

goalie_season_report	<i>Access various reports for a season, game type, and category for all the goalies by season</i>
----------------------	---

Description

goalie_season_report() scrapes various reports for a given set of season, game_type, and category for all the goalies by season.

Usage

```
goalie_season_report(  
  season = season_now(),  
  game_type = game_type_now(),  
  category = "summary"  
)
```


Arguments

season	integer in YYYYYYYY (e.g., 20242025); see seasons() for reference
game_type	integer in 1:3 (where 1 = pre-season, 2 = regular season, 3 = playoff/post-season) OR character of 'pre', 'regular', or playoff'/post'; see seasons() for reference; most functions will NOT support pre-season
category	character (e.g., 'advanced'); see goalie_report_configurations() for reference

Value

data.frame with one row per player

Examples

```
# May take >5s, so skip.
advanced_goalie_season_report_playoffs_20212022 <-
  goalie_season_report(
    season      = 20212022,
    game_type   = 3,
    category    = 'advanced'
  )
```

goalie_season_statistics

Access the statistics for all the goalies by season, game type, and team.

Description

goalie_season_statistics() scrapes the statistics for all the goalies by season, game type, and team.

Usage

```
goalie_season_statistics()
```

```
goalie_season_stats()
```

Value

data.frame with one row per player per season per game type, separated by team if applicable

Examples

```
goalie_season_stats <- goalie_season_statistics()
```

`goalie_series_statistics`*Access the playoff statistics for all the goalies by series*

Description

`goalie_series_statistics()` scrapes the playoff statistics for all the goalies by series.

Usage

```
goalie_series_statistics()
```

```
goalie_series_stats()
```

Value

data.frame with one row per player per series

Examples

```
goalie_series_stats <- goalie_series_statistics()
```

`goalie_statistics`*Access the career statistics for all the goalies*

Description

`goalie_statistics()` scrapes the career statistics for all the goalies.

Usage

```
goalie_statistics()
```

```
goalie_stats()
```

Value

data.frame with one row per player

Examples

```
goalie_stats <- goalie_statistics()
```

 ig_game_cumulative_expected_goals

Save an Instagram (IG) share-able cumulative expected goals (xG) time-series plot for a game

Description

ig_game_cumulative_expected_goals() saves an IG share-able cumulative xG time-series plot for a given game as a PNG.

Usage

```
ig_game_cumulative_expected_goals(game = 2023030417, model = 1, save = TRUE)
```

```
ig_game_cum_xG(game = 2023030417, model = 1)
```

Arguments

game	integer ID (e.g., 2025020275); see games() for reference
model	integer in 1:3 indicating which expected goals model to use (e.g., 1); see calculate_expected_goals_v1 , calculate_expected_goals_v2() , and/or calculate_expected_goals_v3() for reference
save	logical only FALSE for tests

Value

NULL

Examples

```
# May take >5s, so skip.
ig_game_cumulative_expected_goals(
  game = 2023030417,
  model = 1,
  save = FALSE
)
```

 ig_game_shot_locations

Save an Instagram (IG) share-able shot-location plot for a game

Description

ig_game_shot_locations() saves an IG share-able shot location plot for a given game.

Usage

```
ig_game_shot_locations(  
  game = 2023030417,  
  team = "home",  
  model = 1,  
  save = TRUE  
)  
  
ig_game_shot_locs(game = 2023030417, team = "home", model = 1)
```

Arguments

- game integer ID (e.g., 2025020275); see [games\(\)](#) for reference
- team character of 'h'/'home' or 'a'/'away'
- model integer in 1:3 indicating which expected goals model to use (e.g., 1); see [calculate_expected_goals_v1\(\)](#), [calculate_expected_goals_v2\(\)](#), and/or [calculate_expected_goals_v3\(\)](#) for reference
- save logical only FALSE for tests

Value

NULL

Examples

```
# May take >5s, so skip.  
ig_game_shot_locations(  
  game = 2023030417,  
  model = 1,  
  team = 'H',  
  save = FALSE  
)
```

location	<i>Access the location for a zip code</i>
----------	---

Description

location() scrapes the location for a given zip code.

Usage

```
location(zip = 10001)
```

Arguments

- zip integer (e.g., 48304)

Value

data.frame with one row per team

Examples

```
Cranbrook_Schools <- location(48304)
```

lottery_odds	<i>Access the draft lottery odds</i>
--------------	--------------------------------------

Description

lottery_odds() scrapes the draft lottery odds.

Usage

```
lottery_odds()
```

Value

data.frame with one row per draft lottery

Examples

```
lottery_odds <- lottery_odds()
```

normalize_coordinates	<i>Normalize the x and y coordinates for all the events (plays) in a play-by-play</i>
-----------------------	---

Description

normalize_coordinates() normalizes the x and y coordinates for all the events (plays) in a play-by-play such that they all attack towards +x.

Usage

```
normalize_coordinates(play_by_play)
```

Arguments

play_by_play	data.frame of play-by-play(s); see gc_play_by_play() and/or wsc_play_by_play() for reference; must be untouched by non-nhlscraper functions; saves time if flag_is_home() has already been called
--------------	---

Value

data.frame with one row per event (play) and added columns xCoordNorm and yCoordNorm

Examples

```
# May take >5s, so skip.  
  
test          <- gc_play_by_play()  
test_is_home_flagged <- flag_is_home(test)  
test_coords_normalized <- normalize_coordinates(test_is_home_flagged)
```

officials	<i>Access all the officials</i>
-----------	---------------------------------

Description

officials() scrapes all the officials.

Usage

```
officials()
```

Value

data.frame with one row per official

Examples

```
all_officials <- officials()
```

penalty_shots	<i>Access all the penalty shots</i>
---------------	-------------------------------------

Description

penalty_shots() scrapes all the penalty shots.

Usage

```
penalty_shots()
```

```
ps()
```

Value

data.frame with one row per penalty shot

Examples

```
all_ps <- penalty_shots()
```

ping

Ping

Description

ping() is defunct.

Usage

```
ping()
```

players

Access all the players

Description

players() scrapes all the players.

Usage

```
players()
```

Value

data.frame with one row per player

Examples

```
# May take >5s, so skip.  
all_players <- players()
```

player_game_log	<i>Access the game log for a player, season, and game type</i>
-----------------	--

Description

player_game_log() scrapes the game log for a given set of player, season, and game_type.

Usage

```
player_game_log(player = 8478402, season = "now", game_type = "")
```

Arguments

player	integer ID (e.g., 8480039); see players() for reference
season	integer in YYYYYYYY (e.g., 20242025); see seasons() for reference
game_type	integer in 1:3 (where 1 = pre-season, 2 = regular season, 3 = playoff/post-season) OR character of 'pre', 'regular', or playoff'/'post'; see seasons() for reference; most functions will NOT support pre-season

Value

data.frame with one row per game

Examples

```
Martin_Necas_game_log_regular_20242025 <- player_game_log(
  player = 8480039,
  season = 20242025,
  game_type = 2
)
```

player_seasons	<i>Access the season(s) and game type(s) in which a player played</i>
----------------	---

Description

player_seasons() scrapes the season(s) and game type(s) in which a player played in the NHL.

Usage

```
player_seasons(player = 8478402)
```

Arguments

player	integer ID (e.g., 8480039); see players() for reference
--------	---

Value

data.frame with one row per season

Examples

```
Martin_Necas_seasons <- player_seasons(player = 8480039)
```

player_summary	<i>Access the summary for a player</i>
----------------	--

Description

player_summary() scrapes the summary for a given player.

Usage

```
player_summary(player = 8478402)
```

Arguments

player integer ID (e.g., 8480039); see [players\(\)](#) for reference

Value

list with various items

Examples

```
Martin_Necas_summary <- player_summary(player = 8480039)
```

playoff_season_statistics	<i>Access the playoff statistics by season</i>
---------------------------	--

Description

playoff_season_statistics() scrapes the playoff statistics by season.

Usage

```
playoff_season_statistics()
```

```
playoff_season_stats()
```

Value

data.frame with one row per season

Examples

```
playoff_season_stats <- playoff_season_statistics()
```

replay	<i>Access the replay for an event</i>
--------	---------------------------------------

Description

replay() scrapes the replay for a given event.

Usage

```
replay(game = 2023030417, event = 866)
```

Arguments

- game integer ID (e.g., 2025020262); see [games\(\)](#) for reference
- event integer ID (e.g., 751); see [gc_play_by_play\(\)](#) and/or [wsc_play_by_play\(\)](#) for reference; must be a 'goal' event

Value

data.frame with one row per decisecond

Examples

```
Gabriel_Landeskog_first_regular_goal_back_replay <- replay(  
  game = 2025020262,  
  event = 751  
)
```

roster	<i>Access the roster for a team, season, and position</i>
--------	---

Description

roster() scrapes the roster for a given set of team, season, and position.

Usage

```
roster(team = 1, season = "current", position = "forwards")
```

Arguments

team	integer ID (e.g., 21), character full name (e.g., 'Colorado Avalanche'), OR three-letter code (e.g., 'COL'); see teams() for reference; ID is preferable as there now exists duplicate three-letter codes (i.e., 'UTA' for 'Utah Hockey Club' and 'Utah Mammoth')
season	integer in YYYYYYYY (e.g., 20242025); see seasons() for reference
position	character of 'f'/'forwards', 'd'/'defensemen', or 'g'/'goalies'

Value

data.frame with one row per player

Examples

```
COL_defensemen_20242025 <- roster(  
  team    = 21,  
  season  = 20242025,  
  position = 'D'  
)
```

roster_statistics	<i>Access the roster statistics for a team, season, game type, and position</i>
-------------------	---

Description

roster_statistics() scrapes the roster statistics for a given set of team, season, game_type, and position.

Usage

```
roster_statistics(  
  team = 1,  
  season = "now",  
  game_type = "",  
  position = "skaters"  
)  
  
roster_stats(team = 1, season = "now", game_type = "", position = "skaters")
```

Arguments

team	integer ID (e.g., 21), character full name (e.g., 'Colorado Avalanche'), OR three-letter code (e.g., 'COL'); see teams() for reference; ID is preferable as there now exists duplicate three-letter codes (i.e., 'UTA' for 'Utah Hockey Club' and 'Utah Mammoth')
season	integer in YYYYYYYY (e.g., 20242025); see seasons() for reference

game_type	integer in 1:3 (where 1 = pre-season, 2 = regular season, 3 = playoff/post-season) OR character of 'pre', 'regular', or playoff'/'post'; see seasons() for reference; most functions will NOT support pre-season
position	character of 's'/'skaters' or 'g'/'goalies'

Value

data.frame with one row per player

Examples

```
COL_goalies_statistics_regular_20242025 <- roster_statistics(  
  team      = 21,  
  season    = 20242025,  
  game_type = 2,  
  position  = 'G'  
)
```

schedule	<i>Access the schedule for a date</i>
----------	---------------------------------------

Description

schedule() scrapes the schedule for a given date.

Usage

```
schedule(date = Sys.Date())
```

Arguments

date	character in 'YYYY-MM-DD' (e.g., '2025-01-01'); see seasons() for reference
------	---

Value

data.frame with one row per game

Examples

```
schedule_Halloween_2025 <- schedule(date = '2025-10-31')
```

scores	<i>Access the scores for a date</i>
--------	-------------------------------------

Description

scores() scrapes the scores for a given date.

Usage

```
scores(date = "now")
```

Arguments

date	character in 'YYYY-MM-DD' (e.g., '2025-01-01'); see seasons() for reference
------	---

Value

data.frame with one row per game

Examples

```
scores_Halloween_2025 <- scores(date = '2025-10-31')
```

seasons	<i>Access all the seasons</i>
---------	-------------------------------

Description

seasons() scrapes all the seasons.

Usage

```
seasons()
```

Value

data.frame with one row per season

Examples

```
all_seasons <- seasons()
```

season_now	<i>Access the season as of now</i>
------------	------------------------------------

Description

season_now scrapes the current season.

Usage

```
season_now()
```

Value

integer in YYYYYYYYY (e.g., 20242025)

Examples

```
season_now <- season_now()
```

series	<i>Access all the playoff series by game</i>
--------	--

Description

series() scrapes all the playoff series by game.

Usage

```
series()
```

Value

data.frame with one row per game per series

Examples

```
# May take >5s, so skip.  
all_series <- series()
```

series_schedule	<i>Access the playoff schedule for a season and series</i>
-----------------	--

Description

series_schedule() scrapes the playoff schedule for a given set of season and series.

Usage

```
series_schedule(season = season_now(), series = "a")
```

Arguments

season	integer in YYYYYYYY (e.g., 20242025); see seasons() for reference
series	one-letter code (e.g., 'O'); see series() and/or bracket() for reference

Value

data.frame with one row per game

Examples

```
SCF_schedule_20212022 <- series_schedule(
  season = 20212022,
  series = 'O'
)
```

shifts	<i>Access the shift charts for a game</i>
--------	---

Description

shifts() scrapes the shift charts for a given game.

Usage

```
shifts(game = 2023030417)
```

Arguments

game	integer ID (e.g., 2025020275); see games() for reference
------	--

Value

data.frame with one row per shift

Examples

```
shifts_Martin_Necas_legacy_game <- shifts(game = 2025020275)
```

skater_edge_leaders	<i>Access the skater EDGE statistics leaders for a season and game type</i>
---------------------	---

Description

skater_edge_leaders() scrapes the skater EDGE statistics leaders for a given set of season and game_type.

Usage

```
skater_edge_leaders(season = "now", game_type = "")
```

Arguments

season	integer in YYYYYYYYY (e.g., 20242025); see skater_edge_seasons() for reference
game_type	integer in 1:3 (where 1 = pre-season, 2 = regular season, 3 = playoff/post-season) OR character of 'pre', 'regular', or 'playoff'/'post'; see skater_edge_seasons() for reference; most functions will NOT support pre-season

Value

list of various items

Examples

```
skater_EDGE_leaders_regular_20242025 <- skater_edge_leaders(
  season    = 20242025,
  game_type = 2
)
```

skater_edge_seasons	<i>Access the season(s) and game type(s) in which there exists skater EDGE statistics</i>
---------------------	---

Description

skater_edge_seasons scrapes the season(s) and game type(s) in which the NHL recorded skater EDGE statistics.

Usage

```
skater_edge_seasons()
```

Value

data.frame with one row per season

Examples

```
skater_EDGE_seasons <- skater_edge_seasons()
```

```
skater_edge_shot_location
```

Access the EDGE shot location statistics for a skater, season, game type, and category

Description

skater_edge_shot_location() scrapes the EDGE shot location statistics for a given set of skater, season, game_type, and category.

Usage

```
skater_edge_shot_location(
  player = 8478402,
  season = "now",
  game_type = "",
  category = "details"
)
```

Arguments

player	integer ID (e.g., 8480039); see players() for reference
season	integer in YYYYYYYY (e.g., 20242025); see skater_edge_seasons() for reference
game_type	integer in 1:3 (where 1 = pre-season, 2 = regular season, 3 = playoff/post-season) OR character of 'pre', 'regular', or 'playoff'/'post'; see skater_edge_seasons() for reference; most functions will NOT support pre-season
category	character of 'd'/details' or 't'/totals'

Value

data.frame with one row per shot location

Examples

```
Martin_Necas_shot_location_totals_regular_20242025 <-
  skater_edge_shot_location(
    player = 8480039,
    season = 20242025,
    game_type = 2,
    category = 'T'
  )
```

 skater_edge_shot_speed

Access the EDGE shot speed statistics for a skater, season, game type, and category

Description

skater_edge_shot_speed() scrapes the EDGE shot speed statistics for a given set of skater, season, game_type, and category.

Usage

```
skater_edge_shot_speed(
  player = 8478402,
  season = "now",
  game_type = "",
  category = "details"
)
```

Arguments

player	integer ID (e.g., 8480039); see players() for reference
season	integer in YYYYYYYY (e.g., 20242025); see skater_edge_seasons() for reference
game_type	integer in 1:3 (where 1 = pre-season, 2 = regular season, 3 = playoff/post-season) OR character of 'pre', 'regular', or 'playoff'/'post'; see skater_edge_seasons() for reference; most functions will NOT support pre-season
category	character of 'd'/'details' or 'h'/'hardest'

Value

list with six items (category = 'details') or data.frame with one row per shot (category = 'hardest')

Examples

```
Martin_Necas_hardest_shots_regular_20242025 <- skater_edge_shot_speed(
  player   = 8480039,
  season   = 20242025,
  game_type = 2,
  category  = 'H'
)
```

 skater_edge_skating_distance

Access the EDGE skating distance statistics for a skater, season, game type, and category

Description

skater_edge_skating_distance() scrapes the EDGE skating distance statistics for a given set of skater, season, game_type, and category.

Usage

```
skater_edge_skating_distance(
  player = 8478402,
  season = "now",
  game_type = "",
  category = "details"
)
```

Arguments

player	integer ID (e.g., 8480039); see players() for reference
season	integer in YYYYYYYY (e.g., 20242025); see skater_edge_seasons() for reference
game_type	integer in 1:3 (where 1 = pre-season, 2 = regular season, 3 = playoff/post-season) OR character of 'pre', 'regular', or 'playoff'/'post'; see skater_edge_seasons() for reference; most functions will NOT support pre-season
category	character of 'd'/'details' or 'l'/'l10'/'last 10'

Value

data.frame with one row per strength state (category = 'details') or game (category = 'last 10')

Examples

```
Martin_Necas_L10_skating_distance_regular_20242025 <-
  skater_edge_skating_distance(
    player = 8480039,
    season = 20242025,
    game_type = 2,
    category = 'L'
  )
```

 skater_edge_skating_speed

Access the EDGE skating speed statistics for a skater, season, game type, and category

Description

skater_edge_skating_speed() scrapes the EDGE skating speed statistics for a given set of skater, season, game_type, and category.

Usage

```
skater_edge_skating_speed(
  player = 8478402,
  season = "now",
  game_type = "",
  category = "details"
)
```

Arguments

player	integer ID (e.g., 8480039); see players() for reference
season	integer in YYYYYYYY (e.g., 20242025); see skater_edge_seasons() for reference
game_type	integer in 1:3 (where 1 = pre-season, 2 = regular season, 3 = playoff/post-season) OR character of 'pre', 'regular', or 'playoff'/'post'; see skater_edge_seasons() for reference; most functions will NOT support pre-season
category	character of 'd'/'details' or 't'/'top'/'top speeds'

Value

list with four items (category = 'details') or data.frame with one row per burst (category = 'top speeds')

Examples

```
Martin_Necas_top_speeds_regular_20242025 <- skater_edge_skating_speed(
  player    = 8480039,
  season    = 20242025,
  game_type = 2,
  category  = 'T'
)
```

skater_edge_summary *Access the EDGE summary for a skater, season, and game type*

Description

skater_edge_summary() scrapes the EDGE summary for a given set of skater, season, and game_type.

Usage

```
skater_edge_summary(player = 8478402, season = "now", game_type = "")
```

Arguments

player	integer ID (e.g., 8480039); see players() for reference
season	integer in YYYYYYYY (e.g., 20242025); see skater_edge_seasons() for reference
game_type	integer in 1:3 (where 1 = pre-season, 2 = regular season, 3 = playoff/post-season) OR character of 'pre', 'regular', or 'playoff'/'post'; see skater_edge_seasons() for reference; most functions will NOT support pre-season

Value

list of various items

Examples

```
Martin_Necas_EDGE_summary_regular_20242025 <- skater_edge_summary(
  player   = 8480039,
  season   = 20242025,
  game_type = 2
)
```

skater_edge_zone_time *Access the EDGE zone time statistics for a skater, season, game type, and category*

Description

skater_edge_zone_time() scrapes the EDGE zone time statistics for a given set of skater, season, game_type, and category.

Usage

```
skater_edge_zone_time(
  player = 8478402,
  season = "now",
  game_type = "",
  category = "details"
)
```

Arguments

player	integer ID (e.g., 8480039); see players() for reference
season	integer in YYYYYYYYY (e.g., 20242025); see skater_edge_seasons() for reference
game_type	integer in 1:3 (where 1 = pre-season, 2 = regular season, 3 = playoff/post-season) OR character of 'pre', 'regular', or 'playoff'/'post'; see skater_edge_seasons() for reference; most functions will NOT support pre-season
category	character of 'd'/'details' or 's'/'starts'

Value

data.frame with one row per strength state (category = 'details') or list with six items (category = 'starts')

Examples

```
Martin_Necas_starts_regular_20242025 <- skater_edge_zone_time(
  player = 8480039,
  season = 20242025,
  game_type = 2,
  category = 'S'
)
```

skater_game_report	<i>Access various reports for a season, game type, and category for all the skaters by game</i>
--------------------	---

Description

skater_game_report() scrapes various reports for a given set of season, game_type, and category for all the skaters by game.

Usage

```
skater_game_report(
  season = season_now(),
  game_type = game_type_now(),
  category = "summary"
)
```

Arguments

season	integer in YYYYYYYY (e.g., 20242025); see seasons() for reference
game_type	integer in 1:3 (where 1 = pre-season, 2 = regular season, 3 = playoff/post-season) OR character of 'pre', 'regular', or playoff'/'post'; see seasons() for reference; most functions will NOT support pre-season
category	character (e.g., 'puckPossessions'); see skater_report_configurations() for reference

Value

data.frame with one row per game per player

Examples

```
# May take >5s, so skip.
possession_skater_game_report_playoff_20212022 <-
  skater_game_report(
    season      = 20212022,
    game_type   = 3,
    category    = 'puckPossessions'
  )
```

skater_leaders	<i>Access the skater statistics leaders for a season, game type, and category</i>
----------------	---

Description

skater_leaders() scrapes the skater statistics leaders for a given set of season, game_type, and category.

Usage

```
skater_leaders(season = "current", game_type = "", category = "points")
```

Arguments

season	integer in YYYYYYYY (e.g., 20242025); see seasons() for reference
game_type	integer in 1:3 (where 1 = pre-season, 2 = regular season, 3 = playoff/post-season) OR character of 'pre', 'regular', or playoff'/'post'; see seasons() for reference; most functions will NOT support pre-season
category	string of 'a'/'assists', 'g'/'goals', 'shg'/'shorthanded goals', 'ppg'/'powerplay goals', 'p'/'points', 'pim'/'penalty minutes'/'penalty infraction minutes', 'toi'/'time on ice', 'pm'/'plus minus', or 'f'/'faceoffs'

Value

data.frame with one row per player

Examples

```
TOI_leaders_regular_20242025 <- skater_leaders(
  season    = 20242025,
  game_type = 2,
  category  = 'TOI'
)
```

skater_milestones	<i>Access the skaters on milestone watch</i>
-------------------	--

Description

skater_milestones() scrapes the skaters on milestone watch.

Usage

```
skater_milestones()
```

Value

data.frame with one row per player

Examples

```
skater_milestones <- skater_milestones()
```

skater_playoff_statistics	<i>Access the career playoff statistics for all the skaters</i>
---------------------------	---

Description

skater_playoff_statistics() scrapes the career playoff statistics for all the skaters.

Usage

```
skater_playoff_statistics()
```

```
skater_playoff_stats()
```

Value

data.frame with one row per player

Examples

```
skater_playoff_stats <- skater_playoff_statistics()
```

```
skater_regular_statistics
```

Access the career regular season statistics for all the skaters

Description

skater_regular_statistics() scrapes the career regular season statistics for all the skaters.

Usage

```
skater_regular_statistics()
```

```
skater_regular_stats()
```

Value

data.frame with one row per player

Examples

```
skater_regular_stats <- skater_regular_statistics()
```

```
skater_report_configurations
```

Access the configurations for skater reports

Description

skater_report_configurations() scrapes the configurations for [skater_season_report\(\)](#) and [skater_game_report\(\)](#).

Usage

```
skater_report_configurations()
```

```
skater_report_configs()
```

Value

list with various items

Examples

```
skater_report_configs <- skater_report_configurations()
```

skater_season_report	<i>Access various reports for a season, game type, and category for all the skaters by season</i>
----------------------	---

Description

skater_season_report() scrapes various reports for a given set of season, game_type, and category for all the skaters by season.

Usage

```
skater_season_report(
  season = season_now(),
  game_type = game_type_now(),
  category = "summary"
)
```

Arguments

season	integer in YYYYYYYYY (e.g., 20242025); see seasons() for reference
game_type	integer in 1:3 (where 1 = pre-season, 2 = regular season, 3 = playoff/post-season) OR character of 'pre', 'regular', or 'playoff'/'post'; see seasons() for reference; most functions will NOT support pre-season
category	character (e.g., 'puckPossessions'); see skater_report_configurations() for reference

Value

data.frame with one row per player

Examples

```
# May take >5s, so skip.
possession_skater_season_report_playoff_20212022 <-
  skater_season_report(
    season = 20212022,
    game_type = 3,
    category = 'puckPossessions'
  )
```

`skater_season_statistics`*Access the statistics for all the skaters by season, game type, and team*

Description

`skater_season_statistics()` scrapes the statistics for all the skaters by season, game type, and team.

Usage

```
skater_season_statistics()
```

```
skater_season_stats()
```

Value

data.frame with one row per player per season per game type, separated by team if applicable

Examples

```
# May take >5s, so skip.  
skater_season_stats <- skater_season_statistics()
```

`skater_series_statistics`*Access the playoff statistics for all the skaters by series*

Description

`skater_series_statistics()` scrapes the playoff statistics for all the skaters by series.

Usage

```
skater_series_statistics()
```

```
skater_series_stats()
```

Value

data.frame with one row per player per series

Examples

```
# May take >5s, so skip.  
skater_series_stats <- skater_series_statistics()
```

skater_statistics	<i>Access the career statistics for all the skaters</i>
-------------------	---

Description

skater_statistics() scrapes the career statistics for all the skaters.

Usage

```
skater_statistics()
```

```
skater_stats()
```

Value

data.frame with one row per player

Examples

```
skater_stats <- skater_statistics()
```

spotlight_players	<i>Access the spotlight players</i>
-------------------	-------------------------------------

Description

spotlight_players() scrapes the spotlight players.

Usage

```
spotlight_players()
```

Value

data.frame with one row per player

Examples

```
spotlight_players <- spotlight_players()
```

standings	<i>Access the standings for a date</i>
-----------	--

Description

`standings()` scrapes the standings for a given date.

Usage

```
standings(date = "now")
```

Arguments

date	character in 'YYYY-MM-DD' (e.g., '2025-01-01'); see seasons() for reference
------	---

Value

data.frame with one row per team

Examples

```
standings_Halloween_2025 <- standings(date = '2025-10-31')
```

standings_rules	<i>Access the standings rules by season</i>
-----------------	---

Description

`standings_rules()` scrapes the standings rules by season.

Usage

```
standings_rules()
```

Value

data.frame with one row per season

Examples

```
standings_rules <- standings_rules()
```

streams	<i>Access all the streams</i>
---------	-------------------------------

Description

streams() scrapes all the streams.

Usage

```
streams()
```

Value

data.frame with one row per stream

Examples

```
all_streams <- streams()
```

strip_game_id	<i>Strip the game ID into the season ID, game type ID, and game number for all the events (plays) in a play-by-play</i>
---------------	---

Description

strip_game_id() strips the game ID into the season ID, game type ID, and game number for all the events (plays) in a play-by-play.

Usage

```
strip_game_id(play_by_play)
```

Arguments

play_by_play data.frame of play-by-play(s); see [gc_play_by_play\(\)](#) and/or [wsc_play_by_play\(\)](#) for reference; must be untouched by non-nhlscraper functions

Value

data.frame with one row per event (play) and added columns: seasonId, gameId, and gameNumber

Examples

```
# May take >5s, so skip.

test <- gc_play_by_play()
test_game_id_stripped <- strip_game_id(test)
```

strip_situation_code	<i>Strip the situation code into goalie and skater counts, man differential, and strength state for all the events (plays) in a play-by-play by perspective</i>
----------------------	---

Description

strip_situation_code() strip the situation code into goalie and skater counts, man differential, and strength state for all the events (plays) in a play-by-play by perspective.

Usage

```
strip_situation_code(play_by_play)
```

Arguments

play_by_play data.frame of play-by-play(s); see [gc_play_by_play\(\)](#) and/or [wsc_play_by_play\(\)](#) for reference; must be untouched by non-nhlscraper functions; saves time if [flag_is_home\(\)](#) has already been called

Value

data.frame with one row per event (play) and added columns: homeIsEmptyNet, awayIsEmptyNet, homeSkaterCount, awaySkaterCount, isEmptyNetFor, isEmptyNetAgainst, skaterCountFor, skaterCountAgainst, manDifferential, and strengthState

Examples

```
# May take >5s, so skip.

test          <- gc_play_by_play()
test_is_home_flagged <- flag_is_home(test)
test_situation_code_stripped <- strip_situation_code(test_is_home_flagged)
```

strip_time_period	<i>Strip the timestamp and period number into the time elapsed in the period and game for all the events (plays) in a play-by-play</i>
-------------------	--

Description

strip_time_period() strip the timestamp and period number into the time elapsed in the period and game for all the events (plays) in a play-by-play.

Usage

```
strip_time_period(play_by_play)
```

Arguments

play_by_play data.frame of play-by-play(s); see [gc_play_by_play\(\)](#) and/or [wsc_play_by_play\(\)](#) for reference; must be untouched by non-nhlscraper functions; saves time if [strip_game_id\(\)](#) has already been called

Value

data.frame with one row per event (play) and added columns secondsElapsedInPeriod and secondsElapsedInGame

Examples

```
# May take >5s, so skip.  
  
test                              <- gc_play_by_play()  
test_game_id_stripped          <- strip_game_id(test)  
test_time_period_stripped <- strip_time_period(test_game_id_stripped)
```

teams	<i>Access all the teams</i>
-------	-----------------------------

Description

teams() scrapes all the teams.

Usage

```
teams()
```

Value

data.frame with one row per team

Examples

```
all_teams <- teams()
```

team_edge_leaders	<i>Access the team EDGE statistics leaders for a season and game type</i>
-------------------	---

Description

team_edge_leaders() scrapes the team EDGE statistics leaders for a given set of season and game_type.

Usage

```
team_edge_leaders(season = "now", game_type = "")
```

Arguments

- season integer in YYYYYYYY (e.g., 20242025); see [team_edge_seasons\(\)](#) for reference
- game_type integer in 1:3 (where 1 = pre-season, 2 = regular season, 3 = playoff/post-season) OR character of 'pre', 'regular', or 'playoff'/'post'; see [team_edge_seasons\(\)](#) for reference; most functions will NOT support pre-season

Value

list of various items

Examples

```
team_EDGE_leaders_regular_20242025 <- team_edge_leaders(  
  season = 20242025,  
  game_type = 2  
)
```

team_edge_seasons	<i>Access the season(s) and game type(s) in which there exists team EDGE statistics</i>
-------------------	---

Description

team_edge_seasons scrapes the season(s) and game type(s) in which the NHL recorded team EDGE statistics.

Usage

```
team_edge_seasons()
```

Value

data.frame with one row per season

Examples

```
team_EDGE_seasons <- team_edge_seasons()
```

```
team_edge_shot_location
```

Access the EDGE shot location statistics for a team, season, game type, and category

Description

`team_edge_shot_location()` scrapes the EDGE shot location statistics for a given set of team, season, game_type, and category.

Usage

```
team_edge_shot_location(
  team = 1,
  season = "now",
  game_type = "",
  category = "details"
)
```

Arguments

team	integer ID (e.g., 21), character full name (e.g., 'Colorado Avalanche'), OR three-letter code (e.g., 'COL'); see teams() for reference; ID is preferable as there now exists duplicate three-letter codes (i.e., 'UTA' for 'Utah Hockey Club' and 'Utah Mammoth')
season	integer in YYYYYYYY (e.g., 20242025); see team_edge_seasons() for reference
game_type	integer in 1:3 (where 1 = pre-season, 2 = regular season, 3 = playoff/post-season) OR character of 'pre', 'regular', or 'playoff'/'post'; see team_edge_seasons() for reference; most functions will NOT support pre-season
category	character of 'd'/details' or 't'/totals'

Value

data.frame with one row per location (category = 'details') or combination of strength state and position (category = 'totals')

Examples

```
COL_shot_location_totals_regular_20242025 <- team_edge_shot_location(
  team      = 21,
  season    = 20242025,
  game_type = 2,
  category  = 'T'
)
```

team_edge_shot_speed	<i>Access the EDGE shot speed statistics for a team, season, game type, and category</i>
----------------------	--

Description

team_edge_shot_speed() scrapes the EDGE shot speed statistics for a given set of team, season, game_type, and category.

Usage

```
team_edge_shot_speed(
  team = 1,
  season = "now",
  game_type = "",
  category = "details"
)
```

Arguments

team	integer ID (e.g., 21), character full name (e.g., 'Colorado Avalanche'), OR three-letter code (e.g., 'COL'); see teams() for reference; ID is preferable as there now exists duplicate three-letter codes (i.e., 'UTA' for 'Utah Hockey Club' and 'Utah Mammoth')
season	integer in YYYYYYYY (e.g., 20242025); see team_edge_seasons() for reference
game_type	integer in 1:3 (where 1 = pre-season, 2 = regular season, 3 = playoff/post-season) OR character of 'pre', 'regular', or 'playoff'/'post'; see team_edge_seasons() for reference; most functions will NOT support pre-season
category	character of 'd'/'details' or 'h'/'hardest'

Value

data.frame with one row per position (category = 'details') or shot (category = 'hardest')

Examples

```
COL_hardest_shots_regular_20242025 <- team_edge_shot_speed(
  team      = 21,
  season     = 20242025,
  game_type  = 2,
  category   = 'H'
)
```

team_edge_skating_distance

Access the EDGE skating distance statistics for a team, season, game type, and category

Description

team_edge_skating_distance() scrapes the EDGE skating distance statistics for a given set of team, season, game_type, and category.

Usage

```
team_edge_skating_distance(
  team = 1,
  season = "now",
  game_type = "",
  category = "details"
)
```

Arguments

team	integer ID (e.g., 21), character full name (e.g., 'Colorado Avalanche'), OR three-letter code (e.g., 'COL'); see teams() for reference; ID is preferable as there now exists duplicate three-letter codes (i.e., 'UTA' for 'Utah Hockey Club' and 'Utah Mammoth')
season	integer in YYYYYYYY (e.g., 20242025); see team_edge_seasons() for reference
game_type	integer in 1:3 (where 1 = pre-season, 2 = regular season, 3 = playoff/post-season) OR character of 'pre', 'regular', or 'playoff'/'post'; see team_edge_seasons() for reference; most functions will NOT support pre-season
category	character of 'd'/'details' or 'l'/'l10'/'last 10'

Value

data.frame with one row per combination of strength state and position (category = 'details') or game (category = 'last 10') game

Examples

```
COL_L10_skating_distance_regular_20242025 <- team_edge_skating_distance(
  team      = 21,
  season    = 20242025,
  game_type = 2,
  category  = 'L'
)
```

team_edge_skating_speed

Access the EDGE skating speed statistics for a team, season, game type, and category

Description

team_edge_skating_speed() scrapes the EDGE skating speed statistics for a given set of team, season, game_type, and category.

Usage

```
team_edge_skating_speed(
  team = 1,
  season = "now",
  game_type = "",
  category = "details"
)
```

Arguments

team	integer ID (e.g., 21), character full name (e.g., 'Colorado Avalanche'), OR three-letter code (e.g., 'COL'); see teams() for reference; ID is preferable as there now exists duplicate three-letter codes (i.e., 'UTA' for 'Utah Hockey Club' and 'Utah Mammoth')
season	integer in YYYYYYYY (e.g., 20242025); see team_edge_seasons() for reference
game_type	integer in 1:3 (where 1 = pre-season, 2 = regular season, 3 = playoff/post-season) OR character of 'pre', 'regular', or 'playoff'/'post'; see team_edge_seasons() for reference; most functions will NOT support pre-season
category	character of 'd'/'details' or 't'/'top'/'top speeds'

Value

data.frame with one row per position (category = 'details') or burst (category = 'top speeds')

Examples

```
COL_top_speeds_regular_20242025 <- team_edge_skating_speed(
  team      = 21,
  season    = 20242025,
  game_type = 2,
  category  = 'T'
)
```

team_edge_summary	<i>Access the EDGE summary for a team, season, and game type</i>
-------------------	--

Description

team_edge_summary() scrapes the EDGE summary for a given set of team, season, and game_type.

Usage

```
team_edge_summary(team = 1, season = "now", game_type = "")
```

Arguments

team	integer ID (e.g., 21), character full name (e.g., 'Colorado Avalanche'), OR three-letter code (e.g., 'COL'); see teams() for reference; ID is preferable as there now exists duplicate three-letter codes (i.e., 'UTA' for 'Utah Hockey Club' and 'Utah Mammoth')
season	integer in YYYYYYYY (e.g., 20242025); see team_edge_seasons() for reference
game_type	integer in 1:3 (where 1 = pre-season, 2 = regular season, 3 = playoff/post-season) OR character of 'pre', 'regular', or 'playoff'/'post'; see team_edge_seasons() for reference; most functions will NOT support pre-season

Value

list of various items

Examples

```
COL_EDGE_summary_regular_20242025 <- team_edge_summary(  
  team      = 21,  
  season    = 20242025,  
  game_type = 2  
)
```

team_edge_zone_time	<i>Access the EDGE zone time statistics for a team, season, game type, and category</i>
---------------------	---

Description

team_edge_zone_time() scrapes the EDGE zone time statistics for a given set of team, season, game_type, and category.

Usage

```
team_edge_zone_time(  
  team = 1,  
  season = "now",  
  game_type = "",  
  category = "details"  
)
```

Arguments

team	integer ID (e.g., 21), character full name (e.g., 'Colorado Avalanche'), OR three-letter code (e.g., 'COL'); see teams() for reference; ID is preferable as there now exists duplicate three-letter codes (i.e., 'UTA' for 'Utah Hockey Club' and 'Utah Mammoth')
season	integer in YYYYYYYY (e.g., 20242025); see team_edge_seasons() for reference
game_type	integer in 1:3 (where 1 = pre-season, 2 = regular season, 3 = playoff/post-season) OR character of 'pre', 'regular', or 'playoff'/'post'; see team_edge_seasons() for reference; most functions will NOT support pre-season
category	character of 'd'/'details' or 'dS'/'dSOG'/'dShot'/'shot differential'

Value

data.frame with one row per strength state (category = 'details') or list with four items (category = 'shot differential')

Examples

```
COL_dS_regular_20242025 <- team_edge_zone_time(  
  team      = 21,  
  season    = 20242025,  
  game_type = 2,  
  category  = 'dS'  
)
```

team_game_report	<i>Access various reports for a season, game type, and category for all the teams by game</i>
------------------	---

Description

team_game_report() scrapes various reports for a given set of season, game_type, and category for all the teams by game.

Usage

```
team_game_report(  
  season = season_now(),  
  game_type = game_type_now(),  
  category = "summary"  
)
```

Arguments

- season integer in YYYYYYYY (e.g., 20242025); see [seasons\(\)](#) for reference
- game_type integer in 1:3 (where 1 = pre-season, 2 = regular season, 3 = playoff/post-season) OR character of 'pre', 'regular', or playoff'/post'; see [seasons\(\)](#) for reference; most functions will NOT support pre-season
- category character (e.g., 'leadingtrailing'); see [team_report_configurations\(\)](#) for reference

Value

data.frame with one row per game per team

Examples

```
situational_team_game_report_playoffs_20212022 <- team_game_report(  
  season = 20212022,  
  game_type = 3,  
  category = 'leadingtrailing'  
)
```

team_logos	<i>Access all the team logos</i>
------------	----------------------------------

Description

team_logos() scrapes all the team logos.

Usage

```
team_logos()
```

Value

data.frame with one row per logo

Examples

```
all_team_logos <- team_logos()
```

team_month_schedule	<i>Access the schedule for a team and month</i>
---------------------	---

Description

team_month_schedule() scrapes the schedule for a given set of team and month.

Usage

```
team_month_schedule(team = 1, month = "now")
```

Arguments

team	integer ID (e.g., 21), character full name (e.g., 'Colorado Avalanche'), OR three-letter code (e.g., 'COL'); see teams() for reference; ID is preferable as there now exists duplicate three-letter codes (i.e., 'UTA' for 'Utah Hockey Club' and 'Utah Mammoth')
month	character in 'YYYY-MM' (e.g., '2025-01'); see seasons() for reference

Value

data.frame with one row per game

Examples

```
COL_schedule_December_2025 <- team_month_schedule(
  team = 21,
  month = '2025-12'
)
```

team_prospects	<i>Access the prospects for a team and position</i>
----------------	---

Description

team_prospects() scrapes the prospects for a given set of team and position.

Usage

```
team_prospects(team = 1, position = "forwards")
```

Arguments

team	integer ID (e.g., 21), character full name (e.g., 'Colorado Avalanche'), OR three-letter code (e.g., 'COL'); see teams() for reference; ID is preferable as there now exists duplicate three-letter codes (i.e., 'UTA' for 'Utah Hockey Club' and 'Utah Mammoth')
position	character of 'f'/'forwards', 'd'/'defensemen', or 'g'/'goalies'

Value

data.frame with one row per player

Examples

```
COL_forward_prospects <- team_prospects(
  team    = 21,
  position = 'F'
)
```

team_report_configurations

Access the configurations for team reports

Description

team_report_configurations() scrapes the configurations for [team_season_report\(\)](#) and [team_game_report\(\)](#).

Usage

```
team_report_configurations()
```

```
team_report_configs()
```

Value

list with various items

Examples

```
team_report_configs <- team_report_configurations()
```

team_seasons

Access the season(s) and game type(s) in which a team played

Description

team_seasons() scrapes the season(s) and game type(s) in which a team played in the NHL.

Usage

```
team_seasons(team = 1)
```

Arguments

team integer ID (e.g., 21), character full name (e.g., 'Colorado Avalanche'), OR three-letter code (e.g., 'COL'); see [teams\(\)](#) for reference; ID is preferable as there now exists duplicate three-letter codes (i.e., 'UTA' for 'Utah Hockey Club' and 'Utah Mammoth')

Value

data.frame with one row per season

Examples

```
COL_seasons <- team_seasons(team = 21)
```

team_season_report	<i>Access various reports for a season, game type, and category for all the teams by season</i>
--------------------	---

Description

team_season_report() scrapes various reports for a given set of season, game_type, and category for all the teams by season.

Usage

```
team_season_report(  
  season = season_now(),  
  game_type = game_type_now(),  
  category = "summary"  
)
```

Arguments

season integer in YYYYYYYY (e.g., 20242025); see [seasons\(\)](#) for reference

game_type integer in 1:3 (where 1 = pre-season, 2 = regular season, 3 = playoff/post-season) OR character of 'pre', 'regular', or 'playoff'/'post'; see [seasons\(\)](#) for reference; most functions will NOT support pre-season

category character (e.g., 'leadingtrailing'); see [team_report_configurations\(\)](#) for reference

Value

data.frame with one row per team

Examples

```
situational_team_season_report_playoffs_20212022 <- team_season_report(
  season    = 20212022,
  game_type = 3,
  category  = 'leadingtrailing'
)
```

team_season_schedule	<i>Access the schedule for a team and season</i>
----------------------	--

Description

team_season_schedule() scrapes the schedule for a given set of team and season.

Usage

```
team_season_schedule(team = 1, season = "now")
```

Arguments

team	integer ID (e.g., 21), character full name (e.g., 'Colorado Avalanche'), OR three-letter code (e.g., 'COL'); see teams() for reference; ID is preferable as there now exists duplicate three-letter codes (i.e., 'UTA' for 'Utah Hockey Club' and 'Utah Mammoth')
season	integer in YYYYYYYY (e.g., 20242025); see seasons() for reference

Value

data.frame with one row per game

Examples

```
COL_schedule_20252026 <- team_season_schedule(
  team    = 21,
  season  = 20252026
)
```

`team_season_statistics`*Access the statistics for all the teams by season and game type*

Description

`team_season_statistics()` scrapes the statistics for all the teams by season and game type.

Usage

```
team_season_statistics()
```

```
team_season_stats()
```

Value

data.frame with one row per team per season per game type

Examples

```
# May take >5s, so skip.  
team_season_statistics <- team_season_statistics()
```

`team_week_schedule`*Access the schedule for a team and week since a date*

Description

`team_week_schedule()` scrapes the schedule for a given set of team and a week since date.

Usage

```
team_week_schedule(team = 1, date = "now")
```

Arguments

<code>team</code>	integer ID (e.g., 21), character full name (e.g., 'Colorado Avalanche'), OR three-letter code (e.g., 'COL'); see teams() for reference; ID is preferable as there now exists duplicate three-letter codes (i.e., 'UTA' for 'Utah Hockey Club' and 'Utah Mammoth')
<code>date</code>	character in 'YYYY-MM-DD' (e.g., '2025-01-01'); see seasons() for reference

Value

data.frame with one row per game

Examples

```
COL_schedule_Family_Week_2025 <- team_week_schedule(  
  team = 21,  
  date = '2025-10-06'  
)
```

tv_schedule	<i>Access the NHL Network TV schedule for a date</i>
-------------	--

Description

tv_schedule() scrapes the NHL Network TV schedule for a given date.

Usage

```
tv_schedule(date = "now")
```

Arguments

date character in 'YYYY-MM-DD' (e.g., '2025-01-01'); see [seasons\(\)](#) for reference

Value

data.frame with one row per program

Examples

```
tv_schedule_Halloween_2025 <- tv_schedule(date = '2025-10-31')
```

venues	<i>Access all the venues</i>
--------	------------------------------

Description

venues() scrapes all the venues.

Usage

```
venues()
```

Value

data.frame with one row per venue

Examples

```
all_venues <- venues()
```

wsc_play_by_play	<i>Access the World Showcase (WSC) play-by-play for a game</i>
------------------	--

Description

wsc_play_by_play() scrapes the WSC play-by-play for given game.

Usage

```
wsc_play_by_play(game = 2023030417)
```

```
wsc_pbp(game = 2023030417)
```

Arguments

game integer ID (e.g., 2025020275); see [games\(\)](#) for reference

Value

data.frame with one row per event (play)

Examples

```
wsc_pbp_Martin_Necas_legacy_game <- wsc_play_by_play(game = 2025020275)
```

wsc_play_by_plays	<i>Access the World Showcase (WSC) play-by-plays for a season</i>
-------------------	---

Description

wsc_play_by_plays() loads the WSC play-by-plays for a given season.

Usage

```
wsc_play_by_plays(season = 20242025)
```

```
wsc_pbps(season = 20242025)
```

Arguments

season integer in YYYYYYYY (e.g., 20242025); see [seasons\(\)](#) for reference

Value

data.frame with one row per event (play) per game

Examples

```
# May take >5s, so skip.  
wsc_pbps_20212022 <- wsc_play_by_plays(season = 20212022)
```

wsc_summary	<i>Access the World Showcase (WSC) summary for a game</i>
-------------	---

Description

wsc_summary() scrapes the WSC summary for a given game.

Usage

```
wsc_summary(game = 2023030417)
```

Arguments

game integer ID (e.g., 2025020275); see [games\(\)](#) for reference

Value

list of various items

Examples

```
wsc_summary_Martin_Necas_legacy_game <- wsc_summary(game = 2025020275)
```

x_game_cumulative_expected_goals	<i>Save an X (Twitter) share-able cumulative expected goals (xG) time-series plot for a game</i>
----------------------------------	--

Description

x_game_cumulative_expected_goals() saves an X share-able cumulative xG time-series plot for a given game as a PNG.

Usage

```
x_game_cumulative_expected_goals(game = 2023030417, model = 1, save = TRUE)  
  
x_game_cum_xG(game = 2023030417, model = 1)
```


Arguments

game	integer ID (e.g., 2025020275); see games() for reference
model	integer in 1:3 indicating which expected goals model to use (e.g., 1); see calculate_expected_goals_v1() , calculate_expected_goals_v2() , and/or calculate_expected_goals_v3() for reference
save	logical only FALSE for tests

Value

NULL

Examples

```
# May take >5s, so skip.
x_game_cumulative_expected_goals(
  game = 2023030417,
  model = 1,
  save = FALSE
)
```

x_game_shot_locations *Save an X (Twitter) share-able shot-location plot for a game*

Description

x_game_shot_locations() saves an X share-able shot-location plot for a given game.

Usage

```
x_game_shot_locations(game = 2023030417, team = "home", model = 1, save = TRUE)

x_game_shot_locs(game = 2023030417, team = "home", model = 1)
```

Arguments

game	integer ID (e.g., 2025020275); see games() for reference
team	character of 'h'/'home' or 'a'/'away'
model	integer in 1:3 indicating which expected goals model to use (e.g., 1); see calculate_expected_goals_v1() , calculate_expected_goals_v2() , and/or calculate_expected_goals_v3() for reference
save	logical only FALSE for tests

Value

NULL

Examples

```
# May take >5s, so skip.
x_game_shot_locations(
  game = 2023030417,
  model = 1,
  team = 'H',
  save = FALSE
)
```

Index

attendance, 6
attendance(), 34
award_winners, 7
award_winners(), 34
awards, 7
awards(), 34

boxscore, 8
boxscore(), 45
bracket, 8
bracket(), 35, 54, 87

calculate_angle, 9
calculate_angle(), 10–12
calculate_distance, 9
calculate_distance(), 10–12
calculate_expected_goals_v1, 10
calculate_expected_goals_v1(), 75, 76, 121
calculate_expected_goals_v2, 11
calculate_expected_goals_v2(), 75, 76, 121
calculate_expected_goals_v3, 12
calculate_expected_goals_v3(), 75, 76, 121
calculate_xG_v1
 (calculate_expected_goals_v1), 10
calculate_xG_v2
 (calculate_expected_goals_v2), 11
calculate_xG_v3
 (calculate_expected_goals_v3), 12
coach_career_statistics, 13
coach_career_statistics(), 38, 39
coach_career_stats
 (coach_career_statistics), 13
coach_franchise_statistics, 13
coach_franchise_stats
 (coach_franchise_statistics), 13
coaches, 12
coaches(), 38
combine_reports, 14
count_goals_shots, 15
count_goals_shots(), 12
countries, 14
countries(), 30, 35, 49

draft_picks, 16
draft_picks(), 36
draft_prospects, 16
draft_rankings, 17
draft_rankings(), 36
draft_tracker, 17
draft_tracker(), 37
drafts, 15
drafts(), 17, 36, 37
draw_NHL_rink, 18

espn_futures, 18
espn_futures(), 41
espn_game_odds, 19
espn_game_odds(), 40
espn_game_summary, 20
espn_game_summary(), 39
espn_games, 19
espn_games(), 19, 20, 22, 39–41
espn_injuries, 20
espn_injuries(), 41
espn_pbp (espn_play_by_play), 22
espn_play_by_play, 22
espn_play_by_play(), 40
espn_player_summary, 21
espn_player_summary(), 37
espn_players, 21
espn_players(), 21, 38
espn_team_summary, 23

- espn_team_summary(), [42](#)
- espn_teams, [22](#)
- espn_teams(), [23](#), [42](#)
- espn_transactions, [23](#)
- espn_transactions(), [42](#)
- expansion_draft_picks, [24](#)
- expansion_drafts, [24](#)
- flag_is_home, [25](#)
- flag_is_home(), [15](#), [25](#), [26](#), [77](#), [103](#)
- flag_is_rebound, [25](#)
- flag_is_rebound(), [11](#), [12](#)
- flag_is_rush, [26](#)
- flag_is_rush(), [11](#), [12](#)
- franchise_playoff_situational_results, [27](#)
- franchise_season_statistics, [28](#)
- franchise_season_statistics(), [43](#)
- franchise_season_stats
 - (franchise_season_statistics), [28](#)
- franchise_statistics, [28](#)
- franchise_stats (franchise_statistics), [28](#)
- franchise_team_statistics, [29](#)
- franchise_team_statistics(), [44](#)
- franchise_team_stats
 - (franchise_team_statistics), [29](#)
- franchise_versus_franchise, [29](#)
- franchise_versus_franchise(), [44](#)
- franchise_vs_franchise
 - (franchise_versus_franchise), [29](#)
- franchises, [27](#)
- franchises(), [43](#)
- game_odds, [30](#)
- game_odds(), [49](#)
- game_rosters, [31](#)
- game_type_now, [31](#)
- game_type_now(), [53](#)
- games, [30](#)
- games(), [8](#), [31–33](#), [45–47](#), [54](#), [62](#), [75](#), [76](#), [82](#), [87](#), [119–121](#)
- gc_pbp (gc_play_by_play), [32](#)
- gc_pbps (gc_play_by_plays), [32](#)
- gc_play_by_play, [32](#)
- gc_play_by_play(), [9–12](#), [15](#), [25](#), [26](#), [47](#), [77](#), [82](#), [102–104](#)
- gc_play_by_plays, [32](#)
- gc_summary, [33](#)
- gc_summary(), [40](#), [41](#), [46](#)
- general_managers, [33](#)
- get_attendance, [34](#)
- get_award_winners, [34](#)
- get_awards, [34](#)
- get_bracket, [35](#)
- get_configuration, [35](#)
- get_countries, [35](#)
- get_draft_picks, [36](#)
- get_draft_rankings, [36](#)
- get_draft_tracker, [37](#)
- get_drafts, [36](#)
- get_espn_athlete, [37](#)
- get_espn_athletes, [38](#)
- get_espn_coach, [38](#)
- get_espn_coach_career, [39](#)
- get_espn_coaches, [38](#)
- get_espn_event, [39](#)
- get_espn_event_odds, [40](#)
- get_espn_event_officials, [40](#)
- get_espn_event_play_by_play, [40](#)
- get_espn_event_stars, [41](#)
- get_espn_events, [39](#)
- get_espn_futures, [41](#)
- get_espn_injuries, [41](#)
- get_espn_team, [42](#)
- get_espn_teams, [42](#)
- get_espn_transactions, [42](#)
- get_franchise_season_by_season, [43](#)
- get_franchise_team_totals, [44](#)
- get_franchise_vs_franchise, [44](#)
- get_franchises, [43](#)
- get_game_boxscore, [45](#)
- get_game_landing, [46](#)
- get_game_story, [46](#)
- get_games, [45](#)
- get_gc_play_by_play, [47](#)
- get_glossary, [47](#)
- get_goalie_leaders, [48](#)
- get_goalie_milestones, [48](#)
- get_goalie_statistics, [49](#)
- get_goalies, [47](#)
- get_officials, [49](#)
- get_partner_odds, [49](#)
- get_player_game_log, [50](#)
- get_player_landing, [51](#)

get_players, 50
 get_schedule, 51
 get_scoreboards, 52
 get_scores, 52
 get_season_now, 53
 get_seasons, 53
 get_series, 53
 get_series_schedule, 54
 get_shift_charts, 54
 get_skater_leaders, 55
 get_skater_milestones, 56
 get_skater_statistics, 56
 get_skaters, 55
 get_spotlight_players, 56
 get_standings, 57
 get_standings_information, 57
 get_streams, 58
 get_team_prospects, 58
 get_team_roster, 59
 get_team_roster_statistics, 59
 get_team_schedule, 60
 get_team_scoreboard, 60
 get_team_seasons, 61
 get_team_statistics, 61
 get_teams, 58
 get_tv_schedule, 61
 get_venues, 62
 get_wsc_play_by_play, 62
 glossary, 63
 glossary(), 47
 gms (general_managers), 33
 goalie_edge_5_vs_5
 (goalie_edge_five_versus_five),
 63
 goalie_edge_five_versus_five, 63
 goalie_edge_leaders, 64
 goalie_edge_save_percentage, 65
 goalie_edge_seasons, 66
 goalie_edge_seasons(), 64, 65, 67
 goalie_edge_shot_location, 66
 goalie_edge_summary, 67
 goalie_game_report, 68
 goalie_game_report(), 49, 71
 goalie_game_scoring, 69
 goalie_game_statistics, 69
 goalie_game_stats
 (goalie_game_statistics), 69
 goalie_leaders, 70
 goalie_leaders(), 48
 goalie_milestones, 70
 goalie_milestones(), 48
 goalie_regular_statistics, 71
 goalie_regular_stats
 (goalie_regular_statistics), 71
 goalie_report_configs
 (goalie_report_configurations),
 71
 goalie_report_configurations, 71
 goalie_report_configurations(), 35, 68,
 73
 goalie_scoring, 72
 goalie_season_report, 72
 goalie_season_report(), 49, 71
 goalie_season_statistics, 73
 goalie_season_stats
 (goalie_season_statistics), 73
 goalie_series_statistics, 74
 goalie_series_stats
 (goalie_series_statistics), 74
 goalie_statistics, 74
 goalie_stats (goalie_statistics), 74
 graphics::points(), 18

 ig_game_cum_xG
 (ig_game_cumulative_expected_goals),
 75
 ig_game_cumulative_expected_goals, 75
 ig_game_shot_locations, 75
 ig_game_shot_locs
 (ig_game_shot_locations), 75

 location, 76
 lottery_odds, 77

 normalize_coordinates, 77
 normalize_coordinates(), 9, 10

 officials, 78
 officials(), 49

 penalty_shots, 78
 ping, 79
 player_game_log, 80
 player_game_log(), 50
 player_seasons, 80
 player_summary, 81
 player_summary(), 51

- players, [79](#)
- players(), [47](#), [50](#), [51](#), [55](#), [80](#), [81](#), [89–94](#)
- playoff_season_statistics, [81](#)
- playoff_season_stats
 - (playoff_season_statistics), [81](#)
- ps (penalty_shots), [78](#)
- replay, [82](#)
- roster, [82](#)
- roster(), [59](#)
- roster_statistics, [83](#)
- roster_statistics(), [59](#)
- roster_stats (roster_statistics), [83](#)
- schedule, [84](#)
- schedule(), [51](#)
- scores, [85](#)
- scores(), [52](#)
- season_now, [86](#)
- season_now(), [53](#)
- seasons, [85](#)
- seasons(), [8](#), [18](#), [19](#), [32](#), [35](#), [48](#), [50–55](#), [57](#), [59](#), [60](#), [62](#), [68](#), [70](#), [73](#), [80](#), [83–85](#), [87](#), [95](#), [98](#), [101](#), [112](#), [113](#), [115–119](#)
- series, [86](#)
- series(), [54](#), [87](#)
- series_schedule, [87](#)
- series_schedule(), [54](#)
- shifts, [87](#)
- shifts(), [54](#)
- skater_edge_leaders, [88](#)
- skater_edge_seasons, [88](#)
- skater_edge_seasons(), [88–94](#)
- skater_edge_shot_location, [89](#)
- skater_edge_shot_speed, [90](#)
- skater_edge_skating_distance, [91](#)
- skater_edge_skating_speed, [92](#)
- skater_edge_summary, [93](#)
- skater_edge_zone_time, [93](#)
- skater_game_report, [94](#)
- skater_game_report(), [56](#), [97](#)
- skater_leaders, [95](#)
- skater_leaders(), [55](#)
- skater_milestones, [96](#)
- skater_milestones(), [56](#)
- skater_playoff_statistics, [96](#)
- skater_playoff_stats
 - (skater_playoff_statistics), [96](#)
- skater_regular_statistics, [97](#)
- skater_regular_stats
 - (skater_regular_statistics), [97](#)
- skater_report_configs
 - (skater_report_configurations), [97](#)
- skater_report_configurations, [97](#)
- skater_report_configurations(), [35](#), [95](#), [98](#)
- skater_season_report, [98](#)
- skater_season_report(), [56](#), [97](#)
- skater_season_statistics, [99](#)
- skater_season_stats
 - (skater_season_statistics), [99](#)
- skater_series_statistics, [99](#)
- skater_series_stats
 - (skater_series_statistics), [99](#)
- skater_statistics, [100](#)
- skater_stats (skater_statistics), [100](#)
- spotlight_players, [100](#)
- spotlight_players(), [56](#)
- standings, [101](#)
- standings(), [57](#)
- standings_rules, [101](#)
- standings_rules(), [57](#)
- streams, [102](#)
- streams(), [58](#)
- strip_game_id, [102](#)
- strip_game_id(), [104](#)
- strip_situation_code, [103](#)
- strip_situation_code(), [10–12](#)
- strip_time_period, [103](#)
- strip_time_period(), [26](#)
- team_edge_leaders, [105](#)
- team_edge_seasons, [105](#)
- team_edge_seasons(), [105–111](#)
- team_edge_shot_location, [106](#)
- team_edge_shot_speed, [107](#)
- team_edge_skating_distance, [108](#)
- team_edge_skating_speed, [109](#)
- team_edge_summary, [110](#)
- team_edge_zone_time, [110](#)
- team_game_report, [111](#)
- team_game_report(), [61](#), [114](#)
- team_logos, [112](#)
- team_month_schedule, [113](#)
- team_prospects, [113](#)
- team_prospects(), [58](#)

team_report_configs
 (team_report_configurations),
 114
team_report_configurations, 114
team_report_configurations(), 35, 112,
 115
team_season_report, 115
team_season_report(), 61, 114
team_season_schedule, 116
team_season_schedule(), 60
team_season_statistics, 117
team_season_stats
 (team_season_statistics), 117
team_seasons, 114
team_seasons(), 61
team_week_schedule, 117
teams, 104
teams(), 58–61, 83, 106–111, 113, 115–117
tv_schedule, 118
tv_schedule(), 61

venues, 118
venues(), 62

wsc_pbp (wsc_play_by_play), 119
wsc_pbps (wsc_play_by_plays), 119
wsc_play_by_play, 119
wsc_play_by_play(), 9–12, 15, 25, 26, 62,
 77, 82, 102–104
wsc_play_by_plays, 119
wsc_summary, 120
wsc_summary(), 40, 41, 46

x_game_cum_xG
 (x_game_cumulative_expected_goals),
 120
x_game_cumulative_expected_goals, 120
x_game_shot_locations, 121
x_game_shot_locs
 (x_game_shot_locations), 121