



fastRhockey :: CHEAT SHEET

v1.0.0

Tidy R access to hockey play-by-play, box scores, schedules, rosters and stats: the NHL Web API (api-web.nhle.com), NHL Stats REST, NHL EDGE tracking, NHL Records, ESPN and Fox NHL feeds, the PWHL plus AHL / OHL / WHL / QMJHL (HockeyTech), and season loaders from the SportsDataverse data releases. Sister of [hoopR](#) / [wehoop](#) / [cfbfaster](#); Python mirror: [sportsdataverse.nhl](https://sportsdataverse.nhl.com/) and [sportsdataverse.pwhl](https://sportsdataverse.pwhl.com/). Page 1 of 2: NHL. Page 2: PWHL, HockeyTech leagues, PHF, helpers.

R >= 4.1.0 · httr2 · dplyr · data.table
fastRhockey.sportsdataverse.org

GET STARTED

```
install.packages("fastRhockey")
# dev build
remotes::install_github("sportsdataverse/fastRhockey")
library(fastRhockey)
library(dplyr)
```

Every function returns a tibble of class `fastRhockey_data` (or a named list of them). Live NHL wrappers hit api-web.nhle.com/v1/ with no API key. Endpoint family overviews: `?nhl_web` `?nhl_stats_rest` `?nhl_edge` `?nhl_records` `?nhl_loaders` `?nhl_xg` `?pwhl`.

NHL RELEASE LOADERS

Pre-scraped full seasons (2011+) from the **sportsdataverse-data** GitHub releases (tag `nhl_*`). Default seasons = `most_recent_nhl_season()`; optional `dbConnection`, `tablename` write straight to a DB.

```
pbp <- load_nhl_pbp(seasons = 2023:2025)
lite <- load_nhl_pbp_lite(2025)
# slim columns
tb <- load_nhl_team_box(2025)
pb <- load_nhl_player_box(2025)
# + skater / goalie
sch <- load_nhl_schedule(2020:2025)
```

Loader	Release tag
<code>load_nhl_pbp · _pbp_full · _pbp_lite</code>	<code>nhl_pbp_full / nhl_pbp_lite</code>
<code>load_nhl_team_box _player_box _skater_box _goalie_box</code>	<code>nhl_*_boxscores</code>
<code>load_nhl_schedule _rosters _game_rosters _game_info</code>	<code>nhl_schedules ...</code>
<code>load_nhl_scoring _penalties _three_stars _scratches</code>	<code>nhl_scoring ...</code>
<code>load_nhl_linescore _shifts _officials _shots_by_period _shootout</code>	<code>nhl_shifts ...</code>

Python-parity aliases: `load_nhl_schedules()`, `load_nhl_player_boxscores()`, `load_nhl_team_boxscores()`, `load_nhl_skater_boxscores()`, `load_nhl_goalie_boxscores()`.

```
# local SQLite mirror of NHL pbp (needs DBI + RSQLite)
update_nhl_db(dbdir = ".",
  dbname = "fastRhockey_db",
  tblname = "fastRhockey_nhl_pbp",
  force_rebuild = FALSE)
```

NHL WEB API · NHL_*

GAMES & PLAY-BY-PLAY

```
feed <- nhl_game_feed(2023020001)
# list: pbp, game_info, rosters
pbp <- nhl_game_pbp(id,
  include_shifts = TRUE)
box <- nhl_game_boxscore(2023020001)
# game_info, team_box, skater/goalie_stats
nhl_game_shifts(2023020001)
# one row per shift
nhl_gamecenter_landing(id)
nhl_gamecenter_right_rail(id)
```

Also `nhl_game_story / _content`, `nhl_wsc_pbp`, `nhl_ppt_replay / _goal`. `raw = TRUE` returns the JSON list.

SCHEDULE, SCORES, STANDINGS

```
nhl_schedule(day = "2024-01-15")
nhl_schedule(season = 20232024,
  team_abbr = "TOR", game_type = "regular")
nhl_schedule_calendar(date)
nhl_club_schedule("BOS", season)
nhl_scoreboard(date); nhl_scores(date)
nhl_tv_schedule(date)
nhl_standings(date = "2024-03-01")
nhl_standings_season()
nhl_seasons(); nhl_playoff_bracket(2024)
```

Also `nhl_playoff_carousel / _schedule`, `nhl_where_to_watch()`, `nhl_partner_game_odds("US")`.

TEAMS, ROSTERS, PLAYERS

```
nhl_teams(season); nhl_teams_info("TOR")
nhl_teams_roster("TOR", season = 20232024)
nhl_teams_stats("TOR", season, game_type = 2)
nhl_roster_season("TOR")
nhl_player_info(8478402) # McDavid
nhl_player_game_log(id, season, game_type = 2)
nhl_player_stats(id); nhl_player_spotlight()
```

Also `nhl_team_prospects()`, `nhl_club_stats_season()`, `nhl_team_scoreboard()`, `nhl_conferences()` / `nhl_divisions()` + `_info` variants.

Draft

```
nhl_draft() · nhl_draft_year(year, round) ·
nhl_draft_prospects() / _info() ·
nhl_draft_rankings(season, category) ·
nhl_draft_tracker()
```

SEASON STATS & LEADERS

```
nhl_stats_skaters(report_type = "summary",
  season = 20232024, game_type = 2)
nhl_stats_goalies(...); nhl_stats_teams(...)
nhl_skater_stats_leaders(season,
  categories = "points")
nhl_goalie_stats_leaders(season)
nhl_stats_draft(year); nhl_stats_seasons()
```

`report_type`: summary bios realtime penalties powerplay penaltykill scoringRates ...; page with `limit / start / sort`.

Aggregators (multi-call helpers)

```
nhl_game_ids_by_season() ·
nhl_all_players_by_season() ·
nhl_player_career_stats() · nhl_team / _skater /
_goalie_summary_range()
```

STATS REST (API.NHLE.COM/STATS/REST)

```
nhl_stats_misc(endpoint = "glossary")
nhl_stats_players(limit = 100)
nhl_stats_skater_leaders("points")
nhl_stats_franchise(); nhl_stats_country()
nhl_stats_team_listing()
```

Also `nhl_stats_goalie_leaders · _game_listing · _skater / _goalie_milestones · _glossary · _config · _ping · _content_module`. Cayenne filters: `cayenne_exp = "seasonId=20232024"`.

NHL EDGE TRACKING

```
nhl_edge_skater_detail(8478402, season)
nhl_edge_skater_landing(season)
nhl_edge_skater_comparison(id)
nhl_edge_goalie_detail(id)
nhl_edge_team_detail(team_id)
```

35 wrappers: `nhl_edge_skater_* / _goalie_* / _team_*` (`_detail` + `_top_10`) + `nhl_cat_edge_*_detail`.

RECORDS (RECORDS.NHL.COM)

```
nhl_records_franchise(franchise_id = 10)
nhl_records_player_stats(
  cayenne_exp = "playerId=8478402")
nhl_records_draft(); nhl_records_trophy()
nhl_records_hof_players()
```

25 wrappers: `nhl_records_franchise_* _player_* _skater_* _goalie_* _draft_*`, awards, HOF, officials, attendance, venue, combine.

ESPN NHL (ESPN_NHL_*, 126 FNS)

```
espn_nhl_scoreboard(dates = 20240115)
espn_nhl_schedule(team_id = 21, season = 2024)
espn_nhl_teams()
espn_nhl_team_roster(team_id = 21)
espn_nhl_pbp(game_id = 401559593)
espn_nhl_player_box(game_id)
espn_nhl_team_box(game_id)
espn_nhl_summary(id)
espn_nhl_standings(season = 2024)
espn_nhl_athletes_index(active = TRUE,
  limit = 100)
```

Core-API families: `espn_nhl_game_* · espn_nhl_player_* · espn_nhl_season_* · espn_nhl_team_*` + leaders, draft, news, venues, franchises, coaches.

Fox Sports (Bifrost, read-only)

```
fox_nhl_pbp / _boxscore / _odds / _team_roster /
_team_stats / _team_gamelog / _standings /
_league_leaders.
```

EXPECTED GOALS (NHL_XG)

```
pbp <- nhl_game_pbp(2023020001)
pbp <- helper_nhl_calculate_xg(pbp)
# adds xg column
refresh_xg_models() # re-download models
```

Three XGBoost models (5v5, special teams, penalty shot) from `fastRhockey-nhl-data`, cached once. Needs suggested `xgboost`; without it `xg` is NA.

HELPERS & BUNDLED DATA

```
most_recent_nhl_season() # 2025 (end year)
most_recent_nhl_season_api_param() # 20242025
nhl_team_logos # 34 x 11 tibble
```

`nhl_team_logos`: abbrs, colors, ESPN logos; `division != "Relocated"` = 32 active clubs. Also `pwhl_team_logos / phf_team_logos` (p.2).

PYTHON MIRROR · SPORTSDATAVERSE.NHL

```
load_nhl_pbp / _schedule / _player_boxscore /
_team_boxscore (same release tags) · live_nhl_web_*
nhl_edge_* nhl_stats_rest_* nhl_records_*
espn_nhl_* fox_nhl_* · models_nhl_goalie_gsax()
nhl_skater_rapm() nhl_skater_war()
```



PWHL · PWHL_* (HOCKEYTECH FEEDS)

GAMES & PLAY-BY-PLAY

```
library(fastRhockey)
sch <- pwhl_schedule(season = 2025,
  game_type = "both")
pwhl_scorebar(days_back = 3, days_ahead = 3)
pwhl_game_info(game_id = 1)
gs <- pwhl_game_summary(1) # rosters, 3 stars
pbp <- pwhl_pbp(1) # one row per event
box <- pwhl_player_box(1) # per-game box
```

season is the concluding year of XXXX-YY (2025 = 2024-25; 2024 = inaugural). game_type: "regular" · "playoffs" · "both". Feeds: [statviewfeed](#), [modulekit](#), [gc](#) (JSONP stripped internally).

TEAMS, PLAYERS, LEAGUE

```
pwhl_teams(); pwhl_standings(season = 2025)
pwhl_team_roster(team = "Toronto",
  season = 2025)
pwhl_player_info(id); pwhl_player_stats(id)
pwhl_player_game_log(player_id, season = 2025,
  game_type = "both")
pwhl_player_search("Poulin")
pwhl_stats(position = "skater", team = "all",
  season = 2025, regular = TRUE)
pwhl_leaders(position = "skaters",
  season = 2025,
  game_type = "regular",
  limit = 100)
pwhl_streaks(season)
pwhl_transactions(season)
pwhl_playoff_bracket(season)
pwhl_season_id(season)
```

pwhl_stats(position = "goalie") for netminders (team = abbr like "BOS"). pwhl_team_roster(team =) matches the team_label from pwhl_teams(). pwhl_season_id() returns the HockeyTech season table (id, name, season_yr).

ON-ICE ANALYTICS

```
shifts <- pwhl_game_shifts(game_id)
# one row per stint
toi <- pwhl_player_toi(game_id)
# TOI per player
corsi <- pwhl_game_corsi(game_id)
# CF/CA, FF/FA
```

Same trio exists for every HockeyTech league:

```
ahl_game_shifts(), ohl_player_toi(),
whl_game_corsi() ...
```

PWHL RELEASE LOADERS (LOAD_PWHL_*)

Full seasons from the [sportsdataverse-data](#) releases (tags pwhl_*); default seasons = most_recent_pwhl_season(); dbConnection / tablename optional.

```
pbp <- load_pwhl_pbp(seasons = 2024:2025)
load_pwhl_player_box(2025)
load_pwhl_team_box(2025)
load_pwhl_skater_box(2025)
load_pwhl_goalie_box(2025)
load_pwhl_schedule(2025)
load_pwhl_rosters(2025)
load_pwhl_game_rosters(2025)
load_pwhl_game_info(2025)
```

Loader	Release tag
load_pwhl_pbp	pwhl_pbp
load_pwhl_player_box _skater_box _goalie_box _team_box	pwhl_*_boxscore s
load_pwhl_schedule _rosters _game_rosters _game_info	pwhl_schedules ...
load_pwhl_scoring_summary _penalty_summary _three_stars _officials	pwhl_scoring_sum mary ...
load_pwhl_shots_by_period _shootout	pwhl_shots_by_pe riod · pwhl_shootout

Python-parity aliases: load_pwhl_schedules() load_pwhl_player_boxscores() load_pwhl_skater_boxscores() load_pwhl_goalie_boxscores() load_pwhl_team_boxscores().

```
update_pwhl_db(dbdir = ".",
  dbname = "fastRhockey_db",
  force_rebuild = FALSE)
```

BUNDLED DATA

Object	Shape · use
nhl_team_logos	34 x 11 — abbr, names, division, colors, ESPN logos
pwhl_team_logos	14 x 7 — 6 founders twice (2023-24 + rebrand) + Seattle, Vancouver; join on team_abbr
phf_team_logos	6 x 7 — historical PHF clubs
pwhl_team_logos	> dplyr::select(team_abbr, team_color1, team_logo)

HOCKEYTECH · AHL_* OHL_* WHL_* QMJHL_*

ONE SURFACE, FOUR LEAGUES

Each league exports the same 13 callables (thin wrappers over one HockeyTech core, registry in [hockeytech_leagues.R](#)). Swap the prefix; the signatures match.

```
ahl_season_id() # seasons + ids
ahl_schedule(season = 2025) # or season_id =
ahl_standings(season = 2025)
ahl_teams(season = 2025)
ahl_team_roster(team_id, season = 2025)
ahl_leaders(season = 2025)
ahl_player_stats(player_id)
ahl_pbp(game_id); ahl_game_summary(game_id)
ahl_game_shifts(game_id)
ahl_player_toi(game_id)
ahl_game_corsi(game_id)
most_recent_ahl_season()
```

League	Prefix	Notes
PWHL	pwhl_	richer surface (col 1); pbp_style A
AHL	ahl_	league_id 4; canvas A
OHL	ohl_	canvas B
WHL	whl_	league_id 7; canvas B
QMJHL	qmjhl_	client lhjmq, leaguestat host

season = end year resolved to a HockeyTech season_id via the modulekit seasons feed; pass season_id directly to skip the lookup. Override any league key with env SDV_<LEAGUE>_API_KEY (e.g. SDV_AHL_API_KEY); defaults ship in the package.

Overviews: ?ahl ?ohl ?whl ?qmjhl. HockeyTech errors arrive as HTTP 200 with an error body; wrappers return empty frames gracefully.

SEASON HELPERS

```
most_recent_nhl_season()
most_recent_pwhl_season()
most_recent_ahl_season()
most_recent_ohl_season()
most_recent_whl_season()
most_recent_qmjhl_season()
most_recent_phf_season()
most_recent_nhl_season_api_param()
```

All return the season END year (NHL api param = YYYYYYYY). PWHL season spans Nov–May, so Jun–Oct resolves to the just-finished season.

PHF (DEPRECATED — LEAGUE CEASED)

The PHF has ceased operations; every phf_* / load_phf_* call raises lifecycle::deprecate_stop(). Reference only:

Live wrappers	Loaders
phf_schedule phf_game_all phf_game_raw phf_game_details phf_game_summary phf_pbp phf_team_box phf_player_box	load_phf_pbp load_phf_team_box load_phf_player_box load_phf_rosters load_phf_schedule
phf_standings phf_leaders phf_player_stats phf_team_roster phf_team_stats phf_league_info	update_phf_db

Historical PHF seasons live in the [fastRhockey-data](#) repo releases; PWHL (2024+) is the successor surface.

UTILITIES

```
# documented internals (not exported): use :::
fastRhockey:::rds_from_url(url)
# release .rds
fastRhockey:::csv_from_url(url)
# data.table::fread
fastRhockey:::progressively(f, p) # progressor
progressr::with_progress(
  pbp <- load_nhl_pbp(2020:2025))
```

Loaders report progress via progressr. Returns are fastRhockey_data tibbles; the print method shows data type + update timestamp.

PYTHON MIRROR · SPORTSDATAVERSE.PWHL / HOCKEY

```
import sportsdataverse as sdv
sdv.pwhl_pbp(game_id)
sdv.load_pwhl_pbp(seasons=[2025])
sdv.ahl_schedule(season=2025)
sdv.ohl_standings(season=2025)
```

Python ships 20 HockeyTech families (pwhl ahl ohl whl qmjhl echl sphl ushl ...) from the registry the R side mirrors (e.g. echl_pbp). PWHL-only extras: pwhl_shot_xg, pwhl_skater_rapm, pwhl_team_ratings, pwhl_in_game_win_prob.

SPORTSDATAVERSE SIBLINGS

[hoopR](#) / [wehoop](#) (basketball) · [cfbfastR](#) (CFB) · [baseballr](#) · [oddsapiR](#) (lines) · [sportyR](#) (rink plots) · [sportsdataverse-R](#) meta-package · data repos [fastRhockey-data](#), [fastRhockey-nhl-data](#), [fastRhockey-pwhl-data](#), [sportsdataverse-data](#).



fastRhockey :: CHEAT SHEET

v1.0.0

Tidy R access to hockey play-by-play, box scores, schedules, rosters and stats: the NHL Web API (api-web.nhle.com), NHL Stats REST, NHL EDGE tracking, NHL Records, ESPN and Fox NHL feeds, the PWHL plus AHL / OHL / WHL / QMJHL (HockeyTech), and season loaders from the SportsDataverse data releases. Sister of [hoopR](#) / [wehoop](#) / [cfbfaster](#); Python mirror: [sportsdataverse.nhl](#) and [sportsdataverse.pwhl](#) . Page 1 of 2: NHL. Page 2: PWHL, HockeyTech leagues, PHF, helpers.

R >= 4.1.0 · [httr2](#) · [dplyr](#) · [data.table](#)
[fastRhockey.sportsdataverse.org](#)

GET STARTED

```
install.packages("fastRhockey")
# dev build
remotes::install_github("sportsdataverse/fastRhockey")
library(fastRhockey)
library(dplyr)
```

Every function returns a tibble of class `fastRhockey_data` (or a named list of them). Live NHL wrappers hit api-web.nhle.com/v1/ with no API key. Endpoint family overviews: `?nhl_web` `?nhl_stats_rest` `?nhl_edge` `?nhl_records` `?nhl_loaders` `?nhl_xg` `?pwhl`.

NHL RELEASE LOADERS

Pre-scraped full seasons (2011+) from the **sportsdataverse-data** GitHub releases (tag `nhl_*`). Default `seasons = most_recent_nhl_season()`; optional `dbConnection`, `tablename` write straight to a DB.

```
bbp <- load_nhl_bbp(seasons = 2023:2025)
lite <- load_nhl_bbp_lite(2025)
# slim columns
tb <- load_nhl_team_box(2025)
pb <- load_nhl_player_box(2025)
# + skater / goalie
sch <- load_nhl_schedule(2020:2025)
```

Loader	Release tag
<code>load_nhl_bbp</code> · <code>_bbp_full</code> · <code>_bbp_lite</code>	<code>nhl_bbp_full</code> / <code>nhl_bbp_lite</code>
<code>load_nhl_team_box</code> <code>_player_box</code> <code>_skater_box</code> <code>_goalie_box</code>	<code>nhl_*_boxscores</code>
<code>load_nhl_schedule</code> <code>_rosters</code> <code>_game_rosters</code> <code>_game_info</code>	<code>nhl_schedules</code> ...
<code>load_nhl_scoring</code> <code>_penalties</code> <code>_three_stars</code> <code>_scratches</code>	<code>nhl_scoring</code> ...
<code>load_nhl_linescore</code> <code>_shifts</code> <code>_officials</code> <code>_shots_by_period</code> <code>_shootout</code>	<code>nhl_shifts</code> ...

Python-parity aliases: `load_nhl_schedules()`, `load_nhl_player_boxscores()`, `load_nhl_team_boxscores()`, `load_nhl_skater_boxscores()`, `load_nhl_goalie_boxscores()`.

```
# local SQLite mirror of NHL bbp (needs DBI + RSQLite)
update_nhl_db(dbdir = ".",
  dbname = "fastRhockey_db",
  tblname = "fastRhockey_nhl_bbp",
  force_rebuild = FALSE)
```

NHL WEB API · NHL_*

GAMES & PLAY-BY-PLAY

```
feed <- nhl_game_feed(2023020001)
# list: bbp, game_info, rosters
bbp <- nhl_game_bbp(id,
  include_shifts = TRUE)
box <- nhl_game_boxscore(2023020001)
# game_info, team_box, skater/goalie_stats
nhl_game_shifts(2023020001)
# one row per shift
nhl_gamecenter_landing(id)
nhl_gamecenter_right_rail(id)
```

Also `nhl_game_story` / `_content`, `nhl_wsc_bbp`, `nhl_ppt_replay` / `_goal`. `raw = TRUE` returns the JSON list.

SCHEDULE, SCORES, STANDINGS

```
nhl_schedule(day = "2024-01-15")
nhl_schedule(season = 20232024,
  team_abbr = "TOR", game_type = "regular")
nhl_schedule_calendar(date)
nhl_club_schedule("BOS", season)
nhl_scoreboard(date); nhl_scores(date)
nhl_tv_schedule(date)
nhl_standings(date = "2024-03-01")
nhl_standings_season()
nhl_seasons(); nhl_playoff_bracket(2024)
```

Also `nhl_playoff_carousel` / `_schedule`, `nhl_where_to_watch()`, `nhl_partner_game_odds("US")`.

TEAMS, ROSTERS, PLAYERS

```
nhl_teams(season); nhl_teams_info("TOR")
nhl_teams_roster("TOR", season = 20232024)
nhl_teams_stats("TOR", season, game_type = 2)
nhl_roster_season("TOR")
nhl_player_info(8478402) # McDavid
nhl_player_game_log(id, season, game_type = 2)
nhl_player_stats(id); nhl_player_spotlight()
```

Also `nhl_team_prospects()`, `nhl_club_stats_season()`, `nhl_team_scoreboard()`, `nhl_conferences()` / `nhl_divisions()` + `_info` variants.

Draft

```
nhl_draft() · nhl_draft_year(year, round) ·
nhl_draft_prospects() / _info() ·
nhl_draft_rankings(season, category) ·
nhl_draft_tracker().
```

SEASON STATS & LEADERS

```
nhl_stats_skaters(report_type = "summary",
  season = 20232024, game_type = 2)
nhl_stats_goalies(...); nhl_stats_teams(...)
nhl_skater_stats_leaders(season,
  categories = "points")
nhl_goalie_stats_leaders(season)
nhl_stats_draft(year); nhl_stats_seasons()
```

`report_type: summary bios realtime penalties powerplay penaltykill scoringRates ...`; page with `limit` / `start` / `sort`.

Aggregators (multi-call helpers)

```
nhl_game_ids_by_season() ·
nhl_all_players_by_season() ·
nhl_player_career_stats() · nhl_team / _skater /
  _goalie_summary_range().
```

STATS REST (API.NHLE.COM/STATS/REST)

```
nhl_stats_misc(endpoint = "glossary")
nhl_stats_players(limit = 100)
nhl_stats_skater_leaders("points")
nhl_stats_franchise(); nhl_stats_country()
nhl_stats_team_listing()
```

Also `nhl_stats_goalie_leaders` · `_game_listing` · `_skater/_goalie_milestones` · `_glossary` · `_config` · `_ping` · `_content_module`. Cayenne filters: `cayenne_exp = "seasonId=20232024"`.

NHL EDGE TRACKING

```
nhl_edge_skater_detail(8478402, season)
nhl_edge_skater_landing(season)
nhl_edge_skater_comparison(id)
nhl_edge_goalie_detail(id)
nhl_edge_team_detail(team_id)
```

35 wrappers: `nhl_edge_skater_*` / `_goalie_*` / `_team_*` (`_detail` + `_top_10`) + `nhl_cat_edge_*_detail`.

RECORDS (RECORDS.NHL.COM)

```
nhl_records_franchise(franchise_id = 10)
nhl_records_player_stats(
  cayenne_exp = "playerId=8478402")
nhl_records_draft(); nhl_records_trophy()
nhl_records_hof_players()
```

25 wrappers: `nhl_records_franchise_*` `_player*` `_skater*` `_goalie*` `_draft*`, `awards`, `HOF`, `officials`, `attendance`, `venue`, `combine`.

ESPN NHL (ESPN_NHL_*, 126 FNS)

```
espn_nhl_scoreboard(dates = 20240115)
espn_nhl_schedule(team_id = 21, season = 2024)
espn_nhl_teams()
espn_nhl_team_roster(team_id = 21)
espn_nhl_bbp(game_id = 401559593)
espn_nhl_player_box(game_id)
espn_nhl_team_box(game_id)
espn_nhl_summary(id)
espn_nhl_standings(season = 2024)
espn_nhl_athletes_index(active = TRUE,
  limit = 100)
```

Core-API families: `espn_nhl_game_*` · `espn_nhl_player_*` · `espn_nhl_season_*` · `espn_nhl_team_*` + `leaders`, `draft`, `news`, `venues`, `franchises`, `coaches`.

Fox Sports (Bifrost, read-only)

```
fox_nhl_bbp / _boxscore / _odds / _team_roster /
  _team_stats / _team_game_log / _standings /
  _league_leaders.
```

EXPECTED GOALS (NHL_XG)

```
bbp <- nhl_game_bbp(2023020001)
bbp <- helper_nhl_calculate_xg(bbp)
# adds xg column
refresh_xg_models() # re-download models
```

Three XGBoost models (5v5, special teams, penalty shot) from `fastRhockey-nhl-data`, cached once. Needs suggested `xgboost`; without it `xg` is NA.

HELPERS & BUNDLED DATA

```
most_recent_nhl_season() # 2025 (end year)
most_recent_nhl_season_api_param() # 20242025
nhl_team_logos # 34 x 11 tibble
```

`nhl_team_logos`: `abbrs`, `colors`, ESPN logos; `division != "Relocated"` = 32 active clubs. Also `pwhl_team_logos` / `phf_team_logos` (p.2).

PYTHON MIRROR · SPORTSDATAVERSE.NHL

```
load_nhl_bbp / _schedule / _player_boxscore /
  _team_boxscore (same release tags) · live nhl_web_*
nhl_edge_* nhl_stats_rest_* nhl_records_*
espn_nhl_* fox_nhl_* · models nhl_goalie_gsax()
nhl_skater_rapm() nhl_skater_war().
```



Women's & minor-league hockey :: PWHL · HockeyTech · PHF · helpers

Python mirror: [sportsdataverse.pwhl](#) · [sportsdataverse.hockey.<league>](#) · data: [fastRhockey-pwhl-raw](#) / [-pwhl-data](#) · see also: [hockeyR](#) (NHL, non-SDV)

PWHL · PWHL_* (HOCKEYTECH FEEDS)

GAMES & PLAY-BY-PLAY

```
library(fastRhockey)
sch <- pwhl_schedule(season = 2025,
  game_type = "both")
pwhl_scorebar(days_back = 3, days_ahead = 3)
pwhl_game_info(game_id = 1)
gs <- pwhl_game_summary(1) # rosters, 3 stars
pbp <- pwhl_pbp(1) # one row per event
box <- pwhl_player_box(1) # per-game box
```

season is the concluding year of XXXX-YY (2025 = 2024-25; 2024 = inaugural). game_type: "regular" · "playoffs" · "both". Feeds: statviewfeed, modulekit, gc (JSONP stripped internally).

TEAMS, PLAYERS, LEAGUE

```
pwhl_teams(); pwhl_standings(season = 2025)
pwhl_team_roster(team = "Toronto",
  season = 2025)
pwhl_player_info(id); pwhl_player_stats(id)
pwhl_player_game_log(player_id, season = 2025,
  game_type = "both")
pwhl_player_search("Poulin")
pwhl_stats(position = "skater", team = "all",
  season = 2025, regular = TRUE)
pwhl_leaders(position = "skaters",
  season = 2025,
  game_type = "regular",
  limit = 100)
pwhl_streaks(season)
pwhl_transactions(season)
pwhl_playoff_bracket(season)
pwhl_season_id(season)
```

pwhl_stats(position = "goalie") for netminders (team = abbr like "BOS"). pwhl_team_roster(team =) matches the team_label from pwhl_teams(). pwhl_season_id() returns the HockeyTech season table (id, name, season_yr).

ON-ICE ANALYTICS

```
shifts <- pwhl_game_shifts(game_id)
# one row per stint
toi <- pwhl_player_toi(game_id)
# TOI per player
corsi <- pwhl_game_corsi(game_id)
# CF/CA, FF/FA
```

Same trio exists for every HockeyTech league: ahl_game_shifts(), ohl_player_toi(), whl_game_corsi() ...

PWHL RELEASE LOADERS (LOAD_PWHL_*)

Full seasons from the **sportsdataverse-data** releases (tags pwhl_*); default seasons = most_recent_pwhl_season(); dbConnection / tablename optional.

```
pbp <- load_pwhl_pbp(seasons = 2024:2025)
load_pwhl_player_box(2025)
load_pwhl_team_box(2025)
load_pwhl_skater_box(2025)
load_pwhl_goalie_box(2025)
load_pwhl_schedule(2025)
load_pwhl_rosters(2025)
load_pwhl_game_rosters(2025)
load_pwhl_game_info(2025)
```

Loader	Release tag
load_pwhl_pbp	pwhl_pbp
load_pwhl_player_box	pwhl_*_boxscore
_skater_box	s
_goalie_box	
_team_box	
load_pwhl_schedule	pwhl_schedules
_rosters	...
_game_rosters	_game_info
load_pwhl_scoring_summary	pwhl_scoring_sum
_penalty_summary	_three_stars
_officials	mary ...
load_pwhl_shots_by_period	pwhl_shots_by_pe
_shootout	riod ·
	pwhl_shootout

Python-parity aliases: load_pwhl_schedules()

load_pwhl_player_boxscores()

load_pwhl_skater_boxscores()

load_pwhl_goalie_boxscores()

load_pwhl_team_boxscores().

```
update_pwhl_db(dbdir = ".",
  dbname = "fastRhockey_db",
  force_rebuild = FALSE)
```

BUNDLED DATA

Object	Shape · use
nhl_team_logos	34 x 11 — abbr, names, division, colors, ESPN logos
pwhl_team_logos	14 x 7 — 6 founders twice (2023-24 + rebrand) + Seattle, Vancouver; join on team_abbr
phf_team_logos	6 x 7 — historical PHF clubs
pwhl_team_logos	> dplyr::select(team_abbr, team_color1, team_logo)

HOCKEYTECH · AHL_* OHL_* WHL_* QMJHL_*

ONE SURFACE, FOUR LEAGUES

Each league exports the same 13 callables (thin wrappers over one HockeyTech core, registry in hockeytech_leagues.R). Swap the prefix; the signatures match.

```
ahl_season_id() # seasons + ids
ahl_schedule(season = 2025) # or season_id =
ahl_standings(season = 2025)
ahl_teams(season = 2025)
ahl_team_roster(team_id, season = 2025)
ahl_leaders(season = 2025)
ahl_player_stats(player_id)
ahl_pbp(game_id); ahl_game_summary(game_id)
ahl_game_shifts(game_id)
ahl_player_toi(game_id)
ahl_game_corsi(game_id)
most_recent_ahl_season()
```

League	Prefix	Notes
PWHL	pwhl_	richer surface (col 1); pbp_style A
AHL	ahl_	league_id 4; canvas A
OHL	ohl_	canvas B
WHL	whl_	league_id 7; canvas B
QMJHL	qmjhl_	client lhjmq, leaguestat host

season = end year resolved to a HockeyTech season_id via the modulekit seasons feed; pass season_id directly to skip the lookup. Override any league key with env SDV_<LEAGUE>_API_KEY (e.g. SDV_AHL_API_KEY); defaults ship in the package.

Overviews: ?ahl ?ohl ?whl ?qmjhl. HockeyTech errors arrive as HTTP 200 with an error body; wrappers return empty frames gracefully.

SEASON HELPERS

```
most_recent_nhl_season()
most_recent_pwhl_season()
most_recent_ahl_season()
most_recent_ohl_season()
most_recent_whl_season()
most_recent_qmjhl_season()
most_recent_phf_season()
most_recent_nhl_season_api_param()
```

All return the season END year (NHL api param = YYYYYYYY). PWHL season spans Nov–May, so Jun–Oct resolves to the just-finished season.

PHF (DEPRECATED — LEAGUE CEASED)

The PHF has ceased operations; every phf_* / load_phf_* call raises lifecycle::deprecate_stop(). Reference only:

Live wrappers	Loaders
phf_schedule	phf_game_all
phf_game_raw	load_phf_pbp
phf_game_details	load_phf_team_box
phf_game_summary	load_phf_player_box
phf_pbp	load_phf_rosters
phf_team_box	load_phf_schedule
phf_player_box	
phf_standings	phf_leaders
phf_player_stats	update_phf_db
phf_team_roster	
phf_team_stats	
phf_league_info	

Historical PHF seasons live in the **fastRhockey-data** repo releases; PWHL (2024+) is the successor surface.

UTILITIES

```
# documented internals (not exported): use ::
fastRhockey::rds_from_url(url)
# release .rds
fastRhockey::csv_from_url(url)
# data.table::fread
fastRhockey::progressively(f, p) # progressor
progressr::with_progress(
  pbp <- load_nhl_pbp(2020:2025))
```

Loaders report progress via progressr. Returns are fastRhockey_data tibbles; the print method shows data type + update timestamp.

PYTHON MIRROR · SPORTSDATAVERSE.PWHL / HOCKEY

```
import sportsdataverse as sdv
sdv.pwhl_pbp(game_id)
sdv.load_pwhl_pbp(seasons=[2025])
sdv.ahl_schedule(season=2025)
sdv.ohl_standings(season=2025)
```

Python ships 20 HockeyTech families (pwhl ahl ohl whl qmjhl echl sphl ushl ...) from the registry the R side mirrors (e.g. echl_pbp). PWHL-only extras: pwhl_shot_xg, pwhl_skater_rapm, pwhl_team_ratings, pwhl_in_game_win_prob.

SPORTSDATAVERSE SIBLINGS

hoopR / wehoop (basketball) · cfbfastR (CFB) · baseballr · oddsapiR (lines) · sportyR (rink plots) · sportsdataverse-R meta-package · data repos fastRhockey-data, fastRhockey-nhl-data, fastRhockey-pwhl-data, sportsdataverse-data.