Package: PremPredict (via r-universe)

June 3, 2026

Type Package

Title Predict the Premier League

Version 0.4.7

Description Provides functions to predict the outcome of Premier League games and seasons.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

URL https://github.com/p0bs/PremPredict, https://p0bs.github.io/PremPredict/

BugReports https://github.com/p0bs/PremPredict/issues

RoxygenNote 7.3.3

Imports cli, dplyr, forcats, gnm, jsonlite, knitr, lubridate, purrr, readr, rlang, stringr, tidyr

Suggests covr, tibble, testthat (>= 3.0.0)

Config/testthat/edition 3

Depends R (>= 4.1)

Roxygen list(markdown = TRUE)

Config/pak/sysreqs libicu-dev libx11-dev

Repository https://p0bs.r-universe.dev

Date/Publication 2026-02-03 10:30:13 UTC

RemoteUrl https://github.com/p0bs/PremPredict

RemoteRef HEAD

RemoteSha 0b232d6f88859012a3615d78c41221fa1e750055

Contents

`calc_game_latest` *Find the index of the latest game so far in this Premier League season*

Description

This function finds the index of the most recently played game in this Premier League season.

Usage

```
calc_game_latest(results)
```

Arguments

`results`	These are the results generated by running `get_results`.

Examples

```
## Not run:
calc_game_latest(
  results = data_results
  )

## End(Not run)
```

`calc_points_expected_remaining` *Calculate the points expected to be gained by each team in the remainder of this Premier League season*

Description

This function calculates the points expected to be gained by each team in the remainder of this Premier League season.

Usage

```
calc_points_expected_remaining(games_remaining)
```

Arguments

`games_remaining`
: These are the remaining games, with their associated model parameters, as generated by running `model_parameters_unplayed`.

Examples

```
## Not run:
calc_points_expected_remaining(
  games_remaining = data_results
  )

## End(Not run)
```

`calc_points_expected_total` *Calculate the points expected to be gained by each team across this Premier League season*

Description

This function projects the points expected to be gained by each team across this Premier League season.

Usage

```
calc_points_expected_total(table_current, points_expected)
```

Arguments

`table_current`	This is the current table, as generated by running `calc_table_current`.
`points_expected`	These are the expected points per team across the rest of the season, as generated by running `calc_points_expected_remaining`.

Examples

```
## Not run:
calc_points_expected_total(
  table_current = data_table_current,
  points_expected = data_points_expected_remaining
  )

## End(Not run)
```

`calc_points_simulated_match` *Calculate the result of one match, given model parameters and a uniform random variate.*

Description

This function takes the relevant outcome probabilities for a match and calculates its projected result based upon the outcome of a random number generator.

Usage

```
calc_points_simulated_match(
  data_model_parameters_unplayed_slim,
  randoms,
  number_sims,
  value_match
)
```

Arguments

data_model_parameters_unplayed_slim	These are the model parameters assigned to unplayed matches in this Premier League season, as generated by model_parameters_unplayed, with appropriate columns retained.
randoms	This is a vector of uniformly-distributed random numbers, with a length of the product of the number of remaining matches and the number of simulations.
number_sims	This is the number of simulations to use for each game in the remaining season. Defaults to 50,000.
value_match	This is the simulated match to choose within the randoms vector of many simulated matches.

Examples

```
## Not run:
calc_points_simulated_match(
  data_model_parameters_unplayed_slim = data_model_parameters_unplayed,
  randoms = data_randoms,
  number_sims = value_number_sims,
  value_match = 1L
  )

## End(Not run)
```

---

calc_table_current *Calculate the current table for this Premier League season*

Description

This function generates the latest standings of this Premier League season.

Usage

```
calc_table_current(results)
```

Arguments

results	These are the results generated by running get_results_filtered.

Examples

```
## Not run:
calc_table_current(
  results = data_results
  )

## End(Not run)
```

`example_thisSeason` *A dataset containing the results of the Premier League from this season so far (as committed on 2025-04-21).*

Description

A dataset containing the results of the Premier League from this season so far (as committed on 2025-04-21).

Usage

```
example_thisSeason
```

Format

A data frame with many rows (one for each game this season) and 10 variables:

number_match A character of the index for the game in question

number_match_integer The integer version of `number_match`

matchday The date on which the game occurred

homeTeam The `shortName` of the team that played at home in the match

awayTeam The `shortName` of the team that played away in the match

FTHG The goals scored by the team that played at home in the match

FTAG The goals scored by the team that played away in the match

FTR The result of the match, as a factor of "A" (away win), "D" (draw) or "H" (home win)

played A logical indicating if this game has been played yet

year_end The calendar year in which the season ended

Source

https://github.com/openfootball/football.json

`get_footballData` *Get the latest available results for the Premier League in a given season*

Description

This function retrieves the latest data on the Premier League results for a given season.

Usage

```
get_footballData(value_link, table_schedule, table_teams, value_yearEnd)
```

Arguments

`value_link`	This is the link for the data on the web. For example, you could use 'https://www.football-data.co.uk/mmz4281/2526/E0.csv'.
`table_schedule`	This is the location of the schedule data, as generated through an in-built dataset or by using `get_openData_schedule`.
`table_teams`	These are the teams in the season's Premier League, available as the `teams` dataset in this package.
`value_yearEnd`	This is the integer required as the year in which the season ends.

Source

https://www.football-data.co.uk

Examples

```
## Not run:
get_footballData(
  value_link = "https://www.football-data.co.uk/mmz4281/2526/E0.csv",
  table_schedule = schedule_thisSeason,
  table_teams = teams,
  value_yearEnd = 2026L
  )

## End(Not run)
```

get_openData — *Get the latest available results for the Premier League in a given season*

Description

This function retrieves the latest data on the Premier League results for a given season.

Usage

```
get_openData(value_path, table_teams, value_yearEnd)
```

Arguments

value_path	This is the location of the data on GitHub. See the example below for reference and use an address of the form, 'https://raw.githubusercontent.com/openfootball/football.json/refs/heads/m 25/en.1.json'.
table_teams	These are the teams in the season's Premier League, available as the teams dataset in this package.
value_yearEnd	This is the integer required as the year in which the season ends.

Source

https://github.com/openfootball/football.json

Examples

```
## Not run:
get_openData(
  value_path = "https://raw.githubusercontent.com/openfootball/football.json/refs/etc",
  table_teams = teams,
  value_yearEnd = 2025L
  )

## End(Not run)
```

get_openData_schedule — *Get the latest available schedule for the Premier League in a given season*

Description

This function retrieves the latest data on the Premier League results for a given season.

Usage

```
get_openData_schedule(value_path, table_teams, value_yearEnd)
```

Arguments

value_path	This is the location of the data on GitHub. See the example below for reference and use an address of the form, 'https://raw.githubusercontent.com/openfootball/football.json/refs/heads/m 25/en.1.json'. Note that this data is updated with scores later in the season.
table_teams	These are the teams in the season's Premier League, available as the teams dataset in this package.
value_yearEnd	This is the integer required as the year in which the season ends.

Source

https://github.com/openfootball/football.json

Examples

```
## Not run:
get_openData_schedule(
  value_path = "https://raw.githubusercontent.com/openfootball/football.json/refs/etc",
  table_teams = teams,
  value_yearEnd = 2025L
  )

## End(Not run)
```

---

get_results *Get the Premier League results for the desired seasons*

Description

This function takes the latest data on this Premier League season and combines it with corresponding results from previous seasons, if desired.

Usage

```
get_results(results_thisSeason, seasons = 0L)
```

Arguments

results_thisSeason	These are the results generated by running get_openData on the current season, such as at 'https://raw.githubusercontent.com/openfootball/football.json/refs/heads/master/2024-25/en.1.json'.
seasons	This is the integer required for the number of previous seasons to include. It defaults to zero.

Examples

```
## Not run:
get_results(
  results_thisSeason = data_thisSeason,
  seasons = 1L
  )

## End(Not run)
```

`get_results_filtered` *Get the Premier League results for the desired seasons*

Description

This function takes the latest data on this Premier League season and combines it with corresponding results from previous seasons, if desired.

Usage

```
get_results_filtered(results, index_game_latest, lookback_rounds)
```

Arguments

`results`	These are the results from this and possibly earlier seasons, as generated by `get_results`.
`index_game_latest`	This is the index of the latest game played, which can be generated by `calc_game_latest`.
`lookback_rounds`	This is the number of rounds of fixtures to use in a model (so 38 would represent a whole season).

Examples

```
## Not run:
get_results_filtered(
  results = data_results,
  index_game_latest = 280L,
  lookback_rounds = 38L,
  )

## End(Not run)
```

model_extract_parameters

Run the prediction model

Description

This function takes the relevant results generated in modelling the strength of the Premier League teams and extracts the relevant parameters.

Usage

```
model_extract_parameters(model_output)
```

Arguments

model_output	This is the output generated by model_run.

Examples

```
## Not run:
model_extract_parameters(
  model_output = data_model_output
  )

## End(Not run)
```

model_parameters_unplayed

Assign the model parameters for unplayed games in this Premier League season.

Description

This function takes the relevant parameters from the model and assigns them to the approapriate teams in each remaining game of the season.

Usage

```
model_parameters_unplayed(model_parameters, results)
```

Arguments

model_parameters	This is the output generated by model_extract_parameters.
results	These are the results from this and possibly earlier seasons, as generated by get_results_filtered.

Examples

```
## Not run:
model_parameters_unplayed(
  model_parameters = data_model_parameters,
  results = data_results
  )

## End(Not run)
```

`model_prepare_frame` *Prepare the modelframe in order to run the prediction model*

Description

This function takes the relevant filtered results from the Premier League and combines it in a way that the R prediction model can recognise.

Usage

```
model_prepare_frame(results)
```

Arguments

`results`	These are the results from this and possibly earlier seasons, as generated by `get_results_filtered`.

Examples

```
## Not run:
model_prepare_frame(
  results = data_results
  )

## End(Not run)
```

`model_run` *Run the prediction model*

Description

This function takes the relevant filtered results from the Premier League and uses them to model each team's capabilities, both at home and away.

Usage

```
model_run(modelframe)
```

Arguments

modelframe	This is the modelframe generated in model_prepare_frame.

Examples

```
## Not run:
model_run(
  modelframe = data_modelframe
  )

## End(Not run)
```

---

previous_seasons *A dataset containing the results of the Premier League from previous seasons.*

Description

A dataset containing the results of the Premier League from previous seasons.

Usage

```
previous_seasons
```

Format

A data frame with many rows (one for each game in recent history that involves two current Premier League teams) and 10 variables:

number_match A character of the index for the game in question

number_match_integer The integer version of number_match

matchday The date on which the game occurred

homeTeam The shortName of the team that played at home in the match

awayTeam The shortName of the team that played away in the match

FTHG The goals scored by the team that played at home in the match

FTAG The goals scored by the team that played away in the match

FTR The result of the match, as a factor of "A" (away win), "D" (draw) or "H" (home win)

played A logical indicating if this game has been played yet

year_end The calendar year in which the season ended

Source

https://github.com/openfootball/football.json

print_table_current *Print the current table for this Premier League season*

Description

This function prints the latest standings of this Premier League season.

Usage

```
print_table_current(table_latest)
```

Arguments

table_latest	These are the latest standings generated by running calc_table_current.

Examples

```
## Not run:
print_table_current(
  table_latest = data_calc_table_current
  )

## End(Not run)
```

reformat_outcomes *Reformat the outcomes data for improved presentation.*

Description

This function takes the likelihoods of all possible standings for all clubs over this Premier League season and reformats them for improved presentation..

Usage

```
reformat_outcomes(value)
```

Arguments

value	This is the outcome value to be reformatted.

Examples

```
## Not run:
reformat_outcomes(
  value = 0.94
  )

## End(Not run)
```

`run_simulations` *Assign the model parameters for unplayed games in this Premier League season.*

Description

This function takes the relevant parameters from the model and assigns them to the approapriate teams in each remaining game of the season.

Usage

```
run_simulations(
  results_thisSeason,
  number_seasons = 0L,
  lookback_rounds = 19L,
  number_simulations = 25000L,
  value_seed = 120519L
)
```

Arguments

`results_thisSeason`

These are the results generated by running `get_openData` on the current season, such as at 'https://raw.githubusercontent.com/openfootball/football.json/refs/heads/master/2024-25/en.1.json'. The data should have the following columns and names:

number_match The id values of the matches in the dataset as a character, typically from "001"

number_match_integer the integer equivalent of the characters in `number_match`

matchday The date of the match in yyyy-mm-dd format

homeTeam The `shortName` of the home team, consistent with the data in the `teams` table

awayTeam The `shortName` of the away team, consistent with the data in the `teams` table

FTHG The integer number of goals scored in the whole match by the home team

FTAG The integer number of goals scored in the whole match by the away team

FTR The result of the match, as a factor of three levels ... where 'H', 'D' and 'A' represent a home win, a draw and an away win, respectively

played A logical to show if the game has yet been played

year_end The four figure integer value of the year in which the season ends

`number_seasons` This is the integer required for the number of previous seasons to include. It defaults to zero.

`lookback_rounds`

This is the integer number of rounds of fixtures to use in a model (so 38L would represent a whole season). Defaults to half a season (that is, 19L).

number_simulations
: This is the integer number of simulations to use for each game in the remaining season. Defaults to 25000L.

value_seed
: This is the integer seed to use for the random numbers in the simulation. Defaults to 120519L (which, IMHO, was a great footballing day).

Examples

```
## Not run:
run_simulations(
  results_thisSeason = example_thisSeason,
  number_seasons = 1L,
  lookback_rounds = 78L,
  number_simulations = 25000L,
  value_seed = 120519L
  )

## End(Not run)
```

---

schedule_thisSeason *A dataset containing the schedule of the Premier League this.*

Description

A dataset containing the schedule of the Premier League this.

Usage

```
schedule_thisSeason
```

Format

A data frame with many rows (one for each game this season) and 6 variables:

number_match A character of the index for the game in question

number_match_integer The integer version of `number_match`

matchday The date on which the game occurred

homeTeam The `shortName` of the team that played at home in the match

awayTeam The `shortName` of the team that played away in the match

year_end The calendar year in which the season ended

Source

https://github.com/openfootball/football.json

simulate_games *Assign the model parameters for unplayed games in this Premier League season.*

Description

This function takes the relevant parameters from the model and assigns them to the approapriate teams in each remaining game of the season.

Usage

```
simulate_games(
  data_model_parameters_unplayed,
  value_number_sims = 50000,
  value_seed = 120519L
)
```

Arguments

data_model_parameters_unplayed	These are the model parameters assigned to unplayed games in this Premier League season, as generated by model_parameters_unplayed.
value_number_sims	This is the number of simulations to use for each game in the remaining season. Defaults to 50,000.
value_seed	This is the seed to use for the random numbers in the simulation. Defaults to 120519 (which, IMHO, was a great footballing day).

Examples

```
## Not run:
simulate_games(
  data_model_parameters_unplayed = data_model_parameters_unplayed,
  value_number_sims = 1000000,
  value_seed = 120519L
  )

## End(Not run)
```

simulate_outcomes *Show the likelihoods of all possible standings for all clubs over this Premier League season.*

Description

This function takes the relevant data from the simulations and the current table to the likelihoods of all possible standings for all clubs over this Premier League season.

Usage

```
simulate_outcomes(data_standings_simulations, value_number_sims)
```

Arguments

data_standings_simulations	These are the standings for each club in each scenario run, as generated by simulate_standings.
value_number_sims	This is the number of simulations to use for each game in the remaining season. Defaults to 50,000.

Examples

```
## Not run:
simulate_outcomes(
  data_standings_simulations = data_standings_simulations,
  value_number_sims = 1000000
  )

## End(Not run)
```

simulate_standings *Simulate the standings over this Premier League season.*

Description

This function takes the relevant data from the simulations and finds the standings for each team at the end of the season.

Usage

```
simulate_standings(data_game_simulations, data_table_latest)
```

Arguments

`data_game_simulations`
: These are the outcome scenarios of the unplayed games in this Premier League season, as generated by `simulate_games`.

`data_table_latest`
: These are the latest standings generated by running `calc_table_current`.

Examples

```
## Not run:
simulate_standings(
  data_game_simulations = data_game_simulations,
  data_table_latest = data_table_latest
  )

## End(Not run)
```

`teams` *A dataset containing the teams in the latest Premier League season.*

Description

A dataset containing the teams in the latest Premier League season.

Usage

```
teams
```

Format

A data frame with 20 rows (one for each team) and 4 variables:

teamName The long name of the team in question, used in footballData

shortName The three-letter code of the team in question

midName The more-readable name of the team in question

openName The more-readable name of the team in question, used in openData

Source

https://www.premierleague.com/

Index