Table of Contents

[Exported Functions 2](#_Toc10459130)

[Basketball Reference Player Season Tables 2](#_Toc10459131)

[Function 2](#_Toc10459132)

[Arguments 2](#_Toc10459133)

[Value 2](#_Toc10459134)

[Function Walkthrough 2](#_Toc10459135)

[Metrics Leaders (Source: R/league\_leaders.R) 3](#_Toc10459136)

[Function 3](#_Toc10459137)

[Arguments 3](#_Toc10459138)

[Value 4](#_Toc10459139)

[Function Walkthrough 4](#_Toc10459140)

[Hidden and Support Functions 4](#_Toc10459141)

[.get\_season\_metric\_league\_leaders 4](#_Toc10459142)

[Function 4](#_Toc10459143)

[Arguments 4](#_Toc10459144)

[Value 4](#_Toc10459145)

[Function Walkthrough 4](#_Toc10459146)

[Generate\_Season\_Slug (Source: R/new\_ish.R) 5](#_Toc10459147)

[Value 5](#_Toc10459148)

[Function Walkthrough 5](#_Toc10459149)

[Resolve\_nba\_names (Source: R/new\_ish.R) 5](#_Toc10459150)

[Function 5](#_Toc10459151)

[Arguments 5](#_Toc10459152)

[Value 5](#_Toc10459153)

[Function Walkthrough 5](#_Toc10459154)

[Dictionary\_nba\_names 6](#_Toc10459155)

[Munge\_nba\_data 6](#_Toc10459156)

# Exported Functions

## Basketball Reference Player Season Tables

### Function

bref\_players\_stats(seasons = NULL, tables = [c](https://www.rdocumentation.org/packages/base/topics/c)("advanced", "totals"),

include\_all\_nba = F, only\_totals = TRUE, nest\_data = FALSE,

assign\_to\_environment = TRUE, widen\_data = TRUE, join\_data = TRUE,

return\_message = TRUE)

### Arguments

|  |  |
| --- | --- |
| **seasons** | vector of years 1951 to current season |
| **tables** | player table   * totals: Totals * per\_game: Per game * advanced: Advanced * per\_minute: Per 36 minutes * per\_poss: Per Possession |
| **include\_all\_nba** | if TRUE include all\_nba teams |
| **only\_totals** | if TRUE returns only a player's total statistics |
| **nest\_data** | if TRUE returns a nested data frame |
| **assign\_to\_environment** | if `TRUE` assigns to environment |
| **widen\_data** | if `TRUE` widens data |
| **join\_data** | if `TRUE` joins `tibbles` |
| **return\_message** | if TRUE returns a message |

### Value

a tibble

### Function Walkthrough

1. Create *tables* character vector and make lowercase
2. Create *.get\_data\_bref\_player\_seasons\_safe* using **.get\_data\_bref\_player\_seasons**. Wrap in **purrr::possibly**
   1. purrr::possibly(.get\_data\_bref\_player\_seasons, tibble())
3. Create *all data* DF
   1. Tables

## Metrics Leaders (Source: [R/league\_leaders.R](https://github.com/abresler/nbastatR/blob/master/R/league_leaders.R))

### Function

metrics\_leaders(seasons = 2017:2018, metric = "pts",

season\_types = "Regular Season", modes = [c](https://www.rdocumentation.org/packages/base/topics/c)("PerGame", "Totals"),

return\_message = TRUE, nest\_data = FALSE)

### Arguments

|  |  |
| --- | --- |
| **seasons** | vector of seasons |
| **metric** | name of metric to sort on   * pts * min * ast * treb * oreb * dreb * stl * tov * fg3 * fga * fgm * fta * ftm * pf * eff |
| **season\_types** | type of season   * Regular Season * Playoffs * Pre Season |
| **modes** | mode of search   * Totals * Per Game * Per48 |
| **return\_message** | if TRUE returns a message |
| **nest\_data** | if TRUE returns a nested data frame |

### Value

a tibble

### Function Walkthrough

1. Create *input\_df:* a data frame from all combinations of the supplied vectors or factors. Input is Argument Inputs (Season, Metric, Season Type, Modes and Return Message)
2. Create *.get\_season\_metrics\_leage\_leaders\_safe* using [**.get\_season\_metric\_league\_leaders**.](#_.get_season_metric_league_leaders_1)  Wrap in **purrr::possibly**
   1. purrr::possibly(.get\_season\_metric\_league\_leaders, tibble())
3. Create *all\_data* DF
   1. Get number of rows of *input\_df*
   2. Future Map Dfr of *input\_data* into Function
   3. Create *df\_row* and slice data to get the arguments by row (season, metric, etc.)
   4. Run *.get\_season\_metric\_leaague\_leaders\_safe*using arguments from columns of input\_df
4. Nest\_data if Yes

# Hidden and Support Functions

## .get\_data\_bref\_player\_seasons

### Function

(seasons = 2019,

table = "advanced",

only\_totals = TRUE,

return\_message = TRUE)

### Arguments

### Value

a tibble

### Function Walkthrough

1. Create *current season* using **.get\_current\_season**
2. Create *urls* using **.generate\_years\_urls**
3. Create *.parse\_player\_season\_safe*

## .generate\_year\_urls

### Function

## memoise::memoise(function(table = "per\_game",

## seasons = 1951:2017)

### Function Walkthrough

1. Create *tables* vector that includes list of table choices
2. Error message if Table Choice is not in list
3. Create *urls* list using seasons and table
4. Reduce to character vector with purrr::reduce(paste0)
5. Return urls

## .parse\_player\_season

### Arguments

url = <http://www.basketball-reference.com/leagues/NBA_1997_per_game.html>

### Function Walkthrough

1. *page object* created using URL
   1. Read\_html
2. Create url\_df DF
   1. Parse\_url
   2. Flatten into DF
3. url\_path
   1. Get path using replace all on “.html” or “leagues/NBA”
4. Get *year\_season\_end*
   1. str\_split url\_path to split at underscores
   2. Flatten Characters
   3. Take 1st object
   4. Parse as numbers
5. Create *name\_slug (“PerGame”)*
6. Create *id\_season* (“1996-97”)
7. Extract *players*
8. Extract *player\_id*
9. Clean *player\_ids*
10. Create tibble *df\_players*
    1. slugPlayerBREF = player\_ids, namePlayer = players
    2. Distinct
    3. Add numberPlayer column
11. Create df of html table
12. Work on *df*
    1. Mutate all columns except Tm, Player and Pos as numeric
    2. Remove Ranks that are NA
13. *Df\_names* use ***get\_bref\_name\_df()*** to get tibble of bref\_names
14. Create *bref\_names* which is a character vector of the bref\_names
15. Create *actual\_names* object
    1. Function
16. *DF*
    1. use Purrr to set names
    2. isHOF Player column based on \* in name
    3. namePlayer removes star
17. DF
    1. Join df\_players DF
    2. Distinct to remove duplicate rows
    3. Suppress Messages
    4. Add slug\_season = id\_season
    5. yearSeason – year\_season\_end
    6. urlData = url
    7. Select(SlugSeason, yearSeason, Player, everything
    8. Mutate\_at pct to decimals
    9. Add TypeData column

## .get\_season\_metric\_league\_leaders

### Function

(season = 2018, metric = "pts",

season\_types = "Regular Season", modes = “Per48”

return\_message = TRUE)

### Arguments

### Value

a tibble

### Function Walkthrough

1. Slug\_season <- Use Season from input and [**generate\_season\_slug**](#_Generate_Season_Slug_(Source:_R/new)
2. Error Message if Mode not in Mode Slugs
3. Scope\_slug = “s” (not sure why this is used)
4. Create json\_url using glue and arguments from input
5. Glue Acquiring Return Message
6. Take JSON URL and curl. Use From JSON and Simplify Data Frame
7. Actual Names <- Take Column Name from Input and run **resolve\_nba\_names** as character vector
8. Df\_params <- Create tibble of JSON URL Parameters
9. Data <-
   1. Create tibble from values of JSON
   2. Set Names using Actual Names
   3. **Munge NBA Data**
   4. Add numberTable column = 1 so that tables have common variable to join on
   5. Return Data
   6. Left\_join Df\_params Tibble
   7. Use select one of Actual params, then everything
   8. Use Select to remove Number Table and idLeague columns
   9. Suppress Messages

## Generate\_Season\_Slug (Source: [R/new\_ish.R](https://github.com/abresler/nbastatR/blob/master/R/league_leaders.R))

Using Season = 2018.

### Value

"2017-18"

### Function Walkthrough

1. Take Season defined and create character vector (“Season – 1” – “Season”). Format: YYYY - YY

## Resolve\_nba\_names (Source: R/new\_ish.R)

### Function

### Arguments

Takes json\_names as input

### Value

### Function Walkthrough

1. Create Tibble df\_nba\_names using [**dictionary\_nba\_names**](#_Dictionary_nba_names)
2. Take JSON\_Names as input and map characters using the NBA Name column name as the input
3. Run error if the name is not available
4. Return Actual Name

## Dictionary\_nba\_names

## Munge\_nba\_data