

Package: recombinator (via r-universe)

October 31, 2024

Type Package
Title Recombine Nested Lists to Dataframes
Description Turns nested lists into data.frames in an orderly manner.
Version 1.0.1
Maintainer Peter Hurford <peter@peterhurford.com>
License MIT + file LICENSE
LazyData true
Depends R (>= 3.0.1)
Imports stats, crayon
Suggests testthat
RoxygenNote 6.0.1
NeedsCompilation no
Author Peter Hurford [aut, cre], Robert Krzyzanowski [aut]
Date/Publication 2019-01-14 22:50:03 UTC
Repository <https://peterhurford.r-universe.dev>
RemoteUrl <https://github.com/cran/recombinator>
RemoteRef HEAD
RemoteSha 407bb0f6bc7a7f36f769bd9c7e15eacf64a58bbf

Contents

has_names	2
heterogeneous_recombinator	2
homogeneous_recombinator	3
is_heterogeneous	4
is_homogeneous	4
recombinator	5
warn_on_nonstandard_names	6

Index	7
--------------	----------

has_names	<i>Checks if a list has names.</i>
-----------	------------------------------------

Description

Checks if a list has names.

Usage

```
has_names(dat)
```

Arguments

dat	list. The list to verify.
-----	---------------------------

Value

boolean. TRUE if the list is named, FALSE otherwise.

heterogeneous_recombinator	<i>Process heterogeneous batch data.</i>
----------------------------	--

Description

This function turns a list of data obtained from the Avant API in heterogeneous format into a `data.frame`. Here, heterogeneous refers to a list of lists with each element being of possibly different size, but a complete named list of the data for that row.

Usage

```
heterogeneous_recombinator(dat, id = "id")
```

Arguments

dat	list. The list of lists to process. Each row is a named list with the names being variable names and the values being respective variable values.
id	character. Primary key, by default "id".

Details

For example, `list(list(variable_one = 1, variable_two = 'a'), list(variable_one = 2, variable_three = 1))` refers to a data set with three variables with two rows, the first variable having `c(1,2)`, the second `c('a', NA)`, and the third `c(NA, 1)`.

If the list of lists is not formatted in this way, the function performs no error handling and will likely return a malformed `data.frame`.

Value

the formatted `data.frame`

Examples

```
pre_dataframe <-  
  list(list(variable_one = 1, variable_two = 'a'),  
        list(variable_one = 2, variable_three = 1))  
df <- heterogeneous_recombinator(pre_dataframe)  
# 3 by 2 dataframe w/ c(1,2), c('a', NA), c(NA, 1) in the columns, respectively.
```

homogeneous_recombinator

Process homogeneous batch data.

Description

This function turns a list of data obtained from the Avant API in homogeneous format into a `data.frame`. Here, homogeneous refers to a list of lists with the first element of the list being a character vector of column names, and subsequent list elements being lists of values in the correct order and of the same length as the names vector.

Usage

```
homogeneous_recombinator(dat, id = "id")
```

Arguments

<code>dat</code>	list. The list of lists to process. The first list element is a character vector of variable names, and subsequent elements are lists of variable values ordered by these variable names.
<code>id</code>	character. Primary key, by default "id".

Details

For example, `list(c('variable_one', 'variable_two'), list(1, 'a'), list(2, 'b'))` refers to a data set with two variables with two rows, the first variable having `c(1, 2)` and the latter having `'a', 'b'`.

If the list of lists is not formatted in this way, the function performs no error handling and will likely return a malformed `data.frame`.

Value

the formatted `data.frame`

Examples

```
pre_dataframe <- list(c('variable_one', 'variable_two'), list(1, 'a'), list(2, 'b'))
df <- homogeneous_recombinator(pre_dataframe)
# 2 by 2 dataframe w/ c(1,2), c('a','b') in the columns, respectively.
```

is_heterogeneous	<i>Is this heterogeneous data?</i>
------------------	------------------------------------

Description

Is this heterogeneous data?

Usage

```
is_heterogeneous(dat)
```

Arguments

dat list. The list to verify.

Value

boolean. TRUE if the list is heterogeneous, FALSE otherwise.

is_homogeneous	<i>Is this homogeneous data?</i>
----------------	----------------------------------

Description

Is this homogeneous data?

Usage

```
is_homogeneous(dat)
```

Arguments

dat list. The list to verify.

Value

boolean. TRUE if the list is heterogeneous, FALSE otherwise.

recombinator	<i>Turn nested lists into data.frames.</i>
--------------	--

Description

A mini-utility package for turning nested lists into `data.frames`.

A recombinator attempts to convert a depth 2 nested list into a `data.frame`.

Usage

```
recombinator(dat, id = "id")
```

Arguments

<code>dat</code>	list. The list of lists to process. It can be in homogeneous or heterogeneous format (see the description).
<code>id</code>	character. Primary key, by default "id".

Details

There are two supported formats.

1. Homogeneous lists A list where the first list element is a character vector giving the names of the `data.frame`, and the subsequent list elements themselves lists of values.
2. Heterogeneous lists A list where each element is a named list of values. In this format, `plyr::rbind` will be used to take the union of all names and impute the ones missing with NA values.

Value

the converted `data.frame`. If not a list, no changes will be performed.

Note

A warning will be issued if non-standard names (i.e. those containing more than alphanumeric, underscore, and period characters) are used.

warn_on_nonstandard_names

Warn if names will be changed when converting to a data.frame.

Description

Warn if names will be changed when converting to a data.frame.

Usage

```
warn_on_nonstandard_names(data)
```

Arguments

data list. A list to convert to a data.frame.

Value

Nothing, but a warning if the names will be mangled due to R's [make.names](#).

Index

[has_names](#), [2](#)
[heterogeneous_recombinator](#), [2](#)
[homogeneous_recombinator](#), [3](#)

[is_heterogeneous](#), [4](#)
[is_homogeneous](#), [4](#)

[make.names](#), [6](#)

[recombinator](#), [5](#)
[recombinator-package \(recombinator\)](#), [5](#)

[warn_on_nonstandard_names](#), [6](#)