Package ‘loopr’

August 29, 2016

Type Package

Title Uses an Archive to Amend Previous Stages of a Pipe using Current Output

Version 1.0.1

Date 2015-05-27

Author Brandon Taylor

Maintainer Brandon Taylor <Brandon.Taylor221@gmail.com>

Description Remedies a common problem in piping: not having access to intermediate outputs of the pipe. Within a ``loop'', a piping intermediate is stored in a stack archive, data is processed, and then both the stored intermediate and the current output are reintegrated using an ``ending'' function. Two special ending functions are provided: amend and insert. However, any ending function can be specified, including merge functions, join functions, setNames(), etc. This framework allows the following work-flow: focus on a particular aspect or section of a data set, conduct specific operations, and then reintegrate changes into the whole.

Depends R (>= 3.1.3)

Imports plyr (>= 1.8.1), dplyr (>= 0.4.1), magrittr (>= 1.5), lazyeval (>= 0.1.10), R6 (>= 2.0.1)

License MIT + file LICENSE

LazyData true

Suggests knitr (>= 1.9)

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2015-05-28 01:07:44

R topics documented:

amend *Amend a dataframe with new information*

Description

full_join two dataframes. If there are matching columns, amend each data column with the corresponding amendData column using amendColumns.

Usage

```
amend(data, amendData, by = NULL, suffix = "toFix")
```

Arguments

data	A data frame
amendData	A data frame
by	A quoted vector of column names to join by. If set to NULL or unspecified, will default to the grouping columns in data
suffix	A suffix used internally. No existing column names should use this suffix.

Value

An amended tbl_df

amendColumns *Amend variables with new information*

Description

Replace all non-NA values in one set of columns with values from another matching set

Usage

```
amendColumns(data, originalNames, amendNames)
```

Arguments

`data`	A data frame
`originalNames`	A vector of column names with out-of-date information
`amendNames`	A vector of column names with amended information. They will be removed at the end of processing.

Value

An amended `tbl_df`

`fillColumns` *Fill variables with new information*

Description

Replace all NA values in one set of columns with values from another matching set

Usage

```
fillColumns(data, originalNames, fillNames)
```

Arguments

`data`	A data frame
`originalNames`	A vector of column names with out-of-date information
`fillNames`	A vector of column names with new information. They will be removed at the end of processing.

Value

A `tbl_df`

`insert` *Insert new information into a dataframe.*

Description

`anti_join` data with insertData, then `bind_cols` of insertData, then arrange by by variables.

Usage

```
insert(data, insertData, by)
```

Arguments

`data` A data frame

`insertData` A data frame

`by` A quoted vector of column names to join by.

Value

An inserted `tbl_df`

`loopClass` *An implementation of an loop*

Description

An implementation of an loop

Usage

```
loopClass
```

Format

An `R6Class` generator object

Inherits

`stackClass`

Methods

`begin(item, name = "")` Alias for `stackClass$push`

`end(endData, FUN, ...)` Will return `FUN(stackClass$pop, endData, ...)`

`cross(crossData, FUN, ...)` Will return `FUN(crossData, stackClass$pop, ...)`

stackClass *An implementation of a stack*

Description

An implementation of a stack

Usage

```
stackClass
```

Format

An `R6Class` generator object

Fields

`stack` A list which can directly accessed or accessed through the functions below. The bottom (original) item, set to NULL, should not be modified.

Methods

`push(item, name = "")` Append an `item` to the `stack` with an optional `name` and return the item

Active Bindings

`height` This will return the length of the `stack`

`pop` This will remove the last (most recent) item from the `stack` and return it

`pop` This will return, but not remove, the last (most recent) item from the `stack`

Index