Package ‘lplyr’

January 25, 2017

Type Package

Title 'dplyr' Verbs for Lists and Other Verbs for Data Frames

Version 0.1.6

Date 2017-01-23

Description Provides 'dplyr' verbs for lists and other useful verbs for manipulation of data frames. In particular, it includes a mutate_which() function that mutates columns for a specific subset of rows defined by a condition, and fuse() which is a more flexible version of 'tidyr' unite() function.

License MIT + file LICENSE

LazyData TRUE

Depends R (>= 3.1.3)

Imports bazar, dplyr, lazyeval, magrittr

Suggests knitr, rmarkdown, testthat, tidyr

VignetteBuilder knitr

URL https://github.com/paulponcet/lplyr

BugReports https://github.com/paulponcet/lplyr/issues

RoxygenNote 5.0.1

NeedsCompilation no

Author Paul Poncet [aut, cre]

Maintainer Paul Poncet <paulponcet@yahoo.fr>

Repository CRAN

Date/Publication 2017-01-25 10:23:19

R topics documented:

fuse *Fuse multiple columns into one*

Description

fuse is a more flexible version of unite from package **tidyr**.

Usage

```
fuse(.data, col, ..., fun = concat0, remove = TRUE)

fuse_(.data, col, from, fun = concat0, remove = TRUE)

## S3 method for class 'data.frame'
fuse_(.data, col, from, fun = concat0, remove = TRUE)

## S3 method for class 'tbl_df'
fuse_(.data, col, from, fun = concat0, remove = TRUE)
```

Arguments

.data	A tbl or data.frame
col	character. (Bare) name of column to add
...	Specification of columns to fuse.
fun	function. The function to be applied (concat0 by default).
remove	logical. If TRUE (the default), remove input columns from output data frame.
from	character. A vector of the names of columns to fuse.

Note

This function has been inspired by the issue raised at https://github.com/tidyverse/tidyr/issues/203.

See Also

unite and unite_ from package **tidyr**;

concat0 from package **bazar**.

Examples

```
df <- data.frame(x = c(NA, "a", NA),
                 y = c("b", NA, NA))
fuse(df, "z", x, y)

# To be compared with:
tidyr::unite(df, "z", x, y, sep = "")
# The same
fuse(df, "z", x, y, fun = function(x) concat0(x, na.rm = FALSE))
```

`mutate_.list` *Dplyr verbs for lists and pairlists*

Description

We add methods for the verbs `mutate`, `rename`, `select`, and `transmute`.

Usage

```
## S3 method for class 'list'
mutate_(.data, ..., .dots)

## S3 method for class 'pairlist'
mutate_(.data, ..., .dots)

## S3 method for class 'list'
rename_(.data, ..., .dots)

## S3 method for class 'pairlist'
rename_(.data, ..., .dots)

## S3 method for class 'list'
select_(.data, ..., .dots)

## S3 method for class 'pairlist'
select_(.data, ..., .dots)

## S3 method for class 'list'
transmute_(.data, ..., .dots)

## S3 method for class 'pairlist'
transmute_(.data, ..., .dots)
```

Arguments

`.data`	A list or pairlist.
`...`	Comma separated list of unquoted expressions.
`.dots`	Used to work around non-standard evaluation.

Value

A list or a pairlist.

See Also

`mutate`, `rename`, `select`, `transmute` from package **dplyr**.

Examples

```
xs <- list(x1 = 1:3,
           x2 = 2:5,
           x3 = list("alpha", c("beta", "gamma")))

# Non-standard evaluation
mutate(xs, x4 = 4)
rename(xs, x0 = x1)
select(xs, -x3)
transmute(xs, x5 = 5)

# Standard evaluation
mutate_(xs, x4 = ~ 4)
rename_(xs, x0 = ~ x1)
select_(xs, ~ (-x3))
transmute_(xs, x5 = ~ 5)
```

`mutate_which` *Add new variables or modify existing ones on a subset of the data*

Description

The functions `mutate_which` and `transmute_which` are similar to `mutate` and `transmute` from package **dplyr**, except that they work only on a subset of `.data`, this subset being defined by the `.condition`.

The functions `mutate_which_` and `transmute_which_` are standard evaluation versions, similar to `mutate_` and `transmute_`.

Usage

```
mutate_which(.data, .condition, ...)

mutate_which_(.data, .condition, ..., .dots)

transmute_which(.data, .condition, ...)

transmute_which_(.data, .condition, ..., .dots)
```

Arguments

`.data` A tbl or data.frame.

`.condition` A condition defining the subset on which the mutate or transmute operation applies. New variables are initialized to `NA`.

`...` Name-value pairs of expressions. Use `NULL` to drop a variable.

`.dots` Used to work around non-standard evaluation.

Value

A tbl or a data frame, depending on the class of `.data`.

Author(s)

Adapted from G. Grothendieck on StackOverflow, see `http://stackoverflow.com/a/34096575`.

See Also

`mutate`, `mutate_`, `transmute`, `transmute_` from package **dplyr**.

Examples

```
df <- mtcars[1:10,]

# Non-standard evaluation
mutate_which(df, gear==4, carb = 100)
transmute_which(df, gear==4, carb = 100)

# Standard evaluation
mutate_which_(df, ~ gear==4, carb = ~ 100)
transmute_which_(df, ~ gear==4, carb = ~ 100)
```

pull *Column selection*

Description

The function pull selects a column in a data frame and transforms it into a vector. This is useful to use it in combination with **magrittr**'s pipe operator and **dplyr**'s verbs.

Usage

```
pull(.data, j)

pull_(.data, j)

## S3 method for class 'data.frame'
pull_(.data, j)

## S3 method for class 'matrix'
pull_(.data, j)

## S3 method for class 'list'
pull_(.data, j)
```

Arguments

.data	A tbl.
j	integer. The column to be extracted.

Value

A vector of length nrow(.data)

Author(s)

Adapted from Tommy O' Dell, see http://stackoverflow.com/a/24730843/3902976 on StackOverflow.

Examples

```
library(dplyr)
mtcars[["mpg"]]
mtcars %>% pull(mpg)

# more convenient than (mtcars %>% filter(mpg > 20))[[3L]]
mtcars %>%
  filter(mpg > 20) %>%
  pull(3)
```

Index