

Package ‘tbltools’

January 7, 2019

Title Tools for Working with Tibbles

Version 0.0.3

Description Tools for working with tibbles, or data.frame-like objects generated by the 'tibble' package <<https://tibble.tidyverse.org/>>. Functions assist in converting objects to tibbles and creating frequency tables.

License GPL-3

Encoding UTF-8

LazyData true

Imports tfse, tibble, rlang, tidyselect

Suggests testthat, covr

RoxygenNote 6.1.1

URL <https://github.com/mkearney/tbltools>

BugReports <https://github.com/mkearney/tbltools/issues>

NeedsCompilation no

Author Michael W. Kearney [aut, cre] (<<https://orcid.org/0000-0002-0730-4694>>)

Maintainer Michael W. Kearney <kearneymw@missouri.edu>

Repository CRAN

Date/Publication 2019-01-07 19:00:09 UTC

R topics documented:

arrange_rows	2
as_tbl	2
env_tbls	3
filter_rows	4
repos_back	4
repos_front	5
tabsort	6
tidyselector	7

Index	8
--------------	----------

arrange_rows	<i>Arrange rows</i>
--------------	---------------------

Description

Arrange rows via descending or ascending column values

Usage

```
arrange_rows(.data, ..., desc = TRUE)
```

Arguments

.data	data frame
...	The name of up to two column names on which to arrange the rows
desc	Logical indicating whether to arrange by descending (default) or ascending values.

Value

Rearranged data frame

as_tbl	<i>as_tbl</i>
--------	---------------

Description

See `tibble::as_tibble` for details.

Usage

```
as_tbl(x, row_names = FALSE)
```

Arguments

x	Data
row_names	Logical indicating whether to convert non-null row names into the first column.

Examples

```
## data with row names
d <- data.frame(x = rnorm(5), y = rnorm(5), row.names = letters[1:5])

## convert to tibble
as_tbl(d)

## convert to tibble and create row_names variable
as_tbl(d, row_names = TRUE)
```

`env_tbls`*Convert all data frames in environment into tibbles*

Description

Converts data frames found in a given environment into tibbles (`tbl_df`)

Usage

```
env_tbls(env = globalenv(), row_names = TRUE)
```

Arguments

<code>env</code>	Name of environment from which data frames should be converted to tibbles. Defaults to global environment.
<code>row_names</code>	Logical indicating whether to create a <code>row_names</code> variable if non-auto row names are found.

Value

The function will print messages when converting occurs and it will print a final completion message, but otherwise returns nothing.

Examples

```
## data with row names
d <- data.frame(x = rnorm(5), y = rnorm(5), row.names = letters[1:5])

## convert data frames in global environment to tibbles
env_tbls()
```

 filter_rows

Filter rows

Description

Filter rows via integer/numeric position or logical vector

Usage

```
filter_rows(.data, ...)
```

Arguments

.data Data frame or two dimensional array

... This should evaluate and reduce down to a numeric (row number) or logical vector. Row numbers higher than what exists in x will be ignored. Any numeric vector must be either all positive or all negative.

Value

Sliced/filtered data frame

 repos_back

move vars to back

Description

move vars to back

Usage

```
repos_back(data, ...)
```

Arguments

data data frame

... columns to move to back

Value

Reordered data frame.

Examples

```
## data with row names
d <- data.frame(x = rnorm(5), y = rnorm(5), row.names = letters[1:5])

## move x to back
repos_back(d, x)
```

repos_front	<i>move vars to front</i>
-------------	---------------------------

Description

move vars to front

Usage

```
repos_front(data, ...)
```

Arguments

data	data frame
...	columns to move to front

Value

Reordered data frame.

Examples

```
## data with row names
d <- data.frame(x = rnorm(5), y = rnorm(5), row.names = letters[1:5])

## move y to front
repos_front(d, y)
```

tabsort	<i>tabsort</i>
---------	----------------

Description

Returns a sorted (descending) frequency tbl

Usage

```
tabsort(.data, ..., prop = TRUE, na_omit = TRUE, sort = TRUE)
```

```
ntbl(.data, ...)
```

Arguments

<code>.data</code>	Data
<code>...</code>	Unquoted column names of variables to include in table. Default is to use all columns.
<code>prop</code>	Logical indicating whether to include a proportion of total obs column.
<code>na_omit</code>	Logical indicating whether to exclude missing. If all responses are missing, a missing value is used as the single category.
<code>sort</code>	Logical indicating whether to sort the returned object.

Value

Frequency tbl

Examples

```
## generate example data
x <- sample(letters[1:4], 200, replace = TRUE)
y <- sample(letters[5:8], 200, replace = TRUE)

## count and sort frequencies for each vector
tabsort(x)
tabsort(y)

## combine x and y into data frame
dat <- data.frame(x, y)

## select columns and create freq table
tabsort(dat, x)
tabsort(dat, x, y)
```

tidyselector	<i>tidyselector</i>
--------------	---------------------

Description

Select columns using tidy eval

Usage

```
tidyselector(data, ...)
```

Arguments

data	data frame
...	vars to select

Value

Data with selected columns.

Examples

```
## data with row names  
d <- data.frame(x = rnorm(5), y = rnorm(5), row.names = letters[1:5])  
  
## select only x  
tidyselector(d, x)
```

Index

`arrange_rows`, 2

`as_tbl`, 2

`as_tibble`, 2

`env_tbls`, 3

`filter_rows`, 4

`ntbl (tabsort)`, 6

`repos_back`, 4

`repos_front`, 5

`tabsort`, 6

`tidyselector`, 7