Package 'wrangle'

March 29, 2024

Type Package

Title A Systematic Data Wrangling Idiom

Version 0.6.4

Author Tim Bergsma

Maintainer Tim Bergsma <bergsmat@gmail.com>

Description Supports systematic scrutiny, modification, and integration of data. The function status() counts rows that have missing values in grouping columns (returned by na()), have non-unique combinations of grouping columns (returned by dup()), and that are not locally sorted (returned by unsorted()). Functions enumerate() and itemize() give sorted unique combinations of columns, with or without occurrence counts, respectively. Function ignore() drops columns in x that are present in y, and informative() drops columns in x that are entirely NA; constant() returns values that are constant, given a key. Data that have defined unique combinations of grouping values behave more predictably during merge operations.

License GPL-3

BugReports https://github.com/bergsmat/wrangle/issues

Imports dplyr (>= 1.0.2), tidyr, magrittr, rlang

RoxygenNote 7.2.3

NeedsCompilation no

Repository CRAN

Date/Publication 2024-03-29 04:40:02 UTC

R topics documented:

`constant.data.frame` *Identify Constant Features of a Data Frame*

Description

Returns columns of a data.frame whose values do not vary within subsets defined by columns named in .... Defaults to groups(x) if none supplied, or all columns otherwise.

Usage

```
## S3 method for class 'data.frame'
constant(x, ...)
```

Arguments

x	object
...	optional grouping columns (named arguments are ignored)

Value

data.frame (should be same class as x)

See Also

Other constant: `constant()`

Examples

```
library(dplyr)
constant(Theoph)                    # data frame with 0 columns and 1 row
constant(Theoph, Subject)           # Subject Wt Dose Study
Theoph$Study <- 1
constant(Theoph)                    # Study
constant(Theoph, Study)             # Study
constant(Theoph, Study, Subject)    # Subject Wt Dose Study
```

```
Theoph <- group_by(Theoph, Subject)
constant(Theoph)                      # Subject Wt Dose Study
constant(Theoph, Study)               # Study
foo <- data.frame(x = 1)
foo <-  group_by(foo, x)
class(foo) <- c('foo', class(foo))
stopifnot(identical(class(foo), class(constant(foo))))
```

`dup.data.frame` *Show records with duplicate or duplicated values of grouping variables.*

Description

Shows records with duplicate or duplicated values of grouping variables.

Usage

```
## S3 method for class 'data.frame'
dup(x, ...)
```

Arguments

x	data.frame
...	optional grouping columns (named arguments are ignored)

Value

data.frame

See Also

Other dup: dup()

Examples

```
library(dplyr)
dupGroups(mtcars)
dupGroups(group_by(mtcars, mpg))
dup(group_by(mtcars, mpg))
```

`dupGroups.data.frame` *Index records with with duplicate or duplicated values of grouping variables.*

Description

Indexes records with with duplicate or duplicated values of grouping variables. If b follows a and and is the same, then b is a duplicate, a is duplicated, and both are shown.

Usage

```
## S3 method for class 'data.frame'
dupGroups(x, ...)
```

Arguments

`x`	data.frame
`...`	optional grouping columns (named arguments are ignored)

Value

grouped_df

logical

See Also

Other dupGroups: `dupGroups()`

`enumerate` *Count unique combinations of items in specified columns.*

Description

Counts unique combinations of items in specified columns (unquoted).

Usage

```
enumerate(x, ...)
```

Arguments

`x`	data.frame
`...`	columns to show

Value

grouped_df

See Also

Other util: `detect()`, `itemize()`, `static()`

Examples

```
enumerate(mtcars, cyl, gear, carb)
```

`ignore` *Drop columns in x that are present in y.*

Description

Drops columns in x that are present in y.

Usage

```
ignore(x, y, ...)
```

Arguments

x	data.frame
y	data.frame
...	ingored

Value

data.frame

`informative` *Drop columns in x that are entirely NA.*

Description

Drops columns in x that are entirely NA.

Usage

```
informative(x, ...)
```

Arguments

x	object of dispatch
...	passed

See Also

informative.data.frame

Other informative: informative.data.frame()

Examples

```
head(Theoph)
Theoph$Dose <- NA
head(informative(Theoph))
```

---

informative.data.frame

Drop columns in x that are entirely NA.

Description

Drops columns in x that are entirely NA.

Usage

```
## S3 method for class 'data.frame'
informative(x, ...)
```

Arguments

x	data.frame
...	ingored

Value

data.frame

See Also

Other informative: informative()

itemize *Show unique combinations of items in specified columns*

Description

Shows unique combinations of items in specified columns (unquoted).

Usage

```
itemize(x, ...)
```

Arguments

x	data.frame
...	columns to show

Value

grouped_df

See Also

Other util: `detect()`, `enumerate()`, `static()`

Examples

```
itemize(mtcars, cyl, gear, carb)
```

misplaced.data.frame *Index records whose relative positions would change if sorted.*

Description

Indexes records whose relative positions would change if sorted, i.e. records that would not have the same nearest neighbors (before and after). unsorted() returns the records corresponding to this index.

Usage

```
## S3 method for class 'data.frame'
misplaced(x, ...)
```

Arguments

x	data.frame
...	optional grouping columns (named arguments are ignored)

Value

logical with length nrow(x)

See Also

na dup

Other unsorted: misplaced(), unsorted.data.frame(), unsorted()

`na.data.frame` *Show records with NA values of grouping variables.*

Description

Shows records with NA values of grouping variables.

Usage

```
## S3 method for class 'data.frame'
na(x, ...)
```

Arguments

x	data.frame
...	optional grouping columns (named arguments are ignored)

Value

data.frame

See Also

Other na: na()

`naGroups.data.frame` *Index records with NA values of grouping variables.*

Description

Indexes records with NA values of grouping variables.

Usage

```
## S3 method for class 'data.frame'
naGroups(x, ...)
```

Arguments

x	data.frame
...	optional grouping columns (named arguments are ignored)

Value

logical

See Also

Other naGroups: naGroups()

---

safe_join.data.frame *Join Data Frames Safely*

Description

Joins data frames safely. I.e., a left join that cannot alter row order or number. Supports the case where you only intend to augment existing rows with additional columns and are expecting singular matches. Gives an error if row order or number would have been altered by a left join.

Usage

```
## S3 method for class 'data.frame'
safe_join(x, y, ...)
```

Arguments

x	data.frame
y	data.frame
...	passed to dplyr::left_join

See Also

Other safe_join: safe_join()

Examples

```
library(magrittr)
x <- data.frame(code = c('a','b','c'), value = c(1:3))
y <- data.frame(code = c('a','b','c'), roman = c('I','II','III'))
x %>% safe_join(y)
try(
x %>% safe_join(rbind(y,y))
)
```

sort.grouped_df *Arrange by groups.*

Description

As of 0.5, dplyr::arrange ignores groups. This function gives the old behavior as a method for generic base::sort. Borrowed from Ax3man at https://github.com/hadley/dplyr/issues/1206.

Usage

```
## S3 method for class 'grouped_df'
sort(x, decreasing = FALSE, ...)
```

Arguments

x	grouped_df
decreasing	logical (ignored)
...	further sort criteria

Value

grouped_df

Examples

```
library(dplyr)
head(sort(group_by(Theoph, Subject, Time)))
```

static *Find unique records for subset of columns with one unique value.*

Description

Finds unique records for subset of columns with one unique value.

Usage

```
static(x, ...)
```

Arguments

x	data.frame
...	ignored

Value

data.frame

See Also

Other util: `detect()`, `enumerate()`, `itemize()`

`status.data.frame` *Report status with respect to grouping variables.*

Description

Reports status with respect to grouping variables.

Usage

```
## S3 method for class 'data.frame'
status(x, ...)
```

Arguments

x	data.frame
...	optional grouping columns (named arguments are ignored)

Value

returns x invisibly (as originally grouped)

See Also

`na` `dup` `unsorted` `informative` `ignore` `itemize` `enumerate` `sort.grouped_df`

Other status: `status()`

Examples

```
library(dplyr)
status(Theoph)
status(Theoph, Subject)
status(group_by(Theoph, Subject, Time))
```

unsorted.data.frame *Extract records whose relative positions would change if sorted.*

Description

Extracts records whose relative positions would change if sorted, i.e. records that would not have the same nearest neighbors (before and after). misplaced() returns the index that extracts these records.

Usage

```
## S3 method for class 'data.frame'
unsorted(x, ...)
```

Arguments

x	data.frame
...	optional grouping columns (named arguments are ignored)

Value

data.frame, possibly grouped_df

See Also

na dup

Other unsorted: misplaced.data.frame(), misplaced(), unsorted()

weak.data.frame *Show records with NA, duplicate or duplicated values of grouping variables.*

Description

Shows records with NA, duplicate or duplicated values of grouping variables.

Usage

```
## S3 method for class 'data.frame'
weak(x, ...)
```

Arguments

x	data.frame
...	optional grouping columns (named arguments are ignored)

Value

data.frame

See Also

Other weak: `weak()`

Index