

Package ‘wrangle’

January 3, 2022

Type Package

Title A Systematic Data Wrangling Idiom

Version 0.5.7

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

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

RoxygenNote 7.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2022-01-03 18:20:02 UTC

R topics documented:

<code>constant</code>	2
<code>constant.data.frame</code>	3
<code>constant.grouped_df</code>	3
<code>detect</code>	4
<code>dup</code>	5
<code>dup.grouped_df</code>	5
<code>dupGroups</code>	6
<code>dupGroups.grouped_df</code>	6

enumerate	7
ignore	7
informative	8
informative.data.frame	8
itemize	9
key	10
key.grouped_df	10
na	11
na.grouped_df	11
naGroups	12
naGroups.grouped_df	12
safe_join	13
safe_join.data.frame	13
sort.grouped_df	14
static	15
status	15
status.grouped_df	16
unsorted	16
unsorted.grouped_df	17
weak	17
weak.grouped_df	18

Index 19

constant	<i>Identify Constant Features of an Object</i>
----------	--

Description

Identifies constant features of an object. Generic, with methods for `data.frame` and `grouped_df`.

Usage

```
constant(x, ...)
```

Arguments

x	object
...	passed arguments

See Also

Other constant: [constant.data.frame\(\)](#), [constant.grouped_df\(\)](#)

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

Usage

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

Arguments

x	object
...	grouping columns

Value

data.frame

See Also

Other constant: [constant.grouped_df\(\)](#), [constant\(\)](#)

Examples

```
constant(Theoph)  
constant(Theoph, Subject) # Wt Dose  
Theoph$Study <- 1  
constant(Theoph)  
constant(Theoph, Study)  
constant(Theoph, Study, Subject)
```

constant.grouped_df *Identify Constant Features of a Grouped Data Frame*

Description

Returns columns of a grouped_df whose values do not vary within subsets defined by groups. If any grouping arguments (dots) are supplied, existing groups are over-ridden.

Usage

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

Arguments

x object
... grouping columns

Value

grouped data.frame

See Also

Other constant: [constant.data.frame\(\)](#), [constant\(\)](#)

detect *Sort column subsets.*

Description

Sort column subsets.

Usage

```
detect(x, ...)
```

Arguments

x data.frame
... columns to sort

Value

grouped_df

See Also

Other util: [enumerate\(\)](#), [itemize\(\)](#), [static\(\)](#)

dup	<i>Show duplicate or duplicated elements.</i>
-----	---

Description

Shows duplicate or duplicated elements.

Usage

```
dup(x, ...)
```

Arguments

x	object of dispatch
...	other arguments

See Also

[dup.grouped_df na weak unsorted](#)

Other dup: [dup.grouped_df\(\)](#)

dup.grouped_df	<i>Show records with duplicate or duplicated values of grouping variables.</i>
----------------	--

Description

Shows records with duplicate or duplicated values of grouping variables.

Usage

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

Arguments

x	data.frame
...	ignored

Value

grouped_df

See Also

Other dup: [dup\(\)](#)

Examples

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

dupGroups	<i>Calculate dupGroups.</i>
-----------	-----------------------------

Description

Calculates dupGroups.

Usage

```
dupGroups(x, ...)
```

Arguments

x	object of dispatch
...	other arguments

See Also

Other dupGroups: [dupGroups.grouped_df\(\)](#)

dupGroups.grouped_df	<i>Count records with with duplicate or duplicated values of grouping variables.</i>
----------------------	--

Description

Counts 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 'grouped_df'
dupGroups(x, ...)
```

Arguments

x	data.frame
...	ignored

Value

grouped_df

See Also

Other dupGroups: [dupGroups\(\)](#)

enumerate	<i>Count unique combinations of items in specified columns.</i>
-----------	---

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	<i>Drop columns in x that are present in y.</i>
--------	---

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	<i>Show unique combinations of items in specified columns</i>
---------	---

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)
```

key	<i>Fetch the key.</i>
-----	-----------------------

Description

Fetches the key of an object.

Usage

```
key(x, ...)
```

Arguments

x	object of dispatch
...	other arguments

See Also

Other key: [key.grouped_df\(\)](#)

key.grouped_df	<i>Fetch the key for a grouped_df as character vector</i>
----------------	---

Description

Fetches the key for a grouped_df as character vector

Usage

```
## S3 method for class 'grouped_df'
key(x, ...)
```

Arguments

x	data.frame
...	columns to show

Value

character

See Also

Other key: [key\(\)](#)

na	<i>Show na elements.</i>
----	--------------------------

Description

Shows na elements.

Usage

```
na(x, ...)
```

Arguments

x	object of dispatch
...	other arguments

See Also

[na.grouped_df](#) [dup](#) [weak](#) [unsorted](#)

Other na: [na.grouped_df\(\)](#)

na.grouped_df	<i>Show records with NA values of grouping variables.</i>
---------------	---

Description

Shows records with NA values of grouping variables.

Usage

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

Arguments

x	data.frame
...	ignored

Value

grouped_df

See Also

Other na: [na\(\)](#)

naGroups	<i>Calculate naGroups.</i>
----------	----------------------------

Description

Calculates naGroups.

Usage

```
naGroups(x, ...)
```

Arguments

x	object of dispatch
...	other arguments

See Also

Other naGroups: [naGroups.grouped_df\(\)](#)

naGroups.grouped_df	<i>Count records with NA values of grouping variables.</i>
---------------------	--

Description

Counts records with NA values of grouping variables.

Usage

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

Arguments

x	data.frame
...	ignored

Value

numeric

See Also

Other naGroups: [naGroups\(\)](#)

safe_join	<i>Join Data Safely</i>
-----------	-------------------------

Description

Joins data safely. Generic, with method for data.frame.

Usage

```
safe_join(x, ...)
```

Arguments

x	object of dispatch
...	arguments to methods

See Also

[safe_join.data.frame](#)

Other safe_join: [safe_join.data.frame\(\)](#)

Examples

```
example(safe_join.data.frame)
```

safe_join.data.frame	<i>Join Data Frames Safely</i>
----------------------	--------------------------------

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	<i>Arrange by groups.</i>
-----------------	---------------------------

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	<i>Find unique records for subset of columns with one unique value.</i>
--------	---

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	<i>Report status.</i>
--------	-----------------------

Description

Reports the status of an object.

Usage

```
status(x, ...)
```

Arguments

x	object of dispatch
...	other arguments

See Also

Other status: [status.grouped_df\(\)](#)

Examples

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

`status.grouped_df` *Report status with respect to grouping variables.*

Description

Reports status with respect to grouping variables.

Usage

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

Arguments

<code>x</code>	<code>data.frame</code>
<code>...</code>	<code>ignored</code>

Value

returns `x` invisibly

See Also

[na dup unsorted informative ignore itemize enumerate sort.grouped_df](#)
Other status: [status\(\)](#)

Examples

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

`unsorted` *Show unsorted elements.*

Description

Shows unsorted elements.

Usage

```
unsorted(x, ...)
```

Arguments

<code>x</code>	<code>object of dispatch</code>
<code>...</code>	<code>other arguments</code>

See Also[unsorted.grouped_df](#)Other unsorted: [unsorted.grouped_df\(\)](#)

`unsorted.grouped_df` *Find records whose relative positions would change if sorted.*

Description

Finds records whose relative positions would change if sorted, i.e. records that would not have the same nearest neighbors (before and after).

Usage

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

Arguments

<code>x</code>	<code>data.frame</code>
<code>...</code>	<code>ignored</code>

Value

`grouped_df`

See Also[na dup](#)Other unsorted: [unsorted\(\)](#)

`weak` *Show na, duplicate, or duplicated elements.*

Description

Shows na, duplicate, or duplicated elements.

Usage

```
weak(x, ...)
```

Arguments

x object of dispatch
... other arguments

See Also

[weak.grouped_df](#)

Other weak: [weak.grouped_df\(\)](#)

weak.grouped_df	<i>Show records with NA, duplicate or duplicated values of grouping variables.</i>
-----------------	--

Description

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

Usage

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

Arguments

x data.frame
... ignored

Value

grouped_df

See Also

Other weak: [weak\(\)](#)

Index

- * **constant**
 - constant, [2](#)
 - constant.data.frame, [3](#)
 - constant.grouped_df, [3](#)
 - * **dupGroups**
 - dupGroups, [6](#)
 - dupGroups.grouped_df, [6](#)
 - * **dup**
 - dup, [5](#)
 - dup.grouped_df, [5](#)
 - * **ignore**
 - ignore, [7](#)
 - * **informative**
 - informative, [8](#)
 - informative.data.frame, [8](#)
 - * **key**
 - key, [10](#)
 - key.grouped_df, [10](#)
 - * **naGroups**
 - naGroups, [12](#)
 - naGroups.grouped_df, [12](#)
 - * **na**
 - na, [11](#)
 - na.grouped_df, [11](#)
 - * **safe_join**
 - safe_join, [13](#)
 - safe_join.data.frame, [13](#)
 - * **sort**
 - sort.grouped_df, [14](#)
 - * **status**
 - status, [15](#)
 - status.grouped_df, [16](#)
 - * **unsorted**
 - unsorted, [16](#)
 - unsorted.grouped_df, [17](#)
 - * **util**
 - detect, [4](#)
 - enumerate, [7](#)
 - itemize, [9](#)
 - static, [15](#)
 - * **weak**
 - weak, [17](#)
 - weak.grouped_df, [18](#)
- constant, [2, 3, 4](#)
constant.data.frame, [2, 3, 4](#)
constant.grouped_df, [2, 3, 3](#)
- detect, [4, 7, 9, 15](#)
dup, [5, 5, 11, 16, 17](#)
dup.grouped_df, [5, 5](#)
dupGroups, [6, 7](#)
dupGroups.grouped_df, [6, 6](#)
- enumerate, [4, 7, 9, 15, 16](#)
- ignore, [7, 16](#)
informative, [8, 9, 16](#)
informative.data.frame, [8, 8](#)
itemize, [4, 7, 9, 15, 16](#)
- key, [10, 10](#)
key.grouped_df, [10, 10](#)
- na, [5, 11, 11, 16, 17](#)
na.grouped_df, [11, 11](#)
naGroups, [12, 12](#)
naGroups.grouped_df, [12, 12](#)
- safe_join, [13, 14](#)
safe_join.data.frame, [13, 13](#)
sort.grouped_df, [14, 16](#)
static, [4, 7, 9, 15](#)
status, [15, 16](#)
status.grouped_df, [15, 16](#)
- unsorted, [5, 11, 16, 16, 17](#)
unsorted.grouped_df, [17, 17](#)
- weak, [5, 11, 17, 18](#)
weak.grouped_df, [18, 18](#)
wrangle(status.grouped_df), [16](#)