

# Package ‘wrangle’

April 25, 2019

**Type** Package

**Title** A Systematic Data Wrangling Idiom

**Version** 0.5.2

**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

**LazyData** TRUE

**Imports** dplyr, tidyr, lazyeval, magrittr, rlang

**RoxygenNote** 6.0.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2019-04-25 05:00:03 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

dupGroups.grouped_df . . . . .	6
enumerate . . . . .	7
group_by_all . . . . .	7
ignore . . . . .	8
informative . . . . .	8
informative.data.frame . . . . .	9
itemize . . . . .	9
key . . . . .	10
key.grouped_df . . . . .	10
na . . . . .	11
na.grouped_df . . . . .	12
naGroups . . . . .	12
naGroups.grouped_df . . . . .	13
sort.grouped_df . . . . .	13
static . . . . .	14
status . . . . .	14
status.grouped_df . . . . .	15
unsorted . . . . .	16
unsorted.grouped_df . . . . .	16
weak . . . . .	17
weak.grouped_df . . . . .	17

**Index** **18**

---

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)
```

---

group_by_all	<i>Group by all columns.</i>
--------------	------------------------------

---

**Description**

Groups by all columns.

**Usage**

```
group_by_all(x, ...)
```

**Arguments**

x	data.frame
...	ignored

**Value**

grouped\_df

---

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	<i>Drop columns in x that are entirely NA.</i>
-------------	--

---

**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)
```

---

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                    *Show na elements.*

---

**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     *Count records with NA values of grouping variables.*

---

### 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](#)

---

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	<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**

x                    data.frame  
...                  ignored

**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**

x	object of dispatch
...	other arguments

**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**

x	data.frame
...	ignored

**Value**

grouped\_df

**See Also**

[na dup](#)

Other unsorted: [unsorted](#)



---

weak	<i>Show na, duplicate, or duplicated elements.</i>
------	--

---

**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, [2](#), [3](#), [4](#)  
constant.data.frame, [2](#), [3](#), [4](#)  
constant.grouped\_df, [2](#), [3](#), [3](#)

detect, [4](#), [7](#), [10](#), [14](#)  
dup, [5](#), [5](#), [11](#), [15](#), [16](#)  
dup.grouped\_df, [5](#), [5](#)  
dupGroups, [6](#), [7](#)  
dupGroups.grouped\_df, [6](#), [6](#)

enumerate, [4](#), [7](#), [10](#), [14](#), [15](#)

group\_by\_all, [7](#)

ignore, [8](#), [15](#)  
informative, [8](#), [9](#), [15](#)  
informative.data.frame, [8](#), [9](#)  
itemize, [4](#), [7](#), [9](#), [14](#), [15](#)

key, [10](#), [11](#)  
key.grouped\_df, [10](#), [10](#)

na, [5](#), [11](#), [12](#), [15](#), [16](#)  
na.grouped\_df, [11](#), [12](#)  
naGroups, [12](#), [13](#)  
naGroups.grouped\_df, [12](#), [13](#)

sort.grouped\_df, [13](#), [15](#)  
static, [4](#), [7](#), [10](#), [14](#)  
status, [14](#), [15](#)  
status.grouped\_df, [15](#), [15](#)

unsorted, [5](#), [11](#), [15](#), [16](#), [16](#)  
unsorted.grouped\_df, [16](#), [16](#)

weak, [5](#), [11](#), [17](#), [17](#)  
weak.grouped\_df, [17](#), [17](#)  
wrangle(status.grouped\_df), [15](#)