

# Package ‘wrangle’

April 14, 2017

**Type** Package

**Title** A Systematic Data Wrangling Idiom

**Version** 0.4

**Date** 2017-04-13

**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. 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

**RoxygenNote** 6.0.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2017-04-13 22:30:46 UTC

## R topics documented:

<code>detect</code>	2
<code>dup</code>	3
<code>dup.grouped_df</code>	3
<code>dupGroups</code>	4
<code>dupGroups.grouped_df</code>	4

enumerate . . . . .	5
group_by_all . . . . .	5
ignore . . . . .	6
informative . . . . .	6
informative.data.frame . . . . .	7
itemize . . . . .	7
key . . . . .	8
key.grouped_df . . . . .	8
na . . . . .	9
na.grouped_df . . . . .	9
naGroups . . . . .	10
naGroups.grouped_df . . . . .	10
sort.grouped_df . . . . .	11
static . . . . .	11
status . . . . .	12
status.grouped_df . . . . .	12
unsorted . . . . .	13
unsorted.grouped_df . . . . .	13
weak . . . . .	14
weak.grouped_df . . . . .	14

**Index** **15**

---

detect	<i>Sort column subsets.</i>
--------	-----------------------------

---

**Description**

Sort column subsets.

**Usage**

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

**Arguments**

x	data.frame
...	columns to sort

**Value**

grouped\_df

---

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](#)

---

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

**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

---

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

---

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

**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](#)

**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

---

`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

**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

---

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

---

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](#)

---

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

---

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

---

**Description**

Calculates naGroups.

**Usage**

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

**Arguments**

x	object of dispatch
...	other arguments

---

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

---

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

---

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

---

**Description**

Reports the status of an object.

**Usage**

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

**Arguments**

x	object of dispatch
...	other arguments

**Examples**

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

---

status.grouped_df	<i>Report status with respect to grouping variables.</i>
-------------------	--

---

**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](#)

**Examples**

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

---

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

---

**Description**

Shows unsorted elements.

**Usage**

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

**Arguments**

x	object of dispatch
...	other arguments

**See Also**

[unsorted.grouped\\_df](#)

---

unsorted.grouped_df	<i>Find records whose relative positions would change if sorted.</i>
---------------------	--

---

**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](#)


---

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](#)


---

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

# Index

detect, [2](#)  
dup, [3](#), [9](#), [12](#), [14](#)  
dup.grouped\_df, [3](#), [3](#)  
dupGroups, [4](#)  
dupGroups.grouped\_df, [4](#)  
  
enumerate, [5](#), [12](#)  
  
group\_by\_all, [5](#)  
  
ignore, [6](#), [12](#)  
informative, [6](#), [12](#)  
informative.data.frame, [6](#), [7](#)  
itemize, [7](#), [12](#)  
  
key, [8](#)  
key.grouped\_df, [8](#)  
  
na, [3](#), [9](#), [12](#), [14](#)  
na.grouped\_df, [9](#), [9](#)  
naGroups, [10](#)  
naGroups.grouped\_df, [10](#)  
  
sort.grouped\_df, [11](#), [12](#)  
static, [11](#)  
status, [12](#)  
status.grouped\_df, [12](#)  
  
unsorted, [3](#), [9](#), [12](#), [13](#)  
unsorted.grouped\_df, [13](#), [13](#)  
  
weak, [3](#), [9](#), [14](#)  
weak.grouped\_df, [14](#), [14](#)  
wrangle(status.grouped\_df), [12](#)