

# Package ‘tidymargins’

July 23, 2019

**Type** Package

**Title** Tidy Marginal Functions

**Version** 0.1.0

**Maintainer** Andrew Redd <Andrew.Redd@hsc.utah.edu>

**Description** Provides the ability to easily perform functions over all margins of a grouped data frame.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Language** en-US

**LazyData** true

**Imports** assertthat, dplyr, forcats, magrittr, pkgcond, purrr, rlang, tidy, tidyselect, utils

**Suggests** covr, testthat

**RoxygenNote** 6.1.1

**URL** <https://github.com/halpo/tidymargins>

**BugReports** <https://github.com/halpo/tidymargins/issues>

**NeedsCompilation** no

**Author** Andrew Redd [aut, cre] (<<https://orcid.org/000-0002-6149-2438>>)

**Repository** CRAN

**Date/Publication** 2019-07-23 13:40:02 UTC

## R topics documented:

spread_each . . . . .	2
with_margins . . . . .	3

---

spread\_each                      *Spread multiple variables*

---

### Description

This is a multiple variable version of the function `tidyr::spread()`.

### Usage

```
spread_each(data, key, ..., fill = NA, convert = FALSE, drop = FALSE,
            sep = ".", key.first = TRUE)
```

### Arguments

<code>data</code>	A data frame.
<code>key</code>	Column names or positions. This is passed to <code>tidyselect::vars_pull()</code> . These arguments are passed by expression and support quasiquotation (you can unquote column names or column positions).
<code>...</code>	the columns to act as the values to spread out.
<code>fill</code>	If set, missing values will be replaced with this value. Note that there are two types of missingness in the input: explicit missing values (i.e. NA), and implicit missings, rows that simply aren't present. Both types of missing value will be replaced by <code>fill</code> .
<code>convert</code>	If TRUE, <code>type.convert()</code> with <code>asis = TRUE</code> will be run on each of the new columns. This is useful if the value column was a mix of variables that was coerced to a string. If the class of the value column was factor or date, note that will not be true of the new columns that are produced, which are coerced to character before type conversion.
<code>drop</code>	If FALSE, will keep factor levels that don't appear in the data, filling in missing combinations with <code>fill</code> .
<code>sep</code>	the character to use to separate parts of column names.
<code>key.first</code>	If TRUE, the default, the columns are named <code>{key level}{sep}{value column name}</code> , otherwise the format is <code>{value column name}{sep}{key level}{sep}</code>
<code>.</code>	The separator between the key levels and the value column names.

### Value

A wide `tbl_df`, with multiple value columns spread out.

### See Also

- Wide versus long data<sup>1</sup> (also known as narrow data) on Wikipedia.
- `tidyr::spread()` for the single variable version.

<sup>1</sup>[https://en.wikipedia.org/wiki/Wide\\_and\\_narrow\\_data](https://en.wikipedia.org/wiki/Wide_and_narrow_data)

**Examples**

```
library(dplyr)
data <- mutate(expand.grid( x = c( 'a', 'b', 'c')
                           , y = c( 'd', 'e', 'f')
                           , .rep = 1:10
                           ), v = rnorm(90))
long <- summarise(group_by(data, x, y), N=n(), sum=sum(v))

spread_each(long, y, N, sum)
```

---

with_margins	<i>Operate over margins</i>
--------------	-----------------------------

---

**Description**

Alter a function to operate over groups and all possible subsets of groupings, including no grouping.

**Usage**

```
with_margins(FUN, all.name = "(All)")
```

**Arguments**

FUN	the function to wrap.
all.name	the string to use to represent that a variable was marginalized over.

**Value**

Returns a function which alters the given FUN to be run for each possible subset of groupings and row bind the results together.

**See Also**

For understanding marginals:

- Marginal distributions<sup>2</sup> on Wikipedia.
- Marginal frequency<sup>3</sup> on statistics dictionary.

**Examples**

```
library(dplyr)
x <- c( 'a', 'b', 'c')
y <- c( 'd', 'e', 'f')
data <- group_by(mutate(expand.grid( x = x, y = y, .rep = 1:10), v = rnorm(90)), x, y)
with_margins(summarise)(data, N=n(), sum=sum(v))
```

---

<sup>2</sup>[https://en.wikipedia.org/wiki/Marginal\\_distribution](https://en.wikipedia.org/wiki/Marginal_distribution)

<sup>3</sup>[https://stattrek.com/statistics/dictionary.aspx?definition=marginal\\_frequency](https://stattrek.com/statistics/dictionary.aspx?definition=marginal_frequency)