

Package ‘ipmisc’

May 7, 2021

Type Package

Title Miscellaneous Functions for Data Cleaning and Analysis

Version 6.0.2

Maintainer Indrajeet Patil <patilindrajeet.science@gmail.com>

Description Provides functions needed for data cleaning and formatting and forms data cleaning and wrangling backend for the following packages: 'ggstatsplot', 'pairwiseComparisons', and 'statsExpressions'.

License GPL-3 | file LICENSE

URL <https://indrajeetpatil.github.io/ipmisc/>,
<https://github.com/IndrajeetPatil/ipmisc>

BugReports <https://github.com/IndrajeetPatil/ipmisc/issues>

Depends R (>= 3.6.0)

Imports dplyr, magrittr, rlang, tibble, tidyr, zeallot

Suggests ggplot2, rmarkdown, spelling, testthat

Encoding UTF-8

Language en-US

LazyData true

RoxygenNote 7.1.1.9001

Config/testthat/edition 3

Config/testthat/parallel true

NeedsCompilation no

Author Indrajeet Patil [aut, cre, cph]
(<<https://orcid.org/0000-0003-1995-6531>>)

Repository CRAN

Date/Publication 2021-05-07 08:00:03 UTC

R topics documented:

| | |
|----------------------------------|---|
| bugs_long | 2 |
| format_num | 3 |
| iris_long | 4 |
| long_to_wide_converter | 5 |
| stats_type_switch | 6 |

| | |
|--------------|----------|
| Index | 8 |
|--------------|----------|

| | |
|-----------|--|
| bugs_long | <i>Tidy version of the "Bugs" dataset.</i> |
|-----------|--|

Description

Tidy version of the "Bugs" dataset.

Usage

```
bugs_long
```

Format

A data frame with 372 rows and 6 variables

- subject. Dummy identity number for each participant.
- gender. Participant's gender (Female, Male).
- region. Region of the world the participant was from.
- education. Level of education.
- condition. Condition of the experiment the participant gave rating for (**LDLF**: low frighteningness and low disgustingness; **LFHD**: low frighteningness and high disgustingness; **HFHD**: high frighteningness and low disgustingness; **HFHD**: high frighteningness and high disgustingness).
- desire. The desire to kill an arthropod was indicated on a scale from 0 to 10.

Details

This data set, "Bugs", provides the extent to which men and women want to kill arthropods that vary in frighteningness (low, high) and disgustingness (low, high). Each participant rates their attitudes towards all anthropods. Subset of the data reported by Ryan et al. (2013).

Source

<https://www.sciencedirect.com/science/article/pii/S0747563213000277>

Examples

```
dim(bugs_long)
head(bugs_long)
dplyr::glimpse(bugs_long)
```

| | |
|------------|----------------------------------|
| format_num | <i>Formatting numeric values</i> |
|------------|----------------------------------|

Description

Function to format an R object for pretty printing with a specified (k) number of decimal places. The function also allows really small p -values to be denoted as " $p < 0.001$ " rather than " $p = 0.000$ ". Note that if `p.value` is set to `TRUE`, the minimum value of k allowed is 3. If k is set to less than 3, the function will ignore entered k value and use $k = 3$ instead.

Usage

```
format_num(x, k = 3L, p.value = FALSE, ...)
```

Arguments

| | |
|----------------------|--|
| <code>x</code> | A numeric value. |
| <code>k</code> | Number of digits after decimal point (should be an integer) (Default: <code>k = 3L</code>). |
| <code>p.value</code> | Decides whether the number is a p -value (Default: <code>FALSE</code>). |
| <code>...</code> | Currently ignored. |

Value

Formatted numeric value.

Note

This function is **not** vectorized.

Examples

```
format_num(0.0000123, k = 2, p.value = TRUE)
format_num(0.008675, k = 2, p.value = TRUE)
format_num(0.003458, k = 3, p.value = FALSE)
```

`iris_long`*Edgar Anderson's Iris Data in long format.*

Description

Edgar Anderson's Iris Data in long format.

Usage

```
iris_long
```

Format

A data frame with 600 rows and 5 variables

- `id`. Dummy identity number for each flower (150 flowers in total).
- `Species`. The species are *Iris setosa*, *versicolor*, and *virginica*.
- `condition`. Factor giving a detailed description of the attribute (Four levels: "Petal.Length", "Petal.Width", "Sepal.Length", "Sepal.Width").
- `attribute`. What attribute is being measured ("Sepal" or "Petal").
- `measure`. What aspect of the attribute is being measured ("Length" or "Width").
- `value`. Value of the measurement.

Details

This famous (Fisher's or Anderson's) iris data set gives the measurements in centimeters of the variables sepal length and width and petal length and width, respectively, for 50 flowers from each of 3 species of iris. The species are *Iris setosa*, *versicolor*, and *virginica*.

This is a modified dataset from `datasets` package.

Source

<https://stat.ethz.ch/R-manual/R-devel/library/datasets/html/iris.html>

Examples

```
dim(iris_long)
head(iris_long)
dplyr::glimpse(iris_long)
```

`long_to_wide_converter`*Converts dataframe from long/tidy to wide format with NAs removed*

Description

This conversion is helpful mostly for repeated measures design, where removing NAs by participant can be a bit tedious.

It does not make sense to spread the dataframe to wide format when the measure is not repeated, so if `paired = TRUE`, `spread` argument will be ignored.

Usage

```
long_to_wide_converter(  
  data,  
  x,  
  y,  
  subject.id = NULL,  
  paired = TRUE,  
  spread = TRUE,  
  ...  
)
```

Arguments

| | |
|-------------------------|---|
| <code>data</code> | A dataframe (or a tibble) from which variables specified are to be taken. Other data types (e.g., matrix, table, array, etc.) will not be accepted. |
| <code>x</code> | The grouping (or independent) variable from the dataframe <code>data</code> . In case of a repeated measures or within-subjects design, if <code>subject.id</code> argument is not available or not explicitly specified, the function assumes that the data has already been sorted by such an id by the user and creates an internal identifier. So if your data is not sorted, the results <i>can</i> be inaccurate when there are more than two levels in <code>x</code> and there are NAs present. The data is expected to be sorted by user in subject-1, subject-2, ..., pattern. |
| <code>y</code> | The response (or outcome or dependent) variable from the dataframe <code>data</code> . |
| <code>subject.id</code> | Relevant in case of a repeated measures or within-subjects design (<code>paired = TRUE</code> , i.e.), it specifies the subject or repeated measures identifier. Important: Note that if this argument is <code>NULL</code> (which is the default), the function assumes that the data has already been sorted by such an id by the user and creates an internal identifier. So if your data is not sorted and you leave this argument unspecified, the results <i>can</i> be inaccurate when there are more than two levels in <code>x</code> and there are NAs present. |
| <code>paired</code> | Logical that decides whether the experimental design is repeated measures/within-subjects or between-subjects. The default is <code>FALSE</code> . |

| | |
|--------|--|
| spread | Logical that decides whether the dataframe needs to be converted from long/tidy to wide (default: TRUE). Relevant only if paired = TRUE. |
| ... | Currently ignored. |

Value

A dataframe with NAs removed.

Examples

```
# for reproducibility
library(ipmisc)
set.seed(123)

# repeated measures design
long_to_wide_converter(
  data = bugs_long,
  x = condition,
  y = desire,
  subject.id = subject,
  paired = TRUE
)

# independent measures design (spread argument is ignored)
long_to_wide_converter(
  data = ggplot2::msleep,
  x = vore,
  y = brainwt,
  paired = FALSE,
  spread = FALSE
)
```

stats_type_switch *Switch type of statistics.*

Description

Relevant mostly for ggstatsplot and statsExpressions packages, where different statistical approaches are supported via this argument: parametric, non-parametric, robust, and Bayesian. This switch function converts strings entered by users to a common pattern for convenience.

Usage

```
stats_type_switch(type)
```

Arguments

`type` A character specifying the type of statistical approach:

- "parametric"
- "nonparametric"
- "robust"
- "bayes"

This argument also accepts the following abbreviations: "p" (for *parametric*), "np" (for *nonparametric*), "r" (for *robust*), "bf" (for *Bayes Factor* or *Bayesian*).

Examples

```
stats_type_switch("p")  
stats_type_switch("bf")
```

Index

* datasets

bugs_long, 2

iris_long, 4

bugs_long, 2

format_num, 3

iris_long, 4

long_to_wide_converter, 5

stats_type_switch, 6