Package ‘ipmisc’

May 31, 2020

Type Package

Title Miscellaneous Functions for Data Cleaning and Analysis

Version 3.0.0

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: 'broomExtra', 'ggstatsplot', 'groupedstats', 'pairwiseComparisons', 'statsExpressions', and 'tidyBF'.

License GPL-3 | file LICENSE


Depends R (>= 3.6.0)

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

Suggests ggplot2, knitr, rmarkdown, spelling, testthat

Encoding UTF-8

Language en-US

LazyData true

RoxygenNote 7.1.0.9000

NeedsCompilation no

Author Indrajeet Patil [aut, cre, cph]

<https://orcid.org/0000-0003-1995-6531>)

Repository CRAN

Date/Publication 2020-05-31 04:20:19 UTC
R topics documented:

- bugs_long ......................................................... 2
- iris_long .......................................................... 3
- long_to_wide_converter .......................................... 4
- outlier_df ........................................................... 5
- set_cwd ............................................................... 6
- signif_column ....................................................... 6
- specify_decimal_p .................................................. 7
- stats_type_switch .................................................. 8

Index

bugs_long 9

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 freighteningness and low disgustingness; LFHD: low freighteningness and high disgustingness; HFHD: high freighteningness and low disgustingness; HFHD: high freighteningness 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 freighteningness (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

### 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*.
- **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


### Examples

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

Converts long-format dataframe to wide-format dataframe

Description

This conversion is helpful mostly for repeated measures design.

Usage

long_to_wide_converter(data, x, y, paired = TRUE, ...)

Arguments

data A dataframe (or a tibble) from which variables specified are to be taken. A matrix or tables will not be accepted.
x The grouping variable from the dataframe data.
y The response (a.k.a. outcome or dependent) variable from the dataframe data.
paired Logical that decides whether the experimental design is repeated measures/within-subjects or between-subjects. The default is FALSE.
... Currently ignored.

Value

A dataframe in the wide (or Cartesian) format.

Author(s)

Indrajeet Patil

Examples

```r
long_to_wide_converter(
  data = iris_long,
  x = condition,
  y = value,
  paired = TRUE
)
```
### Description

Adding a column to dataframe describing outlier status

### Usage

```r
outlier_df(data, x, y, outlier.label, outlier.coef = 1.5, ...)
```

### Arguments

- **data**: A dataframe (or a tibble) from which variables specified are to be taken. A matrix or tables will **not** be accepted.
- **x**: The grouping variable from the dataframe `data`.
- **y**: The response (a.k.a. outcome or dependent) variable from the dataframe `data`.
- **outlier.label**: Label to put on the outliers that have been tagged. This can’t be the same as `x` argument.
- **outlier.coef**: Coefficient for outlier detection using Tukey’s method. With Tukey’s method, outliers are below (1st Quartile) or above (3rd Quartile) `coef` times the Inter-Quartile Range (IQR) (Default: 1.5).
- **...**: Additional arguments.

### Value

The dataframe entered as `data` argument is returned with two additional columns: `isanoutlier` and `outlier` denoting which observation are outliers and their corresponding labels.

### Examples

```r
# adding column for outlier and a label for that outlier
outlier_df(
  data = morley,
  x = Expt,
  y = Speed,
  outlier.label = Run,
  outlier.coef = 2
)

#>      Expt  Speed isanoutlier outlier outlier.label 1 2 3 Run
#> 1 Expt1  Run1     FALSE     FALSE      False
#> 2 Expt2  Run2     FALSE     FALSE      False
#> 3 Expt3  Run3     FALSE     FALSE      False
#> 4 Expt4  Run4     FALSE     FALSE      False
#> 5 Expt5  Run5     FALSE     FALSE      False
#> 6 Expt6  Run6     FALSE     FALSE      False
#> 7 Expt7  Run7     FALSE     FALSE      False
#> 8 Expt8  Run8     FALSE     FALSE      False
#> 9 Expt9  Run9     FALSE     FALSE      False
#> 10 Expt10 Run10     FALSE     FALSE      False
#> 11 Expt11 Run11     FALSE     FALSE      False
#> 12 Expt12 Run12     FALSE     FALSE      False
#> 13 Expt13 Run13     FALSE     FALSE      False
#> 14 Expt14 Run14     FALSE     FALSE      False
#> 15 Expt15 Run15     FALSE     FALSE      False
#> 16 Expt16 Run16     FALSE     FALSE      False
#> 17 Expt17 Run17     FALSE     FALSE      False
#> 18 Expt18 Run18     FALSE     FALSE      False
#> 19 Expt19 Run19     FALSE     FALSE      False
#> 20 Expt20 Run20     FALSE     FALSE      False
#> 21 Expt21 Run21     FALSE     FALSE      False
#> 22 Expt22 Run22     FALSE     FALSE      False
#> 23 Expt23 Run23     FALSE     FALSE      False
#> 24 Expt24 Run24     FALSE     FALSE      False
#> 25 Expt25 Run25     FALSE     FALSE      False
#> 26 Expt26 Run26     FALSE     FALSE      False
#> 27 Expt27 Run27     FALSE     FALSE      False
#> 28 Expt28 Run28     FALSE     FALSE      False
#> 29 Expt29 Run29     FALSE     FALSE      False
#> 30 Expt30 Run30     FALSE     FALSE      False
```

```r
dplyr::arrange(outlier)
```
### set_cwd

*Setting Working Directory in RStudio to where the R Script is.*

**Description**

This function will change the current working directory to whichever directory the R script you are currently working on is located. This preempts the trouble of setting the working directory manually.

**Usage**

```r
set_cwd()
```

**Value**

Path to changed working directory.

**Note**

This function will work only with RStudio IDE. Reference: [https://eranraviv.com/r-tips-and-tricks-working-directory/](https://eranraviv.com/r-tips-and-tricks-working-directory/)

### signif_column

*Creating a new column with significance labels*

**Description**

This function will add a new column with significance labels to a dataframe containing p-values.

**Usage**

```r
signif_column(data, p, ...)
```

**Arguments**

- `data` Data frame from which variables specified are preferentially to be taken.
- `p` The column containing p-values.
- `...` Currently ignored.

**Value**

Returns the dataframe in tibble format with an additional column corresponding to APA-format statistical significance labels.
specify_decimal_p

Author(s)
Indrajeet Patil

Examples

# preparing a new dataframe
df <- cbind.data.frame(
  x = 1:5,
  y = 1,
  p.value = c(0.1, 0.5, 0.00001, 0.05, 0.01)
)

# dataframe with significance column
signif_column(data = df, p = p.value)

specify_decimal_p  Formatting numeric (p-)values

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. Important: This function is not vectorized.

Usage

specify_decimal_p(x, k = 3, p.value = FALSE)

Arguments

x  A numeric value.

k  Number of digits after decimal point (should be an integer) (Default: k = 3).

p.value  Decides whether the number is a p-value (Default: FALSE).

Value

Formatted numeric value.

Author(s)
Indrajeet Patil

Examples

specify_decimal_p(x = 0.00001, k = 2, p.value = TRUE)
specify_decimal_p(x = 0.008, k = 2, p.value = TRUE)
specify_decimal_p(x = 0.008, k = 3, p.value = FALSE)
stats_type_switch

Switch type of statistics.

Description
Relevant mostly for ggstatsplot and statsExpressions, where there are four types of statistics are supported: parametric, non-parametric, robust, and Bayesian. This switch function converts strings entered by users to a common pattern.

Usage
stats_type_switch(type)

Arguments
type Character string describing the type of statistics.

Examples
stats_type_switch("p")
stats_type_switch("bf")
Index

*Topic datasets
  bugs_long, 2
  iris_long, 3

bugs_long, 2

iris_long, 3

long_to_wide_converter, 4

outlier_df, 5

set_cwd, 6
signif_column, 6
specify_decimal_p, 7
stats_type_switch, 8