Package ‘ipmisc’

March 6, 2020

Type Package

Title Miscellaneous Functions for Data Cleaning and Analysis

Version 1.2.0

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: ‘ggstatsplot’, ‘groupedstats’, ‘pairwiseComparisons’, and ‘statsExpressions’.

License GPL-3 | file LICENSE


Depends R (>= 3.5.0)

Imports broomExtra, crayon, dplyr (>= 0.8.3), forcats, rlang (>= 0.4.2), rstudioapi, tibble (>= 2.1.3), tidyr (>= 1.0.0), zeallot

Suggests ggplot2, knitr, parameters, rmarkdown, spelling, testthat

Encoding UTF-8

Language en-US

LazyData true

RoxygenNote 7.0.2.9000

NeedsCompilation no

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

Repository CRAN

Date/Publication 2020-03-06 08:20:02 UTC
R topics documented:

- bartlett_message
- bugs_long
- ipmisc
- iris_long
- long_to_wide_converter
- movies_long
- movies_wide
- normality_message
- outlier_df
- set_cwd
- signif_column
- sort_xy
- specify_decimal_p

Index

### bartlett_message

Display homogeneity of variance test as a message

#### Description

A note to the user about the validity of assumptions for the default linear model.

#### Usage

```r
bartlett_message(data, x, y, lab = NULL, k = 3, output = "message", ...)
```

#### 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`.
- **lab**
  A character describing label for the variable. If `NULL`, variable name will be used.
- **k**
  Number of digits after decimal point (should be an integer) (Default: `k = 3`).
- **output**
  What output is desired: "message" (default) or "stats" (or "tidy") objects.
- **...**
  Currently ignored.
**Value**

A list of class "htest" containing the following components:

- **statistic**: Bartlett’s K-squared test statistic.
- **parameter**: the degrees of freedom of the approximate chi-squared distribution of the test statistic.
- **p.value**: the p-value of the test.
- **method**: the character string "Bartlett test of homogeneity of variances".
- **data.name**: a character string giving the names of the data.

**Examples**

```r
# getting message
bartlett_message(
  data = iris,
  x = Species,
  y = Sepal.Length,
  lab = "Iris Species"
)

# getting results from the test
bartlett_message(
  data = mtcars,
  x = am,
  y = wt,
  output = "tidy"
)
```

**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 freighteniningness 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


Examples

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

Description

Collection of functions to help with certain aspects of data-wrangling and data analysis that are not covered in the existing R packages.

Details

For more documentation, see README on GitHub.

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*.
- 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 data frame to wide-format data frame*

Description

This conversion is helpful mostly for repeated measures design.

Usage

```r
long_to_wide_converter(data, x, y, paired = TRUE, ...)
```
movies_long

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

long_to_wide_converter(
  data = iris_long,
  x = condition,
  y = value,
  paired = TRUE
)

movies_long

Movie information and user ratings from IMDB.com (long format).

Description

Movie information and user ratings from IMDB.com (long format).

Usage

movies_long

Format

A data frame with 1,579 rows and 8 variables

- title. Title of the movie.
- year. Year of release.
- budget. Total budget (if known) in US dollars
movies_wide

- length. Length in minutes.
- rating. Average IMDB user rating.
- votes. Number of IMDB users who rated this movie.
- mpaa. MPAA rating.
- genre. Different genres of movies (action, animation, comedy, drama, documentary, romance, short).

Details

Modified dataset from ggplot2movies package.


Movies were are identical to those selected for inclusion in movies_wide but this dataset has been constructed such that every movie appears in one and only one genre category.

Source

https://CRAN.R-project.org/package=ggplot2movies

Examples

dim(movies_long)
head(movies_long)
dplyr::glimpse(movies_long)

movies_wide

Movie information and user ratings from IMDB.com (wide format).

Description

Movie information and user ratings from IMDB.com (wide format).

Usage

movies_wide

Format

A data frame with 1,579 rows and 13 variables
- title. Title of the movie.
- year. Year of release.
- budget. Total budget in millions of US dollars
• length. Length in minutes.
• rating. Average IMDB user rating.
• votes. Number of IMDB users who rated this movie.
• mpaa. MPAA rating.
• action, animation, comedy, drama, documentary, romance, short. Binary variables representing if movie was classified as belonging to that genre.
• NumGenre. The number of different genres a film was classified in an integer between one and four

Details

Modified dataset from ggplot2movies package.


Movies were selected for inclusion if they had a known length and had been rated by at least one imdb user. Small categories such as documentaries and NC-17 movies were removed.

Source

https://CRAN.R-project.org/package=ggplot2movies

Examples

dim(movies_wide)
head(movies_wide)
dplyr::glimpse(movies_wide)

normality_message Display normality test result as a message.

Description

A note to the user about the validity of assumptions for the default linear model.

Usage

normality_message(x, lab = NULL, k = 2, output = "message", ...)
outlier_df

Arguments

x: A numeric vector.
lab: A character describing label for the variable. If NULL, a generic "x" label will be used.
k: Number of digits after decimal point (should be an integer) (Default: k = 3).
output: What output is desired: "message" (default) or "stats" (or "tidy") objects.
...: Additional arguments (ignored).

Value

A list with class "htest" containing the following components:

- statistic: the value of the Shapiro-Wilk statistic.
- p.value: an approximate p-value for the test. This is said in Royston (1995) to be adequate for p.value < 0.1.
- method: the character string "Shapiro-Wilk normality test".
- data.name: a character string giving the name(s) of the data.

Examples

# message
normality_message(
  x = anscombe$x1,
  lab = "x1",
  k = 3
)

# statistical test object
normality_message(
  x = anscombe$x2,
  output = "tidy"
)

outlier_df

Adding a column to dataframe describing outlier status

Description

Adding a column to dataframe describing outlier status

Usage

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

**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
) %>%
dplyr::arrange(outlier)
```

---

**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.
signif_column

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

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

Usage

```
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.

Author(s)
Indrajeet Patil

Examples
```r
# 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)
```
sort_xy

Sorting y column in data by x.

Description

Sorting y column in data by x.

Usage

sort_xy(data, x, y, sort = "none", .fun = mean, ...)

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.
.sort If "ascending" (default), x-variable factor levels will be sorted based on increasing values of y-variable. If "descending", the opposite. If "none", no sorting will happen.
.fun n summary function. It should take one vector for fct_reorder, and two vectors for fct_reorder2, and return a single value.
... Currently ignored.

Examples

sort_xy(ggplot2::msleep, vore, brainwt, sort = "ascending")

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)
**specify_decimal_p**

**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**

```r
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)
```
Index

*Topic datasets
  bugs_long, 3
  iris_long, 4
  movies_long, 6
  movies_wide, 7

bartlett_message, 2
bugs_long, 3

ipmisc, 4
iris_long, 4

long_to_wide_converter, 5
movies_long, 6
movies_wide, 7

normality_message, 8

outlier_df, 9

set_cwd, 10
signif_column, 11
sort_xy, 12
specify_decimal_p, 12