Package ‘schoRsch’

November 19, 2024

Type Package

Title Tools for Analyzing Factorial Experiments

Version 1.11

Date 2024-11-19

Author Roland Pfister [aut, cre],
Markus Janczyk [aut]

Maintainer Roland Pfister <`mail@roland-pfister.net`>

Imports utils

Suggests ez

Description Offers a helping hand to psychologists and other behavioral scientists who routinely deal with experimental data from factorial experiments. It includes several functions to format output from other R functions according to the style guidelines of the APA (American Psychological Association). This formatted output can be copied directly into manuscripts to facilitate data reporting. These features are backed up by a toolkit of several small helper functions, e.g., offering out-of-the-box outlier removal. The package lends its name to Georg ``Schorsch'' Schuessler, ingenious technician at the Department of Psychology III, University of Wuerzburg. For details on the implemented methods, see Roland Pfister and Markus Janczyk (2016) <doi:10.20982/tqmp.12.2.p147>.

URL `https://www.tqmp.org/RegularArticles/vol12-2/p147/index.html`

License GPL

NeedsCompilation no

Repository CRAN

Date/Publication 2024-11-19 12:00:02 UTC

Contents

`anova_out` *Format ANOVA Output*

Description

Distilles the most relevant data from an output object of `ezANOVA` and displays the results in a compact format.

Usage

```
anova_out(ezout, print = TRUE, sph.cor = "GG", mau.p = 0.05,
          etasq = "partial", dfsep = ", ", corr.df = FALSE, show.eps = 0)
```

Arguments

`ezout`	Output object created by a call to `ezANOVA`. This call has to have included a detailed output (`detailed=TRUE`).
`print`	Force results to be displayed, even if the function output is assigned to a variable (e.g., `output <- anova_out(...)`; logical; default=`TRUE`).
`sph.cor`	Correction method (one of `"no"`,`"GG"`,`"HF"`; default=`"GG"`).
`mau.p`	Threshold for Mauchly's test of sphericity (numerical; default=`0.05`).
`etasq`	Effect size estimate to be used; either partial eta-squared (`"partial"; default`) or generalized eta-squared (`"generalized"`).
`dfsep`	String that delimits the degrees of freedom of each F-value (default=`", "`).
`corr.df`	Display corrected degrees of freedom when Mauchly's test of sphericity is significant (default=`FALSE`).
`show.eps`	Show epsilon estimates when Mauchly's test of sphericity is significant? `0` = do not show, `1` = print after denominator dfs, `2` = print after F-value, `3` = print after effect size (default=`0`).

Details

The output of a call to `ezANOVA` is formatted according to the guidelines of the APA (American Psychological Association) as well as the DGPs ("Deutsche Gesellschaft fuer Psychologie"; German Psychological Society).

For repeated-measures ANOVAs, sphericity corrections are automatically applied to the p-values of effects that produced a significant result in Mauchly's test of sphericity. The corresponding input arguments (`sph.cor` and `mau.p`) do not affect between-subject designs.

Value

`anoout(ezout,...)` returns a list containing (1) the ANOVA table, (2) sphericity tests and corrections (if applicable), (3) formatted ANOVA results, (4) notes about which correction was applied to which effect.

Author(s)

Roland Pfister, Markus Janczyk

See Also

`ezANOVA`; `aov`; `chi_out`; `cor_out`; `t_out`;

`bimod_coef` *Compute bimodality coefficient*

Description

Computes the bimodality coefficient for a vector of data points (for a tutorial, see Pfister et al., 2013, Frontiers in Quantitative Psychology and Measurement).

Usage

```
bimod_coef(data, moments = FALSE, na.rm = TRUE)
```

Arguments

`data`	A vector containing the data.
`moments`	A logical specifying whether the sample moments skewness and kurtosis should be contained in the output.
`na.rm`	A logical specifying whether NAs should be removed from the data vector.

Value

`bimod_coef(data)` returns a bimodality coefficient for the input data; `bimod_coef(data,moments=TRUE)` returns a vector with three elements: the bimodality coefficient, skewness of the data, and sample kurtosis of the data.

Author(s)

Moritz Schaaf, Roland Pfister

See Also

`rank`; `ntiles`;

Examples

```
# Input slightly bimodal data
data <- c( 3, 5, 5, 5, 5, 7, 10, 17, 18, 18, 19, 19, 20)

# Show histogram
hist(data,breaks=c(0:20),include.lowest=FALSE,
     ylim=c(0,5),xlim=c(0,20))

# Compute bimodality coefficient
bimod_coef(data)

# Get bimodality coefficient, skewness, and kurtosis
bimod_coef(data,moments=TRUE)
```

`cd` *Change Directory*

Description

Performs relative changes of the working directory. Calling `cd("..")` moves one level up in the hierarchy whereas `cd("folder_name")` moves one level down to the designated folder.

Usage

```
cd(x)
```

Arguments

`x`	: A character string corresponding to target directory or `".."`.

Details

`cd` is designed as an equivalent to the DOS command. Contrary to the common use of cd, however, this function does not take absolute paths as input. Use `setwd` instead to navigate to an absulte path.

Value

`cd(x)` returns the new working directory.

Author(s)

Roland Pfister, Markus Janczyk

See Also

`getwd`; `setwd`;

Examples

```
## Create temporary folder
dir.create("a_test_dir")

## Navigate into the new folder...
cd("a_test_dir")
## ... and back again
cd("..")

## Remove temporary folder
unlink("a_test_dir",recursive=TRUE)
```

`chi_out` *Format Chi-Squared Test Output*

Description

Distilles the most relevant data from an output object of `chisq.test` and displays the results in a compact format.

Usage

```
chi_out(chioutput, show.n = FALSE,
print = TRUE)
```

Arguments

`chioutput`	Output object created by a call to `chisq.test`.
`show.n`	Display sample size (logical; default=`FALSE`)
`print`	Force results to be displayed, even if the function output is assigned to a variable (e.g., `output <- chi_out(...)`; logical; default=`TRUE`).

Details

The output of a call to `chi_out` is formatted according to the guidelines of the APA (American Psychological Association) as well as the DGPs ("Deutsche Gesellschaft fuer Psychologie"; German Psychological Society).

Value

`chi_out(chioutput,...)` returns a data.frame containing (1) a description of the test and (2) a line with formatted results.

Author(s)

Daniel Gromer

See Also

`chisq.test`; `anova_out`; `cor_out`; `t_out`;

`clear` *Clear Global Workspace*

Description

The global workspace is cleared; `clear` is a shortcut for the usual `rm(list=ls())`.

Usage

```
clear()
```

Author(s)

Roland Pfister, Markus Janczyk

See Also

`rm`; `ls`;

Examples

```
## Declare variables
a <- 1
b <- "abc"
ls()

## Clear workspace
clear()
ls()
```

`clear_all_but` *Clear Parts of Global Workspace*

Description

The global workspace is cleared while keeping (only) selected variables.

Usage

```
clear_all_but(keep = NULL)
```

Arguments

`keep`	Variables to keep. Specified as a vector of strings.

Details

An R version of the eponymous custom MATLAB function (https://de.mathworks.com/matlabcentral/fileexchange/25339-clear-all-but).

Author(s)

Moritz Schaaf

See Also

`clear`; `rm`; `ls`;

Examples

```
## Declare variables
a <- 1
b <- "abc"
c <- NA
ls()

## Clear workspace
clear_all_but(c("a","b"))
ls()
```

`cor_out` *Format Correlation Test Statistics*

Description

Distilles the most relevant data from an output object of `cor.test` and displays the results in a compact format.

Usage

```
cor_out(coroutput, stats = FALSE, print = TRUE, df = TRUE)
```

Arguments

`coroutput`	Output object created by a call to `cor.test`.
`stats`	If `TRUE`, the output includes t-values and corresponding degrees of freedom (default=`FALSE`.
`print`	Force results to be displayed, even if the function output is assigned to a variable (e.g., `output` <- `cor_out(...)`; logical; default=`TRUE`).
`df`	Show degrees of freedom (df) rather than sample size (N) in parentheses after the Pearson correlation coefficient to match APA Style; logical; default=`TRUE`) as of schoRsch v1.11.

Details

The output of a call to `cor.test` is formatted according to the guidelines of the APA (American Psychological Association) as well as the DGPs ("Deutsche Gesellschaft fuer Psychologie"; German Psychological Society).

Value

`cor_out(coroutput,...)` returns a line containing the formatted correlation results.

Author(s)

Markus Janczyk, Roland Pfister

See Also

`cor`; `cor.test`; `anova_out`; `chi_out`; `t_out`;

`ntiles` *Split distribution into quantiles*

Description

The data of a variable are rank-ordered and split to bins of (approximately) equal size. When tied ranks span across category borders, the function assigns all values to the lowest possible bin. This procedure can result in slightly different results as the corresponding function `Rank Cases` of SPSS with option `Ntiles`.

Usage

```
ntiles(data, dv,
       factors = NaN,
       bins = 5,
       res.labels = FALSE)
```

Arguments

`data`	A data frame containing the data relevant variable and possible factors that can be used to split the data frame into separate compartments.
`dv`	Character string specifying the name of the variable within `data` that is to be cut in bins. Alternatively, `dv` can be the appropriate column index.
`factors`	A string or vector of strings (e.g., `c("subject","condition")`) stating the conditions that should be used for splitting the data.
`bins`	The number of bins to be generated. Alternatively, a vector of cut-points can be specified according to the `break` argument of the function `cut`.
`res.labels`	The default value `FALSE` returns the bin number for each observation whereas `TRUE` returns the corresponding interval borders (in ranks).

Value

`ntiles(data, dv, ...)` returns a vector of bins.

Author(s)

Roland Pfister; Markus Janczyk

See Also

`cut`; `rank`; `split`; `lapply`;

Examples

```
## Build data frame
var1 <- c(1:9)
var2 <- c(1,1,1,2,2,2,3,3,3)
tmpdata <- data.frame(cbind(var1,var2))
tmpdata$var2 <- as.factor(tmpdata$var2)

## Get overall bins and display result
tmpdata$bins <- ntiles(tmpdata, dv = "var1", bins=3)
tmpdata

## Get bins separately for each factor level
## and display result
tmpdata$bins2 <- ntiles(tmpdata, dv = "var1", bins=3, factors = "var2")
tmpdata
```

`outlier` *Screen Data for Outliers*

Description

A chosen column of a data frame is screened for outliers, outliers are marked and/or eliminated. Either absolute lower and upper limits are applied, or outliers are identified based on z-transformed data. Either exact limits and/or cutoffs based on z-values need to be entered.

Usage

```
outlier(data, dv,
        todo = "na", res.name = "outlier",
        upper.limit = NaN, lower.limit = NaN,
        limit.exact = FALSE,
        upper.z = NaN, lower.z = NaN,
        z.exact = FALSE, factors = NaN,
        z.keep = TRUE, z.name = "zscores",
        vsj = FALSE,
        print.summary = TRUE)
```

Arguments

`data`	A data frame containing the data to be screened as well was appropriate condition variables.
`dv`	Character string specifying the name of the variable within `data` that is to be screened for outlier. Alternatively, `dv` can be the appropriate column index.
`todo`	Character string specifying the fate of outliers: `"na"` - outliers are turned into NAs, `"elim"` - rows containing outliers are deleted from dataframe, `"nothing"` - nothing happens, default=`todo = "na"`.

`res.name`	Character string specifying the name of the variable to be used for marking outliers, default=`res.name = "outlier"`.
`upper.limit`	An optional numerical specifying the absolute upper limit defining outliers.
`lower.limit`	An optional numerical specifying the absolute lower limit defining outliers.
`limit.exact`	Logical, if `TRUE` values equal to `lower.limit`/`upper.limit` are deemed outlier.
`upper.z`	An optional numerical specifying how much standard deviations within a cell a value must exceed to be identified as an outlier.
`lower.z`	An optional numerical specifying how much standard deviations within a cell a value must undercut to be identified as an outlier.
`factors`	A string or vector of strings (e.g., `c("subject","condition")`) stating the conditions that should be used for splitting the data.
`z.exact`	Logical, if `TRUE` z-values equal to `lower.z`/`upper.z` are deemed outlier.
`z.keep`	Logical, if `TRUE`, z-scores are stored in an additional column. If `FALSE`, z-scores are discarded after outlier correction.
`z.name`	Character string, specifying a name for the variable that should be used for storing z-scores.
`vsj`	To be implemented in a future version...
`print.summary`	Logical, if `TRUE`, a short summary on identified outliers is printed.

Details

If both, absolute limits and z-limits are specified, absolute limits are processed first and z-scores are computed for the remaining data points.

Value

`outlier(data,...)` returns the original data frame with the outlier correction applied. This data frame also has one additional column containing flags for outliers (`0` = not suspicious, `1` = outlier). If z-scores are requested, these scores are retured as an additional column.

Author(s)

Markus Janczyk, Roland Pfister

See Also

`split`; `zscores`;

`schoRsch` *Tools for Analyzing Factorial Experiments*

Description

Offers a helping hand to psychologists and other behavioral scientists who routinely deal with experimental data from factorial experiments. It includes several functions to format output from other R functions according to the style guidelines of the APA (American Psychological Association). This formatted output can be copied directly into manuscripts to facilitate data reporting. These features are backed up by a toolkit of several small helper functions, e.g., offering out-of-the-box outlier removal. The package lends its name to Georg "Schorsch" Schuessler, ingenious technician at the Department of Psychology III, University of Wuerzburg.

Details

Package:	schoRsch
Type:	Package
Version:	1.11
Date:	2024-11-19
License:	GPL-3

This package contains the following functions:

`anova_out`: Formats the output object from `ezANOVA` to the APA style (requires the `ez` package).

`bimod_coef`: Computes the bimodality coefficient for a data distribution.

`cor_out`: Formats the output object from `cor.test` to the APA style.

`chi_out`: Formats the output object from `chisq.test` to the APA style.

`t_out`: Formats the output object from `t.test` to the APA style.

`outlier`: Screens data for outliers, based on absolute values or z-scores. Outliers can either be marked or eliminated.

`ntiles`: Split distribution into quantiles for distribution analysis.

`zscores`: Computes z-scores of values separately for defined design cells.

`cd`: To easily change the current working directory.

`toclipboard`: Write data to clipboard (Windows only).

`clear`: Clears the whole workspace (i.e., like `rm(list=ls())`).

`clear_all_but`: Clears the whole workspace while keeping named variables.

Version history:

v1.11 | 2024-11-19 | Added `bimod_coef` as contributed by Moritz Schaaf. Buxfix for `t_out`, which was missing a newline when printing output. Added an option to output degrees of freedom rather than sample size for `cor_out`, with r(df) being the new default.

v1.10 | 2022-11-01 | Added `clear_all_but` as kindly contributed by Moritz Schaaf.

v1.9 | 2020-12-11 | Added argument `clipwarning` to `toclipboard`. Thanks to Moritz Schaaf for the feature request (v1.9.1 provided an instant bugfix to the new code).

v1.8 | 2020-09-23 | Fix for the changed behavior of factor levels for strings as introduced in R 4.0 (relevant for `anova_out`). Thanks to Valentin Koob for sending in the bug report.

v1.7 | 2019-11-12 | Bugfix for `anova_out` which crashed when assembling corrected degrees of freedom in certain cases after violations of sphericity. Thanks to Mirela Dubravac for sending in the bug report.

v1.6 | 2019-05-02 | Bugfix for `cor_out` which did not display negative correlations with 0 > r > -0.1 correctly. Thanks to Mario Reutter for the bug report.

v1.5 | 2018-12-15 | Default value for correcting effect sizes for paired-samples t-tests changed to `FALSE`; the use of corrections is now displayed as feedback message. Also: New options for `anova_out`; it is now possible to display corrected degrees of freedom for violations of the sphericity assumption and corresponding epsilon estimates. Thanks to Onur Asci for the feature request. Additional bugfix for the dfsep argument of `anova_out`.

v1.4 | 2017-02-14 | Bugfix for `cor_out` that no longer displays leading zeros for correlation coefficients; thanks to Juan Ramon Barrada for sending in the bug report.

v1.3 | 2016-09-13 | Overall documentation update based on comments from Vincent LeBlanc.

v1.2 | 2015-07-05 | Bugfix for the `print` option of `anova_out`; thanks to Sylvain Clement for sending in the bug report. Minor code changes.

v1.1 | 2014-07-30 | New functions `chi_out` (contributed by Daniel Gromer) and `toclipboard`; bugfixes when `anova_out` is called without detailed=TRUE. Updated help files.

v1.0 | 2013-03-20 | Package release.

Author(s)

Roland Pfister <mail(at)roland-pfister.net>, Markus Janczyk;

References

Pfister, R., & Janczyk, M. (2016). schoRsch: An R package for analyzing and reporting factorial experiments. *The Quantitative Methods for Psychology*, *12*(2), 147-151. doi: 10.20982/tqmp.12.2.p147

`toclipboard` *Copy Data to Clipboard*

Description

A data frame of variable is written to the clipboard, allowing easy pasting to MS Excel and Open/Libre Office Calc. This function is a wrapper to `write.table` with pre-specified options for plug-and-play usage. Most options of `write.table` are also supported by `toclipboard`. Note: The current version of `toclipboard` only supports Windows systems; the function will not run under Linux or Mac OS.

Usage

```
toclipboard(data,
            sep = "\t", quote = FALSE,
            eol = "\n",  na = "NA",
            dec = ".", row.names = FALSE,
            col.names = TRUE,
clipwarning = FALSE)
```

Arguments

`data`	The first argument should be the data frame or variable that is to be written to the clipboard. Data frames are copied with column names but without row names and columns are separated by tabs. This behavior can be customized with the following optional arguments (passed to `write.table`).
`sep`	Delimiter string.
`quote`	Put quotes around strings?
`eol`	End-of-line character.
`na`	How should NA-values be written?
`dec`	Decimal separator.
`row.names`	Should row names be written?
`col.names`	Should column names be written?
`clipwarning`	Determine if warnings should overwrite clipboard content (especially in case of buffer overflow when attempting to copy large datasets to the clipboard.

Author(s)

Roland Pfister

See Also

`write.table`

Examples

```
## Build data frame
var1 <- c(1:9)
var2 <- c(1,1,1,2,2,2,3,3,3)
tmpdata <- data.frame(cbind(var1,var2))

## Write data frame to clipboard
toclipboard(tmpdata)

## -> The data frame can now be pasted
## into any other application.
```

`t_out` *Format t-Test Output*

Description

Distilles the most relevant data from an output object of `t.test` and displays the results in a compact format.

Usage

```
t_out(toutput, n.equal = TRUE,
welch.df.exact = TRUE, welch.n = NA,
d.corr = FALSE, print = TRUE)
```

Arguments

`toutput`	Output object created by a call to `t.test`.
`n.equal`	Only applicable to two-sample t-tests. If sample sizes are not equal, `n.equal` specifies a vector of sample sizes, e.g., `n.equal = c(12,8)`.
`welch.df.exact`	Only applicable to Welch-tests. Indicates whether Welch-adjusted or unadjusted degrees of freedom (dfs) are reported (default=`TRUE`, i.e., Welch-adjusted dfs). If set to `FALSE`, the parameter `welch.n` has to be set as well.
`welch.n`	Only applicable to Welch-tests with unadjusted degrees of freedom. Parameter should be equal to the total sample size `n=n_1+n_2`.
`d.corr`	Only applicable to one-sample or paired-samples t-tests. If `TRUE`, Cohen's ds are computed using `sqrt(2)`-correction. Default changed to `FALSE` from version 1.5 onwards with an additional feedback message showing the use of corrections.
`print`	Force results to be displayed, even if the function output is assigned to a variable (e.g., `output <- t_out(...)`; logical; default=`TRUE`).

Details

The output of a call to `t_out` is formatted according to the guidelines of the APA (American Psychological Association) as well as the DGPs ("Deutsche Gesellschaft fuer Psychologie"; German Psychological Society).

Value

`t_out(toutput,...)` returns a list containing (1) a description of the t-test (two-sample t-test, Welch-test, paired-samples t-test, one-sample t-test) and (2) a line with formatted results.

Author(s)

Roland Pfister, Markus Janczyk

See Also

`t.test`; `anova_out`; `chi_out`; `cor_out`;

`zscores` *Compute z-Scores by Condition*

Description

Data of an input vector is transformed to z-scores (mean = 0, sd = 1). The function operates on single vectors as well as on specified columns of a data frame.

Usage

```
zscores(data, factors=NaN, dv=NaN)
```

Arguments

`data`	Either a data frame containing the data of interest or a single vector.
`factors`	If called with `factors=NaN` (default), the entire data is processed according to its grand mean and total variance. If `data` is a vector, `factors` can be a list of variables for splitting the variable into separate compartments. If `data` is a data frame, `factors` has to be specified as a character vector of column names or column indices.
`dv`	If `data` is a single vector, `dv` does not have to be specified. If `data` is a data frame, `dv` indicates the column of the data frame which contains the variable for z-transformation (e.g., `dv="rt"`) or its column index (e.g., `dv=15`).

Details

`zscores` computes z-score of a vector or a specified column within a dataframe. Computation can be done separately for combinations of factors.

Value

`zscores()` returns a vector containing the requested z-scores.

Author(s)

Roland Pfister, Markus Janczyk

See Also

`scale`; `split`; `outlier`;

Examples

```
# Create input vector and compute z-scores
measurements <- c(3,12,5,4,2,23,1,6)
zscores(measurements)

# Compute z-scores separately
# for conditions
cond1 <- c(1,1,1,1,2,2,2,2)
cond2 <- c(1,1,2,2,1,1,2,2)
zscores(measurements,list(cond1))
zscores(measurements,list(cond1,cond2))

# Calling zscores for data frames
data <- data.frame(measurements,
cond1,cond2)
zscores(data,dv="measurements",
factors=c("cond1","cond2"))

# Operating on column indices
zscores(data,dv=1,
factors=3)
```

Index