Package ‘TOSTER’

May 8, 2024

Version 0.8.3
Title Two One-Sided Tests (TOST) Equivalence Testing
Description Two one-sided tests (TOST) procedure to test equivalence for t-tests, correlations, differences between proportions, and meta-analyses, including power analysis for t-tests and correlations. Allows you to specify equivalence bounds in raw scale units or in terms of effect sizes. See: Lakens (2017) <doi:10.1177/1948550617697177>.

Maintainer Aaron Caldwell <arncaldwell149@gmail.com>

URL https://aaroncaldwell.us/TOSTERpkg/
License GPL-3
Imports stats, graphics, jmvcore (>= 0.9.6.4), ggplot2, ggdist, distributional, cowplot, tidyr, utils, R6, lifecycle
Suggests knitr, rmarkdown, broom, car, afex, testthat (>= 3.0.0), spelling
VignetteBuilder knitr
Encoding UTF-8
LazyData true
Config/testthat/edition 3
Depends R (>= 3.5)
RoxygenNote 7.2.3
Language en-US
NeedsCompilation no
Author Daniel Lakens [aut], Aaron Caldwell [aut, cre]
Repository CRAN
Date/Publication 2024-05-08 16:40:02 UTC
R topics documented:

as.htest ........................................... 3
boot_compare_cor .................................. 4
boot_compare_smd .................................. 5
boot.cor_test ....................................... 7
boot_log_TOST ...................................... 8
boot_ses_calc ...................................... 11
boot_smd_calc ...................................... 13
boot_t_test ......................................... 15
boot_t_TOST ......................................... 17
brunner_munzel ..................................... 20
compare_cor ......................................... 22
compare_smd ......................................... 24
corsum_test ......................................... 25
dataTOSTone ......................................... 27
dataTOSTpaired ..................................... 28
dataTOSTr ........................................... 30
dataTOSTtwo ......................................... 31
datatosttwoprop .................................. 33
equ_anova ............................................ 34
equ_ftest ............................................. 36
extract_r_paired ................................... 37
hawthorne ............................................ 38
htest-helpers ....................................... 38
log_TOST ............................................ 39
plot_cor ............................................. 41
plot_pes ............................................. 42
plot_smd ............................................. 43
powerTOSTone ...................................... 45
powerTOSTpaired .................................. 46
powerTOSTtwo ...................................... 48
power_eq_f ......................................... 49
power_twoprop ..................................... 50
power_t_TOST ...................................... 52
power_z_cor ........................................ 54
rbs ..................................................... 55
ses_calc ............................................. 57
simple_htest ........................................ 58
smd_calc ............................................ 61
TOSTmeta ............................................ 63
TOSTnp-methods .................................... 64
TOSTone ............................................. 65
TOSTpaired .......................................... 66
TOSTr ................................................ 68
TOSTt-methods ...................................... 69
TOSTtwo ............................................. 70
TOSTtwo.prop ....................................... 72
as_h test

Convert to class 'htest'

Description

[Maturing]
Convert a TOSTER result object of class 'TOSTt' or 'TOSTnp' to a list of class 'htest'.

Usage

as_h test(TOST)

Arguments

TOST A TOSTER result object of class 'TOSTt' or 'TOSTnp'.

Value

Returns a list containing a list of class 'htest' for the result of each test with the following elements:

data.name A character string giving the names of the data.
estimate Estimated difference in raw units.
method A character string indicating the performed test.
null.value Equivalence bound.
alternative A character string describing the alternative hypothesis.
parameter The degrees of freedom of the distribution of the test statistic.
statistic The value of the test statistic.
p.value The p-value of the test.
conf.int The confidence interval of the difference.

See Also

Other htest: htest-helper s, simple_h test()

Examples

res1 = t_TOST(formula = extra ~ group, data = sleep, eqb = .5, smd_ci = "goulet")
as_h test(res1)
Comparing Correlations between independent studies with bootstrapping

Description

[Stable]

A function to compare standardized mean differences (SMDs) between studies. This function is intended to be used to compare the compatibility of original studies with replication studies (lower p-values indicating lower compatibility).

Usage

```r
boot_compare_cor(
  x1, y1,
  x2, y2,
  alternative = c("two.sided", "less", "greater", "equivalence", "minimal.effect"),
  method = c("pearson", "kendall", "spearman", "winsorized", "bendpercent"),
  alpha = 0.05,
  null = 0,
  R = 1999,
  ...
)
```

Arguments

- `x1, y1` numeric vectors of data values. x and y must have the same length from study 1.
- `x2, y2` numeric vectors of data values. x and y must have the same length from study 2.
- `alternative` a character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater", "less", "equivalence" (TOST), or "minimal.effect" (TOST). You can specify just the initial letter.
- `method` a character string indicating which correlation coefficient is to be used for the test. One of "winsorized", "bendpercent","pearson", "kendall", or "spearman", can be abbreviated.
- `alpha` alpha level (default = 0.05)
- `null` a number indicating the null hypothesis. Default is a correlation of zero.
- `R` number of bootstrap replicates
- `...` further arguments to be passed to or from methods.
Value

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

- "p.value": numeric scalar containing the p-value for the test under the null hypothesis.
- "estimate": difference in correlations between studies.
- "conf.int": percentile (bootstrap) confidence interval for difference in correlations.
- "null.value": the specified hypothesized value for the null hypothesis.
- "alternative": character string indicating the alternative hypothesis (the value of the input argument alternative). Possible values are "greater", "less", or "two-sided".
- "method": a character string indicating how the association was measured.
- "data.name": Names of input values.
- "boot_res": List of bootstrapped results.
- "call": the matched call.

See Also

Other compare studies: `boot_compare_smd()`, `compare_cor()`, `compare_smd()`

---

**Description**

[Stable]

A function to compare standardized mean differences (SMDs) between studies. This function is intended to be used to compare the compatibility of original studies with replication studies (lower p-values indicating lower compatibility).

**Usage**

```r
boot_compare_smd(
x1,
y1 = NULL,
x2,
y2 = NULL,
null = 0,
paired = FALSE,
alternative = c("two.sided", "less", "greater"),
R = 1999,
alpha = 0.05
)
```
Arguments

- `x1`: a (non-empty) numeric vector of data values from study 1.
- `y1`: an optional (non-empty) numeric vector of data values from study 1.
- `x2`: a (non-empty) numeric vector of data values from study 2.
- `y2`: an optional (non-empty) numeric vector of data values from study 2.
- `null`: a number indicating the null hypothesis. For TOST, this would be equivalence bound.
- `paired`: a logical indicating whether the SMD is from a paired or independent samples design.
- `alternative`: a character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater" or "less". You can specify just the initial letter.
- `R`: number of bootstrap replicates
- `alpha`: alpha level (default = 0.05)

Value

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

- "statistic": z-score.
- "p.value": numeric scalar containing the p-value for the test under the null hypothesis.
- "estimate": difference in SMD between studies
- "conf.int": percentile (bootstrap) confidence interval for difference in SMDs
- "null.value": the specified hypothesized value for the null hypothesis.
- "alternative": character string indicating the alternative hypothesis (the value of the input argument alternative). Possible values are "greater", "less", or "two-sided".
- "method": Type of SMD.
- "data.name": "Boostrapped" to denote summary statistics were utilized to obtain results.
- "smd": SMDs input for the function.
- "df_ci": Data frame of confidence intervals.
- "boot_res": List of bootstrapped results.
- "call": the matched call.

See Also

Other compare studies: `boot_compare_cor()`, `compare_cor()`, `compare_smd()`
boot_cor_test

Description

[Stable]
A function for a bootstrap, percentile, method for correlation coefficients.

Usage

boot_cor_test(
  x,
  y,
  alternative = c("two.sided", "less", "greater", "equivalence", "minimal.effect"),
  method = c("pearson", "kendall", "spearman", "winsorized", "bendpercent"),
  alpha = 0.05,
  null = 0,
  boot_ci = c("basic", "perc"),
  R = 1999,
  ...
)

Arguments

x a (non-empty) numeric vector of data values.
y an optional (non-empty) numeric vector of data values.
alternative a character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater", "less", "equivalence" (TOST), or "minimal.effect" (TOST). You can specify just the initial letter.
method a character string indicating which correlation coefficient is to be used for the test. One of "winsorized", "bendpercent", "pearson", "kendall", or "spearman", can be abbreviated.
alpha alpha level (default = 0.05)
null a number indicating the null hypothesis. Default is a correlation of zero.
boot_ci type of bootstrap confidence interval. Options include studentized (stud), empirical/basic (basic) and percentile (perc) confidence intervals.
R number of bootstrap replicates
... further arguments to be passed to or from methods.

Details

This function uses a percentile bootstrap methods for the confidence intervals. The returned p-values are calculated from a re-sampled null distribution (similar to boot_t_TOST). See vignette("correlations") for more details.
The bootstrap correlation methods in this package offer two other correlations: a Winsorized correlation and a percentage bend correlation (see Wilcox 2017). These two can modified by adding the trim (Winsorized) or beta (percentage bend) arguments. The default for both arguments is 0.2 but can be modified at the user’s discretion. These calculations are based on Rand Wilcox’s R functions for his book (Wilcox, 2017), and adapted from their implementation in Guillaume Rousselet’s R package “bootcorci”.

Value

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

- "p.value": the p-value of the test.
- "estimate": the estimated measure of association, with name "pb", "wincor", "cor", "tau", or "rho" corresponding to the method employed.
- "null.value": the value of the association measure under the null hypothesis.
- "alternative": character string indicating the alternative hypothesis (the value of the input argument alternative).
- "method": a character string indicating how the association was measured.
- "data.name": a character string giving the names of the data.
- "call": the matched call.

References


See Also

Other Correlations: corsum_test(), plot_cor(), power_z_cor(), z_cor_test()
Usage

boot_log_TOST(x, ...)

## Default S3 method:
boot_log_TOST(
  x,
  y = NULL,
  hypothesis = c("EQU", "MET"),
  paired = FALSE,
  var.equal = FALSE,
  eqb = 1.25,
  alpha = 0.05,
  null = 1,
  boot_ci = c("stud", "basic", "perc"),
  R = 1999,
  ...
)

## S3 method for class 'formula'
boot_log_TOST(formula, data, subset, na.action, ...)

Arguments

x      a (non-empty) numeric vector of data values.
...    further arguments to be passed to or from methods.
y      an optional (non-empty) numeric vector of data values.
hypothesis 'EQU' for equivalence (default), or 'MET' for minimal effects test, the alternative hypothesis.
paired a logical indicating whether you want a paired t-test.
var.equal a logical variable indicating whether to treat the two variances as being equal. If TRUE then the pooled variance is used to estimate the variance otherwise the Welch (or Satterthwaite) approximation to the degrees of freedom is used.
eqb     Equivalence bound. Can provide 1 value (negative value is taken as the lower bound) or 2 specific values that represent the upper and lower equivalence bounds.
alpha   alpha level (default = 0.05)
null    Null hypothesis value for a two-tailed test (default is 1).
boot_ci type of bootstrap confidence interval. Options include studentized (stud), empirical/basic (basic) and percentile (perc) confidence intervals.
R       number of bootstrap replicates
formula a formula of the form lhs ~ rhs where lhs is a numeric variable giving the data values and rhs either 1 for a one-sample or paired test or a factor with two levels giving the corresponding groups. If lhs is of class "Pair" and rhs is 1, a paired test is done.
data an optional matrix or data frame (or similar: see model.frame) containing the variables in the formula formula. By default the variables are taken from environment(formula).

subset an optional vector specifying a subset of observations to be used.

na.action a function which indicates what should happen when the data contain NAs. Defaults to getOption("na.action").

Details

The implemented test(s) corresponds to the proposal of Chapter 16 of Efron and Tibshirani (1994), and recommended by He et al (2022). Returns TOSTt class object with bootstrapped based results. This is approximately equivalent to the percentile bootstrap method mentioned by He et al (2014).

For two-sample tests, the test is of $\bar{\log(x)} - \bar{\log(y)}$ (mean of x minus mean of y). For paired samples, the test is of the difference scores ($z$), wherein $z = \log(x) - \log(y) = \log(x)/\log(y)$, and the test is of $\bar{z}$ (mean of the difference/ratio scores).

For details on the calculations in this function see vignette("robustTOST").

Value

An S3 object of class "TOSTt" is returned containing the following slots:

- "TOST": A table of class "data.frame" containing two-tailed t-test and both one-tailed results.
- "eqb": A table of class "data.frame" containing equivalence bound settings.
- "effsize": table of class "data.frame" containing effect size estimates.
- "hypothesis": String stating the hypothesis being tested.
- "smd": List containing the results of the means ratio calculation.
  - Items include: d (means ratio estimate), dlow (lower CI bound), dhigh (upper CI bound), d_df (degrees of freedom for SMD), d_sigma (SE), d_lambda (non-centrality), J (bias correction), smd_label (type of SMD), d_denom (denominator calculation)
- "alpha": Alpha level set for the analysis.
- "method": Type of t-test.
- "decision": List included text regarding the decisions for statistical inference.
- "boot": List containing the bootstrap samples.

References


**boot_ses_calc**  

**Description**  

**[Maturing]**  
Standardized effect size (SES), these are the effect sizes not considered SMDs.

**Usage**

```r
boot_ses_calc(
  x,
  ...,  # further arguments to be passed to or from methods.
  paired = FALSE,
  ses = "rb",
  alpha = 0.05,
  boot_ci = c("basic", "stud", "perc"),
  R = 1999
)
```

```r
## Default S3 method:
boot_ses_calc(
  x,
  y = NULL,
  paired = FALSE,
  ses = c("rb", "odds", "logodds", "cstat"),
  alpha = 0.05,
  boot_ci = c("basic", "stud", "perc"),
  R = 1999,
  ...
)
```

```r
## S3 method for class 'formula'
boot_ses_calc(formula, data, subset, na.action, ...)
```

**Arguments**

- **x**
  a (non-empty) numeric vector of data values.

- **...**
  further arguments to be passed to or from methods.

- **paired**
  a logical indicating whether you want a paired t-test.

**See Also**

Other Robust tests: `boot_t_TOST()`, `boot_t_test()`, `brunner_munzel()`, `log_TOST()`, `wilcoxon_TOST()`  
Other TOST: `boot_t_TOST()`, `simple_htest()`, `t_TOST()`, `tsum_TOST()`, `wilcoxon_TOST()`
boot_ses_calc

ses  Standardized effect size. Default is "rb" for rank-biserial correlation. Options also include "cstat" for concordance probability, or "odds" for Wilcoxon-Mann-Whitney odds (otherwise known as Agresti’s generalized odds ratio).

alpha  alpha level (default = 0.05)

boot_ci  type of bootstrap confidence interval. Options include studentized (stud), empirical/basic (basic) and percentile (perc) confidence intervals.

R  number of bootstrap replicates

y  an optional (non-empty) numeric vector of data values.

formula  a formula of the form lhs ~ rhs where lhs is a numeric variable giving the data values and rhs either 1 for a one-sample or paired test or a factor with two levels giving the corresponding groups. If lhs is of class "Pair" and rhs is 1, a paired test is done.

data  an optional matrix or data frame (or similar: see model.frame) containing the variables in the formula formula. By default the variables are taken from environment(formula).

subset  an optional vector specifying a subset of observations to be used.

na.action  a function which indicates what should happen when the data contain NAs. Defaults to getOption("na.action").

Details

For details on the calculations in this function see vignette("robustTOST").

Value

A data frame containing the standardized effect size.

See Also

Other effect sizes: boot_smd_calc(), ses_calc(), smd_calc()

Examples

## Not run:
boot_ses_calc(formula = extra ~ group, data = sleep, paired = TRUE, ses = "r")

## End(Not run)
**boot_smd_calc**  

*Bootstrapped SMD Calculation*

---

**Description**

**[Maturing]**

A function to only calculate standardized mean differences with bootstrap confidence intervals.

**Usage**

```r
boot_smd_calc(  
x,  
...,  
paired = FALSE,  
var.equal = FALSE,  
alpha = 0.05,  
bias_correction = TRUE,  
rm_correction = FALSE,  
glass = NULL,  
boot_ci = c("stud", "basic", "perc"),  
R = 1999  
)
```

## Default S3 method:

```r
boot_smd_calc(  
x,  
y = NULL,  
paired = FALSE,  
var.equal = FALSE,  
alpha = 0.05,  
mu = 0,  
bias_correction = TRUE,  
rm_correction = FALSE,  
glass = NULL,  
boot_ci = c("stud", "basic", "perc"),  
R = 1999,  
...  
)
```

## S3 method for class 'formula'

```r
boot_smd_calc(formula, data, subset, na.action, ...)  
```

**Arguments**

- `x`  
  a (non-empty) numeric vector of data values.

- `...`  
  further arguments to be passed to or from methods.
boot_smd_calc

paired a logical indicating whether you want a paired t-test.

var.equal a logical variable indicating whether to treat the two variances as being equal. If TRUE then the pooled variance is used to estimate the variance otherwise the Welch (or Satterthwaite) approximation to the degrees of freedom is used.

alpha alpha level (default = 0.05)

bias_correction Apply Hedges’ correction for bias (default is TRUE).

rm_correction Repeated measures correction to make standardized mean difference Cohen’s d(rm). This only applies to repeated/paired samples. Default is FALSE.

glass A option to calculate Glass’s delta as an alternative to Cohen’s d type SMD. Default is NULL to not calculate Glass’s delta, ”glass1” will use the first group’s SD as the denominator whereas ”glass2” will use the 2nd group’s SD.

boot_ci type of bootstrap confidence interval. Options include studentized (stud), empirical/basic (basic) and percentile (perc) confidence intervals.

R number of bootstrap replicates

y an optional (non-empty) numeric vector of data values.

mu Null value. Deviating from zero will give the x-y-mu.

formula a formula of the form lhs ~ rhs where lhs is a numeric variable giving the data values and rhs either 1 for a one-sample or paired test or a factor with two levels giving the corresponding groups. If lhs is of class ”Pair” and rhs is 1, a paired test is done.

data an optional matrix or data frame (or similar: see model.frame) containing the variables in the formula formula. By default the variables are taken from environment(formula).

subset an optional vector specifying a subset of observations to be used.

na.action a function which indicates what should happen when the data contain NAs. Defaults to getOption(”na.action”).

Details

For details on the calculations in this function see vignette(”SMD_calcs”).

Value

A data frame containing the SMD estimates.

See Also

Other effect sizes: boot_ses_calc(), ses_calc(), smd_calc()

Examples

## Not run:
boot_smd_calc(formula = extra ~ group,data = sleep, paired = TRUE, smd_ci = ”nct”)

## End(Not run)
boot_t_test

Bootstrapped t-test

Description

[Stable]
A function for a bootstrap method for t-tests.

Usage

boot_t_test(x, ...)

## Default S3 method:
boot_t_test(
  x,
  y = NULL,
  var.equal = FALSE,
  paired = FALSE,
  alternative = c("two.sided", "less", "greater", "equivalence", "minimal.effect"),
  mu = 0,
  alpha = 0.05,
  boot_ci = c("stud", "basic", "perc"),
  R = 1999,
  ...
)

## S3 method for class 'formula'
boot_t_test(formula, data, subset, na.action, ...)

Arguments

x
a (non-empty) numeric vector of data values.

... further arguments to be passed to or from methods.

y
an optional (non-empty) numeric vector of data values.

var.equal
a logical variable indicating whether to treat the two variances as being equal. If TRUE then the pooled variance is used to estimate the variance otherwise the Welch (or Satterthwaite) approximation to the degrees of freedom is used.

paired
a logical indicating whether you want a paired t-test.

alternative
a character string specifying the alternative hypothesis. must be one of "two.sided" (default), "greater", "less", "equivalence" (TOST), or "minimal.effect" (TOST). You can specify just the initial letter.

mu
a number specifying an optional parameter used to form the null hypothesis. See 'Details'.

alpha alpha level (default = 0.05)
boot_t_test

boot_ci  type of bootstrap confidence interval. Options include studentized (stud), empirical/basic (basic) and percentile (perc) confidence intervals.

R  number of bootstrap replicates

formula  a formula of the form lhs ~ rhs where lhs is a numeric variable giving the data values and rhs either 1 for a one-sample or paired test or a factor with two levels giving the corresponding groups. If lhs is of class "Pair" and rhs is 1, a paired test is done.

data  an optional matrix or data frame (or similar: see model.frame) containing the variables in the formula formula. By default the variables are taken from environment(formula).

subset  an optional vector specifying a subset of observations to be used.

na.action  a function which indicates what should happen when the data contain NAs. Defaults to getOption("na.action").

Details

The implemented test(s) corresponds to the proposal of Chapter 16 of Efron and Tibshirani (1994).

For two-sample tests, the test is of \( \bar{x} - \bar{y} \) (mean of x minus mean of y). For paired samples, the test is of the difference scores (z), wherein \( z = x - y \), and the test is of \( \bar{z} \) (mean of the difference scores). For one-sample tests, the test is of \( \bar{x} \) (mean of x).

For details on the calculations in this function see vignette("robustTOST").

Value

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

- "statistic": the value of the t-statistic.
- "parameter": the degrees of freedom for the t-statistic.
- "p.value": the p-value for the test.
- "conf.int": a confidence interval for the mean appropriate to the specified alternative hypothesis.
- "estimate": the estimated mean or difference in means depending on whether it was a one-sample test or a two-sample test.
- "null.value": the specified hypothesized value of the mean or mean difference. May be 2 values.
- "stderr": the standard error of the mean (difference), used as denominator in the t-statistic formula.
- "alternative": a character string describing the alternative hypothesis.
- "method": a character string indicating what type of t-test was performed.
- "data.name": a character string giving the name(s) of the data.

References

See Also

Other Robust tests: `boot_log_TOST()`, `boot_t_TOST()`, `brunner_munzel()`, `log_TOST()`, `wilcoxon_TOST()`

Examples

```
# example code
boot_t_test(extra ~ group, data = sleep)
```

---

### Description

[Stable]

A function for a bootstrap method for TOST with all types of t-tests.

#### Usage

```r
boot_t_TOST(x, ...)  
```

```r  
## Default S3 method:
boot_t_TOST(
  x,
  y = NULL,
  hypothesis = "EQU",
  paired = FALSE,
  var.equal = FALSE,
  eqb,
  low_eqbound,
  high_eqbound,
  eqbound_type = "raw",
  alpha = 0.05,
  bias_correction = TRUE,
  rm_correction = FALSE,
  glass = NULL,
  mu = 0,
  R = 1999,
  boot_ci = c("stud", "basic", "perc"),
  ...
)
```

```r  
## S3 method for class 'formula'
boot_t_TOST(formula, data, subset, na.action, ...)
```
Arguments

\begin{itemize}
\item \texttt{x} \quad a (non-empty) numeric vector of data values.
\item \texttt{...} \quad further arguments to be passed to or from methods.
\item \texttt{y} \quad an optional (non-empty) numeric vector of data values.
\item \texttt{hypothesis} \quad 'EQU' for equivalence (default), or 'MET' for minimal effects test, the alternative hypothesis.
\item \texttt{paired} \quad a logical indicating whether you want a paired t-test.
\item \texttt{var.equal} \quad a logical variable indicating whether to treat the two variances as being equal. If TRUE then the pooled variance is used to estimate the variance otherwise the Welch (or Satterthwaite) approximation to the degrees of freedom is used.
\item \texttt{eqb} \quad Equivalence bound. Can provide 1 value (negative value is taken as the lower bound) or 2 specific values that represent the upper and lower equivalence bounds.
\item \texttt{low_eqbound} \quad lower equivalence bounds (deprecated).
\item \texttt{high_eqbound} \quad upper equivalence bounds (deprecated).
\item \texttt{eqbound_type} \quad Type of equivalence bound. Can be set to "SMD" for standardized mean difference (i.e., Cohen’s d) or "raw" for the mean difference. Default is "raw". Raw is strongly recommended as SMD bounds will produce biased results.
\item \texttt{alpha} \quad alpha level (default = 0.05)
\item \texttt{bias_correction} \quad Apply Hedges’ correction for bias (default is TRUE).
\item \texttt{rm_correction} \quad Repeated measures correction to make standardized mean difference Cohen’s d(rm). This only applies to repeated/paired samples. Default is FALSE.
\item \texttt{glass} \quad A option to calculate Glass’s delta as an alternative to Cohen’s d type SMD. Default is NULL to not calculate Glass’s delta, "glass1" will use the first group’s SD as the denominator whereas "glass2" will use the 2nd group’s SD.
\item \texttt{mu} \quad a number indicating the true value of the mean for the two tailed test (or difference in means if you are performing a two sample test).
\item \texttt{R} \quad number of bootstrap replicates
\item \texttt{boot_ci} \quad type of bootstrap confidence interval. Options include studentized (stud), empirical/basic (basic) and percentile (perc) confidence intervals.
\item \texttt{formula} \quad a formula of the form lhs ~ rhs where lhs is a numeric variable giving the data values and rhs either 1 for a one-sample or paired test or a factor with two levels giving the corresponding groups. If lhs is of class "Pair" and rhs is 1, a paired test is done.
\item \texttt{data} \quad an optional matrix or data frame (or similar: see model.frame) containing the variables in the formula formula. By default the variables are taken from environment(formula).
\item \texttt{subset} \quad an optional vector specifying a subset of observations to be used.
\item \texttt{na.action} \quad a function which indicates what should happen when the data contain NAs. Defaults to getOption("na.action").
\end{itemize}
Details

The implemented test(s) corresponds to the proposal of Chapter 16 of Efron and Tibshirani (1994). Returns TOSTt class object with bootstrapped based results. Please note that the repeated measures "corrected" effect size is not available.

For two-sample tests, the test is of $\bar{x} - \bar{y}$ (mean of x minus mean of y). For paired samples, the test is of the difference scores ($z$, wherein $z = x - y$), and the test is of $\bar{z}$ (mean of the difference scores). For one-sample tests, the test is of $\bar{x}$ (mean of x).

For details on the calculations in this function see vignette("robustTOST").

Value

An S3 object of class "TOSTt" is returned containing the following slots:

- "TOST": A table of class "data.frame" containing two-tailed t-test and both one-tailed results.
- "eqb": A table of class "data.frame" containing equivalence bound settings.
- "effsize": table of class "data.frame" containing effect size estimates.
- "hypothesis": String stating the hypothesis being tested
- "smd": List containing the results of the standardized mean difference calculations (e.g., Cohen's d).
  - Items include: d (estimate), dlow (lower CI bound), dhigh (upper CI bound), d_df (degrees of freedom for SMD), d_sigma (SE), d_lambda (non-centrality), J (bias correction), smd_label (type of SMD), d_denom (denominator calculation)
- "alpha": Alpha level set for the analysis.
- "method": Type of t-test.
- "decision": List included text regarding the decisions for statistical inference.
- "boot": List containing the bootstrap samples.

References


See Also

Other Robust tests: boot_log_TOST(), boot_t_test(), brunner_munzel(), log_TOST(), wilcoxon_TOST()
Other TOST: boot_log_TOST(), simple_htest(), t_TOST(), tsum_TOST(), wilcoxon_TOST()
brunner_munzel  Brunner-Munzel Test

Description

[Maturing]
This is a generic function that performs a generalized asymptotic Brunner-Munzel test in a fashion similar to t.test.

Usage

brunner_munzel(
  x,
  ..., 
  paired = FALSE, 
  alternative = c("two.sided", "less", "greater"), 
  mu = 0.5, 
  alpha = 0.05, 
  perm = FALSE, 
  max_n_perm = 10000 
)

## Default S3 method:
brunner_munzel(
  x, 
  y, 
  paired = FALSE, 
  alternative = c("two.sided", "less", "greater"), 
  mu = 0.5, 
  alpha = 0.05, 
  perm = FALSE, 
  max_n_perm = 10000, 
  ... 
)

## S3 method for class 'formula'
brunner_munzel(formula, data, subset, na.action, ...)

Arguments

x  a (non-empty) numeric vector of data values.
...

paired  a logical indicating whether you want a paired test.
alternative  a character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater" or "less". You can specify just the initial letter.
mu  a number specifying an optional parameter used to form the null hypothesis (Default = 0.5). This can be thought of as the null in terms of the relative effect, \( p = P(X < Y) + 0.5 \cdot P(X = Y) \); See 'Details'.

alpha  alpha level (default = 0.05)

perm  a logical indicating whether or not to perform a permutation test over approximate t-distribution based test (default is FALSE). Highly recommend to set perm = TRUE when sample size per condition is less than 15.

max_n_perm  the maximum number of permutations (default is 10000).

y  an optional (non-empty) numeric vector of data values.

formula  a formula of the form lhs ~ rhs where lhs is a numeric variable giving the data values and rhs either 1 for a one-sample or paired test or a factor with two levels giving the corresponding groups. If lhs is of class "Pair" and rhs is 1, a paired test is done.

data  an optional matrix or data frame (or similar: see model.frame) containing the variables in the formula formula. By default the variables are taken from environment(formula).

subset  an optional vector specifying a subset of observations to be used.

na.action  a function which indicates what should happen when the data contain NAs. Defaults to getOption("na.action").

Details

This function is made to provide a test of stochastic equality between two samples (paired or independent), and is referred to as the Brunner-Munzel test.

This tests the hypothesis that the relative effect, discussed below, is equal to the null value (default is \( \mu = 0.5 \)).

The estimate of the relative effect, which can be considered as value similar to the probability of superiority, refers to the following:

\[
\hat{p} = p(X < Y) + \frac{1}{2} \cdot P(X = Y)
\]

Note, for paired samples, this does not refer to the probability of an increase/decrease in paired sample but rather the probability that a randomly sampled value of X. This is also referred to as the "relative" effect in the literature. Therefore, the results will differ from the concordance probability provided by the ses_calc function.

The brunner_munzel function is based on the npar.t.test and npar.t.test.paired functions within the nparcomp package (Konietschke et al. 2015).

Value

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

- "statistic": the value of the test statistic.
- "parameter": the degrees of freedom for the test statistic.
- "p.value": the p-value for the test.
• “conf.int”: a confidence interval for the relative effect appropriate to the specified alternative hypothesis.
• “estimate”: the estimated relative effect.
• “null.value”: the specified hypothesized value of the relative effect.
• “stderr”: the standard error of the relative effect.
• “alternative”: a character string describing the alternative hypothesis.
• “method”: a character string indicating what type of test was performed.
• “data.name”: a character string giving the name(s) of the data.

References

See Also
Other Robust tests: `boot_log_TOST()`, `boot_t_TOST()`, `boot_t_test()`, `log_TOST()`, `wilcoxon_TOST()`

Examples
```
data(mtcars)
brunner_munzel(mpg ~ am, data = mtcars)
```

Description

[Stable]
A function to compare correlations between studies. This function is intended to be used to compare the compatibility of original studies with replication studies (lower p-values indicating lower compatibility).
compare_cor

Usage

```r
compare_cor(
  r1,  
  df1,  
  r2,  
  df2,  
  method = c("fisher", "kraatz"),  
  alternative = c("two.sided", "less", "greater", "equivalence", "minimal.effect"),  
  null = 0
)
```

Arguments

- **r1**: Correlation study 1.
- **df1**: Degrees of freedom from study 1 (if a simple correlation the df is N-2).
- **r2**: Correlation study 2.
- **df2**: Degrees of freedom from study 2 (if a simple correlation the df is N-2).
- **method**: Method for determining differences. Default, "z", will use Fisher’s transformation, while "Kraatz" will use the Kraatz method.
- **alternative**: a character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater", "less", "equivalence" (TOST), or "minimal.effect" (TOST). You can specify just the initial letter.
- **null**: a number indicating the null hypothesis. For TOST, this would be equivalence bound.

Details

This function tests for differences between correlations. @return A list with class "htest" containing the following components:

- "statistic": z-score
- "p.value": numeric scalar containing the p-value for the test under the null hypothesis.
- "estimate": difference in SMD between studies.
- "null.value": the specified hypothesized value for the null hypothesis.
- "alternative": character string indicating the alternative hypothesis (the value of the input argument alternative). Possible values are "greater", "less", or "two-sided".
- "method": Type of SMD.
- "data.name": "Summary Statistics" to denote summary statistics were utilized to obtain results.
- "cor": Correlation input for the function.
- "call": the matched call.
References

See Also
Other compare studies: `boot_compare_cor()`, `boot_compare_smd()`, `compare_smd()`

---

**compare_smd**

Comparing SMDs between independent studies

**Description**

A function to compare standardized mean differences (SMDs) between studies. This function is intended to be used to compare the compatibility of original studies with replication studies (lower p-values indicating lower compatibility).

**Usage**

```r
close_smd(
smd1,
n1,
se1 = NULL,
smd2,
n2,
se2 = NULL,
paired = FALSE,
alternative = c("two.sided", "less", "greater", "equivalence", "minimal.effect"),
null = 0,
TOST = FALSE
)
```

**Arguments**

- `smd1`, `smd2`: SMDs from study 1 & 2, respectively.
- `n1`, `n2`: Sample size(s) from study 1 & 2, respectively (can be 1 number or vector of 2 numbers).
- `se1`, `se2`: User supplied standard errors (SEs). This will override the internal calculations.
- `paired`: A logical indicating whether the SMD is from a paired or independent samples design. If a one-sample design, then paired must be set to TRUE.
alternative  a character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater", "less", "equivalence" (TOST), or "minimal.effect" (TOST). You can specify just the initial letter.

null  a number indicating the null hypothesis. For TOST, this would be equivalence bound.

TOST  Defunct: use alternative argument. Logical indicator (default = FALSE) to perform two one-sided tests of equivalence (TOST).

Details  This function tests for differences between SMDs from independent studies (e.g., original vs replication).

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

- "statistic": z-score.
- "p.value": numeric scalar containing the p-value for the test under the null hypothesis.
- "estimate": difference in SMD between studies.
- "null.value": the specified hypothesized value for the null hypothesis.
- "alternative": character string indicating the alternative hypothesis (the value of the input argument alternative). Possible values are "greater", "less", or "two-sided".
- "method": Type of SMD.
- "data.name": "Summary Statistics" to denote summary statistics were utilized to obtain results.
- "smd": SMDs input for the function.
- "sample_sizes": Sample sizes input for the function.
- "call": the matched call.

See Also  Other compare studies: boot_compare_cor(), boot_compare_smd(), compare_cor()
Usage
corsum_test(  
r,  
n,  
alternative = c("two.sided", "less", "greater", "equivalence", "minimal.effect"),  
method = c("pearson", "kendall", "spearman"),  
alpha = 0.05,  
null = 0  
)

Arguments

r observed correlation
n number of pairs of observations
alternative a character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater", "less", "equivalence" (TOST), or "minimal.effect" (TOST). You can specify just the initial letter.
method a character string indicating which correlation coefficient is to be used for the test. One of "pearson", "kendall", or "spearman", can be abbreviated.
alpha alpha level (default = 0.05)
null a number indicating the null hypothesis. Default is a correlation of zero.

Details

This function uses Fisher’s z transformation for the correlations, but uses Fieller’s correction of the standard error for Spearman’s ρ and Kendall’s τ.

Value

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

- "statistic": z-score.
- "p.value": the p-value of the test.
- "estimate": the estimated measure of association, with name "cor", "tau", or "rho" corresponding to the method employed.
- "null.value": the value of the association measure under the null hypothesis.
- "alternative": character string indicating the alternative hypothesis (the value of the input argument alternative).
- "method": a character string indicating how the association was measured.
- "data.name": a character string giving the names of the data.
- "call": the matched call.

References

See Also

Other Correlations: `boot_cor_test()`, `plot_cor()`, `power_z_cor()`, `z.cor_test()`

---

**dataTOSTone**

### TOST One Sample T-Test

**Description**

TOST One Sample T-Test in `jamovi`. This function is not meant to be utilized in R. See `t_TOST` function.

**Usage**

```r
dataTOSTone(
  data,
  vars,
  mu = 0,
  hypothesis = "EQU",
  low_eqbound = -0.5,
  high_eqbound = 0.5,
  eqbound_type = "raw",
  alpha = 0.05,
  desc = FALSE,
  plots = FALSE,
  low_eqbound_d = -999999999,
  high_eqbound_d = -999999999,
  smd_type = "g"
)
```

**Arguments**

- `data` the data as a data frame
- `vars` a vector of strings naming variables of interest in `data`
- `mu` a number (default: 0) to compare against
- `hypothesis` 'EQU' for equivalence (default), or 'MET' for minimal effects test, the alternative hypothesis;
- `low_eqbound` a number (default: -0.5) the lower equivalence bounds
- `high_eqbound` a number (default: 0.5) the upper equivalence bounds
- `eqbound_type` 'SMD' (default) or 'raw'; whether the bounds are specified in Cohen’s d or raw units respectively
- `alpha` alpha level (default = 0.05)
- `desc` TRUE or FALSE (default), provide descriptive statistics
- `plots` TRUE or FALSE (default), provide plots
Value

A results object containing:

- `results$text` an html
- `results$tost` a table
- `results$eqb` a table
- `results$effsize` a table
- `results$desc` a table
- `results$plots` an array of images

Tables can be converted to data frames with `asDF` or `as.data.frame`. For example:

```r
results$tost$asDF
as.data.frame(results$tost)
```

Examples

```r
library("TOSTER")
dataTOSTone(data=iris, vars="Sepal.Width", mu=3, low_eqbound=-0.3, high_eqbound=0.3, alpha=0.05, desc=TRUE, plots=TRUE)
TOSTone(m=3.05733, mu=3, sd=0.4358663, n=150, low_eqbound_d=-0.3, high_eqbound_d=0.3, alpha=0.05)
```

Description

TOST Paired Samples T-Test in jamovi. This function is not meant to be utilized in R. See `t_TOST` function.

Usage

```r
dataTOSTpaired(
  data,
  pair1,
  pair2,
  hypothesis = "EQU",
)```
low_eqbound = -0.5,
high_eqbound = 0.5,
eqbound_type = "raw",
alpha = 0.05,
desc = FALSE,
plots = FALSE,
low_eqbound_dz = -999999999,
high_eqbound_dz = -999999999,
indplot = FALSE,
diffplot = FALSE,
smd_type = "g"
)

Arguments

data the data as a data frame
pair1 A string naming the first part of the pair
pair2 A string naming the second part of the pair
hypothesis 'EQU' for equivalence (default), or 'MET' for minimal effects test, the alternative hypothesis.
low_eqbound a number (default: 0.5) the lower equivalence bounds
high_eqbound a number (default: 0.5) the upper equivalence bounds
eqbound_type 'SMD' (default) or 'raw'; whether the bounds are specified in standardized mean difference (Cohen’s dz) or raw units respectively
alpha alpha level (default = 0.05)
desc TRUE or FALSE (default), provide descriptive statistics
plots TRUE or FALSE (default), provide plots
low_eqbound_dz deprecated
high_eqbound_dz deprecated
indplot TRUE or FALSE (default), provide plot of paired data.
diffplot TRUE or FALSE (default), provide plot of difference scores.
smd_type 'd' (default) or 'g'; whether the calculated effect size is biased (d) or bias-corrected (g).

Value

A results object containing:

result$text a html
result$tost a table
result$eqb a table
result$effsize a table
result$desc a table
result$plots an image
Tables can be converted to data frames with `asDF` or `as.data.frame`. For example:

```r
results$tost$asDF
as.data.frame(results$tost)
```

**References**


**Examples**

```r
## Not run:
library("TOSTER")

dataTOSTpaired(data = randu, pair1 = "x", pair2="y", low_eqbound = -0.3,
               high_eqbound = 0.3, alpha = 0.05, desc = TRUE, plots = TRUE)

## End(Not run)
```

---

dataTOSTr

### Description

TOST for correlations in jamovi. This function is not meant to be utilized in R.

### Usage

```r
dataTOSTr(
  data,
  pairs,
  cor_type = "pearson",
  hypothesis = "EQU",
  low_eqbound_r = -0.3,
  high_eqbound_r = 0.3,
  alpha = 0.05,
  desc = FALSE,
  plots = FALSE
)
```
dataTOSTtwo

Arguments

- **data**: the data as a data frame
- **pairs**: a list of vectors of strings naming variables to correlate from `data`
- **cor_type**: a character string indicating which correlation coefficient is to be used for the test. One of "pearson", "kendall", or "spearman", can be abbreviated.
- **hypothesis**: 'EQU' for equivalence (default), or 'MET' for minimal effects test, the alternative hypothesis.
- **low_eqbound_r**: lower equivalence bounds (e.g., -0.3) expressed in a correlation effect size
- **high_eqbound_r**: upper equivalence bounds (e.g., 0.3) expressed in a correlation effect size
- **alpha**: alpha level (default = 0.05)
- **desc**: TRUE or FALSE (default), provide descriptive statistics
- **plots**: TRUE or FALSE (default), provide plots

Value

A results object containing:

- `results$text`: a preformatted
- `results$tost`: a table
- `results$desc`: a table
- `results$plots`: an array of images

Tables can be converted to data frames with `asDF` or `as.data.frame`. For example:

```r
results$tost$asDF
as.data.frame(results$tost)
```

---

**dataTOSTtwo**  
**TOST Independent Samples T-Test**

Description

TOST Independent Samples T-Test for jamovi. This function is not meant to be utilized in R. See `t_TOST` function.

Usage

```r
dataTOSTtwo(
  data,  
deps,  
group,  
var_equal = FALSE,  
hypothesis = "EQU",
```
```r
low_eqbound = -0.5,
high_eqbound = 0.5,
eqbound_type = "raw",
alpha = 0.05,
desc = FALSE,
plots = FALSE,
descplots = FALSE,
low_eqbound_d = -999999999,
high_eqbound_d = -999999999,
smd_type = "g"
)
```

**Arguments**

- `data`: the data as a data frame
- `deps`: a vector of strings naming dependent variables in `data`
- `group`: a string naming the grouping variable in `data`; must have two levels
- `var_equal`: TRUE or FALSE (default), assume equal variances
- `hypothesis`: 'EQU' for equivalence (default), or 'MET' for minimal effects test, the alternative hypothesis.
- `low_eqbound`: a number (default: -0.5) the lower equivalence/MET bounds
- `high_eqbound`: a number (default: 0.5) the upper equivalence/MET bounds
- `eqbound_type`: 'SMD' (default) or 'raw'; whether the bounds are specified in Cohen’s d or raw units respectively
- `alpha`: alpha level (default = 0.05)
- `desc`: TRUE or FALSE (default), provide descriptive statistics
- `plots`: TRUE or FALSE (default), provide effect size plots
- `descplots`: TRUE or FALSE (default), provide plots
- `low_eqbound_d`: deprecated
- `high_eqbound_d`: deprecated
- `smd_type`: 'd' (default) or 'g'; whether the calculated effect size is biased (d) or bias-corrected (g).

**Value**

A results object containing:

- `results$text`: a html
- `results$tost`: a table
- `results$eqb`: a table
- `results$effsize`: a table
- `results$desc`: a table
- `results$plots`: an array of images
- `results$descplots`: an array of images
Tables can be converted to data frames with asDF or \texttt{as.data.frame}. For example:

\begin{verbatim}
results$to$t$asDF
as.data.frame(results$to$t)
\end{verbatim}

\section*{References}


\section*{Examples}

\begin{verbatim}
library(TOSTER)

## Load iris dataset, remove one of the three groups so two are left
data<-iris[which(iris$Species!="versicolor"),]

## TOST procedure on the raw data
dataTOSTtwo(data, deps="Sepal.Width", group="Species", var_equal = TRUE, low_eqbound = -0.5, high_eqbound = 0.5, alpha = 0.05, desc = TRUE, plots = TRUE)
\end{verbatim}

\begin{verbatim}
TOST Two Proportions
\end{verbatim}

\section*{Description}

TOST Two Proportions for jamovi. This function is not meant to be utilized in R.

\section*{Usage}

\begin{verbatim}
datatosttwo(prop(data, var, level, group, hypothesis = "EQU", low_eqbound = -0.1, high_eqbound = 0.1, alpha = 0.05, desc = FALSE, plot = FALSE)
\end{verbatim}
Arguments

data
var
level
group
hypothesis 'EQU' for equivalence (default), or 'MET' for minimal effects test, the alternative hypothesis.
low_eqbound a number (default: -0.1) the lower equivalence bounds
high_eqbound a number (default: 0.1) the upper equivalence bounds
alpha alpha level (default = 0.05)
desc TRUE or FALSE (default), provide descriptive statistics
plot TRUE or FALSE (default), provide plot

Value
A results object containing:

results$tost a table
results$eqb a table
results$desc a table
results$plot an image

Tables can be converted to data frames with asDF or as.data.frame. For example:
results$tost$asDF
as.data.frame(results$tost)

equ_anova

Equivalence Test for ANOVA Results

Description

[Stable]
Performs equivalence test on the partial eta-squared (pes) value from ANOVA results.

Usage
equ_anova(object, eqbound, MET = FALSE, alpha = 0.05)
equ_anova

Arguments

  object: an object of returned by either Anova, aov, or afex_aov
  eqbound: Equivalence bound for the partial eta-squared.
  MET: logical indicator to perform a minimal effect test rather than equivalence test (default is FALSE).
  alpha: alpha used for the test (e.g., 0.05).

Details

For details on the calculations in this function see vignette("the_ftestTOSTER").

Value

Returns a data frame containing the ANOVA results with equivalence tests added.

The following abbreviations are used in the table:

- effect: name of the effect.
- df1: Degrees of Freedom in the numerator (i.e. DF effect).
- df2: Degrees of Freedom in the denominator (i.e., DF error).
- F: F-value.
- p.null: p-value (probability of the data given the null hypothesis).
- pes: partial Eta-Squared measure of effect size.
- eqbound: equivalence bound.
- p.equ: p-value (probability of the data given the equivalence hypothesis)

References


See Also

Other f-test: equ_ftest()
equ_ftest

Equivalence Test using an F-test

Description

[Stable]

Performs equivalence test on the partial eta-squared (pes) value for using an F-test.

Usage

equ_ftest(Fstat, df1, df2, eqbound = NULL, eqb, MET = FALSE, alpha = 0.05)

Arguments

Fstat The F-statistic from the F-test.
df1 Degrees of freedom for the numerator.
df2 Degrees of freedom for the denominator.
eqbound Defunct argument for equivalence bound for the partial eta-squared. Default is NULL.
eqb Defunct argument for equivalence bound for the partial eta-squared.
MET logical indicator to perform a minimal effect test rather than equivalence test (default is FALSE).
alpha alpha used for the test (e.g., 0.05).

Details

For details on the calculations in this function see vignette("the_ftestTOSTER").

Value

Object of class "htest"

- "statistic": The value of the F-statistic.
- "parameter": The degrees of freedom for the F-statistic.
- "p.value": The p-value for the test.
- "conf.int": A confidence interval for the partial eta-squared statistic.
- "estimate": Estimate of partial eta-squared.
- "null.value": The specified for the equivalence test.
- "method": A string indicating the type of F-test.
- "data.name": A required string indicating that this was calculated from summary statistics.
extract_r_paired

References

See Also
Other f-test: equ_anova()

---

extract_r_paired  Extract Paired Correlation

Description
[Stable]
A function for estimating the correlation from a paired samples t-test. Useful for when using tsum_TOST and the correlation is not available.

Usage
extract_r_paired(m1, sd1, m2, sd2 = NULL, n, tstat = NULL, pvalue = NULL)

Arguments
m1  mean of group 1.
sd1  standard deviation of group 1.
m2  mean of group 2.
sd2  standard deviation of group 2.
n  Sample size (number of pairs)
tstat  The t-value from a paired samples t-test
pvalue  The two-tailed p-value from a paired samples t-test

Value
An estimate of the correlation.

References
Description

A dataset from a study on the Hawthorne effect published by McCambridge et al. The dataset has 5 variables (participant_ID, totaldrinking.x, group, totaldrinking.y, totaldrinking.diff)

Usage

hawthorne

Format

An object of class data.frame with 5474 rows and 5 columns.

Source


Description

Functions to help interpret or display objects of the class htest

Usage

df_htest(htest, test_statistics = TRUE, show_ci = TRUE, extract_names = TRUE)

describe_htest(htest, alpha = NULL, digits = 3)

Arguments

htest A S3 object of the class htest

test_statistics A logical variable to display the test statistics.

show_ci A logical variable to display the confidence interval.

extract_names A logical variable to take the names from the S3 object (i.e., statistic for t.test would be "t")

alpha alpha level (default = 0.05)

digits integer indicating the number of decimal places.
log_TOST

See Also
Other htest: as_htest(), simple_htest()

Examples

# simple example with t-test
tres = t.test(extra ~ group, data = sleep)

# As a data frame
df_htest(tres)

# Describe t-test results
describe_htest(tres)

---

log_TOST

TOST with log transformed t-tests

Description

[Stable]
A function for TOST on the log-transformed data using parametric t-tests.

Usage

log_TOST(
  x,
  ...
  hypothesis = "EQU",
  paired = FALSE,
  var.equal = FALSE,
  eqb = 1.25,
  alpha = 0.05,
  null = 1
)

## Default S3 method:
log_TOST(
  x,
  y = NULL,
  hypothesis = c("EQU", "MET"),
  var.equal = FALSE,
  paired = FALSE,
  eqb = 1.25,
  alpha = 0.05,
  null = 1,
  ...
)
## S3 method for class 'formula'

`log_TOST(formula, data, subset, na.action, ...)`

### Arguments

- **x**
  - a (non-empty) numeric vector of data values.
- **...**
  - further arguments to be passed to or from methods.
- **hypothesis**
  - 'EQU' for equivalence (default), or 'MET' for minimal effects test, the alternative hypothesis.
- **paired**
  - a logical indicating whether you want a paired t-test.
- **var.equal**
  - a logical variable indicating whether to treat the two variances as being equal. If TRUE then the pooled variance is used to estimate the variance otherwise the Welch (or Satterthwaite) approximation to the degrees of freedom is used.
- **eqb**
  - Equivalence bound; default is 1.25 (FDA guidelines). Can provide 1 value (reciprocal value is taken as the lower bound) or 2 specific values that represent the upper and lower equivalence bounds.
- **alpha**
  - alpha level (default = 0.05)
- **null**
  - Null hypothesis value for a two-tailed test (default is 1).
- **y**
  - an optional (non-empty) numeric vector of data values.
- **formula**
  - a formula of the form lhs ~ rhs where lhs is a numeric variable giving the data values and rhs either 1 for a one-sample or paired test or a factor with two levels giving the corresponding groups. If lhs is of class "Pair" and rhs is 1, a paired test is done.
- **data**
  - an optional matrix or data frame (or similar: see model.frame) containing the variables in the formula formula. By default the variables are taken from environment(formula).
- **subset**
  - an optional vector specifying a subset of observations to be used.
- **na.action**
  - a function which indicates what should happen when the data contain NAs. Defaults to getOption("na.action").

### Details

For details on the calculations in this function see vignette("robustTOST").

For two-sample tests, the test is of $\bar{\log}(x) - \bar{\log}(y)$ (mean of x minus mean of y). For paired samples, the test is of the difference scores (z), wherein $z = \log(x) - \log(y)$, and the test is of $\bar{z}$ (mean of the difference/ratio scores).

### Value

An S3 object of class "TOSTt" is returned containing the following slots:

- "TOST": A table of class "data.frame" containing two-tailed t-test and both one-tailed results.
- "eqb": A table of class "data.frame" containing equivalence bound settings.
- "effsize": Table of class "data.frame" containing effect size estimates.
- "hypothesis": String stating the hypothesis being tested
- "smd": List containing the results of the means ratio calculation.
  - Items include: d (means ratio estimate), dlow (lower CI bound), dhigh (upper CI bound),
    d_df (degrees of freedom for SMD), d_sigma (SE), d_lambda (non-centrality), J (bias
    correction), smd_label (type of SMD), d_denom (denominator calculation)
- "alpha": Alpha level set for the analysis.
- "method": Type of t-test.
- "decision": List included text regarding the decisions for statistical inference.

References


See Also

Other Robust tests: boot_log_TOST(), boot_t_TOST(), boot_t_test(), brunner_munzel(), wilcox_TOST()

Examples

data(mtcars)
# Default FDA bioequivalence bounds
log_TOST(mpg ~ am,
data = mtcars)

---

**plot_cor**

*Plot correlation coefficients*

**Description**

[Stable]

Function to produce plots of the distribution of standard correlation coefficients.
Usage

plot_cor(
  r,
  n,
  method = c("pearson", "spearman", "kendall"),
  type = c("c", "cd"),
  levels = c(0.68, 0.9, 0.95, 0.999)
)

Arguments

- **r**: The observed correlation coefficient.
- **n**: Total number of observations (sample size).
- **method**: The method by which the coefficient was calculated: pearson, spearman, or kendall (default is "pearson")
- **type**: Choose whether to plot a "consonance" function ("c"), consonance density ("cd"), or both (c("c","cd"); default option).
- **levels**: Numeric vector of confidence levels to display

Details

This function was created so that users could create consonance plots of Pearson’s correlation coefficient. These types of plots are discussed by Schweder T, Hjort NL. (2016, ISBN:9781316445051) and Rafi Z, Greenland S. (2020) doi:10.1186/s12874-020-01105-9.

Value

Returns plot of the distribution of the correlation coefficient.

See Also

Other Correlations: `boot_cor_test()`, `corsum_test()`, `power_z_cor()`, `z_cor_test()

Other plotting functions: `plot_pes()`, `plot_smd()`

---

**plot_pes**

*Plot partial eta-squared*

Description

[Stable]

Function to produce plots of the distribution of partial eta-squared.
**plot_smd**

**Usage**

```r
plot_pes(
  Fstat,
  df1,
  df2,
  type = c("c", "cd"),
  levels = c(0.68, 0.9, 0.95, 0.999)
)
```

**Arguments**

- `Fstat`  The F-statistic from the F-test.
- `df1`  Degrees of freedom for the numerator.
- `df2`  Degrees of freedom for the denominator.
- `type`  Choose whether to plot a "consonance" function ("c"), consonance density ("cd"), or both (c("c","cd")); default option.
- `levels` Numeric vector of confidence levels to display

**Details**

This function was created so that users could create consonance plots of partial eta-squared from ANOVA-level effects. These types of plots are discussed by Schweder T, Hjort NL. (2016, ISBN:9781316445051) and Rafi Z, Greenland S. (2020) doi:10.1186/s12874-020-01105-9.

**Value**

Returns plot of the distribution of partial eta-squared

**See Also**

Other plotting functions: `plot_cor()`, `plot_smd()`

---

**plot_smd**  
*Plot Distribution of a SMD*

**Description**

[Stable]

Function to produce plots of the distribution of the standardized mean difference
Usage

plot_smd(
  d,
  df,
  lambda = NULL,
  sigma = NULL,
  smd_ci = c("goulet", "nct", "t", "z"),
  smd_label = "SMD",
  type = c("c", "cd"),
  levels = c(0.5, 0.9, 0.95, 0.999)
)

Arguments

d Estimate of the standardized mean difference

df degrees of freedom for the standardized mean difference

lambda The non-centrality parameter for the standardized mean difference

sigma The standard error for the standardized mean difference

smd_ci Method for calculating SMD confidence intervals. Methods include Goulet, noncentral t (nct), central t (t), and normal method (z).

smd_label Label for the x-axis indicating the SMD measure

type Choose whether to plot a "consonance" function ("c"), consonance density ("cd"), or both (c("c","cd"); default option).

levels Numeric vector of confidence levels to display

Details

This function was created so that users could create plots from their own SMD calculations and were inspired by the concurve R package. The difficulty is that specific information must be past onto this function. The calculations for the standardized mean difference can be found in the vignettes of this package. These types of plots are discussed by Schweder T, Hjort NL. (2016, ISBN:9781316445051) and Rafi Z, Greenland S. (2020) doi:10.1186/s12874-020-01105-9.

Value

Returns plot of the distribution of the standardized mean difference.

See Also

Other plotting functions: plot_cor(), plot_pes()
Description

[Superseded]
Power analysis for TOST for one-sample t-test (Cohen’s d). This function is no longer maintained please use power_t_TOST.

Usage

powerTOSTone(alpha, statistical_power, N, low_eqbound_d, high_eqbound_d)

powerTOSTone.raw(alpha, statistical_power, N, sd, low_eqbound, high_eqbound)

Arguments

alpha alpha used for the test (e.g., 0.05)
statistical_power desired power (e.g., 0.8)
N sample size (e.g., 108)
low_eqbound_d lower equivalence bounds (e.g., -0.5) expressed in standardized mean difference (Cohen’s d)
high_eqbound_d upper equivalence bounds (e.g., 0.5) expressed in standardized mean difference (Cohen’s d)
sd standard deviation.
low_eqbound lower equivalence bounds (e.g., -0.5) expressed in raw scores
high_eqbound upper equivalence bounds (e.g., 0.5) expressed in raw scores

Value

Calculate either achieved power, equivalence bounds, or required N, assuming a true effect size of 0. Returns a string summarizing the power analysis, and a numeric variable for number of observations, equivalence bounds, or power.

References

Examples

```r
## Sample size for alpha = 0.05, 90% power, equivalence bounds of
## Cohen's d = -0.3 and Cohen's d = 0.3, and assuming true effect = 0
powerTOSTone(alpha=0.05, statistical_power=0.9, low_eqbound_d=-0.3, high_eqbound_d=0.3)

## Power for sample size of 121, alpha = 0.05, equivalence bounds of
## Cohen's d = -0.3 and Cohen's d = 0.3, and assuming true effect = 0
powerTOSTone(alpha=0.05, N=121, low_eqbound_d=-0.3, high_eqbound_d=0.3)

## Equivalence bounds for sample size of 121, alpha = 0.05, statistical power of
## 0.9, and assuming true effect d = 0
powerTOSTone(alpha=0.05, N=121, statistical_power=.9)

## Sample size for alpha = 0.05, 90% power, equivalence bounds of -0.3 and 0.3 in
## raw units, assuming pooled standard deviation of 1, and assuming true effect d = 0
powerTOSTone.raw(alpha=0.05, statistical_power=0.9, sd = 1, low_eqbound=-0.3, high_eqbound=0.3)

## Power for sample size of 121, alpha = 0.05, equivalence bounds of
## -0.3 and 0.3 in raw units, assuming pooled standard deviation of 1, and assuming true effect = 0
powerTOSTone.raw(alpha=0.05, N=121, sd = 1, low_eqbound=-0.3, high_eqbound=0.3)

## Power for sample size of 121, alpha = 0.05, statistical power of
## 0.9, and assuming true effect = 0
powerTOSTone.raw(alpha=0.05, N=121, statistical_power=.9, sd=1)
```

---

### powerTOSTpaired

**Power Paired Sample t-test**

#### Description

[Superseded]

Power analysis for TOST for dependent t-test (Cohen’s dz). This function is no longer maintained please use `power_t_TOST`

#### Usage

```r
powerTOSTpaired(alpha, statistical_power, N, low_eqbound_dz, high_eqbound_dz)

powerTOSTpaired.raw(
  alpha,
  statistical_power,
  low_eqbound,
  high_eqbound,
  sdif,
  N
)
```
powerTOSTpaired

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>alpha</td>
<td>alpha used for the test (e.g., 0.05)</td>
</tr>
<tr>
<td>statistical_power</td>
<td>desired power (e.g., 0.8)</td>
</tr>
<tr>
<td>N</td>
<td>number of pairs (e.g., 96)</td>
</tr>
<tr>
<td>low_eqbound_dz</td>
<td>lower equivalence bounds (e.g., -0.5) expressed in standardized mean difference (Cohen’s $d_z$)</td>
</tr>
<tr>
<td>high_eqbound_dz</td>
<td>upper equivalence bounds (e.g., 0.5) expressed in standardized mean difference (Cohen’s $d_z$)</td>
</tr>
<tr>
<td>low_eqbound</td>
<td>lower equivalence bounds (e.g., -0.5) expressed in raw mean difference</td>
</tr>
<tr>
<td>high_eqbound</td>
<td>upper equivalence bounds (e.g., 0.5) expressed in raw mean difference</td>
</tr>
<tr>
<td>sdif</td>
<td>standard deviation of the difference scores</td>
</tr>
</tbody>
</table>

Value

Calculate either achieved power, equivalence bounds, or required $N$, assuming a true effect size of 0. Returns a string summarizing the power analysis, and a numeric variable for number of observations, equivalence bounds, or power.

References


Examples

```r
## Sample size for alpha = 0.05, 80% power, equivalence bounds of Cohen's $d_z$ = -0.3 and Cohen's $d$ = 0.3, and assuming true effect = 0
powerTOSTpaired(alpha=0.05,statistical_power=0.8,low_eqbound_dz=-0.3,high_eqbound_dz=0.3)

## Sample size for alpha = 0.05, N = 96 pairs, equivalence bounds of Cohen's $d_z$ = -0.3 and Cohen's $d$ = 0.3, and assuming true effect = 0
powerTOSTpaired(alpha=0.05,N=96,low_eqbound_dz=-0.3,high_eqbound_dz=0.3)

## Equivalence bounds for alpha = 0.05, N = 96 pairs, statistical power of 0.8, and assuming true effect = 0
powerTOSTpaired(alpha=0.05,N=96,statistical_power=0.8)

## Sample size for alpha = 0.05, 80% power, equivalence bounds of -3 and 3 in raw units
## and assuming a standard deviation of the difference scores of 10, and assuming a true effect = 0
powerTOSTpaired.raw(alpha=0.05,statistical_power=0.8,low_eqbound=-3,high_eqbound=3, sdif=10)

## Sample size for alpha = 0.05, N = 96 pairs, equivalence bounds of -3 and 3 in raw units
## and assuming a standard deviation of the difference scores of 10, and assuming a true effect = 0
powerTOSTpaired.raw(alpha=0.05,N=96,low_eqbound=-3, high_eqbound=3, sdif=10)

## Equivalence bounds for alpha = 0.05, N = 96 pairs, statistical power of 0.8
## and assuming a standard deviation of the difference scores of 10, and assuming a true effect = 0
```
Description

[Superseded]
Power analysis for TOST for independent t-test (Cohen’s d). This function is no longer maintained please use `power_t_TOST`.

Usage

```r
powerTOSTtwo(alpha, statistical_power, N, low_eqbound_d, high_eqbound_d)

powerTOSTtwo.raw(
  alpha,
  statistical_power,
  N,
  sdpooled,
  low_eqbound,
  high_eqbound,
  delta = 0
)
```

Arguments

- `alpha` alpha used for the test (e.g., 0.05)
- `statistical_power` desired power (e.g., 0.8)
- `N` sample size per group (e.g., 108)
- `low_eqbound_d` lower equivalence bounds (e.g., -0.5) expressed in standardized mean difference (Cohen’s d)
- `high_eqbound_d` upper equivalence bounds (e.g., 0.5) expressed in standardized mean difference (Cohen’s d)
- `sdpooled` specify the pooled standard deviation
- `low_eqbound` lower equivalence bounds (e.g., -0.5) expressed in raw scale units (e.g., scale-points)
- `high_eqbound` upper equivalence bounds (e.g., 0.5) expressed in raw scale units (e.g., scale-points)
- `delta` hypothesized true value for the difference between the 2 means. Default is zero.

Value

Calculate either achieved power, equivalence bounds, or required N, assuming a true effect size of 0. Returns a string summarizing the power analysis, and a numeric variable for number of observations, equivalence bounds, or power.
References

Examples
```r
## Sample size for alpha = 0.05, 80% power, equivalence bounds of
## Cohen’s d = -0.4 and Cohen’s d = 0.4, assuming true effect = 0
powerTOSTtwo(alpha=0.05, statistical_power=0.8, low_eqbound_d=-0.4, high_eqbound_d=0.4)

## Statistical power for alpha = 0.05, N = 108 per group, equivalence bounds of
## Cohen’s d = -0.4 and Cohen’s d = 0.4, assuming true effect = 0
powerTOSTtwo(alpha=0.05, N=108, low_eqbound_d=-0.4, high_eqbound_d=0.4)

## Equivalence bounds for alpha = 0.05, N = 108 per group, statistical power of
## 0.8, assuming true effect = 0
powerTOSTtwo(alpha=0.05, N=108, statistical_power=0.8)

## Sample size for alpha = 0.05, 80% power, equivalence bounds of -200 and 200 in raw
## units, assuming pooled standard deviation of 350, and assuming true effect = 0
powerTOSTtwo.raw(alpha=0.05, statistical_power=0.8, low_eqbound=-200, high_eqbound=200, sdpooled=350)

## Power for alpha = 0.05, N = 53 per group, equivalence bounds of
## -200 and 200 in raw units, assuming sdpooled = 350 and true effect = 0
powerTOSTtwo.raw(alpha=0.05, N=53, low_eqbound=-200, high_eqbound=200, sdpooled=350)

## Equivalence bounds for alpha = 0.05, N = 108 per group, statistical power of
## 0.8, assuming true effect = 0
powerTOSTtwo.raw(alpha=0.05, N=53, statistical_power=0.8, sdpooled=350)
```

---

**power_eq_f**

*F-test Power*

**Description**

[Stable] Power analysis for TOST for an F-test

**Usage**

```r
power_eq_f(alpha = 0.05, df1, df2, eqbound)
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>alpha</td>
<td>alpha used for the test (e.g., 0.05)</td>
</tr>
<tr>
<td>df1</td>
<td>Degrees of freedom for the numerator</td>
</tr>
<tr>
<td>df2</td>
<td>Degrees of freedom for the denominator</td>
</tr>
<tr>
<td>eqbound</td>
<td>Equivalence bound for the partial eta-squared</td>
</tr>
</tbody>
</table>
Value

Object of class ""power.htest"

References


See Also

Other power: power_t_TOST(), power_z_cor()

Examples

## Statistical power for alpha = 0.05, 3 groups, n = 80 per group, equivalence bound of
## partial eta squared = 0.01, assuming true effect = 0.
## df1 = number of groups - 1 = 3 - 1 = 2.
## df2 = Total N - number of groups = 240 - 3 = 237.
##
## power_eq_f(alpha=0.05, df1=3, df2 = 237, eqbound = 0.01)

power_twoprop

TOST Power for Tests of Two Proportions

Description

[Maturing]

Power analysis for TOST for difference between two proportions using Z-test (pooled)

Usage

powerTOSTtwo.prop(
  alpha,
  statistical_power,
  prop1,
  prop2,
  N,
  low_eqbound_prop,
  high_eqbound_prop
)

power_twoprop(
  p1,
  p2,
  n = NULL,
  null = 0,
  alpha = NULL,
power = NULL,
  alternative = c("two.sided", "one.sided", "equivalence")
)

Arguments

alpha          a priori alpha-level (i.e., significance level).
statistical_power
  Deprecated. desired power (e.g., 0.8)
prop1          Deprecated. expected proportion in group 1.
prop2          Deprecated. expected proportion in group 2.
N              Deprecated. sample size (e.g., 108)
low_eqbound_prop
  Deprecated. lower equivalence bounds (e.g., -0.05) expressed in proportion
high_eqbound_prop
  Deprecated. upper equivalence bounds (e.g., 0.05) expressed in proportion
p1, p2         Proportions in each respective group.
n              Sample size per group.
null           the null hypothesis value.
power          statistical power (1-beta).
alternative    equivalence, one-sided, or two-sided test. Can be abbreviated.

Value

Calculate either achieved power, equivalence bounds, or required N, assuming a true effect size of 0. Returns a string summarizing the power analysis, and a numeric variable for number of observations, equivalence bounds, or power.

References


Examples

```r
## Sample size for alpha = 0.05, 90% power, assuming true effect prop1 = prop 2 = 0.5,
## equivalence bounds of 0.4 and 0.6 (so low_eqbound_prop = -0.1 and high_eqbound_prop = 0.1)

#powerTOSTtwo.prop(alpha = 0.05, statistical_power = 0.9, prop1 = 0.5, prop2 = 0.5,
#                    low_eqbound_prop = -0.1, high_eqbound_prop = 0.1)
```
power_t_TOST

Power calculations for TOST for t-tests

Description

[Stable]
Calculates the exact power of two one sided t-tests (TOST) for one, two, and paired samples.

Usage

```r
power_t_TOST(
  n = NULL,
  delta = 0,
  sd = 1,
  eqb,
  low_eqbound = NULL,
  alpha = 0.05, power = 0.9, p1 = 0.5, p2 = 0.5,
  null = 0.1, alternative = "e"
)
```

## Power for alpha = 0.05, N 542, assuming true effect prop1 = prop 2 = 0.5,
## equivalence bounds of 0.4 and 0.6 (so low_eqbound-prop = -0.1 and high_eqbound-prop = 0.1)

```r
powerTOSTtwo.prop(alpha = 0.05, N = 542, prop1 = 0.5, prop2 = 0.5,
                   low_eqbound_prop = -0.1, high_eqbound_prop = 0.1)
```

# Example 4.2.4 from Chow, Wang, & Shao (2007, p. 93)
# powerTOSTtwo.prop(alpha=0.05, statistical_power=0.8, prop1 = 0.75, prop2 = 0.8,
#                   low_eqbound_prop = -0.2, high_eqbound_prop = 0.2)

```r
power_towoprop(alpha = 0.05, power = 0.8, p1 = 0.75, p2 = 0.8,
                null = 0.2, alternative = "e")
```

# Example 5 from Julio & Campbell (2012, p. 2932)
# powerTOSTtwo.prop(alpha=0.025, statistical_power=0.9, prop1 = 0.8, prop2 = 0.8,
#                   low_eqbound_prop = -0.1, high_eqbound_prop = 0.1)

```r
power_towoprop(alpha = 0.025, power = 0.9, p1 = 0.8, p2 = 0.8,
                null = 0.1, alternative = "e")
```

# Example 9.4b equivalence of two proportions (p. 113) #
# powerTOSTtwo.prop(alpha=0.010, statistical_power=0.8, prop1 = 0.5, prop2 = 0.5,
#                   low_eqbound_prop = -0.2, high_eqbound_prop = 0.2)/2

```r
power_towoprop(alpha = 0.01, power = 0.8, p1 = 0.5, p2 = 0.5,
                null = 0.2, alternative = "e")
```
power_t_TOST

    high_eqbound = NULL,
    alpha = NULL,
    power = NULL,
    type = "two.sample"
    )

Arguments

n number of observations per group. 2 sample sizes, in a vector, can be provided for the two sample case.
delta true difference in means (default is 0).
sd population standard deviation. Standard deviation of the differences for paired samples.
eqb Equivalence bound. Can provide 1 value (negative value is taken as the lower bound) or 2 specific values that represent the upper and lower equivalence bounds.
low_eqbound Lower equivalence bounds. Deprecated use eqb.
high_eqbound Upper equivalence bounds. Deprecated use eqb.
alpha a priori alpha-level (i.e., significance level).
power power of the TOST procedure (1-beta).
type string specifying the type of t-test.

Details

The exact calculations of power are based on Owen’s Q-function or by direct integration of the bivariate non-central t-distribution (inspired by the PowerTOST package). Approximate power is implemented via the non-central t-distribution or the ‘shifted’ central t-distribution.

Note

The power function in this package is limited. Please see the PowerTOST R package for more options.

References


See Also

Other power: power_eq_f(), power_z_cor()
Description

[Maturing]
Calculates the approximate power for a z-test based on a Pearson product-moment correlation.

Usage

```r
power_z_cor(
  n = NULL,
  rho = NULL,
  power = NULL,
  null = 0,
  alpha = NULL,
  alternative = c("two.sided", "less", "greater", "equivalence")
)
```

```r
powerTOSTr(alpha, statistical_power, N, low_eqbound_r, high_eqbound_r)
```

Arguments

- `n`: number of observations.
- `rho`: true correlation value (alternative hypothesis).
- `power`: statistical power (1-beta).
- `null`: the null hypothesis value.
- `alpha`: a priori alpha-level (i.e., significance level).
- `alternative`: a character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater", "less", or "equivalence" (TOST). You can specify just the initial letter.
- `statistical_power`: Deprecated. desired power (e.g., 0.8)
- `N`: Deprecated. number of pairs (e.g., 96)
- `low_eqbound_r`: Deprecated. lower equivalence bounds (e.g., -0.3) expressed in a correlation effect size
- `high_eqbound_r`: Deprecated. upper equivalence bounds (e.g., 0.3) expressed in a correlation effect size

Value

An object of the class power.htest. This will include the sample size (n), power, beta (1-power), alpha (significance level), null value(s), alternative hypothesis, and a text string detailing the method.

`powerTOSTr` has been replaced by the `power_z_cor` function. The function is only retained for historical purposes.
rbs

Other Correlations: boot_cor_test(), corsum_test(), plot_cor(), z_cor_test()

Other power: power_eq_f(), power_t_TOST()

Examples

## Sample size for alpha = 0.05, 90% power, equivalence bounds of
## r = -0.1 and r = 0.1, assuming true effect = 0
#powerTOSTr(alpha=0.05, statistical_power=0.9, low_eqbound_r=-0.1, high_eqbound_r=0.1)
power_z_cor(alternative = "equivalence", alpha = .05, null = .1, power = .9, rho = 0)

## Sample size for alpha = 0.05, N=536, equivalence bounds of
## r = -0.1 and r = 0.1, assuming true effect = 0
#powerTOSTr(alpha=0.05, N=536, low_eqbound_r=-0.1, high_eqbound_r=0.1)
power_z_cor(alternative = "equivalence", alpha = .05, null = .1, n = 536, rho = 0)

## Equivalence bounds for alpha = 0.05, N=536, statistical power of
## 0.9, assuming true effect = 0
#powerTOSTr(alpha=0.05, N=536, statistical_power=0.9)

---

rbs Non-parametric standardized effect sizes (replicates of ses_calc)

Description

[Superseded]

Effect sizes for simple (one or two sample) non-parametric tests. Suggested to use ses_calc function instead.

Usage

rbs(x, y = NULL, mu = 0, conf.level = 0.95, paired = FALSE)

np_ses(
  x,
  y = NULL,
  mu = 0,
  conf.level = 0.95,
  paired = FALSE,
  ses = c("rb", "odds", "cstat")
)
Arguments

- \(x\) a (non-empty) numeric vector of data values.
- \(y\) an optional (non-empty) numeric vector of data values.
- \(\mu\) a number indicating the value around which (a-)symmetry (for one-sample or paired samples) or shift (for independent samples) is to be estimated. See stats::wilcox.test.
- \(\text{conf.level}\) confidence level of the interval.
- \(\text{paired}\) a logical indicating whether you want to calculate a paired test.
- \(\text{ses}\) Rank-biserial (rb), odds (odds), and concordance probability (cstat).

Details

This method was adapted from the effectsize R package. The rank-biserial correlation is appropriate for non-parametric tests of differences - both for the one sample or paired samples case, that would normally be tested with Wilcoxon’s Signed Rank Test (giving the matched-pairs rank-biserial correlation) and for two independent samples case, that would normally be tested with Mann-Whitney’s \(U\) Test (giving Glass’ rank-biserial correlation). See stats::wilcox.test. In both cases, the correlation represents the difference between the proportion of favorable and unfavorable pairs / signed ranks (Kerby, 2014). Values range from \(-1\) indicating that all values of the second sample are smaller than the first sample, to \(+1\) indicating that all values of the second sample are larger than the first sample.

In addition, the rank-biserial correlation can be transformed into a concordance probability (i.e., probability of superiority) or into a generalized odds (WMW odds or Agresti’s generalized odds ratio).

Ties:

When tied values occur, they are each given the average of the ranks that would have been given had no ties occurred. No other corrections have been implemented yet.

Value

Returns a list of results including the rank biserial correlation, logical indicator if it was a paired method, setting for \(\mu\), and confidence interval.

Confidence Intervals

Confidence intervals for the standardized effect sizes are estimated using the normal approximation (via Fisher’s transformation).

References


---

**ses_calc**

**SES Calculation**

---

**Description**

**[Stable]**

Standardized effect size (SES), these are the effect sizes not considered SMDs.

**Usage**

```r
ses_calc(x, ..., paired = FALSE, ses = "rb", alpha = 0.05)
```

### Default S3 method:

```r
nen

ses_calc(
    x,
    y = NULL,
    paired = FALSE,
    ses = c("rb", "odds", "logodds", "cstat"),
    alpha = 0.05,
    mu = 0,
    ...
)
```

### S3 method for class 'formula'

```r
nen

ses_calc(formula, data, subset, na.action, ...)
```

**Arguments**

- **x** a (non-empty) numeric vector of data values.
- **...** further arguments to be passed to or from methods.
- **paired** a logical indicating whether you want a paired t-test.
- **ses** Standardized effect size. Default is "rb" for rank-biserial correlation. Options also include "cstat" for concordance probability, or "odds" for Wilcoxon-Mann-Whitney odds (otherwise known as Agresti’s generalized odds ratio).
- **alpha** alpha level (default = 0.05)
- **y** an optional (non-empty) numeric vector of data values.
- **mu** number indicating the value around which (a-)symmetry (for one-sample or paired samples) or shift (for independent samples) is to be estimated. See `stats::wilcox.test`.
**simple_htest**

One, two, and paired samples hypothesis tests

---

**Description**

[Maturing]

Performs one or two sample t-tests or Wilcoxon-Mann-Whitney rank-based tests with expanded options compared to `t.test`, `brunner_munzel`, or `wilcox.test`.

**Usage**

```r
simple_htest(
  x,
  ..., 
  paired = FALSE,
  alternative = c("two.sided", "less", "greater", "equivalence", "minimal.effect"),
  mu = NULL,
```
alpha = 0.05

## Default S3 method:
simple_h-test(
  x,
  y = NULL,
  test = c("t.test", "wilcox.test", "brunner_munzel"),
  paired = FALSE,
  alternative = c("two.sided", "less", "greater", "equivalence", "minimal.effect"),
  mu = NULL,
  alpha = 0.05,
  ...
)

## S3 method for class 'formula'
simple_h-test(formula, data, subset, na.action, ...)

Arguments

x  
a (non-empty) numeric vector of data values.

...  
further arguments to be passed to or from methods.

paired  
a logical indicating whether you want a paired t-test.

alternative  
a character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater", "less", "equivalence" (TOST), or "minimal.effect" (TOST). You can specify just the initial letter.

mu  
a number specifying an optional parameter used to form the null hypothesis. See 'Details'.

alpha  
alpha level (default = 0.05)

y  
an optional (non-empty) numeric vector of data values.

test  
a character string specifying what type of hypothesis test to use. Options are limited to "wilcox.test", "t.test", or "brunner_munzel". You can specify just the initial letter.

formula  
a formula of the form lhs ~ rhs where lhs is a numeric variable giving the data values and rhs either 1 for a one-sample or paired test or a factor with two levels giving the corresponding groups. If lhs is of class "Pair" and rhs is 1, a paired test is done.

data  
an optional matrix or data frame (or similar: see model.frame) containing the variables in the formula formula. By default the variables are taken from environment(formula).

subset  
an optional vector specifying a subset of observations to be used.

na.action  
a function which indicates what should happen when the data contain NAs. Defaults togetOption("na.action").
Details

The type of test, t-test/Wilcoxon-Mann-Whitney/Brunner-Munzel, can be selected with the "test" argument.

alternative = "greater" is the alternative that x is larger than y (on average). If alternative = "equivalence" then the alternative is that the difference between x and y is between the two null values mu. If alternative = "minimal.effect" then the alternative is that the difference between x and y is less than the lowest null value or greater than the highest.

For more details on each possible test (brunner_munzel, stats::t.test, or stats::wilcox.test), please read their individual documentation.

Value

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

• statistic: the value of the t-statistic.
• parameter: the degrees of freedom for the t-statistic.
• p.value: the p-value for the test.
• conf.int: a confidence interval for the mean appropriate to the specified alternative hypothesis.
• estimate: the estimated mean or difference in means depending on whether it was a one-sample test or a two-sample test.
• null.value: the specified hypothesized value of the mean or mean difference. May be 2 values.
• stderr: the standard error of the mean (difference), used as denominator in the t-statistic formula.
• alternative: a character string describing the alternative hypothesis.
• method: a character string indicating what type of t-test was performed.
• data.name: a character string giving the name(s) of the data.

See Also

Other TOST: boot_log_TOST(), boot_t_TOST(), t_TOST(), tsum_TOST(), wilcox_TOST()

Other htest: as_hptest(), htest-helpers

Examples

data(mtcars)
simple_hptest(mpg ~ am,
data = mtcars,
alternative = "e",
mu = 3)
**smd_calc**

---

**SMD Calculation**

**Description**

[Stable]

A function to only calculate standardized mean differences.

**Usage**

```r
smd_calc(
  x,
  ..., 
  paired = FALSE,
  var.equal = FALSE,
  alpha = 0.05,
  bias_correction = TRUE,
  rm_correction = FALSE,
  glass = NULL,
  smd_ci = c("nct", "goulet", "t", "z")
)
```

```r
## Default S3 method:
smd_calc(
  x,
  y = NULL,
  paired = FALSE,
  var.equal = FALSE,
  alpha = 0.05,
  mu = 0,
  bias_correction = TRUE,
  rm_correction = FALSE,
  glass = NULL,
  smd_ci = c("nct", "goulet", "t", "z"),
  ...
)
```

```r
## S3 method for class 'formula'
smd_calc(formula, data, subset, na.action, ...)
```

**Arguments**

- `x` a (non-empty) numeric vector of data values.
- `...` further arguments to be passed to or from methods.
- `paired` a logical indicating whether you want a paired t-test.
smd_calc

var.equal  a logical variable indicating whether to treat the two variances as being equal. If TRUE then the pooled variance is used to estimate the variance otherwise the Welch (or Satterthwaite) approximation to the degrees of freedom is used.
alpha     alpha level (default = 0.05)
bias_correction  Apply Hedges’ correction for bias (default is TRUE).
rm_correction  Repeated measures correction to make standardized mean difference Cohen’s d(rm). This only applies to repeated/paired samples. Default is FALSE.
glass      A option to calculate Glass’s delta as an alternative to Cohen’s d type SMD. Default is NULL to not calculate Glass’s delta, ”glass1” will use the first group’s SD as the denominator whereas ”glass2” will use the 2nd group’s SD.
smd_ci     Method for calculating SMD confidence intervals. Methods include Goulet, noncentral t (nct), central t (t), and normal method (z).
y         an optional (non-empty) numeric vector of data values.
mu        Null value. Deviating from zero will give the x-y-mu.
formula   a formula of the form lhs ~ rhs where lhs is a numeric variable giving the data values and rhs either 1 for a one-sample or paired test or a factor with two levels giving the corresponding groups. If lhs is of class ”Pair” and rhs is 1, a paired test is done.
data      an optional matrix or data frame (or similar: see model.frame) containing the variables in the formula formula. By default the variables are taken from environment(formula).
subset    an optional vector specifying a subset of observations to be used.
na.action a function which indicates what should happen when the data contain NAs. Defaults to getOption(”na.action”).

Details
For details on the calculations in this function see vignette(”SMD_calcs”).

Value
A data frame containing the SMD estimates.

See Also
Other effect sizes: boot_ses_calc(), boot_smd_calc(), ses_calc()

Examples
## Not run:
smd_calc(formula = extra ~ group, data = sleep, paired = TRUE, smd_ci = ”nct”)
## End(Not run)
TOSTmeta

TOST function for meta-analysis

Description

[Stable]
A function for providing TOST tests of equivalence from meta-analysis results.

Usage

TOSTmeta(
  ES,
  var,
  se,
  low_eqbound_d,
  high_eqbound_d,
  alpha,
  plot = TRUE,
  verbose = TRUE
)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ES</td>
<td>meta-analytic effect size</td>
</tr>
<tr>
<td>var</td>
<td>meta-analytic variance</td>
</tr>
<tr>
<td>se</td>
<td>standard error</td>
</tr>
<tr>
<td>low_eqbound_d</td>
<td>lower equivalence bounds (e.g., -0.5) expressed in standardized mean difference (Cohen’s d)</td>
</tr>
<tr>
<td>high_eqbound_d</td>
<td>upper equivalence bounds (e.g., 0.5) expressed in standardized mean difference (Cohen’s d)</td>
</tr>
<tr>
<td>alpha</td>
<td>alpha level (default = 0.05)</td>
</tr>
<tr>
<td>plot</td>
<td>set whether results should be plotted (plot = TRUE) or not (plot = FALSE) - defaults to TRUE</td>
</tr>
<tr>
<td>verbose</td>
<td>logical variable indicating whether text output should be generated (verbose = TRUE) or not (verbose = FALSE) - default to TRUE</td>
</tr>
</tbody>
</table>

Value

Returns TOST Z-value 1, TOST p-value 1, TOST Z-value 2, TOST p-value 2, alpha, low equivalence bound d, high equivalence bound d, Lower limit confidence interval TOST, Upper limit confidence interval TOST

References

Examples

```r
## Run TOSTmeta by specifying the standard error
TOSTmeta(ES=0.12, se=0.09, low_eqbound_d=-0.2, high_eqbound_d=0.2, alpha=0.05)
## Run TOSTmeta by specifying the variance
TOSTmeta(ES=0.12, var=0.0081, low_eqbound_d=-0.2, high_eqbound_d=0.2, alpha=0.05)
## If both variance and se are specified, TOSTmeta will use standard error and ignore variance
TOSTmeta(ES=0.12, var=9999, se = 0.09, low_eqbound_d=-0.2, high_eqbound_d=0.2, alpha=0.05)
```

TOSTnp-methods

Methods for TOSTnp objects

Description

Methods defined for objects returned from the wilcox_TOST function.

Usage

```r
## S3 method for class 'TOSTnp'
print(x, digits = 4, ...)
## S3 method for class 'TOSTnp'
describe(x, digits = 3, ...)
```

Arguments

- `x` object of class TOSTnp.
- `digits` Number of digits to print for p-values
- `...` further arguments passed through, see description of return value for details.

Value

- `print`: Prints short summary of the tests.
- `describe`: Verbose description of results.

Examples

```r
# example code
data(mtcars)
res1 = wilcox_TOST(mpg ~ am, data = mtcars, eqb = 3)

# PRINT
print(res1)

# DESCRIPTION
describe(res1)
```
Description

[Superseded]
Development on this function is complete, and for new code we recommend switching to tsum_TOST, which is easier to use, more featureful, and still under active development.

Usage

```
TOSTone(
  m, 
  mu, 
  sd, 
  n, 
  low_eqbound_d, 
  high_eqbound_d, 
  alpha, 
  plot = TRUE, 
  verbose = TRUE
)
```

```
TOSTone.raw(
  m, 
  mu, 
  sd, 
  n, 
  low_eqbound, 
  high_eqbound, 
  alpha, 
  plot = TRUE, 
  verbose = TRUE
)
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>m</td>
<td>mean</td>
</tr>
<tr>
<td>mu</td>
<td>value to compare against</td>
</tr>
<tr>
<td>sd</td>
<td>standard deviation</td>
</tr>
<tr>
<td>n</td>
<td>sample size</td>
</tr>
<tr>
<td>low_eqbound_d</td>
<td>lower equivalence bounds (e.g., -0.5) expressed in standardized mean difference (Cohen’s d)</td>
</tr>
<tr>
<td>high_eqbound_d</td>
<td>upper equivalence bounds (e.g., 0.5) expressed in standardized mean difference (Cohen’s d)</td>
</tr>
</tbody>
</table>
alpha level (default = 0.05)

set whether results should be plotted (plot = TRUE) or not (plot = FALSE) - defaults to TRUE

logical variable indicating whether text output should be generated (verbose = TRUE) or not (verbose = FALSE) - default to TRUE

lower equivalence bounds (e.g., -0.5) expressed in raw units

upper equivalence bounds (e.g., 0.5) expressed in raw units

Value

Returns TOST t-value 1, TOST p-value 1, TOST t-value 2, TOST p-value 2, degrees of freedom, low equivalence bound, high equivalence bound, Lower limit confidence interval TOST, Upper limit confidence interval TOST

Examples

```r
## Test observed mean of 0.54 and standard deviation of 1.2 in sample of 100 participants
## against 0.5 given equivalence bounds of Cohen's d = -0.3 and 0.3, with an alpha = 0.05.
TOSTone(m=0.54,mu=0.5,sd=1.2,n=100,low_eqbound_d=-0.3, high_eqbound_d=0.3, alpha=0.05)
```

Description

[Superseded]

Development on this function is complete, and for new code we recommend switching to tsum_TOST, which is easier to use, more featureful, and still under active development.

Usage

```r
TOSTpaired(
  n,
  m1,
  m2,
  sd1,
  sd2,
  r12,
  low_eqbound_dz,
  high_eqbound_dz,
  alpha,
  plot = TRUE,
  verbose = TRUE
)
```

TOSTpaired.raw
n, m1, m2, sd1, sd2, r12, low_eqbound, high_eqbound, alpha, plot = TRUE, verbose = TRUE
)

Arguments

n sample size (pairs)
m1 mean of group 1
m2 mean of group 2
sd1 standard deviation of group 1
sd2 standard deviation of group 2
r12 correlation of dependent variable between group 1 and group 2
low_eqbound_dz lower equivalence bounds (e.g., -0.5) expressed in standardized mean difference (Cohen’s dz)
high_eqbound_dz upper equivalence bounds (e.g., 0.5) expressed in standardized mean difference (Cohen’s dz)
alpha alpha level (default = 0.05)
plot set whether results should be plotted (plot = TRUE) or not (plot = FALSE) - defaults to TRUE
verbose logical variable indicating whether text output should be generated (verbose = TRUE) or not (verbose = FALSE) - default to TRUE
low_eqbound lower equivalence bounds (e.g., -0.5) expressed in raw scores
high_eqbound upper equivalence bounds (e.g., 0.5) expressed in raw scores

Value

Returns TOST t-value 1, TOST p-value 1, TOST t-value 2, TOST p-value 2, degrees of freedom, low equivalence bound, high equivalence bound, low equivalence bound in dz, high equivalence bound in dz, Lower limit confidence interval TOST, Upper limit confidence interval TOST

References

Examples

```r
## Test means of 5.83 and 5.75, standard deviations of 1.17 and 1.29 in sample of 65 pairs
## with correlation between observations of 0.75 using equivalence bounds in Cohen's dz of
## -0.4 and 0.4 (with default alpha setting of 0.05).
TOSTpaired(n=65, m1=5.83, m2=5.75, sd1=1.17, sd2=1.29, r12=0.75, low_eqbound_dz=-0.4, high_eqbound_dz=0.4)
```

TOSTr

### TOST function for a correlation

[Superseded]

Development on TOSTr is complete, and for new code we recommend switching to `corsum_test`, which is easier to use, more featureful, and still under active development.

Usage

```r
TOSTr(n, r, low_eqbound_r, high_eqbound_r, alpha, plot = TRUE, verbose = TRUE)
```

Arguments

- `n`: number of pairs of observations
- `r`: observed correlation
- `low_eqbound_r`: lower equivalence bounds (e.g., -0.3) expressed in a correlation effect size
- `high_eqbound_r`: upper equivalence bounds (e.g., 0.3) expressed in a correlation effect size
- `alpha`: alpha level (default = 0.05)
- `plot`: set whether results should be plotted (plot = TRUE) or not (plot = FALSE) - defaults to TRUE
- `verbose`: logical variable indicating whether text output should be generated (verbose = TRUE) or not (verbose = FALSE) - default to TRUE

Value

Returns TOST p-value 1, TOST p-value 2, alpha, low equivalence bound r, high equivalence bound r, Lower limit confidence interval TOST, Upper limit confidence interval TOST

References


Examples

```r
TOSTr(n=100, r = 0.02, low_eqbound_r=-0.3, high_eqbound_r=0.3, alpha=0.05)
```
TOSTt-methods

Methods for TOSTt objects

Description

Methods defined for objects returned from the t_TOST and boot_t_TOST functions.

Usage

```r
## S3 method for class 'TOSTt'
print(x, digits = 4, ...)

## S3 method for class 'TOSTt'
plot(
x,
    type = c("cd", "c", "tnull"),
    estimates = c("raw", "SMD"),
    ci_lines,
    ci_shades,
    ...
)

describe(x, ...)

## S3 method for class 'TOSTt'
describe(x, digits = 3, ...)
```

Arguments

- `x` object of class TOSTt.
- `digits` Number of digits to print for p-values.
- `...` further arguments passed through, see description of return value for details.
- `type` Type of plot to produce. Default is a consonance density plot "cd". Consonance plots (type = "cd") and null distribution plots (type = "tnull") can also be produced. Note: null distribution plots only available for estimates = "raw".
- `estimates` indicator of what estimates to plot; options include "raw" or "SMD". Default is both: c("raw","SMD").
- `ci_lines` Confidence interval lines for plots. Default is 1-alpha*2 (e.g., alpha = 0.05 is 90%)
- `ci_shades` Confidence interval shades when plot type is "ed".

Value

- `print`: Prints short summary of the tests.
- `plot`: Returns a plot of the effects.
- `describe`: Verbose description of results.
Examples

# example code
# Print

res1 = t_TOST(mpg ~ am, data = mtcars, eqb = 3)

res1
# Print with more digits
print(res1, digits = 6)

# Plot with density plot - only raw values (SLOW)
#plot(res1, type = "cd", estimates = "raw")
# Plot with consonance - only raw values (SLOW)
#plot(res1, type = "c", estimates = "raw")
# Plot null distribution - only raw values
#plot(res1, type = "tnull", estimates = "raw")

# Get description of the results
describe(res1)

TOSTtwo

TOST function for an independent t-test (Cohen's d)

Description

[Superseded]
Development on TOSTtwo is complete, and for new code we recommend switching to tsum_TOST, which is easier to use, more featureful, and still under active development.

Usage

TOSTtwo(
  m1,
  m2,
  sd1,
  sd2,
  n1,
  n2,
  low_eqbound_d,
  high_eqbound_d,
  alpha,
  var.equal,
  plot = TRUE,
  verbose = TRUE
)

TOSTtwo.raw(}
m1, m2, sd1, sd2, n1, n2, low_eqbound, high_eqbound, alpha, var.equal, plot = TRUE, verbose = TRUE
)

Arguments

m1 mean of group 1
m2 mean of group 2
sd1 standard deviation of group 1
sd2 standard deviation of group 2
n1 sample size in group 1
n2 sample size in group 2
low_eqbound_d lower equivalence bounds (e.g., -0.5) expressed in standardized mean difference (Cohen’s d)
high_eqbound_d upper equivalence bounds (e.g., 0.5) expressed in standardized mean difference (Cohen’s d)
alpha alpha level (default = 0.05)
var.equal logical variable indicating whether equal variances assumption is assumed to be TRUE or FALSE. Defaults to FALSE.
plot set whether results should be plotted (plot = TRUE) or not (plot = FALSE) - defaults to TRUE
verbose logical variable indicating whether text output should be generated (verbose = TRUE) or not (verbose = FALSE) - default to TRUE
low_eqbound lower equivalence bounds (e.g., -0.5) expressed in raw scale units (e.g., scale-points)
high_eqbound upper equivalence bounds (e.g., 0.5) expressed in raw scale units (e.g., scale-points)

Value

Returns TOST t-value 1, TOST p-value 1, TOST t-value 2, TOST p-value 2, degrees of freedom, low equivalence bound, high equivalence bound, low equivalence bound in Cohen’s d, high equivalence bound in Cohen’s d, Lower limit confidence interval TOST, Upper limit confidence interval TOST
References


Examples

```r
## Eskine (2013) showed that participants who had been exposed to organic
## food were substantially harsher in their moral judgments relative to
## those exposed to control (d = 0.81, 95% CI: [0.19, 1.45]). A
## replication by Moery & Calin-Jageman (2016, Study 2) did not observe
## a significant effect (Control: n = 95, M = 5.25, SD = 0.95, Organic
## Food: n = 89, M = 5.22, SD = 0.83). Following Simonsohn's (2015)
## recommendation the equivalence bound was set to the effect size the
## original study had 33% power to detect (with n = 21 in each condition,
## this means the equivalence bound is d = 0.48, which equals a
## difference of 0.384 on a 7-point scale given the sample sizes and a
## pooled standard deviation of 0.894). Using a TOST equivalence test
## with default alpha = 0.05, not assuming equal variances, and equivalence
## bounds of d = -0.43 and d = 0.43 is significant, t(182) = -2.69,
## p = 0.004. We can reject effects larger than d = 0.43.

TOSTtwo(m1=5.25,m2=5.22,sd1=0.95,sd2=0.83,n1=95,n2=89,low_eqbound_d=-0.43,high_eqbound_d=0.43)
```

Description

[Superseded]

Development on TOSTtwo.prop is complete, and for new code we recommend switching to twoprop_test, which is easier to use, more featureful, and still under active development.

Usage

```r
TOSTtwo.prop(
  prop1, prop2, n1, n2, low_eqbound, high_eqbound, alpha,
  ci_type = "normal", plot = TRUE, verbose = TRUE
)
```
Arguments

- `prop1`: proportion of group 1
- `prop2`: proportion of group 2
- `n1`: sample size in group 1
- `n2`: sample size in group 2
- `low_eqbound`: lower equivalence bounds (e.g., -0.1) expressed in proportions
- `high_eqbound`: upper equivalence bounds (e.g., 0.1) expressed in proportions
- `alpha`: alpha level (default = 0.05)
- `ci_type`: confidence interval type (default = "normal"). "wilson" produces Wilson score intervals with a Yates continuity correction while "normal" calculates the simple asymptotic method with no continuity correction.
- `plot`: set whether results should be plotted (plot = TRUE) or not (plot = FALSE) - defaults to TRUE
- `verbose`: logical variable indicating whether text output should be generated (verbose = TRUE) or not (verbose = FALSE) - default to TRUE

Value

Returns TOST z-value 1, TOST p-value 1, TOST z-value 2, TOST p-value 2, low equivalence bound, high equivalence bound, Lower limit confidence interval TOST, Upper limit confidence interval TOST

References


Examples

```r
## Equivalence test for two independent proportions equal to .65 and .70, with 100 samples per group, lower equivalence bound of -0.1, higher equivalence bound of 0.1, and alpha of 0.05.

TOSTtwo.prop(prop1 = .65, prop2 = .70, n1 = 100, n2 = 100,
             low_eqbound = -0.1, high_eqbound = 0.1, alpha = .05)
```
Description

[Stable]
A function for TOST with all types of t-tests from summary statistics.

Usage

tsum_TOST(
    m1,  # mean of group 1.
    sd1,  # standard deviation of group 1.
    n1,  # sample size in group 1.
    m2 = NULL,  # mean of group 2.
    sd2 = NULL,  # standard deviation of group 2.
    n2 = NULL,  # sample size in group 2.
    r12 = NULL,  # correlation of dependent variable between group 1 and group 2.
    hypothesis = c("EQU", "MET"),  # 'EQU' for equivalence (default), or 'MET' for minimal effects test, the alternative hypothesis.
    paired = FALSE,  # TRUE for paired t-test.
    var.equal = FALSE,  # TRUE for equal variances.
    eqb,  # equivalence bounds.
    low_eqbound,  # lower equivalence bound.
    high_eqbound,  # upper equivalence bound.
    mu = 0,  # null hypothesis mean.
    eqbound_type = c("raw", "SMD"),  # type of equivalence bounds.
    alpha = 0.05,  # significance level.
    bias_correction = TRUE,  # TRUE for bias correction.
    rm_correction = FALSE,  # TRUE for random effects correction.
    glass = NULL,  # Glass's Delta.
    smd_ci = c("nct", "goulet", "t", "z")  # confidence intervals for SMD.
)

Arguments

m1  # mean of group 1.
sd1  # standard deviation of group 1.
n1  # sample size in group 1.
m2  # mean of group 2.
sd2  # standard deviation of group 2.
n2  # sample size in group 2.
r12  # correlation of dependent variable between group 1 and group 2.
hypothesis  # 'EQU' for equivalence (default), or 'MET' for minimal effects test, the alternative hypothesis.
paired: a logical indicating whether you want a paired t-test.
var.equal: a logical variable indicating whether to treat the two variances as being equal. If TRUE then the pooled variance is used to estimate the variance otherwise the Welch (or Satterthwaite) approximation to the degrees of freedom is used.
eqb: Equivalence bound. Can provide 1 value (negative value is taken as the lower bound) or 2 specific values that represent the upper and lower equivalence bounds.
low_eqbound: lower equivalence bounds (deprecated).
high_eqbound: upper equivalence bounds (deprecated).
mu: a number indicating the true value of the mean for the two tailed test (or difference in means if you are performing a two sample test).
eqb_type: Type of equivalence bound. Can be set to "SMD" for standardized mean difference (i.e., Cohen's d) or "raw" for the mean difference. Default is "raw". Raw is strongly recommended as SMD bounds will produce biased results.
alpha: alpha level (default = 0.05)
bias_correction: Apply Hedges' correction for bias (default is TRUE).
rm_correction: Repeated measures correction to make standardized mean difference Cohen's d(rm). This only applies to repeated/paired samples. Default is FALSE.
glass: A option to calculate Glass's delta as an alternative to Cohen's d type SMD. Default is NULL to not calculate Glass's delta, "glass1" will use the first group's SD as the denominator whereas "glass2" will use the 2nd group's SD.
smd_ci: Method for calculating SMD confidence intervals. Methods include Goulet, noncentral t (nct), central t (t), and normal method (z).

Details
For details on the calculations in this function see vignette("IntroTOSTt") & vignette("SMD_calcs").
For two-sample tests, the test is of \( m_1 - m_2 \) (mean of 1 minus mean of 2). For paired samples, the test is of the difference scores (z), wherein \( z = m_1 - m_2 \), and the test is of \( \bar{z} \) (mean of the difference scores). For one-sample tests, the test is of \( \bar{m}_1 \) (mean of group 1).

Value
An S3 object of class "TOSTt" is returned containing the following slots:
- "TOST": A table of class "data.frame" containing two-tailed t-test and both one-tailed results.
- "eqb": A table of class "data.frame" containing equivalence bound settings.
- "effsize": table of class "data.frame" containing effect size estimates.
- "hypothesis": String stating the hypothesis being tested.
- "smd": List containing the results of the standardized mean difference calculations (e.g., Cohen's d).
  - Items include: d (estimate), dlow (lower CI bound), dhigh (upper CI bound), d_df (degrees of freedom for SMD), d_sigma (SE), d_lambda (non-centrality), J (bias correction), smd_label (type of SMD), d_denom (denominator calculation)
• "alpha": Alpha level set for the analysis.
• "method": Type of t-test.
• "decision": List included text regarding the decisions for statistical inference.

See Also
Other TOST: boot_log_TOST(), boot_t_TOST(), simple_hatest(), t_TOST(), wilcoxon_TOST()

Examples

# example code
# One sample test
tsum_TOST(m1 = 0.55, n1 = 18, sd1 = 4, eqb = 2)

twoprop_test
Test of Proportions between 2 Independent Groups

Description

[Maturing]
This is a hypothesis testing function that mimics prop.test, but focuses only on testing differences in proportions between two groups. This function utilizes a z-test to calculate the p-values (may be inaccurate with small sample sizes).

Usage

twoprop_test(
  p1,
  p2,
  n1,
  n2,
  null = NULL,
  alpha = 0.05,
  alternative = c("two.sided", "less", "greater", "equivalence", "minimal.effect"),
  effect_size = c("difference", "odds.ratio", "risk.ratio")
)

Arguments

p1, p2
  Proportions in each respective group.

n1, n2
  Sample size in each respective group.

null
  A number indicating the null hypothesis of the difference in proportions between two groups.

alpha
  Alpha level (default = 0.05)
alternative a character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater", "less", "equivalence" (TOST), or "minimal.effect" (TOST). You can specify just the initial letter.

effect_size the effect size estimate, and confidence intervals, to calculate. Options include the difference between both proportions ("difference"), odds ratio ("odds.ratio"), or risk ratio ("risk.ratio").

Details

The hypothesis test for differences in proportions can be made on the raw proportions scale, the odds ratio, or the risk ratio (details below). This function uses the large sample size asymptotic approximations for both the p-value and confidence interval calculations. There should be a good deal of caution when sample sizes are small. The p-values for the differences in proportions will differ from base prop.test due to the use of the unpooled standard error (see below).

Differences in Proportions:

Differences in proportions test is based on the following calculation:

\[ d = p_1 - p_2 \]

The standard error of \( d \) is calculated as the following:

\[ se(d) = \sqrt{\frac{p_1 \cdot (1 - p_1)}{n_1} + \frac{p_2 \cdot (1 - p_2)}{n_2}} \]

The z-test, with \( d_0 \) being the null value, is then calculated as the following (standard normal distribution evaluated to calculate p-value):

\[ z = \frac{d - d_0}{se(d)} \]

The confidence interval can then be calculated as the following:

\[ d_{lower}, d_{upper} = d \pm z_{\alpha} \cdot se(d) \]

Risk Ratio:

The ratio between proportions test is based on the following calculation:

\[ \phi = \frac{p_1}{p_2} \]

The standard error of \( ln(\phi) \) is calculated as the following:

\[ se(ln(\phi)) = \sqrt{\frac{1 - p_1}{n_1 \cdot p_1} + \frac{1 - p_2}{n_2 \cdot p_2}} \]

The z-test, with \( \phi_0 \) being the null value, is then calculated as the following (standard normal distribution evaluated to calculate p-value):

\[ z = \frac{ln(\phi) - ln(\phi_0)}{se(ln(\phi))} \]
The confidence interval can then be calculated as the following:

\[ \hat{\phi}_{lower} = \phi \cdot e^{-z_\alpha \cdot \text{se}(\ln(\phi))} \]

\[ \hat{\phi}_{upper} = \phi \cdot e^{z_\alpha \cdot \text{se}(\ln(\phi))} \]

**Odds Ratio:**
The ratio between proportions test is based on the following calculation: \((p_1/q_1) / (p_2/q_2)\)

\[ OR = \frac{p_1}{1-p_1} / \frac{p_2}{1-p_2} \]

The standard error of \(\ln(OR)\) is calculated as the following:

\[ \text{se}(\ln(OR)) = \sqrt{\frac{1}{n_1 \cdot p_1 + 0.5} + \frac{1}{n_1 \cdot (1-p_1) + 0.5} + \frac{1}{n_2 \cdot p_2 + 0.5} + \frac{1}{n_2 \cdot (1-p_2) + 0.5}} \]

The z-test, with \(OR_0\) being the null value, is then calculated as the following (standard normal distribution evaluated to calculate p-value):

\[ z = \frac{\ln(OR) - \ln(OR_0)}{\text{se}(\ln(OR))} \]

The confidence interval can then be calculated as the following:

\[ OR_{lower}, OR_{upper} = \exp(\ln(OR) \pm z_\alpha \cdot \text{se}(\ln(OR))) \]

**Value**
An S3 object of the class htest.

**References**


Description

[Stable]
A function for TOST with all types of t-tests.

Usage

```
t_TOST(
  x,
  ...,  
hypothesis = "EQU",  
paired = FALSE,  
var.equal = FALSE,  
eqb,  
low_eqbound,  
high_eqbound,  
eqbound_type = "raw",  
alpha = 0.05,  
bias_correction = TRUE,  
rm_correction = FALSE,  
glass = NULL,  
smd_ci = c("nct", "goulet", "t", "z")
)
```

## Default S3 method:
```
t_TOST(  
  x,
  y = NULL,
  hypothesis = c("EQU", "MET"),  
paired = FALSE,  
var.equal = FALSE,  
eqb,  
low_eqbound,  
high_eqbound,  
eqbound_type = c("raw", "SMD"),  
alpha = 0.05,  
mu = 0,  
bias_correction = TRUE,  
rm_correction = FALSE,  
glass = NULL,  
smd_ci = c("nct", "goulet", "t", "z"),  
...)
```
## S3 method for class 'formula'
\texttt{t\_TOST(formula, data, subset, na.action, ...)}

### Arguments

\begin{itemize}
\item \textbf{x} \quad \text{a (non-empty) numeric vector of data values.}
\item \textbf{...} \quad \text{further arguments to be passed to or from methods.}
\item \textbf{hypothesis} \quad 'EQU' for equivalence (default), or 'MET' for minimal effects test, the alternative hypothesis.
\item \textbf{paired} \quad \text{a logical indicating whether you want a paired t-test.}
\item \textbf{var.equal} \quad \text{a logical variable indicating whether to treat the two variances as being equal. If TRUE then the pooled variance is used to estimate the variance otherwise the Welch (or Satterthwaite) approximation to the degrees of freedom is used.}
\item \textbf{eqb} \quad \text{Equivalence bound. Can provide 1 value (negative value is taken as the lower bound) or 2 specific values that represent the upper and lower equivalence bounds.}
\item \textbf{low_eqbound} \quad \text{lower equivalence bounds (deprecated).}
\item \textbf{high_eqbound} \quad \text{upper equivalence bounds (deprecated).}
\item \textbf{eqbound_type} \quad \text{Type of equivalence bound. Can be set to "SMD" for standardized mean difference (i.e., Cohen's d) or "raw" for the mean difference. Default is "raw". Raw is strongly recommended as SMD bounds will produce biased results.}
\item \textbf{alpha} \quad \text{alpha level (default = 0.05)}
\item \textbf{bias\_correction} \quad \text{Apply Hedges' correction for bias (default is TRUE).}
\item \textbf{rm\_correction} \quad \text{Repeated measures correction to make standardized mean difference Cohen's d(rm). This only applies to repeated/paired samples. Default is FALSE.}
\item \textbf{glass} \quad \text{A option to calculate Glass's delta as an alternative to Cohen's d type SMD. Default is NULL to not calculate Glass's delta, "glass1" will use the first group's SD as the denominator whereas "glass2" will use the 2nd group's SD.}
\item \textbf{smd\_ci} \quad \text{Method for calculating SMD confidence intervals. Methods include Goulet, noncentral t (nct), central t (t), and normal method (z).}
\item \textbf{y} \quad \text{an optional (non-empty) numeric vector of data values.}
\item \textbf{mu} \quad \text{a number indicating the true value of the mean for the two tailed test (or difference in means if you are performing a two sample test).}
\item \textbf{formula} \quad \text{a formula of the form \texttt{lhs ~ rhs} where \texttt{lhs} is a numeric variable giving the data values and \texttt{rhs} either 1 for a one-sample or paired test or a factor with two levels giving the corresponding groups. If \texttt{lhs} is of class "Pair" and \texttt{rhs} is 1, a paired test is done.}
\item \textbf{data} \quad \text{an optional matrix or data frame (or similar: see \texttt{model.frame}) containing the variables in the formula \texttt{formula}. By default the variables are taken from \texttt{environment(formula)}.}
\item \textbf{subset} \quad \text{an optional vector specifying a subset of observations to be used.}
\item \textbf{na.action} \quad \text{a function which indicates what should happen when the data contain NAs. Defaults to getOption("na.action").}
\end{itemize}
Details

For details on the calculations in this function see vignette("IntroTOSTt") & vignette("SMD_calcs").

For two-sample tests, the test is of $\bar{x} - \bar{y}$ (mean of x minus mean of y). For paired samples, the test is of the difference scores (z), wherein $z = x - y$, and the test is of $\bar{z}$ (mean of the difference scores). For one-sample tests, the test is of $\bar{x}$ (mean of x).

Value

An S3 object of class "TOSTt" is returned containing the following slots:

- "TOST": A table of class "data.frame" containing two-tailed t-test and both one-tailed results.
- "eqb": A table of class "data.frame" containing equivalence bound settings.
- "effsize": table of class "data.frame" containing effect size estimates.
- "hypothesis": String stating the hypothesis being tested
- "smd": List containing the results of the standardized mean difference calculations (e.g., Cohen's d).
  - Items include: d (estimate), dlow (lower CI bound), dhigh (upper CI bound), df (degrees of freedom for SMD), d_sigma (SE), d_lambda (non-centrality), J (bias correction), smd_label (type of SMD), d_denom (denominator calculation)
- "alpha": Alpha level set for the analysis.
- "method": Type of t-test.
- "decision": List included text regarding the decisions for statistical inference.

See Also

Other TOST: boot_log_TOST(), boot_t_TOST(), simple_hptest(), tsum_TOST(), wilcox_TOST()

Examples

data(mtcars)
t_TOST(mpg ~ am,
data = mtcars,
eqb = 3)

---

wilcox_TOST  
TOST with Wilcoxon-Mann-Whitney tests

Description

[Stable]

A function for TOST using the non-parametric methods of the Wilcoxon-Mann-Whitney family of tests. This function uses the normal approximation and applies a continuity correction automatically.
Usage

```r
wilcox_TOST(
  x,
  ..., 
  hypothesis = "EQU",
  paired = FALSE,
  eqb,
  low_eqbound,
  high_eqbound,
  ses = "rb",
  alpha = 0.05
)
```

```r
## Default S3 method:
wilcox_TOST(
  x,
  y = NULL,
  hypothesis = "EQU",
  paired = FALSE,
  eqb,
  low_eqbound,
  high_eqbound,
  ses = c("rb", "odds", "cstat"),
  alpha = 0.05,
  mu = 0,
  ...
)
```

```r
## S3 method for class 'formula'
wilcox_TOST(formula, data, subset, na.action, ...)
```

Arguments

- **x**: a (non-empty) numeric vector of data values.
- **...**: further arguments to be passed to or from methods.
- **hypothesis**: 'EQU' for equivalence (default), or 'MET' for minimal effects test, the alternative hypothesis.
- **paired**: a logical indicating whether you want a paired t-test.
- **eqb**: Equivalence bound. Can provide 1 value (negative value is taken as the lower bound) or 2 specific values that represent the upper and lower equivalence bounds.
- **low_eqbound** and **high_eqbound**: lower and upper equivalence bounds (deprecated).
- **ses**: Standardized effect size. Default is "rb" for rank-biserial correlation. Options also include "cstat" for concordance probability, or "odds" for Wilcoxon-Mann-Whitney odds (otherwise known as Agresti’s generalized odds ratio).
- **alpha**: alpha level (default = 0.05)
y an optional (non-empty) numeric vector of data values.
mu number indicating the value around which (a-)symmetry (for one-sample or paired samples) or shift (for independent samples) is to be estimated. See stats::wilcox.test.
formula a formula of the form lhs ~ rhs where lhs is a numeric variable giving the data values and rhs either 1 for a one-sample or paired test or a factor with two levels giving the corresponding groups. If lhs is of class "Pair" and rhs is 1, a paired test is done.
data an optional matrix or data frame (or similar: see model.frame) containing the variables in the formula formula. By default the variables are taken from environment(formula).
subset an optional vector specifying a subset of observations to be used.
na.action a function which indicates what should happen when the data contain NAs. Defaults to getOption("na.action").

details For details on the calculations in this function see vignette("robustTOST").

If only x is given, or if both x and y are given and paired is TRUE, a Wilcoxon signed rank test of the null that the distribution of x (in the one sample case) or of x - y (in the paired two sample case) is symmetric about mu is performed.

Otherwise, if both x and y are given and paired is FALSE, a Wilcoxon rank sum test (equivalent to the Mann-Whitney test: see the Note) is carried out. In this case, the null hypothesis is that the distributions of x and y differ by a location shift.

value An S3 object of class "TOSTnp" is returned containing the following slots:

• "TOST": A table of class "data.frame" containing two-tailed wilcoxon signed rank test and both one-tailed results.
• "eqb": A table of class "data.frame" containing equivalence bound settings.
• "effsize": table of class "data.frame" containing effect size estimates.
• "hypothesis": String stating the hypothesis being tested.
• "smd": List containing information on standardized effect size.
• "alpha": Alpha level set for the analysis.
• "method": Type of non-parametric test.
• "decision": List included text regarding the decisions for statistical inference.


See Also

Other Robust tests: boot_log_TOST(), boot_t_TOST(), boot_t_test(), brunner_munzel(), log_TOST()

Other TOST: boot_log_TOST(), boot_t_TOST(), simple_hptest(), t_TOST(), tsum_TOST()

Examples

data(mtcars)
wilcox_TOST(mpg ~ am,
data = mtcars,
eqb = 3)

---

z_cor_test Test for Association/Correlation Between Paired Samples

Description

[Stable]

Test for association between paired samples, using one of Pearson’s product moment correlation coefficient, Kendall’s τ (tau) or Spearman’s ρ (rho). Unlike the stats version of cor.test, this function allows users to set the null to a value other than zero.

Usage

z_cor_test(
x, y,
alternative = c("two.sided", "less", "greater", "equivalence", "minimal.effect"),
method = c("pearson", "kendall", "spearman"),
alpha = 0.05,
nul = 0
)

Arguments

- x, y: numeric vectors of data values. x and y must have the same length.
- alternative: a character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater", "less", "equivalence" (TOST), or "minimal.effect" (TOST). You can specify just the initial letter.
- method: a character string indicating which correlation coefficient is to be used for the test. One of "pearson", "kendall", or "spearman", can be abbreviated.
- alpha: alpha level (default = 0.05)
- null: a number indicating the null hypothesis. Default is a correlation of zero.
Details
This function uses Fisher's z transformation for the correlations, but uses Fieller's correction of
the standard error for Spearman's $\rho$ and Kendall's $\tau$. See vignette("correlations") for more
details.

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

- "p.value": the p-value of the test.
- "estimate": the estimated measure of association, with name "pb", "wincor", "cor", "tau", or
  "rho" corresponding to the method employed.
- "null.value": the value of the association measure under the null hypothesis.
- "alternative": character string indicating the alternative hypothesis (the value of the input ar-
gument alternative).
- "method": a character string indicating how the association was measured.
- "data.name": a character string giving the names of the data.
- "call": the matched call.

References
Goertzen, J. R., & Cribbie, R. A. (2010). Detecting a lack of association: An equivalence testing ap-
proach. British Journal of Mathematical and Statistical Psychology, 63(3), 527-537. https://doi.org/10.1348/000711009X475853,
formula page 531.

See Also
Other Correlations: `boot_cor_test()`, `corsum_test()`, `plot_cor()`, `power_z_cor()`

Examples
```r
# example code
x <- c(44.4, 45.9, 41.9, 53.3, 44.7, 44.1, 50.7, 45.2, 60.1)
y <- c(2.6, 3.1, 2.5, 5.0, 3.6, 4.0, 5.2, 2.8, 3.8)
# Sig test
z_cor_test(x, y, method = "kendall", alternative = "t", null = 0)
# MET test
z_cor_test(x, y, method = "kendall", alternative = "min", null = .2)
```
Index

* Correlations
  - boot_cor_test, 7
  - corsum_test, 25
  - plot_cor, 41
  - power_z_cor, 54
  - z_cor_test, 84
* Robust tests
  - boot_log_TOST, 8
  - boot_t_TOST, 17
  - brunner_munzel, 20
  - log_TOST, 39
  - wilcox_TOST, 81
* TOST
  - boot_log_TOST, 8
  - boot_t_TOST, 17
  - simple_htest, 58
  - t_TOST, 79
  - tsum_TOST, 74
  - wilcox_TOST, 81
* compare studies
  - boot_compare_cor, 4
  - boot_compare_smd, 5
  - compare_cor, 22
  - compare_smd, 24
* datasets
  - hawthorne, 38
* effect sizes
  - boot_ses_calc, 11
  - boot_smd_calc, 13
  - ses_calc, 57
  - smd_calc, 61
* f-test
  - equ_anova, 34
  - equ_ftest, 36
* htest
  - as_htest, 3
  - htest_helpers, 38
  - simple_htest, 58
* plotting functions
  - plot_cor, 41
  - plot_pes, 42
  - plot_smd, 43
* power
  - power_eq_f, 49
  - power_t_TOST, 52
  - power_z_cor, 54
as.data.frame, 28, 30, 31, 33, 34
as_htest, 3, 39, 60
boot_compare_cor, 4, 6, 24, 25
boot_compare_smd, 5, 5, 24, 25
boot_cor_test, 7, 27, 42, 55, 85
boot_log_TOST, 8, 17, 19, 22, 41, 60, 76, 81, 84
boot_ses_calc, 11, 14, 58, 62
boot_smd_calc, 12, 13, 58, 62
boot_t_TOST, 11, 15, 19, 22, 41, 84
boot_t_TOST, 7, 11, 17, 17, 22, 41, 60, 76, 81, 84
brunner_munzel, 11, 17, 19, 20, 41, 58, 60, 84
compare_cor, 5, 6, 22, 25
compare_smd, 5, 6, 24, 24
corsum_test, 8, 25, 42, 55, 68, 85
dataTOSTone, 27
dataTOSTpaired, 28
dataTOSTr, 30
dataTOSTtwo, 31
datatosttwoprop, 33
describe (TOSTt-methods), 69
describe.TOSTnp (TOSTnp-methods), 64
describe_htest (htest-helpers), 38
df_htest (htest-helpers), 38
equ_anova, 34, 37
equ_ftest, 35, 36
extract_r_paired, 37
INDEX

hawthorne, 38
htest-helpers, 38

log_TOST, 11, 17, 19, 22, 39, 84

np_ses (rbs), 55

plot.TOSTt (TOSTt-methods), 69
plot_cor, 8, 27, 41, 43, 44, 55, 85
plot_pes, 42, 42, 44
plot_smd, 42, 43, 43
power_eq_f, 49, 53, 55
power_t_TOST, 45, 46, 48, 50, 52, 55
power_twoprop, 50
power_z_cor, 8, 27, 42, 50, 53, 54, 85
powerTOSTone, 45
powerTOSTpaired, 46
powerTOSTr (power_z_cor), 54
powerTOSTtwo, 48
powerTOSTtwo.prop (power_twoprop), 50
print.TOSTnp (TOSTnp-methods), 64
print.TOSTt (TOSTt-methods), 69
prop.test, 76, 77

rbs, 55

ses_calc, 12, 14, 55, 57, 62
simple_htest, 3, 11, 19, 39, 58, 76, 81, 84
smd_calc, 12, 14, 58, 61
stats::t.test, 60
stats::wilcox.test, 56, 57, 60, 83

t.test, 20, 58
t_TOST, 11, 19, 60, 76, 79, 84
TOSTmeta, 63
TOSTnp-methods, 64, 64
TOSTone, 65
TOSTpaired, 66
TOSTr, 25, 68
TOSTt-methods, 69
TOSTtwo, 70
TOSTtwo.prop, 72
tsum_TOST, 11, 19, 37, 60, 63, 66, 70, 74, 81, 84
twoprop_test, 72, 76

wilcox.test, 58
wilcox_TOST, 11, 17, 19, 22, 41, 60, 76, 81

z_cor_test, 8, 27, 42, 55, 84