Package ‘TOSTER’

February 5, 2022

Version 0.4.0
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 <arcaldwell49@gmail.com>
URL https://aaroncaldwell.us/TOSTRpkg/
License GPL-3
RoxygenNote 7.1.2
Imports stats, graphics, jmvcore (>= 0.9.6.4), jmv, ggplot2, R6, 
ggdist, distributional, cowplot, tidyr, utils
Suggests knitr, rmarkdown, broom, car, afex, testthat (>= 3.0.0)
VignetteBuilder knitr
Encoding UTF-8
LazyData true
Config/testthat/edition 3
Depends R (>= 3.5)
NeedsCompilation no
Author Daniel Lakens [aut],
   Aaron Caldwell [aut, cre]
Repository CRAN
Date/Publication 2022-02-05 00:30:26 UTC

R topics documented:

  boot_t_TOST .................................................. 2
  dataTOSTone ................................................... 5
  dataTOSTpaired .............................................. 6
  dataTOSTr .................................................... 8
boot_t_TOST

Description

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

boot_t_TOST(x, ...)

## Default S3 method:
boot_t_TOST(
  x,
  y = NULL,
  hypothesis = "EQU",
  paired = FALSE,
  var.equal = FALSE,
  low_eqbound,
  high_eqbound,
  eqbound_type = "raw",
  alpha = 0.05,
  bias_correction = TRUE,
  mu = 0,
  R = 1999,
  ...
)

## S3 method for class 'formula'
boot_t_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.
  low_eqbound            lower equivalence bounds
  high_eqbound           upper equivalence bounds
  eqbound_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).
  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).
  R                      number of bootstrap replicates
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.

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).

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

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

The implemented test(s) corresponds to the proposal of Chapter 16 of Efron and Tibshirani (1993). Returns TOSTt class object with bootstrapped based results. Please note that the repeated measures "corrected" effect size is not available at this time.

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.

dataTOSTone

TOST One Sample T-Test

Description

TOST One Sample T-Test

Usage

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
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$text`: a preformatted
- `results$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)
```

---

**dataTOSTpaired**

*TOST Paired Samples T-Test*

Description

TOST Paired Samples T-Test

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:

results$text a preformatted
results$tost a table
results$eqb a table
results$effsize a table
results$desc a table
results$plots an image
results$indplot an image
results$diffplot an image

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

## Description

TOST Correlation

## 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
)
```

## 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.
**dataTOSTtwo**

*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:

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

---

**dataTOSTtwo**

*TOST Independent Samples T-Test*

**Description**

TOST Independent Samples T-Test

**Usage**

```
dataTOSTtwo(
  data,
  deps,
  group,
  var_equal = FALSE,
  hypothesis = "EQU",
  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$text  a preformatted
  results$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 as.data.frame. For example:

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

References

Welch’s t-test on page 135

Examples

```r
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)
```

### datatosttwoprop

TOST Two Proportions

Description

TOST Two Proportions

Usage

```r
datatosttwoprop(data, var, level, group, hypothesis = "EQU", low_eqbound = -0.1, high_eqbound = 0.1, alpha = 0.05, desc = FALSE, plot = FALSE)
```

Arguments

data

var

level

group

hypothesis

low_eqbound

'\texttt{EQU}' for equivalence (default), or '\texttt{MET}' for minimal effects test, the alternative hypothesis.
a number (default: -0.1) the lower equivalence bounds
equ_anova

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

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

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).

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

equ_ftest
Equivalence Test using an F-test

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

Usage
equ_ftest(Fstat, df1, df2, eqbound, 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 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).

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.
References


---

hawthorne  

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


---

plot_cor  

Function to produce plots of the distribution of standard correlation coefficients

Description

Function to produce plots of the distribution of standard correlation coefficients

Usage

plot_cor(
  r,
  n,
  method = "pearson",
  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"); defualt 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.

---

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

Description

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

Usage

```
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"); defualt option).
- **levels**: Numeric vector of confidence levels to display
plot_smd

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

Description

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

Usage

```
plot_smd(d, df, lambda, 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
- `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.

Description

Power analysis for TOST for one-sample t-test (Cohen’s d).

Usage

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

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)

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 = 0

\[
\text{powerTOSTone}(\alpha=0.05, N=121, \text{statistical\_power}=0.9)
\]

---

**Description**

Power analysis for TOST for one-sample t-test (raw scores).

**Usage**

\[
\text{powerTOSTone.raw}(\alpha, \text{statistical\_power}, N, \text{sd}, \text{low\_eqbound}, \text{high\_eqbound})
\]

**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)
- **sd**: population 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**

\[
\text{## Sample size for alpha = 0.05, 90% power, equivalence bounds of -0.3 and 0.3 in raw units, assuming pooled standard deviation of 1, and assuming true effect = 0}
\text{powerTOSTone.raw(\alpha=0.05, statistical\_power=0.9, sd = 1, low\_eqbound=-0.3, high\_eqbound=0.3)}
\]

\[
\text{## Power for sample size of 121, alpha = 0.05, equivalence bounds of -0.3 and 0.3 in raw units, assuming pooled standard deviation of 1, and assuming true effect = 0}
\text{powerTOSTone.raw(\alpha=0.05, N=121, sd = 1, low\_eqbound=-0.3, high\_eqbound=0.3)}
\]
## 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 analysis for TOST for dependent t-test (Cohen’s dz).*

**Description**

Power analysis for TOST for dependent t-test (Cohen’s dz).

**Usage**

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

**Arguments**

- `alpha`: alpha used for the test (e.g., 0.05)
- `statistical_power`: desired power (e.g., 0.8)
- `N`: number of pairs (e.g., 96)
- `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)

**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**

```
## Sample size for alpha = 0.05, 80% power, equivalence bounds of Cohen’s dz = -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 dz = -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)

powerTOSTpaired.raw  Power analysis for TOST for dependent t-test (raw scores).

Description

Power analysis for TOST for dependent t-test (raw scores).

Usage

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

Arguments

alpha          alpha used for the test (e.g., 0.05)
statistical_power          desired power (e.g., 0.8)
N          number of pairs (e.g., 96)
sdif          standard deviation of the difference scores
low_eqbound          lower equivalence bounds (e.g., -0.5) expressed in raw mean difference
high_eqbound          upper equivalence bounds (e.g., 0.5) expressed in raw mean difference

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 -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, sdiff=10)
```

```r
## 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, sdiff=10)
```

```r
## 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
powerTOSTpaired.raw(alpha=0.05, N=96, statistical_power=0.8, sdiff=10)
```

`powerTOSTr`

Power analysis for TOST for correlations.

Description

Power analysis for TOST for correlations.

Usage

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

Arguments

- `alpha`: alpha used for the test (e.g., 0.05)
- `statistical_power`: desired power (e.g., 0.8)
- `N`: number of pairs (e.g., 96)
- `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

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.

Examples

```r
## 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)
```

```r
## Sample size for alpha = 0.05, N=536, equivalence bounds of
## r = -0.1 and r = 0.1, assuming true effect = 0
```
powerTOSTtwo

Description

Power analysis for TOST for independent t-test (Cohen's d).

Usage

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

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)

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

## 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)

Description

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

Usage

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

Arguments

- `alpha`: alpha used for the test (e.g., 0.05)
- `statistical_power`: desired power (e.g., 0.8)
- `prop1`: expected proportion in control condition
- `prop2`: expected proportion in the experimental condition
- `N`: sample size (e.g., 108)
- `low_eqbound_prop`: lower equivalence bounds (e.g., -0.05) expressed in proportion
- `high_eqbound_prop`: upper equivalence bounds (e.g., 0.05) expressed in proportion

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

## Equivalence bounds for alpha = 0.05, N 542 , assuming true effect prop1 = prop 2 = 0.5,
## and 90% power
powerTOSTtwo.prop(alpha=0.05, statistical_power=0.9, N=542, prop1 = 0.5, prop2 = 0.5)

#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)

# Example 5 from Julious & 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)

# 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
```

Description
Power analysis for TOST for independent t-test (raw scores).
powerTOSTtwo.raw

Usage

powerTOSTtwo.raw(
alpha,
statistical_power,
N,
sd_pooled,
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)
- **sd_pooled**: 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 -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,sd_pooled=350)
```

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

```r
## 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, sd_pooled=350)
```
Power analysis for TOST for an F-test

Description

Power analysis for TOST for an F-test

Usage

\[
\text{power_eq_f}(\alpha = 0.05, \text{df1}, \text{df2}, \text{eqbound})
\]

Arguments

- \(\alpha\): alpha used for the test (e.g., 0.05)
- \(\text{df1}\): Degrees of freedom for the numerator
- \(\text{df2}\): Degrees of freedom for the denominator
- \(\text{eqbound}\): Equivalence bound for the partial eta-squared

Value

Object of class "power.htest"

References


Examples

```r
## 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.
# powerTOST_f(alpha=0.05, df1=3, df2 = 237, eqbound = 0.01)
```
Description

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,
  low_eqbound = NULL,
  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
- `low_eqbound`: The lower equivalence bound (raw units)
- `high_eqbound`: The upper equivalence bound (raw units)
- `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


rbs

**Rank-Biserial Correlation**

Description

Rank-Biserial Correlation

Usage

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

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`.
- `conf.level`: confidence level of the interval.
- `paired`: a logical indicating whether you want to calculate a paired test.

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.

## 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.

# Confidence Intervals
Confidence intervals for the rank-biserial correlation are estimated using the normal approximation (via Fisher’s transformation).

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.
References


---

**TOSTmeta**

*TOST function for meta-analysis*

**Description**

TOST function for meta-analysis

**Usage**

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

**Arguments**

- `ES`  
  meta-analytic effect size
- `var`  
  meta-analytic variance
- `se`  
  standard error
- `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)
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, alpha, low
equivalence bound d, high equivalence bound d, Lower limit confidence interval TOST, Upper limit
certainty interval TOST

References
between two experimental groups. Psychological Bulletin, 113(3), 553, formula page 557.

Examples
## 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 defined for TOSTnp objects

Description
Methods defined for objects returned from the agree functions.

Usage
## S3 method for class 'TOSTnp'
print(x, digits = getOption("digits"), ...)

Arguments

x object of class TOSTnp as returned from the reli_stats function
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 Limits of Agreement
TOSTone

TOST function for a one-sample t-test (Cohen's d)

Description

TOST function for a one-sample t-test (Cohen’s d)

Usage

TOSTone(
  m,  # mean
  mu,  # value to compare against
  sd,  # standard deviation
  n,  # sample size
  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)
  plot = TRUE,  # set whether results should be plotted (plot = TRUE) or not (plot = FALSE) - defaults to TRUE
  verbose = TRUE  # logical variable indicating whether text output should be generated (verbose = TRUE) or not (verbose = FALSE) - default to TRUE
)

Arguments

- `m`: mean
- `mu`: value to compare against
- `sd`: standard deviation
- `n`: sample size
- `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)
- `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 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)
```

---

TOSTone.raw  

TOST function for a one-sample t-test (raw scores)

Description

TOST function for a one-sample t-test (raw scores)

Usage

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

Arguments

- `m`: mean
- `mu`: value to compare against
- `sd`: standard deviation
- `n`: sample size
- `low_eqbound`: lower equivalence bounds (e.g., -0.5) expressed in raw units
- `high_eqbound`: upper equivalence bounds (e.g., 0.5) expressed in raw units
- `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 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
## Test observed mean of 0.52 and standard deviation of 0.52 in sample of 300 participants against 0.5 given equivalence bounds in raw units of −0.1 and 0.1, with an alpha = 0.05.

TOSTraw(m=0.52, mu=0.5, sd=0.5, n=300, low_eqbound=-0.1, high_eqbound=0.1, alpha=0.05)

---

### Description

TOST function for a dependent t-test (Cohen’s dz)

### Usage

```r
TOSTpaired(
  n,  
m1,  
m2,  
sd1,  
sd2,  
r12,  
low_eqbound_dz,  
high_eqbound_dz,  
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
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)
```

Description

TOST function for a dependent t-test (raw scores)

Usage

```r
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
TOSTr

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  lower equivalence bounds (e.g., -0.5) expressed in raw scores
high_eqbound  upper equivalence bounds (e.g., 0.5) expressed in raw scores
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 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

References


Examples

## Test means of 5.83 and 5.75, standard deviations of 1.17 and 1.30 in sample of 65 pairs
## with correlation between observations of 0.745 using equivalence bounds in raw units of
## -0.34 and 0.34, (with default alpha setting of = 0.05).
TOSTpaired.raw(n=65,m1=5.83,m2=5.75,sd1=1.17,sd2=1.30,r12=0.745,low_eqbound=-0.34,high_eqbound=0.34)
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

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 = getOption("digits"), ...)

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

Value

- `print`: Prints short summary of the Limits of Agreement
- `plot`: Returns a plot of the data points used in the reliability analysis

Description

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

Usage

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

- \( m_1 \): mean of group 1
- \( m_2 \): mean of group 2
- \( sd_1 \): standard deviation of group 1
- \( sd_2 \): standard deviation of group 2
- \( n_1 \): sample size in group 1
- \( n_2 \): 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

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)

---

### TOSTtwo.prop

**TOST function for two proportions (raw scores)**

**Description**

TOST function for two proportions (raw scores)

**Usage**

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

TOST function for an independent t-test (raw scores)

Usage

```r
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  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)
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

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

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 alpha = 0.05, 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.raw(m1=5.25,m2=5.22,sd1=0.95,sd2=0.83,n1=95,n2=89,low_eqbound=-0.384,high_eqbound=0.384)

---

**tsum_TOST**  
*TOST* with Summary Statistics

---

**Description**

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

**Usage**

```r
tsum_TOST(
m1,  
sd1,  
n1,  
m2 = NULL,  
sd2 = NULL,  
n2 = NULL,  
r12 = NULL,  
hypothesis = "EQU",  
paired = FALSE,  
var.equal = FALSE,  
low_eqbound,  
high_eqbound,  
mu = 0,  
eqbound_type = "raw",  
alpha = 0.05,  
bias_correction = TRUE,  
rm_correction = FALSE
)
```

**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.
t_TOST

Description

A function for TOST with all types of t-tests.
Usage

t_TOST(
  x,
  ...,
  hypothesis = "EQU",
  paired = FALSE,
  var.equal = FALSE,
  low_eqbound,
  high_eqbound,
  eqbound_type = "raw",
  alpha = 0.05,
  bias_correction = TRUE,
  rm_correction = FALSE
)

## Default S3 method:
## t_TOST(
##   x,
##   y = NULL,
##   hypothesis = "EQU",
##   paired = FALSE,
##   var.equal = FALSE,
##   low_eqbound,
##   high_eqbound,
##   eqbound_type = "raw",
##   alpha = 0.05,
##   mu = 0,
##   bias_correction = TRUE,
##   rm_correction = FALSE,
##   ...
## )

## S3 method for class 'formula'
## t_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.
- **low_eqbound**: lower equivalence bounds
- **high_eqbound**: upper equivalence bounds
eqbound_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.

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

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).

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").

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.
Description

A function for TOST using the non-parametric methods of the Wilcoxon signed rank test. This function uses the normal approximation and applies continuity correction automatically.

Usage

wilcox_TOST(
  x,
  ..., 
  hypothesis = "EQU",
  paired = FALSE,
  low_eqbound,
  high_eqbound,
  alpha = 0.05
)

## Default S3 method:
wilcox_TOST(
  x,
  y = NULL,
  hypothesis = "EQU",
  paired = FALSE,
  low_eqbound,
  high_eqbound,
  alpha = 0.05,
  mu = 0,
  ...
)

## 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 to calculate a paired test.

low_eqbound: lower equivalence bounds.

high_eqbound: upper equivalence bounds.
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").

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.

References


Index

* datasets
  hawthorne, 14
  as.data.frame, 6, 7, 9, 10, 12
  boot_t_TOST, 2
  dataTOSTone, 5
  dataTOSTpaired, 6
  dataTOSTr, 8
  dataTOSTtwo, 9
  datatostttwoprop, 11
  equ_anova, 12
  equ_ftest, 13
  hawthorne, 14
  plot.TOSTt (TOSTt-methods), 36
  plot_cor, 14
  plot_pes, 15
  plot_smd, 16
  power_eq_f, 26
  power_t_TOST, 27
  powerTOSTone, 17
  powerTOSTone.raw, 18
  powerTOSTpaired, 19
  powerTOSTpaired.raw, 20
  powerTOSTr, 21
  powerTOSTtwo, 22
  powerTOSTtwo.prop, 23
  powerTOSTtwo.raw, 24
  print.TOSTnp (TOSTnp-methods), 30
  print.TOSTt (TOSTt-methods), 36

  rbs, 28
  t_TOST, 43
  TOSTmeta, 29
  TOSTnp-methods, 30
  TOSTone, 31

TOSTone.raw, 32
TOSTpaired, 33
TOSTpaired.raw, 34
TOSTr, 35
TOSTt-methods, 36
TOSTtwo, 37
TOSTtwo.prop, 39
TOSTtwo.raw, 40
tsum_TOST, 42
wilcox_TOST, 46