

Package ‘TOSTER’

August 22, 2025

Version 0.8.6

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](https://doi.org/10.1177/1948550617697177)>.

Maintainer Aaron Caldwell <ar Caldwell@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 7.3.2

Language en-US

NeedsCompilation no

Author Daniel Lakens [aut],
Aaron Caldwell [aut, cre]

Repository CRAN

Date/Publication 2025-08-22 19:30:02 UTC

Contents

| | |
|----------------------------|----|
| as_hstest | 3 |
| boot_compare_cor | 4 |
| boot_compare_smd | 7 |
| boot_cor_test | 10 |
| boot_log_TOST | 14 |
| boot_ses_calc | 17 |
| boot_smd_calc | 20 |
| boot_t_test | 24 |
| boot_t_TOST | 28 |
| brunner_munzel | 31 |
| compare_cor | 34 |
| compare_smd | 37 |
| corsum_test | 40 |
| dataTOSTone | 42 |
| dataTOSTpaired | 44 |
| dataTOSTr | 46 |
| dataTOSTtwo | 47 |
| datatosttwoprop | 49 |
| equ_anova | 50 |
| equ_ftest | 52 |
| extract_r_paired | 54 |
| hawthorne | 54 |
| hstest-helpers | 55 |
| log_TOST | 57 |
| plot_cor | 59 |
| plot_pes | 61 |
| plot_smd | 63 |
| powerTOSTone | 66 |
| powerTOSTpaired | 67 |
| powerTOSTtwo | 69 |
| power_eq_f | 71 |
| power_twoprop | 72 |
| power_t_TOST | 75 |
| power_z_cor | 76 |
| rbs | 77 |
| ses_calc | 79 |
| simple_hstest | 82 |
| smd_calc | 85 |
| TOSTmeta | 88 |
| TOSTnp-methods | 89 |
| TOSTone | 90 |
| TOSTpaired | 92 |
| TOSTr | 93 |
| TOSTt-methods | 94 |
| TOSTtwo | 96 |
| TOSTtwo.prop | 98 |

| | |
|------------------------|-----|
| tsum_TOST | 99 |
| twoprop_test | 102 |
| t_TOST | 105 |
| wilcox_TOST | 109 |
| z_cor_test | 112 |

| | |
|--------------|------------|
| Index | 115 |
|--------------|------------|

| | |
|----------|--|
| as_htest | <i>Convert TOSTER Results to Class 'htest'</i> |
|----------|--|

Description

[Maturing]

Converts a TOSTER result object of class 'TOSTt' or 'TOSTnp' to a list of class 'htest', making it compatible with standard R hypothesis testing functions and workflows.

Usage

```
as_htest(TOST)
```

Arguments

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

Details

This function allows you to convert the specialized TOSTER result objects to the standard 'htest' class used by most R hypothesis testing functions (e.g., `t.test()`, `cor.test()`). This enables:

- Integration with other statistical functions that expect 'htest' objects
- Using helper functions like `df_htest()` or `describe_htest()`
- Consistent reporting and interpretation of results

Value

Returns a list of class 'htest' containing the following components:

- **statistic**: The value of the test statistic (t for TOSTt, WMW for TOSTnp).
- **parameter**: The degrees of freedom of the test statistic (df for TOSTt, NULL for TOSTnp).
- **p.value**: The p-value of the test.
- **estimate**: Estimated difference in raw units.
- **null.value**: Equivalence bounds.
- **alternative**: A character string describing the alternative hypothesis ("equivalence" or "minimal.effect").
- **method**: A character string indicating the performed test.
- **data.name**: A character string giving the names of the data.
- **conf.int**: The confidence interval of the difference.

See Also

Other htest: [htest-helpers](#), [simple_htest\(\)](#)

Examples

```
# Example 1: Converting TOST t-test results to htest
res1 <- t_TOST(formula = extra ~ group, data = sleep, eqb = .5, smd_ci = "goulet")
htest_result <- as_htest(res1)
htest_result # Print the htest object

# Example 2: Using the converted result with htest helpers
describe_htest(htest_result)
df_htest(htest_result)

# Example 3: Converting a non-parametric TOST result
res2 <- wilcox_TOST(extra ~ group, data = sleep, eqb = 2)
as_htest(res2)
```

boot_compare_cor

Comparing Correlations Between Independent Studies with Bootstrapping

Description**[Stable]**

A function to compare correlation coefficients between independent studies using bootstrap methods. This function is intended to be used to compare the compatibility of original studies with replication studies (lower p-values indicating lower compatibility).

Usage

```
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 from study 1. x1 and y1 must have the same length. |
| x2, y2 | Numeric vectors of data values from study 2. x2 and y2 must have the same length. |
| alternative | a character string specifying the alternative hypothesis: <ul style="list-style-type: none"> • "two.sided": correlation is not equal to null (default) • "greater": correlation is greater than null • "less": correlation is less than null • "equivalence": correlation is within the equivalence bounds (TOST) • "minimal.effect": correlation is outside the equivalence bounds (TOST) <p>You can specify just the initial letter.</p> |
| method | a character string indicating which correlation coefficient to use: <ul style="list-style-type: none"> • "pearson": standard Pearson product-moment correlation • "kendall": Kendall's tau rank correlation • "spearman": Spearman's rho rank correlation • "winsorized": Winsorized correlation (robust to outliers) • "bendpercent": percentage bend correlation (robust to marginal outliers) <p>Can be abbreviated.</p> |
| alpha | alpha level (default = 0.05) |
| null | a number or vector indicating the null hypothesis value(s): <ul style="list-style-type: none"> • For standard tests: a single value (default = 0) • For equivalence/minimal effect tests: either a single value (symmetric bounds \pmvalue will be created) or a vector of two values representing the lower and upper bounds |
| R | number of bootstrap replications (default = 1999). |
| ... | Additional arguments passed to the correlation functions. |

Details

This function tests for differences between correlation coefficients from independent studies using bootstrap resampling methods. Unlike the `compare_cor` function, which uses Fisher's z transformation or the Kraatz method with summary statistics, this function works with raw data and uses bootstrapping to estimate confidence intervals and p-values.

It is particularly useful for:

- Comparing correlations when assumptions for parametric tests may not be met
- Obtaining robust confidence intervals for the difference between correlations
- Comparing an original study with its replication using raw data
- Testing if correlations from different samples are equivalent

The function supports multiple correlation methods:

- Standard correlation coefficients (Pearson, Kendall, Spearman)
- Robust correlation measures (Winsorized, percentage bend)

The function also supports both standard hypothesis testing and equivalence/minimal effect testing:

- For standard tests (two.sided, less, greater), the function tests whether the difference between correlations differs from the null value (typically 0).
- For equivalence testing ("equivalence"), it determines whether the difference falls within the specified bounds, which can be set asymmetrically.
- For minimal effect testing ("minimal.effect"), it determines whether the difference falls outside the specified bounds.

When performing equivalence or minimal effect testing:

- If a single value is provided for null, symmetric bounds \pm value will be used
- If two values are provided for null, they will be used as the lower and upper bounds

Value

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

- **p.value**: The p-value for the test under the null hypothesis.
- **parameter**: Sample sizes from each study.
- **conf.int**: Bootstrap confidence interval for the difference in correlations.
- **estimate**: Difference in correlations between studies.
- **stderr**: Standard error of the difference (estimated from bootstrap distribution).
- **null.value**: The specified hypothesized value(s) for the null hypothesis.
- **alternative**: Character string indicating the alternative hypothesis.
- **method**: Description of the correlation method used.
- **data.name**: Names of the input data vectors.
- **boot_res**: List containing the bootstrap samples for the difference and individual correlations.
- **call**: The matched call.

See Also

Other compare studies: [boot_compare_smd\(\)](#), [compare_cor\(\)](#), [compare_smd\(\)](#)

Examples

```
# Example 1: Comparing Pearson correlations (standard test)
set.seed(123)
x1 <- rnorm(30)
y1 <- x1 * 0.6 + rnorm(30, 0, 0.8)
x2 <- rnorm(25)
y2 <- x2 * 0.3 + rnorm(25, 0, 0.9)

# Two-sided test with Pearson correlation (use fewer bootstraps for example)
```

```

boot_compare_cor(x1, y1, x2, y2, method = "pearson",
                 alternative = "two.sided", R = 500)

# Example 2: Testing for equivalence with Spearman correlation
# Testing if the difference in correlations is within ±0.2
boot_compare_cor(x1, y1, x2, y2, method = "spearman",
                 alternative = "equivalence", null = 0.2, R = 500)

# Example 3: Testing with robust correlation measure
# Using percentage bend correlation for non-normal data
boot_compare_cor(x1, y1, x2, y2, method = "bendpercent",
                 alternative = "greater", R = 500)

# Example 4: Using asymmetric bounds for equivalence testing
boot_compare_cor(x1, y1, x2, y2, method = "pearson",
                 alternative = "equivalence", null = c(-0.1, 0.3), R = 500)

```

| | |
|------------------|--|
| boot_compare_smd | <i>Comparing Standardized Mean Differences (SMDs) Between Independent Studies with Bootstrapping</i> |
|------------------|--|

Description

[Stable]

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

Usage

```

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

```

Arguments

| | |
|----|---|
| x1 | A numeric vector of data values from study 1 (first group for two-sample designs, or the only group for one-sample/paired designs). |
|----|---|

| | |
|-------------|--|
| y1 | An optional numeric vector of data values from study 1 (second group for two-sample designs, or second measurement for paired designs). Set to NULL for one-sample designs. |
| x2 | A numeric vector of data values from study 2 (first group for two-sample designs, or the only group for one-sample/paired designs). |
| y2 | An optional numeric vector of data values from study 2 (second group for two-sample designs, or second measurement for paired designs). Set to NULL for one-sample designs. |
| null | A number or vector indicating the null hypothesis value(s): <ul style="list-style-type: none"> • For standard tests: a single value representing the null difference (default = 0) • For equivalence/minimal effect tests: either a single value (symmetric bounds \pmvalue will be created) or a vector of two values representing the lower and upper bounds |
| paired | A logical indicating whether the SMD is from a paired or independent samples design. If a one-sample design, then paired should be set to TRUE. |
| alternative | A character string specifying the alternative hypothesis: <ul style="list-style-type: none"> • "two.sided": difference is not equal to null (default) • "greater": difference is greater than null • "less": difference is less than null • "equivalence": difference is within the equivalence bounds (TOST) • "minimal.effect": difference is outside the equivalence bounds (TOST) <p>You can specify just the initial letter.</p> |
| R | Number of bootstrap replications (default = 1999). |
| alpha | Alpha level (default = 0.05). |

Details

This function tests for differences between standardized mean differences (SMDs) from independent studies using bootstrap resampling methods. Unlike the `compare_smd` function, which works with summary statistics, this function works with raw data and uses bootstrapping to estimate confidence intervals and p-values.

The function supports both paired/one-sample designs and independent samples designs:

- For **paired/one-sample designs** (`paired = TRUE`):
 - If `y1` and `y2` are provided, the function calculates differences between paired measures
 - If `y1` and `y2` are NULL, the function treats `x1` and `x2` as one-sample data
 - SMDs are calculated as Cohen's d_z (mean divided by standard deviation of differences)
- For **independent samples designs** (`paired = FALSE`):
 - Requires `x1`, `y1`, `x2`, and `y2` (first and second groups for both studies)
 - If `y1` and `y2` are NULL, the function treats `x1` and `x2` as one-sample data with `paired = TRUE`
 - SMDs are calculated as Cohen's d_s (mean difference divided by pooled standard deviation)

The function supports both standard hypothesis testing and equivalence/minimal effect testing:

- For standard tests (two.sided, less, greater), the function tests whether the difference between SMDs differs from the null value (typically 0).
- For equivalence testing ("equivalence"), it determines whether the difference falls within the specified bounds, which can be set asymmetrically.
- For minimal effect testing ("minimal.effect"), it determines whether the difference falls outside the specified bounds.

When performing equivalence or minimal effect testing:

- If a single value is provided for null, symmetric bounds \pm value will be used
- If two values are provided for null, they will be used as the lower and upper bounds

The bootstrap procedure follows these steps:

1. Calculate SMDs for both studies using the original data
2. Calculate the difference between SMDs and its standard error
3. Generate R bootstrap samples by resampling with replacement
4. Calculate SMDs and their difference for each bootstrap sample
5. Calculate test statistics for each bootstrap sample
6. Calculate confidence intervals using the percentile method
7. Compute p-values by comparing the observed test statistics to their bootstrap distributions

Note on p-value calculation: The function uses the bootstrap distribution of test statistics (z-scores) rather than the raw differences to calculate p-values. This approach is analogous to traditional hypothesis testing and estimates the probability of obtaining test statistics as extreme as those observed in the original data under repeated sampling.

Value

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

- **statistic:** z-score (observed) with name "z (observed)"
- **p.value:** The p-value for the test under the null hypothesis
- **conf.int:** Bootstrap confidence interval for the difference in SMDs
- **estimate:** Difference in SMD between studies
- **null.value:** The specified hypothesized value(s) for the null hypothesis
- **alternative:** Character string indicating the alternative hypothesis
- **method:** Description of the SMD type and design used
- **df_ci:** Data frame containing confidence intervals for the difference and individual SMDs
- **boot_res:** List containing the bootstrap samples for SMDs, their difference, and test statistics
- **data.name:** "Bootstrapped" to indicate bootstrap methods were used
- **call:** The matched call

See Also

Other compare studies: [boot_compare_cor\(\)](#), [compare_cor\(\)](#), [compare_smd\(\)](#)

Examples

```
# Example 1: Comparing two independent samples SMDs (standard test)
set.seed(123)
# Study 1 data
x1 <- rnorm(30, mean = 0)
y1 <- rnorm(30, mean = 0.5, sd = 1)
# Study 2 data
x2 <- rnorm(25, mean = 0)
y2 <- rnorm(25, mean = 0.3, sd = 1)

# Two-sided test for independent samples (use fewer bootstraps for example)
boot_compare_smd(x1, y1, x2, y2, paired = FALSE,
                 alternative = "two.sided", R = 99)

# Example 2: Testing for equivalence between SMDs
# Testing if the difference between SMDs is within ±0.2
boot_compare_smd(x1, y1, x2, y2, paired = FALSE,
                 alternative = "equivalence", null = 0.2, R = 99)

# Example 3: Testing for minimal effects
# Testing if the difference between SMDs is outside ±0.3
boot_compare_smd(x1, y1, x2, y2, paired = FALSE,
                 alternative = "minimal.effect", null = 0.3, R = 99)

# Example 4: Comparing paired samples SMDs
# Study 1 data (pre-post measurements)
pre1 <- rnorm(20, mean = 10, sd = 2)
post1 <- rnorm(20, mean = 12, sd = 2)
# Study 2 data (pre-post measurements)
pre2 <- rnorm(25, mean = 10, sd = 2)
post2 <- rnorm(25, mean = 11, sd = 2)

# Comparing paired designs
boot_compare_smd(x1 = pre1, y1 = post1, x2 = pre2, y2 = post2,
                 paired = TRUE, alternative = "greater", R = 99)

# Example 5: Using asymmetric bounds for equivalence testing
boot_compare_smd(x1, y1, x2, y2, paired = FALSE,
                 alternative = "equivalence", null = c(-0.1, 0.3), R = 99)
```

Description**[Stable]**

A function for bootstrap-based correlation tests using various correlation coefficients including Pearson's, Kendall's, Spearman's, Winsorized, and percentage bend correlations. This function supports standard, equivalence, and minimal effect testing with robust bootstrap methods.

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: <ul style="list-style-type: none"> • "two.sided": correlation is not equal to null (default) • "greater": correlation is greater than null • "less": correlation is less than null • "equivalence": correlation is within the equivalence bounds (TOST) • "minimal.effect": correlation is outside the equivalence bounds (TOST) <p>You can specify just the initial letter.</p> |
| method | a character string indicating which correlation coefficient to use: <ul style="list-style-type: none"> • "pearson": standard Pearson product-moment correlation • "kendall": Kendall's tau rank correlation • "spearman": Spearman's rho rank correlation • "winsorized": Winsorized correlation (robust to outliers) • "bendpercent": percentage bend correlation (robust to marginal outliers) <p>Can be abbreviated.</p> |
| alpha | alpha level (default = 0.05) |
| null | a number or vector indicating the null hypothesis value(s): <ul style="list-style-type: none"> • For standard tests: a single value (default = 0) • For equivalence/minimal effect tests: either a single value (symmetric bounds \pmvalue will be created) or a vector of two values representing the lower and upper bounds |

| | |
|---------|---|
| boot_ci | type of bootstrap confidence interval: <ul style="list-style-type: none"> • "basic": basic/empirical bootstrap CI • "perc": percentile bootstrap CI (default) |
| R | number of bootstrap replications (default = 1999). |
| ... | additional arguments passed to correlation functions, such as: <ul style="list-style-type: none"> • tr: trim for Winsorized correlation (default = 0.2) • beta: for percentage bend correlation (default = 0.2) |

Details

This function uses bootstrap methods to calculate correlation coefficients and their confidence intervals. P-values are calculated from a re-sampled null distribution.

The bootstrap correlation methods in this package offer two robust correlations beyond the standard methods:

1. **Winsorized correlation:** Replaces extreme values with less extreme values before calculating the correlation. The `trim` parameter (default: `tr = 0.2`) determines the proportion of data to be Winsorized.
2. **Percentage bend correlation:** A robust correlation that downweights the influence of outliers. The `beta` parameter (default = 0.2) determines the bending constant.

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".

The function supports both standard hypothesis testing and equivalence/minimal effect testing:

- For standard tests (`two.sided`, `less`, `greater`), the function tests whether the correlation differs from the null value (typically 0).
- For equivalence testing ("`equivalence`"), it determines whether the correlation falls within the specified bounds, which can be set asymmetrically.
- For minimal effect testing ("`minimal.effect`"), it determines whether the correlation falls outside the specified bounds.

When performing equivalence or minimal effect testing:

- If a single value is provided for `null`, symmetric bounds \pm value will be used
- If two values are provided for `null`, they will be used as the lower and upper bounds

See `vignette("correlations")` for more details.

Value

A list with class "`hstest`" containing the following components:

- **p.value:** the bootstrap p-value of the test.
- **parameter:** the number of observations used in the test.
- **conf.int:** a bootstrap confidence interval for the correlation coefficient.

- **estimate**: the estimated correlation coefficient, with name "cor", "tau", "rho", "pb", or "win-cor" corresponding to the method employed.
- **stderr**: the bootstrap standard error of the correlation coefficient.
- **null.value**: the value(s) of the correlation under the null hypothesis.
- **alternative**: character string indicating the alternative hypothesis.
- **method**: a character string indicating which bootstrapped correlation was measured.
- **data.name**: a character string giving the names of the data.
- **boot_res**: vector of bootstrap correlation estimates.
- **call**: the matched call.

References

Wilcox, R.R. (2009) Comparing Pearson Correlations: Dealing with Heteroscedasticity and Non-normality. *Communications in Statistics - Simulation and Computation*, 38, 2220–2234.

Wilcox, R.R. (2017) *Introduction to Robust Estimation and Hypothesis Testing*, 4th edition. Academic Press.

See Also

Other Correlations: [corsum_test\(\)](#), [plot_cor\(\)](#), [power_z_cor\(\)](#), [z_cor_test\(\)](#)

Examples

```
# Example 1: Standard bootstrap test with Pearson correlation
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)
boot_cor_test(x, y, method = "pearson", alternative = "two.sided",
              R = 999) # Fewer replicates for example

# Example 2: Equivalence test with Spearman correlation
# Testing if correlation is equivalent to zero within ±0.3
boot_cor_test(x, y, method = "spearman", alternative = "equivalence",
              null = 0.3, R = 999)

# Example 3: Using robust correlation methods
# Using Winsorized correlation with custom trim
boot_cor_test(x, y, method = "winsorized", tr = 0.1,
              R = 999)

# Example 4: Using percentage bend correlation
boot_cor_test(x, y, method = "bendpercent", beta = 0.2,
              R = 999)

# Example 5: Minimal effect test with asymmetric bounds
# Testing if correlation is outside bounds of -0.1 and 0.4
boot_cor_test(x, y, method = "pearson", alternative = "minimal.effect",
              null = c(-0.1, 0.4), R = 999)
```

boot_log_TOST

*Bootstrapped TOST with Log Transformed t-tests***Description****[Stable]**

Performs equivalence testing using the Two One-Sided Tests (TOST) procedure with bootstrapped log-transformed t-tests. This approach is particularly useful for ratio-scale data where the equivalence bounds are expressed as ratios (e.g., bioequivalence studies).

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 positive data values on a ratio scale. |
| ... | further arguments to be passed to or from methods. |
| y | an optional (non-empty) numeric vector of positive data values on a ratio scale. |
| hypothesis | 'EQU' for equivalence (default), or 'MET' for minimal effects test. |
| 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. |
| eqb | Equivalence bound expressed as a ratio. Can provide 1 value (e.g., 1.25 for bounds of 0.8 and 1.25) or 2 specific values that represent the lower and upper equivalence bounds (e.g., c(0.8, 1.25)). |
| alpha | alpha level (default = 0.05). |

| | |
|-----------|---|
| null | the ratio value under the null hypothesis (default = 1). |
| boot_ci | method for bootstrap confidence interval calculation: "stud" (studentized, default), "basic" (basic bootstrap), or "perc" (percentile bootstrap). |
| R | number of bootstrap replications (default = 1999). |
| 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 indicating what should happen when the data contain NAs. |

Details

The function implements a bootstrap method for log-transformed TOST as recommended by He et al. (2022) and corresponds to the proposal in Chapter 16 of Efron and Tibshirani (1994). This is approximately equivalent to the percentile bootstrap method mentioned by He et al. (2014).

For two-sample tests, the test is of $\bar{\log}(x) - \bar{\log}(y)$ which corresponds to testing the ratio of geometric means. For paired samples, the test is of difference scores on the log scale, $z = \log(x) - \log(y) = \log(x/y)$, which also corresponds to a ratio test.

The bootstrap procedure follows these steps:

- Log-transform the data
- Perform resampling with replacement to generate bootstrap samples
- For each bootstrap sample, calculate test statistics and effect sizes
- Use the distribution of bootstrap results to compute p-values and confidence intervals
- Back-transform for the ratio of means

Note that all input data must be positive (ratio scale with a true zero) since log transformation is applied. The function will stop with an error if any negative values are detected.

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.

Purpose

Use this function when:

- Your data is on a ratio scale (all values must be positive)
- You want to establish equivalence based on the ratio of means rather than their difference
- Traditional parametric methods may not be appropriate due to skewed distributions
- You need to analyze bioequivalence data where bounds are expressed as ratios

References

Efron, B., & Tibshirani, R. J. (1994). An introduction to the bootstrap. CRC press.

He, Y., Deng, Y., You, C., & Zhou, X. H. (2022). Equivalence tests for ratio of means in bioequivalence studies under crossover design. *Statistical Methods in Medical Research*, 09622802221093721.

Food and Drug Administration (2014). Bioavailability and Bioequivalence Studies Submitted in NDAs or INDs — General Considerations. Center for Drug Evaluation and Research. Docket: FDA-2014-D-0204. <https://www.fda.gov/regulatory-information/search-fda-guidance-documents/bioavailability-and-bioequivalence-studies-submitted-ndas-or-ind-general-considerations>

See Also

Other Robust tests: [boot_t_TOST\(\)](#), [boot_t_test\(\)](#), [brunner_munzel\(\)](#), [log_TOST\(\)](#), [wilcox_TOST\(\)](#)

Other TOST: [boot_t_TOST\(\)](#), [simple_hstest\(\)](#), [t_TOST\(\)](#), [tsum_TOST\(\)](#), [wilcox_TOST\(\)](#)

Examples

```
# Example 1: Two-Sample Test for Bioequivalence
# Generate ratio scale data (e.g., drug concentrations)
test_group <- rlnorm(30, meanlog = 3.5, sdlog = 0.4)
ref_group <- rlnorm(30, meanlog = 3.6, sdlog = 0.4)

# FDA standard bioequivalence bounds (80% to 125%)
result <- boot_log_TOST(x = test_group,
                       y = ref_group,
                       eqb = 1.25, # Creates bounds of 0.8 and 1.25
                       R = 999)   # Reduce for demonstration

# Example 2: Paired Sample Test
# Generate paired ratio scale data
n <- 20
```



```

baseline <- rlnorm(n, meanlog = 4, sdlog = 0.3)
followup <- baseline * rlnorm(n, meanlog = 0.05, sdlog = 0.2)

# Test with asymmetric bounds
result <- boot_log_TOST(x = followup,
                       y = baseline,
                       paired = TRUE,
                       eqb = c(0.85, 1.20),
                       boot_ci = "perc")

```

boot_ses_calc

Bootstrapped Standardized Effect Size (SES) Calculation

Description

[Maturing]

Calculates non-SMD standardized effect sizes with bootstrap confidence intervals. This function provides more robust confidence intervals for rank-based and probability-based effect size measures through resampling methods.

Usage

```

boot_ses_calc(
  x,
  ...,
  paired = FALSE,
  ses = "rb",
  alpha = 0.05,
  boot_ci = c("basic", "stud", "perc"),
  R = 1999
)

## 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,
  ...
)

## 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. |
| ses | a character string specifying the effect size measure to calculate: - "rb": rank-biserial correlation (default) - "odds": Wilcoxon-Mann-Whitney odds - "logodds": Wilcoxon-Mann-Whitney log-odds - "cstat": concordance statistic (C-statistic/AUC) |
| alpha | alpha level (default = 0.05) |
| boot_ci | method for bootstrap confidence interval calculation: "stud" (studentized, default), "basic" (basic bootstrap), or "perc" (percentile bootstrap). |
| R | number of bootstrap replications (default = 1999). |
| 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 calculates bootstrapped confidence intervals for rank-based and probability-based effect size measures. It is an extension of the `ses_calc()` function that uses resampling to provide more robust confidence intervals, especially for small sample sizes.

The function implements the following bootstrap approach:

- Calculate the raw effect size using the original data
- Create R bootstrap samples by resampling with replacement from the original data
- Calculate the effect size for each bootstrap sample
- Apply the Fisher z-transformation to stabilize variance for rank-biserial correlation values
- Calculate confidence intervals using the specified method
- Back-transform the confidence intervals to the original scale
- Convert to the requested effect size measure (if not rank-biserial)

Three bootstrap confidence interval methods are available:

- **Basic bootstrap ("basic")**: Uses the empirical distribution of bootstrap estimates
- **Studentized bootstrap ("stud")**: Accounts for the variability in standard error estimates

- **Percentile bootstrap ("perc")**: Uses percentiles of the bootstrap distribution directly

The function supports three study designs:

- One-sample design: Compares a single sample to a specified value
- Two-sample independent design: Compares two independent groups
- Paired samples design: Compares paired observations

Note that extreme values (perfect separation between groups) can produce infinite values during the bootstrapping process. This happens often if the sample size is very small. The function will issue a warning if this occurs, as it may affect the accuracy of the confidence intervals. Additionally, this affects the ability to calculate bias and SE estimates from the bootstrap samples. If the number of infinite values is small (less than 10% of the bootstrap samples) then the infinite values are replaced with the nearest next value (only for the SE and bias estimates, not confidence intervals).

For detailed information on calculation methods, see `vignette("robustTOST")`.

Value

A data frame containing the following information:

- `estimate`: The effect size estimate calculated from the original data
- `bias`: Estimated bias (difference between original estimate and median of bootstrap estimates)
- `SE`: Standard error estimated from the bootstrap distribution
- `lower.ci`: Lower bound of the bootstrap confidence interval
- `upper.ci`: Upper bound of the bootstrap confidence interval
- `conf.level`: Confidence level (1-alpha)
- `boot_ci`: The bootstrap confidence interval method used

Purpose

Use this function when:

- You need more robust confidence intervals for non-parametric effect sizes
- You prefer resampling-based confidence intervals over asymptotic approximations
- You need to quantify uncertainty in rank-based effect sizes more accurately

See Also

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

Examples

```
# Example 1: Independent groups comparison with basic bootstrap CI
set.seed(123)
group1 <- c(1.2, 2.3, 3.1, 4.6, 5.2, 6.7)
group2 <- c(3.5, 4.8, 5.6, 6.9, 7.2, 8.5)

# Use fewer bootstrap replicates for a quick example
```

```

result <- boot_ses_calc(x = group1, y = group2,
                      ses = "rb",
                      boot_ci = "basic",
                      R = 99)

# Example 2: Using formula notation to calculate concordance statistic
data(mtcars)
result <- boot_ses_calc(formula = mpg ~ am,
                      data = mtcars,
                      ses = "cstat",
                      boot_ci = "perc",
                      R = 99)

# Example 3: Paired samples with studentized bootstrap CI
data(sleep)
with(sleep, boot_ses_calc(x = extra[group == 1],
                        y = extra[group == 2],
                        paired = TRUE,
                        ses = "rb",
                        boot_ci = "stud",
                        R = 99))

# Example 4: Comparing different bootstrap CI methods
## Not run:
# Basic bootstrap
basic_ci <- boot_ses_calc(x = group1, y = group2, boot_ci = "basic")

# Percentile bootstrap
perc_ci <- boot_ses_calc(x = group1, y = group2, boot_ci = "perc")

# Studentized bootstrap
stud_ci <- boot_ses_calc(x = group1, y = group2, boot_ci = "stud")

# Compare the results
rbind(basic_ci, perc_ci, stud_ci)

## End(Not run)

```

boot_smd_calc

Bootstrapped Standardized Mean Difference (SMD) Calculation

Description

[Maturing]

Calculates standardized mean differences (SMDs) with bootstrap confidence intervals. This function provides more robust confidence intervals for Cohen's *d*, Hedges' *g*, and other SMD measures through resampling methods.

Usage

```

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:
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'
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. |
| 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 | Option to calculate Glass's delta instead of Cohen's d style SMD ('glass1' uses first group's SD, 'glass2' uses second group's SD). |
| boot_ci | method for bootstrap confidence interval calculation: "stud" (studentized, default), "basic" (basic bootstrap), or "perc" (percentile bootstrap). |
| R | number of bootstrap replications (default = 1999). |
| y | an optional (non-empty) numeric vector of data values. |
| mu | null value to adjust the calculation. If non-zero, the function calculates $x-y-\mu$ (default = 0). |
| 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 indicating what should happen when the data contain NAs. |

Details

This function calculates bootstrapped confidence intervals for standardized mean differences. It is an extension of the `smd_calc()` function that uses resampling to provide more robust confidence intervals, especially for small sample sizes or when data violate assumptions of parametric methods.

The function implements the following bootstrap approach:

- Calculate the raw SMD and its standard error using the original data
- Create R bootstrap samples by resampling with replacement from the original data
- Calculate the SMD and its standard error for each bootstrap sample
- Calculate confidence intervals using the specified method

Three bootstrap confidence interval methods are available:

- **Studentized bootstrap ("stud")**: Accounts for the variability in standard error estimates. Usually provides the most accurate coverage probability and is set as the default.
- **Basic bootstrap ("basic")**: Uses the empirical distribution of bootstrap estimates. Simple approach that works well for symmetric distributions.
- **Percentile bootstrap ("perc")**: Uses percentiles of the bootstrap distribution directly. More robust to skewness in the bootstrap distribution.

The function supports various SMD variants:

- Classic standardized mean difference (`bias_correction = FALSE`)
- Bias-corrected version (`bias_correction = TRUE`)
- Glass's delta: Uses only one group's standard deviation as the denominator (`glass = "glass1"` or `"glass2"`)

- Repeated measures d: Accounts for correlation in paired designs (`rm_correction = TRUE`)

The function supports three study designs:

- One-sample design: Standardizes the difference between the sample mean and zero (or other specified value)
- Two-sample independent design: Standardizes the difference between two group means
- Paired samples design: Standardizes the mean difference between paired observations

For detailed information on calculation methods, see `vignette("SMD_calcs")`.

Value

A data frame containing the following information:

- `estimate`: The SMD calculated from the original data
- `bias`: Estimated bias (difference between original estimate and median of bootstrap estimates)
- `SE`: Standard error estimated from the bootstrap distribution
- `lower.ci`: Lower bound of the bootstrap confidence interval
- `upper.ci`: Upper bound of the bootstrap confidence interval
- `conf.level`: Confidence level (1-alpha)
- `boot_ci`: The bootstrap confidence interval method used

Purpose

Use this function when:

- You need more robust confidence intervals for standardized mean differences
- You want to account for non-normality or heterogeneity in your effect size estimates
- Sample sizes are small or standard error approximations may be unreliable
- You prefer resampling-based confidence intervals over parametric approximations
- You need to quantify uncertainty in SMD estimates more accurately

See Also

Other effect sizes: [boot_ses_calc\(\)](#), [ses_calc\(\)](#), [smd_calc\(\)](#)

Examples

```
# Example 1: Independent groups comparison with studentized bootstrap CI
set.seed(123)
group1 <- rnorm(30, mean = 100, sd = 15)
group2 <- rnorm(30, mean = 110, sd = 18)

# Use fewer bootstrap replicates for a quick example
result <- boot_smd_calc(x = group1, y = group2,
                       boot_ci = "stud",
                       R = 999)
```

```

# Example 2: Using formula notation with basic bootstrap and Hedges' g
df <- data.frame(
  value = c(group1, group2),
  group = factor(rep(c("A", "B"), each = 30))
)
result <- boot_smd_calc(formula = value ~ group,
  data = df,
  boot_ci = "basic",
  bias_correction = TRUE,
  R = 999)

# Example 3: Paired samples with percentile bootstrap
set.seed(456)
before <- rnorm(30)
after <- rnorm(30)
result <- boot_smd_calc(x = before,
  y = after,
  paired = TRUE,
  boot_ci = "perc",
  R = 999)

# Example 4: Glass's delta with homogeneous variances
set.seed(456)
control <- rnorm(25, mean = 50, sd = 10)
treatment <- rnorm(25, mean = 60, sd = 10)
result <- boot_smd_calc(x = control,
  y = treatment,
  glass = "glass1",
  boot_ci = "stud",
  R = 999)

```

boot_t_test

Bootstrapped t-test

Description

[Stable]

Performs t-tests with bootstrapped p-values and confidence intervals. This function supports standard hypothesis testing alternatives as well as equivalence and minimal effect testing, all with the familiar htest output structure.

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 the underlying test functions. |
| 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 | the alternative hypothesis: * "two.sided": different from mu (default) * "less": less than mu * "greater": greater than mu * "equivalence": between specified bounds * "minimal.effect": outside specified bounds |
| mu | a number or vector specifying the null hypothesis value(s): * For standard alternatives: a single value (default = 0) * For equivalence/minimal.effect: two values representing the lower and upper bounds |
| alpha | alpha level (default = 0.05) |
| boot_ci | method for bootstrap confidence interval calculation: "stud" (studentized, default), "basic" (basic bootstrap), or "perc" (percentile bootstrap). |
| R | number of bootstrap replications (default = 1999). |
| 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 performs bootstrapped t-tests, providing more robust inference than standard parametric t-tests. It supports one-sample, two-sample (independent), and paired designs, as well as five different alternative hypotheses.

The bootstrap procedure follows these steps:

- Calculate the test statistic from the original data
- Generate R bootstrap samples by resampling with replacement
- Calculate the test statistic for each bootstrap sample
- Compute the p-value by comparing the original test statistic to the bootstrap distribution
- Calculate confidence intervals using the specified bootstrap method

Three bootstrap confidence interval methods are available:

- *Studentized bootstrap* ("*stud*"): Accounts for the variability in standard error estimates
- *Basic bootstrap* ("*basic*"): Uses the empirical distribution of bootstrap estimates
- *Percentile bootstrap* ("*perc*"): Uses percentiles of the bootstrap distribution directly

For different alternatives, the p-values are calculated as follows:

- "two.sided": Proportion of bootstrap statistics at least as extreme as the observed statistic (in either direction), multiplied by 2
- "less": Proportion of bootstrap statistics less than or equal to the observed statistic
- "greater": Proportion of bootstrap statistics greater than or equal to the observed statistic
- "equivalence": Maximum of two one-sided p-values (for lower and upper bounds)
- "minimal.effect": Minimum of two one-sided p-values (for lower and upper bounds)

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).

Unlike the `t_TOST` function, this function returns a standard `htest` object for compatibility with other R functions, while still providing the benefits of bootstrapping.

For detailed information on calculation methods, see `vignette("robustTOST")`.

Value

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

- "p.value": the bootstrapped p-value for the test.
- "stderr": the bootstrapped standard error.
- "conf.int": a bootstrapped confidence interval for the mean appropriate to the specified alternative hypothesis.
- "estimate": the estimated mean or difference in means.
- "null.value": the specified hypothesized value(s) of the mean or mean difference.
- "alternative": a character string describing the alternative hypothesis.

- "method": a character string indicating what type of bootstrapped t-test was performed.
- "boot": the bootstrap samples of the mean or mean difference.
- "data.name": a character string giving the name(s) of the data.
- "call": the matched call.

Purpose

Use this function when:

- You need more robust inference than provided by standard t-tests
- Your data don't meet the assumptions of normality or homogeneity
- You want to perform equivalence or minimal effect testing with bootstrap methods
- Sample sizes are small or standard parametric approaches may be unreliable
- You prefer the standard htest output format for compatibility with other R functions

References

Efron, B., & Tibshirani, R. J. (1994). An introduction to the bootstrap. CRC press.

See Also

Other Robust tests: [boot_log_TOST\(\)](#), [boot_t_TOST\(\)](#), [brunner_munzel\(\)](#), [log_TOST\(\)](#), [wilcox_TOST\(\)](#)

Examples

```
# Example 1: Basic two-sample test with formula notation
data(sleep)
result <- boot_t_test(extra ~ group, data = sleep)
result # Standard htest output format

# Example 2: One-sample bootstrapped t-test
set.seed(123)
x <- rnorm(20, mean = 0.5, sd = 1)
boot_t_test(x, mu = 0, R = 999) # Using fewer replicates for demonstration

# Example 3: Paired samples test with percentile bootstrap CI
before <- c(5.1, 4.8, 6.2, 5.7, 6.0, 5.5, 4.9, 5.8)
after <- c(5.6, 5.2, 6.7, 6.1, 6.5, 5.8, 5.3, 6.2)
boot_t_test(x = before, y = after,
            paired = TRUE,
            alternative = "less", # Testing if before < after
            boot_ci = "perc",
            R = 999)

# Example 4: Equivalence testing with bootstrapped t-test
# Testing if the effect is within ±0.5 units
data(mtcars)
boot_t_test(mpg ~ am, data = mtcars,
            alternative = "equivalence",
            mu = c(-0.5, 0.5),
```

```

boot_ci = "stud",
R = 999)

# Example 5: Minimal effect testing with bootstrapped t-test
# Testing if the effect is outside ±3 units
boot_t_test(mpg ~ am, data = mtcars,
            alternative = "minimal.effect",
            mu = c(-3, 3),
            R = 999)

```

boot_t_TOST

Bootstrapped TOST with t-tests

Description

[Stable]

Performs equivalence testing using the Two One-Sided Tests (TOST) procedure with bootstrapped t-tests. This provides a robust alternative to traditional TOST when data may not meet all parametric assumptions.

Usage

```

boot_t_TOST(x, ...)

## 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"),
  ...
)

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

```

Arguments

| | |
|------------------------------|---|
| <code>x</code> | a (non-empty) numeric vector of data values. |
| <code>...</code> | further arguments to be passed to or from methods. |
| <code>y</code> | an optional (non-empty) numeric vector of data values. |
| <code>hypothesis</code> | 'EQU' for equivalence (default), or 'MET' for minimal effects test. |
| <code>paired</code> | a logical indicating whether you want a paired t-test. |
| <code>var.equal</code> | 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. |
| <code>eqb</code> | Equivalence bound. Can provide 1 value (symmetric bound, negative value is taken as the lower bound) or 2 specific values that represent the upper and lower equivalence bounds. |
| <code>low_eqbound</code> | lower equivalence bounds (deprecated, use eqb instead). |
| <code>high_eqbound</code> | upper equivalence bounds (deprecated, use eqb instead). |
| <code>eqbound_type</code> | Type of equivalence bound. Can be '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. |
| <code>alpha</code> | alpha level (default = 0.05) |
| <code>bias_correction</code> | Apply Hedges' correction for bias (default is TRUE). |
| <code>rm_correction</code> | Repeated measures correction to make standardized mean difference Cohen's d(rm). This only applies to repeated/paired samples. Default is FALSE. |
| <code>glass</code> | Option to calculate Glass's delta instead of Cohen's d style SMD ('glass1' uses first group's SD, 'glass2' uses second group's SD). |
| <code>mu</code> | a number indicating the true value of the mean for the two-tailed test (default = 0). |
| <code>R</code> | number of bootstrap replications (default = 1999). |
| <code>boot_ci</code> | method for bootstrap confidence interval calculation: "stud" (studentized, default), "basic" (basic bootstrap), or "perc" (percentile bootstrap). |
| <code>formula</code> | 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. |
| <code>data</code> | 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). |
| <code>subset</code> | an optional vector specifying a subset of observations to be used. |
| <code>na.action</code> | a function indicating what should happen when the data contain NAs. |

Details

The function implements a bootstrap method for TOST as described in Chapter 16 of Efron and Tibshirani (1994). This approach provides a robust alternative to traditional parametric TOST when data distributions may not meet standard assumptions.

The bootstrap procedure follows these steps:

- Resample with replacement from the original data to create R bootstrap samples
- For each bootstrap sample, calculate test statistics and effect sizes
- Use the distribution of bootstrap results to compute p-values and confidence intervals
- Combine results using the specified bootstrap confidence interval method

Three types of bootstrap confidence intervals are available:

- Studentized ("stud"): Accounts for the variability in the standard error estimate
- Basic/Empirical ("basic"): Uses the empirical distribution of bootstrap estimates
- Percentile ("perc"): Uses percentiles of the bootstrap distribution

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 for SMD and raw effect sizes.

Purpose

Use this function when:

- You want more robust confidence intervals for your effect sizes
- Sample sizes are small and parametric assumptions may not hold
- You want to avoid relying on asymptotic approximations

References

Efron, B., & Tibshirani, R. J. (1994). An introduction to the bootstrap. CRC press.

See Also

Other Robust tests: `boot_log_TOST()`, `boot_t_test()`, `brunner_munzel()`, `log_TOST()`, `wilcox_TOST()`

Other TOST: `boot_log_TOST()`, `simple_hstest()`, `t_TOST()`, `tsum_TOST()`, `wilcox_TOST()`

Examples

```
## Not run:
# Example 1: Two-Sample Test with Symmetric Bounds
set.seed(1234)
group1 <- rnorm(30, mean = 5, sd = 2)
group2 <- rnorm(30, mean = 5.5, sd = 2.2)

# Using symmetric bounds of ±1.5
result <- boot_t_TOST(x = group1,
                     y = group2,
                     eqb = 1.5,
                     R = 999) # Using fewer replications for demonstration

# Example 2: Paired Sample Test with Percentile Bootstrap
set.seed(5678)
pre <- rnorm(25, mean = 100, sd = 15)
post <- pre + rnorm(25, mean = 3, sd = 10)

result <- boot_t_TOST(x = pre,
                     y = post,
                     paired = TRUE,
                     eqb = c(-5, 8), # Asymmetric bounds
                     boot_ci = "perc")

# Example 3: One Sample Test
set.seed(9101)
scores <- rnorm(40, mean = 0.3, sd = 1)

# Testing if mean is equivalent to zero within ±0.5 units
result <- boot_t_TOST(x = scores,
                     eqb = 0.5,
                     boot_ci = "basic")

## End(Not run)
```

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

| | |
|--------------------------|--|
| <code>x</code> | a (non-empty) numeric vector of data values. |
| <code>...</code> | further arguments to be passed to or from methods. |
| <code>paired</code> | a logical indicating whether you want a paired test. |
| <code>alternative</code> | a character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater" or "less". You can specify just the initial letter. |
| <code>mu</code> | 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, $p = P(X < Y) + 0.5 * P(X = Y)$; See 'Details'. |
| <code>alpha</code> | 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

Brunner, E., Munzel, U. (2000). The Nonparametric Behrens-Fisher Problem: Asymptotic Theory and a Small Sample Approximation. *Biometrical Journal* 42, 17 -25.

Neubert, K., Brunner, E., (2006). A Studentized Permutation Test for the Nonparametric Behrens-Fisher Problem. *Computational Statistics and Data Analysis*.

Munzel, U., Brunner, E. (2002). An Exact Paired Rank Test. *Biometrical Journal* 44, 584-593.

Konietschke, F., Placzek, M., Schaarschmidt, F., & Hothorn, L. A. (2015). nparcomp: an R software package for nonparametric multiple comparisons and simultaneous confidence intervals. *Journal of Statistical Software* 64 (2015), Nr. 9, 64(9), 1-17. <http://www.jstatsoft.org/v64/i09/>

See Also

Other Robust tests: [boot_log_TOST\(\)](#), [boot_t_TOST\(\)](#), [boot_t_test\(\)](#), [log_TOST\(\)](#), [wilcox_TOST\(\)](#)

Examples

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

compare_cor

Comparing Two Independent Correlation Coefficients

Description

[Stable]

A function to compare correlations between independent 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

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

Arguments

| | |
|-------------|--|
| r1 | Correlation from study 1. |
| df1 | Degrees of freedom from study 1 (if a simple correlation the df is N-2). |
| r2 | Correlation from study 2. |
| df2 | Degrees of freedom from study 2 (if a simple correlation the df is N-2). |
| method | Method for determining differences: <ul style="list-style-type: none"> • "fisher": uses Fisher's z transformation (default) • "kraatz": uses the Kraatz method |
| alternative | A character string specifying the alternative hypothesis: <ul style="list-style-type: none"> • "two.sided": difference is not equal to null (default) • "greater": difference is greater than null • "less": difference is less than null • "equivalence": difference is within the equivalence bounds (TOST) • "minimal.effect": difference is outside the equivalence bounds (TOST) <p>You can specify just the initial letter.</p> |
| null | A number or vector indicating the null hypothesis value(s): <ul style="list-style-type: none"> • For standard tests: a single value representing the null difference (default = 0) • For equivalence/minimal effect tests: either a single value (symmetric bounds \pmvalue will be created) or a vector of two values representing the lower and upper bounds |

Details

This function tests for differences between correlation coefficients from independent studies. It is particularly useful for:

- Comparing an original study with its replication
- Meta-analytic comparisons between studies
- Testing if correlations from different samples are equivalent

The function offers two methods for comparing correlations:

1. **Fisher's z transformation** (default): Transforms correlations to stabilize variance
2. **Kraatz method**: Uses a direct approach that may be more appropriate for larger correlations

The function supports both standard hypothesis testing and equivalence/minimal effect testing:

- For standard tests (two.sided, less, greater), the function tests whether the difference between correlations differs from the null value (typically 0).
- For equivalence testing ("equivalence"), it determines whether the difference falls within the specified bounds, which can be set asymmetrically.
- For minimal effect testing ("minimal.effect"), it determines whether the difference falls outside the specified bounds.

When performing equivalence or minimal effect testing:

- If a single value is provided for null, symmetric bounds \pm value will be used
- If two values are provided for null, they will be used as the lower and upper bounds

Value

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

- **statistic**: z-score with name "z"
- **p.value**: numeric scalar containing the p-value for the test under the null hypothesis
- **estimate**: difference in correlation coefficients between studies
- **null.value**: the specified hypothesized value(s) for the null hypothesis
- **alternative**: character string indicating the alternative hypothesis
- **method**: description of the method used for comparison
- **data.name**: "Summary Statistics" to denote summary statistics were utilized
- **cor**: list containing the correlation coefficients used in the comparison
- **call**: the matched call

References

Counsell, A., & Cribbie, R. A. (2015). Equivalence tests for comparing correlation and regression coefficients. *The British journal of mathematical and statistical psychology*, 68(2), 292-309. <https://doi.org/10.1111/bmsp.12045>

Anderson, S., & Hauck, W. W. (1983). A new procedure for testing equivalence in comparative bioavailability and other clinical trials. *Communications in Statistics-Theory and Methods*, 12(23), 2663-2692.

See Also

Other compare studies: [boot_compare_cor\(\)](#), [boot_compare_smd\(\)](#), [compare_smd\(\)](#)

Examples

```
# Example 1: Comparing two correlations (standard test)
compare_cor(r1 = 0.45, df1 = 48, r2 = 0.25, df2 = 58,
            method = "fisher", alternative = "two.sided")

# Example 2: Testing for equivalence between correlations
# Testing if the difference between correlations is within ±0.15
compare_cor(r1 = 0.42, df1 = 38, r2 = 0.38, df2 = 42,
            method = "fisher", alternative = "equivalence", null = 0.15)

# Example 3: Testing for minimal effects using Kraatz method
# Testing if the difference between correlations is outside ±0.2
compare_cor(r1 = 0.53, df1 = 28, r2 = 0.22, df2 = 32,
            method = "kraatz", alternative = "minimal.effect", null = 0.2)
```

```
# Example 4: One-sided test (are correlations different in a specific direction?)
compare_cor(r1 = 0.65, df1 = 48, r2 = 0.45, df2 = 52,
            method = "fisher", alternative = "greater")

# Example 5: Using asymmetric bounds for equivalence testing
compare_cor(r1 = 0.35, df1 = 48, r2 = 0.25, df2 = 52,
            method = "fisher", alternative = "equivalence", null = c(-0.05, 0.2))
```

| | |
|-------------|---|
| compare_smd | <i>Comparing Standardized Mean Differences (SMDs) Between Independent Studies</i> |
|-------------|---|

Description

[Stable]

A function to compare standardized mean differences (SMDs) between independent 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

```
compare_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 a single number (total sample size) or a vector of 2 numbers (group sizes) for independent samples designs. |
| se1, se2 | User supplied standard errors (SEs). This will override the internal calculations for the standard error. |
| 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: |

- "two.sided": difference is not equal to null (default)
- "greater": difference is greater than null
- "less": difference is less than null
- "equivalence": difference is within the equivalence bounds (TOST)
- "minimal.effect": difference is outside the equivalence bounds (TOST)

You can specify just the initial letter.

| | |
|------|---|
| null | A number or vector indicating the null hypothesis value(s): <ul style="list-style-type: none"> • For standard tests: a single value representing the null difference (default = 0) • For equivalence/minimal effect tests: either a single value (symmetric bounds \pmvalue will be created) or a vector of two values representing the lower and upper bounds |
| 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). It is particularly useful for:

- Comparing effect sizes between an original study and its replication
- Meta-analytic comparisons between studies
- Testing if effect sizes from different samples are equivalent

The function handles both paired/one-sample designs and independent samples designs:

- For paired/one-sample designs (`paired = TRUE`), standard errors are calculated for Cohen's d_z , and `n1` and `n2` must be single values.
- For independent samples designs (`paired = FALSE`), standard errors are calculated for Cohen's d_s , and `n1` and `n2` can be either single values (total sample size) or vectors of length 2 (group sizes).
- For all other SMDs, you should supply your own standard errors using the `se1` and `se2` arguments.

The function supports both standard hypothesis testing and equivalence/minimal effect testing:

- For standard tests (`two.sided`, `less`, `greater`), the function tests whether the difference between SMDs differs from the null value (typically 0).
- For equivalence testing ("`equivalence`"), it determines whether the difference falls within the specified bounds, which can be set asymmetrically.
- For minimal effect testing ("`minimal.effect`"), it determines whether the difference falls outside the specified bounds.

When performing equivalence or minimal effect testing:

- If a single value is provided for `null`, symmetric bounds \pm value will be used
- If two values are provided for `null`, they will be used as the lower and upper bounds

Value

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

- **statistic**: z-score with name "z"
- **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(s) for the null hypothesis
- **alternative**: character string indicating the alternative hypothesis
- **method**: description of the method used for comparison
- **data.name**: "Summary Statistics" to denote summary statistics were utilized
- **smd**: list containing the SMDs used in the comparison
- **sample_sizes**: list containing the sample sizes used in the comparison
- **call**: the matched call

See Also

Other compare studies: [boot_compare_cor\(\)](#), [boot_compare_smd\(\)](#), [compare_cor\(\)](#)

Examples

```
# Example 1: Comparing two independent samples SMDs (standard test)
compare_smd(smd1 = 0.5, n1 = c(30, 30),
            smd2 = 0.3, n2 = c(25, 25),
            paired = FALSE, alternative = "two.sided")

# Example 2: Comparing two paired samples SMDs
compare_smd(smd1 = 0.6, n1 = 40,
            smd2 = 0.4, n2 = 45,
            paired = TRUE, alternative = "two.sided")

# Example 3: Testing for equivalence between SMDs
# Testing if the difference between SMDs is within ±0.2
compare_smd(smd1 = 0.45, n1 = c(25, 25),
            smd2 = 0.35, n2 = c(30, 30),
            paired = FALSE, alternative = "equivalence", null = 0.2)

# Example 4: Testing for minimal effects
# Testing if the difference between SMDs is outside ±0.3
compare_smd(smd1 = 0.7, n1 = 30,
            smd2 = 0.3, n2 = 35,
            paired = TRUE, alternative = "minimal.effect", null = 0.3)

# Example 5: Using asymmetric bounds for equivalence testing
compare_smd(smd1 = 0.45, n1 = c(30, 30),
            smd2 = 0.35, n2 = c(25, 25),
            paired = FALSE, alternative = "equivalence", null = c(-0.1, 0.3))

# Example 6: Using user-supplied standard errors
```

```
compare_smd(smd1 = 0.5, n1 = 50, se1 = 0.15,
            smd2 = 0.7, n2 = 45, se2 = 0.16,
            paired = TRUE, alternative = "two.sided")
```

corsum_test

Association/Correlation Test from Summary Statistics

Description

[Stable]

Test for association between paired samples using only the correlation coefficient and sample size. Supports Pearson's product moment correlation, Kendall's τ (tau), or Spearman's ρ (rho). This is the updated version of the TOSTr function.

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 | correlation coefficient (the estimated value) |
| n | sample size (number of pairs) |
| alternative | a character string specifying the alternative hypothesis: <ul style="list-style-type: none"> "two.sided": correlation is not equal to null (default) "greater": correlation is greater than null "less": correlation is less than null "equivalence": correlation is within the equivalence bounds (TOST) "minimal.effect": correlation is outside the equivalence bounds (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 or vector indicating the null hypothesis value(s): <ul style="list-style-type: none"> For standard tests: a single value (default = 0) For equivalence/minimal effect tests: either a single value (symmetric bounds \pmvalue will be created) or a vector of two values representing the lower and upper bounds |

Details

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

Unlike `z_cor_test`, which requires raw data, this function only needs the correlation value and sample size. This is particularly useful when:

- You only have access to summary statistics (correlation coefficient and sample size)
- You want to reanalyze published results within an equivalence testing framework

The function supports both standard hypothesis testing and equivalence/minimal effect testing:

- For standard tests (two.sided, less, greater), the function tests whether the correlation differs from the null value (typically 0).
- For equivalence testing ("equivalence"), it determines whether the correlation falls within the specified bounds, which can be set asymmetrically.
- For minimal effect testing ("minimal.effect"), it determines whether the correlation falls outside the specified bounds.

When performing equivalence or minimal effect testing:

- If a single value is provided for `null`, symmetric bounds \pm value will be used
- If two values are provided for `null`, they will be used as the lower and upper bounds

Value

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

- **statistic**: z -score with name "z".
- **p.value**: the p -value of the test.
- **parameter**: the sample size with name "N".
- **conf.int**: a confidence interval for the correlation appropriate to the specified alternative hypothesis.
- **estimate**: the estimated correlation coefficient, with name "cor", "tau", or "rho" corresponding to the method employed.
- **stderr**: the standard error of the test statistic.
- **null.value**: the value(s) of the correlation coefficient under the null hypothesis.
- **alternative**: character string indicating the alternative hypothesis.
- **method**: a character string indicating how the correlation 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 approach. *British Journal of Mathematical and Statistical Psychology*, 63(3), 527-537. <https://doi.org/10.1348/000711009X475> formula page 531.

See Also

Other Correlations: [boot_cor_test\(\)](#), [plot_cor\(\)](#), [power_z_cor\(\)](#), [z_cor_test\(\)](#)

Examples

```
# Example 1: Standard significance test for Pearson correlation
corsum_test(r = 0.45, n = 30, method = "pearson", alternative = "two.sided")
```

```
# Example 2: Equivalence test for Spearman correlation
# Testing if correlation is equivalent to zero within ±0.3
corsum_test(r = 0.15, n = 40, method = "spearman",
            alternative = "equivalence", null = 0.3)
```

```
# Example 3: Minimal effect test for Kendall's tau
# Testing if correlation is meaningfully different from ±0.25
corsum_test(r = 0.42, n = 50, method = "kendall",
            alternative = "minimal.effect", null = 0.25)
```

```
# Example 4: One-sided test with non-zero null
# Testing if correlation is greater than 0.3
corsum_test(r = 0.45, n = 35, method = "pearson",
            alternative = "greater", null = 0.3)
```

```
# Example 5: Using asymmetric bounds for equivalence testing
corsum_test(r = 0.1, n = 60, method = "pearson",
            alternative = "equivalence", null = c(-0.2, 0.3))
```

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

```
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 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:

```

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

```

Examples

```
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 | <i>TOST Paired Samples T-Test</i> |
|----------------|-----------------------------------|

Description

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

Usage

```
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

| | |
|---------------------------|---|
| <code>data</code> | the data as a data frame |
| <code>pair1</code> | A string naming the first part of the pair |
| <code>pair2</code> | A string naming the second part of the pair |
| <code>hypothesis</code> | 'EQU' for equivalence (default), or 'MET' for minimal effects test, the alternative hypothesis. |
| <code>low_eqbound</code> | a number (default: 0.5) the lower equivalence bounds |
| <code>high_eqbound</code> | 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 html |
| 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
as.data.frame(results$tost)
```

References

Mara, C. A., & Cribbie, R. A. (2012). Paired-Samples Tests of Equivalence. *Communications in Statistics - Simulation and Computation*, 41(10), 1928-1943. formula page 1932. Note there is a typo in the formula: n-1 should be n (personal communication, 31-08-2016)

Examples

```
library("TOSTER")

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

dataTOSTr

*TOST Correlation***Description**

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

Usage

```
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. |
| 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 | <i>TOST Independent Samples T-Test</i> |
|-------------|--|

Description

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

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

| | |
|---------------------------|---|
| <code>data</code> | the data as a data frame |
| <code>deps</code> | a vector of strings naming dependent variables in data |
| <code>group</code> | a string naming the grouping variable in data; must have two levels |
| <code>var_equal</code> | TRUE or FALSE (default), assume equal variances |
| <code>hypothesis</code> | 'EQU' for equivalence (default), or 'MET' for minimal effects test, the alternative hypothesis. |
| <code>low_eqbound</code> | a number (default: -0.5) the lower equivalence/MET bounds |
| <code>high_eqbound</code> | a number (default: 0.5) the upper equivalence/MET bounds |
| <code>eqbound_type</code> | '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 `as.data.frame`. For example:

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

References

Berger, R. L., & Hsu, J. C. (1996). Bioequivalence Trials, Intersection-Union Tests and Equivalence Confidence Sets. *Statistical Science*, 11(4), 283-302.

Gruman, J. A., Cribbie, R. A., & Arpin-Cribbie, C. A. (2007). The effects of heteroscedasticity on tests of equivalence. *Journal of Modern Applied Statistical Methods*, 6(1), 133-140, formula for Welch's t-test on page 135

Examples

```
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)
```

datatostwoprop *TOST Two Proportions*

Description

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

Usage

```
datatostwoprop(
  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 | '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\$text | a html |
| 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 or minimal effect testing on the partial eta-squared (η^2) value from ANOVA results to determine if effects are practically equivalent to zero or meaningfully different from zero.

Usage

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

Arguments

| | |
|----------------------|---|
| <code>object</code> | An object returned by either <code>Anova</code> , <code>aov</code> , or <code>afex_aov</code> . |
| <code>eqbound</code> | Equivalence bound for the partial eta-squared. This value represents the smallest effect size considered meaningful or practically significant. |
| <code>MET</code> | Logical indicator to perform a minimal effect test rather than equivalence test (default is <code>FALSE</code>). When <code>TRUE</code> , the alternative hypothesis becomes that the effect is larger than the equivalence bound. |
| <code>alpha</code> | Alpha level used for the test (default = 0.05). |

Details

This function tests whether ANOVA effects are practically equivalent to zero (when `MET = FALSE`) or meaningfully different from zero (when `MET = TRUE`) using the approach described by Campbell & Lakens (2021).

The function works by:

1. Extracting ANOVA results from the input object
2. Converting the equivalence bound for partial eta-squared to a non-centrality parameter
3. Performing an equivalence test or minimal effect test for each effect in the ANOVA

For equivalence tests (`MET = FALSE`), a significant result ($p < \alpha$) indicates that the effect is statistically equivalent to zero (smaller than the equivalence bound).

For minimal effect tests (`MET = TRUE`), a significant result ($p < \alpha$) indicates that the effect is meaningfully different from zero (larger than the equivalence bound).

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 columns are included 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.value**: F-value.
- **p.null**: p-value for the traditional null hypothesis test (probability of the data given the null hypothesis).
- **pes**: Partial eta-squared measure of effect size.
- **eqbound**: Equivalence bound used for testing.
- **p.equ**: p-value for the equivalence or minimal effect test.

References

Campbell, H., & Lakens, D. (2021). Can we disregard the whole model? Omnibus non-inferiority testing for R² in multi-variable linear regression and in ANOVA. *British Journal of Mathematical and Statistical Psychology*, 74(1), 64-89. doi: 10.1111/bmsp.12201

See Also

Other f-test: [equ_ftest\(\)](#)

Examples

```
# One-way ANOVA
data(iris)
anova_result <- aov(Sepal.Length ~ Species, data = iris)

# Equivalence test with bound of 0.1
equ_anova(anova_result, eqbound = 0.1)

# Minimal effect test with bound of 0.1
equ_anova(anova_result, eqbound = 0.1, MET = TRUE)

# Two-way ANOVA with lower equivalence bound
anova_result2 <- aov(Sepal.Length ~ Species * Petal.Width, data = iris)
equ_anova(anova_result2, eqbound = 0.05)
```

equ_ftest *Equivalence Test using an F-test*

Description

[Stable]

Performs equivalence or minimal effect testing on the partial eta-squared (pes) value using an F-test. This function provides a low-level interface that works directly with F statistics rather than ANOVA objects.

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 (effect degrees of freedom). |
| df2 | Degrees of freedom for the denominator (error degrees of freedom). |
| eqbound | Equivalence bound for the partial eta-squared. This value represents the smallest effect size considered meaningful or practically significant. |
| eqb | Defunct argument for equivalence bound, use eqbound instead. |
| MET | Logical indicator to perform a minimal effect test rather than equivalence test (default is FALSE). When TRUE, the alternative hypothesis becomes that the effect is larger than the equivalence bound. |
| alpha | Alpha level used for the test (default = 0.05). |

Details

This function tests whether an effect is practically equivalent to zero (when MET = FALSE) or meaningfully different from zero (when MET = TRUE) using the approach described by Campbell & Lakens (2021).

The function works by:

1. Converting the F-statistic to a partial eta-squared value
2. Converting the equivalence bound for partial eta-squared to a non-centrality parameter
3. Computing the confidence interval for the partial eta-squared
4. Performing an equivalence test or minimal effect test based on the non-central F distribution

For equivalence tests (MET = FALSE), a significant result ($p < \alpha$) indicates that the effect is statistically equivalent to zero (smaller than the equivalence bound).

For minimal effect tests (MET = TRUE), a significant result ($p < \alpha$) indicates that the effect is meaningfully different from zero (larger than the equivalence bound).

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

Value

Object of class "htest" containing the following components:

- **statistic**: The value of the F-statistic with name "F".
- **parameter**: The degrees of freedom for the F-statistic (df1 and df2).
- **p.value**: The p-value for the equivalence or minimal effect test.
- **conf.int**: A confidence interval for the partial eta-squared statistic.
- **estimate**: Estimate of partial eta-squared.
- **null.value**: The specified equivalence bound.
- **alternative**: NULL (not used in this test).
- **method**: A string indicating the type of test ("Equivalence Test from F-test" or "Minimal Effect Test from F-test").
- **data.name**: A string indicating that this was calculated from summary statistics.

References

Campbell, H., & Lakens, D. (2021). Can we disregard the whole model? Omnibus non-inferiority testing for R2 in multi-variable linear regression and in ANOVA. *British Journal of Mathematical and Statistical Psychology*, 74(1), 64-89. doi: 10.1111/bmsp.12201

See Also

Other f-test: [equ_anova\(\)](#)

Examples

```
# Example 1: Equivalence test with a small effect
# F = 2.5, df1 = 2, df2 = 100, equivalence bound = 0.1
equ_ftest(Fstat = 2.5, df1 = 2, df2 = 100, eqbound = 0.1)

# Example 2: Minimal effect test with a large effect
# F = 12, df1 = 3, df2 = 80, equivalence bound = 0.1
equ_ftest(Fstat = 12, df1 = 3, df2 = 80, eqbound = 0.1, MET = TRUE)

# Example 3: Equivalence test with a very small effect
# F = 0.8, df1 = 1, df2 = 50, equivalence bound = 0.05
equ_ftest(Fstat = 0.8, df1 = 1, df2 = 50, eqbound = 0.05)
```

| | |
|------------------|-----------------------------------|
| extract_r_paired | <i>Extract Paired Correlation</i> |
|------------------|-----------------------------------|

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 (not required for one-sample tests). |
| sd2 | standard deviation of group 2 (not required for one-sample tests). |
| 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

Lajeunesse, M. J. (2011). On the meta-analysis of response ratios for studies with correlated and multi-group designs. *Ecology*, 92(11), 2049-2055

| | |
|-----------|-------------|
| hawthorne | <i>Data</i> |
|-----------|-------------|

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

McCambridge, J., Wilson, A., Attia, J., Weaver, N., & Kypri, K. (2019). Randomized trial seeking to induce the Hawthorne effect found no evidence for any effect on self-reported alcohol consumption online. *Journal of Clinical Epidemiology*, 108, 102–109.

 hstest-helpers

Helper Functions for Working with 'hstest' Objects

Description

A collection of utility functions designed to help interpret, display, and standardize information from objects of class `'hstest'` (hypothesis test results). These functions make it easier to extract, format, and report statistical results from various test functions in R.

Usage

```
df_hstest(hstest, test_statistics = TRUE, show_ci = TRUE, extract_names = TRUE)
```

```
describe_hstest(hstest, alpha = NULL, digits = 3)
```

Arguments

| | |
|------------------------------|---|
| <code>hstest</code> | An S3 object of class <code>'hstest'</code> , such as (but not limited to) output from <code>t.test()</code> , <code>cor.test()</code> , <code>wilcox.test()</code> , or TOSTER functions converted with <code>as_hstest()</code> . |
| <code>test_statistics</code> | A logical variable indicating whether to display the test statistics in the output (default = TRUE). |
| <code>show_ci</code> | A logical variable indicating whether to display the confidence interval in the output (default = TRUE). |
| <code>extract_names</code> | A logical variable indicating whether to take the names from the S3 object (i.e., statistic for <code>t.test()</code> would be "t") (default = TRUE). |
| <code>alpha</code> | The significance level to use for determining statistical significance. If NULL (default), it will be extracted from the confidence interval of the <code>hstest</code> object or default to 0.05. |
| <code>digits</code> | Integer indicating the number of decimal places to display in the output (default = 3). |

Details

The package provides two main helper functions:

1. `df_hstest()`: Converts an 'hstest' object to a data frame with standardized columns, making it easier to combine multiple test results or export them for further analysis.
2. `describe_hstest()`: Generates a formatted text description of the test results, following APA style guidelines and providing a complete statistical report with test statistics, p-values, effect sizes, and confidence intervals.

These functions work with standard R hypothesis tests (e.g., `t.test()`, `wilcox.test()`, `cor.test()`) as well as TOSTER-specific tests that have been converted to 'hstest' format using the `as_hstest()` function.

Value

- `df_hstest()`: Returns a data frame containing the formatted test information.
- `describe_hstest()`: Returns a character string with a formatted description of the test results.

See Also

Other hstest: [as_hstest\(\)](#), [simple_hstest\(\)](#)

Examples

```
# Example 1: Working with a standard t-test
t_result <- t.test(extra ~ group, data = sleep)

# Convert to data frame
df_hstest(t_result)

# Generate formatted description
describe_hstest(t_result)

# Example 2: Working with a TOST result
tost_result <- t_TOST(extra ~ group, data = sleep, eqb = 1)
hstest_conv <- as_hstest(tost_result)
describe_hstest(hstest_conv)

# Example 3: Customizing output format
df_hstest(t_result, test_statistics = TRUE, show_ci = FALSE)
describe_hstest(t_result, alpha = 0.01, digits = 2)

# Example 4: Working with correlation tests
cor_result <- cor.test(mtcars$mpg, mtcars$wt)
df_hstest(cor_result)
describe_hstest(cor_result)
```

| | |
|----------|--|
| log_TOST | <i>TOST with log transformed t-tests</i> |
|----------|--|

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

This approach is particularly useful for:

- Bioequivalence studies where FDA guidelines recommend ratio-based bounds
- Data with a multiplicative nature, where ratio comparisons are more meaningful
- Skewed data where log transformation helps normalize the residuals

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

He, Y., Deng, Y., You, C., & Zhou, X. H. (2022). Equivalence tests for ratio of means in bioequivalence studies under crossover design. *Statistical Methods in Medical Research*, 09622802221093721.

Food and Drug Administration (2014). Bioavailability and Bioequivalence Studies Submitted in NDAs or INDs — General Considerations. Center for Drug Evaluation and Research. Docket: FDA-2014-D-0204. <https://www.fda.gov/regulatory-information/search-fda-guidance-documents/bioavailability-and-bioequivalence-studies-submitted-ndas-or-inds-general-considerations>

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]

Creates consonance plots (confidence curves and/or consonance density functions) for correlation coefficients, allowing visualization of uncertainty around correlation estimates.

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: <ul style="list-style-type: none"> "pearson": Pearson's product-moment correlation (default) "spearman": Spearman's rank correlation |

| | |
|--------|--|
| | <ul style="list-style-type: none"> • "kendall": Kendall's tau |
| type | Choose which plot(s) to create: <ul style="list-style-type: none"> • "c": consonance function only (p-values across potential parameter values) • "cd": consonance density function only (distribution of plausible parameter values) • c("c", "cd"): both plots together (default) |
| levels | Numeric vector of confidence levels to display (default: c(.68, .9, .95, .999)). These correspond to the confidence intervals shown on the plot. |

Details

Consonance plots provide a graphical representation of the full range of confidence intervals for correlation coefficients at different confidence levels. These plots help visualize the uncertainty around correlation estimates and go beyond the traditional approach of reporting only a single confidence interval (typically 95%).

The function creates two types of visualizations:

1. **Consonance function** ("c"): Shows how p-values change across different possible values of the correlation coefficient. The x-axis represents possible correlation values, and the y-axis represents the corresponding p-values from two-sided hypothesis tests.
2. **Consonance density** ("cd"): Shows the distribution of plausible values for the correlation coefficient. This can be interpreted as showing where the "weight of evidence" is concentrated.

These plots are particularly useful for:

- Visualizing uncertainty around correlation estimates
- Understanding the precision of correlation estimates
- Comparing the relative plausibility of different correlation values
- Going beyond the binary "significant vs. non-significant" interpretation

These types of plots are discussed by Schweder & Hjort (2016) and Rafi & Greenland (2020).

Value

A ggplot2 object or plot grid from cowplot.

References

Schweder, T., & Hjort, N. L. (2016). *Confidence, likelihood, probability: Statistical inference with confidence distributions*. Cambridge University Press. ISBN: 9781316445051

Rafi, Z., & Greenland, S. (2020). Semantic and cognitive tools to aid statistical science: Replace confidence and significance by compatibility and surprise. *BMC Medical Research Methodology*, 20, 244. doi:10.1186/s12874-020-01105-9

See Also

Other Correlations: [boot_cor_test\(\)](#), [corsum_test\(\)](#), [power_z_cor\(\)](#), [z_cor_test\(\)](#)

Other plotting functions: [plot_pes\(\)](#), [plot_smd\(\)](#)

Examples

```

# Example 1: Basic consonance plot for Pearson correlation
# For a correlation of r = 0.45 with n = 30
plot_cor(r = 0.45, n = 30)

# Example 2: Consonance function only for Spearman correlation
plot_cor(r = 0.6, n = 25, method = "spearman", type = "c")

# Example 3: Consonance density only for Kendall's tau
plot_cor(r = 0.3, n = 40, method = "kendall", type = "cd")

# Example 4: Custom confidence levels
plot_cor(r = 0.5, n = 50, levels = c(0.5, 0.8, 0.95))

# Example 5: Saving and further customizing the plot
library(ggplot2)
p <- plot_cor(r = 0.45, n = 30)
p + theme_minimal() +
  labs(title = "Consonance Plot for Correlation r = 0.45, n = 30")

```

plot_pes

*Plot Partial Eta-Squared***Description****[Stable]**

Creates consonance plots (confidence curves and/or consonance density functions) for partial eta-squared values from ANOVA models, allowing visualization of uncertainty around effect size estimates.

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 (effect degrees of freedom). |
| df2 | Degrees of freedom for the denominator (error degrees of freedom). |
| type | Choose which plot(s) to create: |

- "c": consonance function only (p-values across potential parameter values)
 - "cd": consonance density function only (distribution of plausible parameter values)
 - c("c", "cd"): both plots together (default)
- levels Numeric vector of confidence levels to display (default: c(.68, .9, .95, .999)). These correspond to the confidence intervals shown on the plot.

Details

Consonance plots provide a graphical representation of the full range of confidence intervals for partial eta-squared values at different confidence levels. These plots help visualize the uncertainty around effect size estimates and go beyond the traditional approach of reporting only a single confidence interval (typically 95%).

Partial eta-squared (η^2) is a measure of effect size commonly used in ANOVA, representing the proportion of variance in the dependent variable attributed to a specific factor, while controlling for other factors in the model. Values range from 0 to 1, with larger values indicating stronger effects.

The function creates two types of visualizations:

1. **Consonance function** ("c"): Shows how p-values change across different possible values of partial eta-squared. The x-axis represents possible parameter values, and the y-axis represents the corresponding p-values from two-sided hypothesis tests.
2. **Consonance density** ("cd"): Shows the distribution of plausible values for the partial eta-squared. This can be interpreted as showing where the "weight of evidence" is concentrated.

These plots are particularly useful for:

- Visualizing uncertainty around effect size estimates
- Understanding the precision of effect size estimates
- Comparing the relative plausibility of different effect sizes
- Going beyond the binary "significant vs. non-significant" interpretation

The required inputs (F-statistic, df1, df2) can typically be extracted from standard ANOVA output in R, such as from `aov()`, `Anova()`, or `afex_aov()` functions.

These types of plots are discussed by Schweder & Hjort (2016) and Rafi & Greenland (2020).

Value

A `ggplot2` object or plot grid from `cowplot`.

References

- Schweder, T., & Hjort, N. L. (2016). *Confidence, likelihood, probability: Statistical inference with confidence distributions*. Cambridge University Press. ISBN: 9781316445051
- Rafi, Z., & Greenland, S. (2020). Semantic and cognitive tools to aid statistical science: Replace confidence and significance by compatibility and surprise. *BMC Medical Research Methodology*, 20, 244. doi:10.1186/s12874-020-01105-9

See Also

Other plotting functions: [plot_cor\(\)](#), [plot_smd\(\)](#)

Examples

```
## Not run:
# Example 1: Basic consonance plot for partial eta-squared
# For an F-statistic of 4.5 with df1 = 2, df2 = 60
plot_pes(Fstat = 4.5, df1 = 2, df2 = 60)

# Example 2: Consonance function only (p-value curve)
plot_pes(Fstat = 3.2, df1 = 1, df2 = 45, type = "c")

# Example 3: Consonance density only
plot_pes(Fstat = 6.8, df1 = 3, df2 = 80, type = "cd")

# Example 4: Custom confidence levels
plot_pes(Fstat = 5.1, df1 = 2, df2 = 50, levels = c(0.5, 0.8, 0.95))

# Example 5: Using with actual ANOVA results
# aov_result <- aov(DV ~ IV, data = your_data)
# aov_summary <- summary(aov_result)[[1]]
# F_value <- aov_summary$"F value"[1]
# df1 <- aov_summary$Df[1]
# df2 <- aov_summary$Df[2]
# plot_pes(Fstat = F_value, df1 = df1, df2 = df2)

# Example 6: Saving and further customizing the plot
library(ggplot2)
p <- plot_pes(Fstat = 4.5, df1 = 2, df2 = 60)
p + theme_minimal() +
  labs(title = "Consonance Plot for Partial Eta-Squared",
        subtitle = "F(2, 60) = 4.5")

## End(Not run)
```

plot_smd

Plot Distribution of Standardized Mean Difference (SMD)

Description**[Stable]**

Creates consonance plots (confidence curves and/or consonance density functions) for standardized mean differences (SMDs), allowing visualization of uncertainty around effect size estimates.

Usage

```
plot_smd(
  d,
```

```

df,
lambda = NULL,
sigma = NULL,
smd_ci = c("t", "z", "goulet", "nct"),
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 (Cohen's d, Hedges' g, etc.). |
| df | Degrees of freedom for the standardized mean difference. |
| lambda | The non-centrality parameter for the standardized mean difference. Required when <code>smd_ci = "goulet"</code> . |
| sigma | The standard error for the standardized mean difference. Required when <code>smd_ci</code> is "t" or "z". |
| smd_ci | Method for calculating SMD confidence intervals: <ul style="list-style-type: none"> • "t": central t-distribution method • "z": normal distribution method • "goulet": Goulet-Pelletier method • "nct": noncentral t-distribution method (not currently supported) |
| smd_label | Label for the x-axis indicating the SMD measure (default: "SMD"). Common labels include "Cohen's d", "Hedges' g", or "Glass's delta". |
| type | Choose which plot(s) to create: <ul style="list-style-type: none"> • "c": consonance function only (p-values across potential parameter values) • "cd": consonance density function only (distribution of plausible parameter values) • c("c", "cd"): both plots together (default) |
| levels | Numeric vector of confidence levels to display (default: c(.5, .9, .95, .999)). These correspond to the confidence intervals shown on the plot. |

Details

Consonance plots provide a graphical representation of the full range of confidence intervals for standardized mean differences at different confidence levels. These plots help visualize the uncertainty around effect size estimates and go beyond the traditional approach of reporting only a single confidence interval (typically 95%).

The function creates two types of visualizations:

1. **Consonance function** ("c"): Shows how p-values change across different possible values of the SMD. The x-axis represents possible parameter values, and the y-axis represents the corresponding p-values from two-sided hypothesis tests.
2. **Consonance density** ("cd"): Shows the distribution of plausible values for the SMD. This can be interpreted as showing where the "weight of evidence" is concentrated.

This function requires specific input parameters depending on the chosen confidence interval method:

- For **"goulet"** method: `d`, `df`, and `lambda` must be provided
- For **"t"** and **"z"** methods: `d`, `df`, and `sigma` must be provided
- The **"nct"** method is not currently supported

The required parameters can typically be extracted from the results of functions like `t_TOST()`, `smd_calc()`, or from the `smd` component of these function results.

These plots are particularly useful for:

- Visualizing uncertainty around SMD estimates
- Understanding the precision of effect size estimates
- Comparing the relative plausibility of different effect sizes
- Going beyond the binary "significant vs. non-significant" interpretation

These types of plots are discussed by Schweder & Hjort (2016) and Rafi & Greenland (2020).

Value

A `ggplot2` object or plot grid from `cowplot`.

References

Schweder, T., & Hjort, N. L. (2016). Confidence, likelihood, probability: Statistical inference with confidence distributions. Cambridge University Press. ISBN: 9781316445051

Rafi, Z., & Greenland, S. (2020). Semantic and cognitive tools to aid statistical science: Replace confidence and significance by compatibility and surprise. *BMC Medical Research Methodology*, 20, 244. doi:10.1186/s12874-020-01105-9

See Also

Other plotting functions: [plot_cor\(\)](#), [plot_pes\(\)](#)

Examples

```
# Example 1: Basic consonance plot for Cohen's d using z-method
plot_smd(d = 0.5, df = 40, sigma = 0.164, smd_ci = "z", smd_label = "Cohen's d")
```

```
# Example 2: Consonance function only for Hedges' g using t-method
plot_smd(d = 0.45, df = 28, sigma = 0.192, smd_ci = "t",
        smd_label = "Hedges' g", type = "c")
```

```
# Example 3: Consonance density only using Goulet method
# Note: lambda parameter required for Goulet method
plot_smd(d = 0.6, df = 35, lambda = 3.6, smd_ci = "goulet",
        type = "cd")
```

```
# Example 4: Custom confidence levels
plot_smd(d = 0.8, df = 50, sigma = 0.145, smd_ci = "z",
        levels = c(0.5, 0.8, 0.95))
```

```

# Example 5: Using with TOSTER results (requires extracting needed parameters)
# tost_result <- t_TOST(x = group1, y = group2, eqb = 0.5)
# plot_smd(d = tost_result$smd$d,
#         df = tost_result$smd$d_df,
#         sigma = tost_result$smd$d_sigma,
#         smd_ci = "z",
#         smd_label = tost_result$smd$smd_label)

# Example 6: Saving and further customizing the plot
## Not run:
library(ggplot2)
p <- plot_smd(d = 0.5, df = 40, sigma = 0.164, smd_ci = "z")
p + theme_minimal() +
  labs(title = "Consonance Plot for Cohen's d = 0.5",
       subtitle = "df = 40")

## End(Not run)

```

powerTOSTone

Power One Sample t-test

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

Chow, S.-C., Wang, H., & Shao, J. (2007). Sample Size Calculations in Clinical Research, Second Edition - CRC Press Book. Formula 3.1.9

Examples

```
## 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

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

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

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) |
| 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 |
| sdif | standard deviation of the difference 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

Chow, S.-C., Wang, H., & Shao, J. (2007). *Sample Size Calculations in Clinical Research*, Second Edition - CRC Press Book. Formula 3.1.9

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)

## 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
```

| | |
|--------------|--------------------------------|
| powerTOSTtwo | <i>Power Two Sample t-test</i> |
|--------------|--------------------------------|

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

```
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

Chow, S.-C., Wang, H., & Shao, J. (2007). Sample Size Calculations in Clinical Research, Second Edition - CRC Press Book. Formula 3.2.4 with $k = 1$

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)

## 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)
```

Description

[Stable]

Performs power analysis for equivalence testing with F-tests (ANOVA models). This function calculates statistical power, sample size, equivalence bound, or alpha level when the other parameters are specified.

Usage

```
power_eq_f(alpha = 0.05, df1 = NULL, df2 = NULL, eqbound = NULL, power = NULL)
```

Arguments

| | |
|---------|--|
| alpha | Significance level (Type I error rate). Default is 0.05. |
| df1 | Numerator degrees of freedom (e.g., groups - 1 for one-way ANOVA). |
| df2 | Denominator degrees of freedom (e.g., N - groups for one-way ANOVA), where N is the total sample size. |
| eqbound | Equivalence bound for partial eta-squared. This represents the threshold for what effect size would be considered practically insignificant. |
| power | Desired statistical power (1 - Type II error rate). Default is NULL. |

Details

This function provides power analysis for the omnibus non-inferiority testing procedure described by Campbell & Lakens (2021). Exactly one of the parameters `alpha`, `df1`, `df2`, `eqbound`, or `power` must be `NULL`, and the function will solve for that parameter.

For one-way ANOVA:

- `df1` = number of groups - 1
- `df2` = total N - number of groups

Common equivalence bounds (we do not recommend their use for choosing equivalence bounds) for partial eta-squared based on Cohen's benchmarks:

- Small effect: 0.01
- Medium effect: 0.06
- Large effect: 0.14

Note that this function is primarily validated for one-way ANOVA designs; use with caution for more complex designs.

Value

An object of class "power.htest" containing the following components:

- **df1**: Numerator degrees of freedom
- **df2**: Denominator degrees of freedom
- **eqbound**: Equivalence bound for partial eta-squared
- **sig.level**: Significance level (alpha)
- **power**: Statistical power
- **method**: Description of the test

References

Campbell, H., & Lakens, D. (2021). Can we disregard the whole model? Omnibus non-inferiority testing for R² in multi-variable linear regression and in ANOVA. *British Journal of Mathematical and Statistical Psychology*, 74(1), 64-89. doi: 10.1111/bmsp.12201

See Also

Other power: [power_t_TOST\(\)](#), [power_z_cor\(\)](#)

Examples

```
# Example 1: Calculate power given degrees of freedom and equivalence bound
# For a one-way ANOVA with 3 groups, 80 subjects per group, and equivalence bound of 0.01
power_eq_f(df1 = 2, df2 = 237, eqbound = 0.01)

# Example 2: Calculate required denominator df (related to sample size)
# for 80% power with equivalence bound of 0.05
power_eq_f(df1 = 2, power = 0.8, eqbound = 0.05)

# Example 3: Calculate detectable equivalence bound with 80% power
power_eq_f(df1 = 2, df2 = 100, power = 0.8)

# Example 4: Calculate required alpha level for 90% power
power_eq_f(df1 = 2, df2 = 100, eqbound = 0.05, power = 0.9, alpha = NULL)
```

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

- Silva, G. T. da, Logan, B. R., & Klein, J. P. (2008). Methods for Equivalence and Noninferiority Testing. *Biology of Blood and Marrow Transplantation: Journal of the American Society for Blood and Marrow Transplantation*, 15(1 Suppl), 120-127. <https://doi.org/10.1016/j.bbmt.2008.10.004>
- Julious, S. A. & Campell, M. J. (2012). Tutorial in biostatistics: sample sizes for parallel group clinical trials with binary data. *Statistics in Medicine*, 31:2904-2936.
- Chow, S.-C., Wang, H., & Shao, J. (2007). *Sample Size Calculations in Clinical Research*, Second Edition (2 edition). Boca Raton: Chapman and Hall/CRC.

Examples

```
## 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_twoprop(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)

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

power_twoprop(alpha = 0.05, n = 542, p1 = 0.5, p2 = 0.5,
  null = 0.1, alternative = "e")

#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)

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

# 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)
power_twoprop(alpha = 0.025, power = 0.9, p1 = 0.8, p2 = 0.8,
  null = 0.1, alternative = "e")
# From Machin, D. (Ed.). (2008). Sample size tables for clinical studies (3rd ed).

# 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
power_twoprop(alpha = 0.01, power = 0.8, p1 = 0.5, p2 = 0.5,
  null = 0.2, alternative = "e")
```

power_t_TOST

Power calculations for TOST with t-tests

Description

[Stable]

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

Usage

```
power_t_TOST(  
  n = NULL,  
  delta = 0,  
  sd = 1,  
  eqb,  
  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. |
| 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

Phillips KF. Power of the Two One-Sided Tests Procedure in Bioequivalence. *J Pharmacokin Biopharm.* 1990;18(2):137–44. doi: 10.1007/BF01063556

Diletti D, Hauschke D, Steinijans VW. Sample Size Determination for Bioequivalence Assessment by Means of Confidence Intervals. *Int J Clin Pharmacol Ther Toxicol.* 1991;29(1):1–8.

See Also

Other power: [power_eq_f\(\)](#), [power_z_cor\(\)](#)

power_z_cor

Power Calculations for Correlations

Description**[Maturing]**

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

Usage

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

```
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.

See Also

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", "logodds", "cstat")
)
```

Arguments

| | |
|-------------------------|--|
| <code>x</code> | a (non-empty) numeric vector of data values. |
| <code>y</code> | an optional (non-empty) numeric vector of data values. |
| <code>mu</code> | 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 . |
| <code>conf.level</code> | confidence level of the interval. |
| <code>paired</code> | a logical indicating whether you want to calculate a paired test. |
| <code>ses</code> | 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

- Cureton, E. E. (1956). Rank-biserial correlation. *Psychometrika*, 21(3), 287-290.
- Glass, G. V. (1965). A ranking variable analogue of biserial correlation: Implications for short-cut item analysis. *Journal of Educational Measurement*, 2(1), 91-95.
- Kendall, M.G. (1948) Rank correlation methods. London: Griffin.
- Kerby, D. S. (2014). The simple difference formula: An approach to teaching nonparametric correlation. *Comprehensive Psychology*, 3, 11-17.
- King, B. M., & Minium, E. W. (2008). *Statistical reasoning in the behavioral sciences*. John Wiley & Sons Inc.
- Cliff, N. (1993). Dominance statistics: Ordinal analyses to answer ordinal questions. *Psychological bulletin*, 114(3), 494.
- Tomczak, M., & Tomczak, E. (2014). The need to report effect size estimates revisited. An overview of some recommended measures of effect size.

 ses_calc

Standardized Effect Size (SES) Calculation

Description

[Stable]

Calculates non-SMD standardized effect sizes for group comparisons. This function focuses on rank-based and probability-based effect size measures, which are especially useful for non-parametric analyses and when data do not meet normality assumptions.

Usage

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

```
## Default S3 method:
ses_calc(
  x,
  y = NULL,
  paired = FALSE,
  ses = c("rb", "odds", "logodds", "cstat"),
  alpha = 0.05,
  mu = 0,
  ...
)

## S3 method for class 'formula'
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 | a character string specifying the effect size measure to calculate: - "rb": rank-biserial correlation (default) - "odds": Wilcoxon-Mann-Whitney odds - "logodds": Wilcoxon-Mann-Whitney log-odds - "cstat": concordance statistic (C-statistic, equivalent to the area under the ROC curve) |
| alpha | alpha level for confidence interval calculation (default = 0.05). |
| y | an optional (non-empty) numeric vector of data values. |
| mu | number indicating the value around which asymmetry (for one-sample or paired samples) or shift (for independent samples) is to be estimated (default = 0). |
| 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 calculates standardized effect sizes that are not standardized mean differences (SMDs). These effect sizes are particularly useful for non-parametric analyses or when data violate assumptions of normality.

The available effect size measures are:

- **Rank-biserial correlation ("rb")**: A correlation coefficient based on ranks, ranging from -1 to 1. It can be interpreted as the difference between the proportion of favorable pairs and the proportion of unfavorable pairs. For independent samples, this is equivalent to Cliff's delta.
- **Wilcoxon-Mann-Whitney odds ("odds")**: The ratio of the probability that a randomly selected observation from group 1 exceeds a randomly selected observation from group 2, to the probability of the reverse. Values range from 0 to infinity, with 1 indicating no effect.
- **Wilcoxon-Mann-Whitney log-odds ("logodds")**: The natural logarithm of the WMW odds. This transforms the odds scale to range from negative infinity to positive infinity, with 0 indicating no effect.
- **Concordance statistic ("cstat")**: The probability that a randomly selected observation from group 1 exceeds a randomly selected observation from group 2. Also known as the common language effect size or the area under the ROC curve. Values range from 0 to 1, with 0.5 indicating no effect.

The function supports three study designs:

- One-sample design: Compares a single sample to a specified value
- Two-sample independent design: Compares two independent groups
- Paired samples design: Compares paired observations

For detailed information on calculation methods, see `vignette("robustTOST")`.

Value

A data frame containing the following information:

- `estimate`: The effect size estimate
- `lower.ci`: Lower bound of the confidence interval
- `upper.ci`: Upper bound of the confidence interval
- `conf.level`: Confidence level (1-alpha)

Purpose

Use this function when:

- You want to report non-parametric effect size measures
- You need to quantify the magnitude of differences using ranks or probabilities
- Your outcome variable is ordinal
- You want to complement results from Wilcoxon-Mann-Whitney type test

See Also

Other effect sizes: `boot_ses_calc()`, `boot_smd_calc()`, `smd_calc()`

Examples

```
# Example 1: Independent groups comparison (rank-biserial correlation)
set.seed(123)
group1 <- c(1.2, 2.3, 3.1, 4.6, 5.2, 6.7)
group2 <- c(3.5, 4.8, 5.6, 6.9, 7.2, 8.5)
ses_calc(x = group1, y = group2, ses = "rb")

# Example 2: Using formula notation to calculate WMW odds
data(mtcars)
ses_calc(formula = mpg ~ am, data = mtcars, ses = "odds")

# Example 3: Paired samples with concordance statistic
data(sleep)
with(sleep, ses_calc(x = extra[group == 1],
                    y = extra[group == 2],
                    paired = TRUE,
                    ses = "cstat"))
```

| | |
|---------------|--|
| simple_hstest | <i>One, Two, and Paired Samples Hypothesis Tests with Extended Options</i> |
|---------------|--|

Description

[Maturing]

Performs statistical hypothesis tests with extended functionality beyond standard implementations. Supports t-tests, Wilcoxon-Mann-Whitney tests, and Brunner-Munzel tests with additional alternatives such as equivalence and minimal effect testing.

Usage

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

## Default S3 method:
simple_hstest(
  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_hstest(formula, data, subset, na.action, ...)
```

Arguments

| | |
|-------------|---|
| x | a (non-empty) numeric vector of data values. |
| ... | further arguments to be passed to or from the underlying test functions. |
| paired | a logical indicating whether you want a paired t-test. |
| alternative | the alternative hypothesis: - "two.sided": different from mu (default) - "less": less than mu - "greater": greater than mu - "equivalence": between specified bounds - "minimal.effect": outside specified bounds |

| | |
|------------------------|--|
| <code>mu</code> | a number or vector specifying the null hypothesis value(s): - For standard alternatives (<code>two.sided</code> , <code>less</code> , <code>greater</code>): a single value (default: 0 for <code>t.test</code> / <code>wilcox.test</code> , 0.5 for <code>brunner_munzel</code>) - For equivalence/ <code>minimal.effect</code> : either a single value (symmetric bounds will be created) or a vector of two values representing the lower and upper bounds |
| <code>alpha</code> | alpha level (default = 0.05) |
| <code>y</code> | an optional (non-empty) numeric vector of data values. |
| <code>test</code> | a character string specifying the type of hypothesis test to use: - <code>"t.test"</code> : Student's t-test (parametric, default) - <code>"wilcox.test"</code> : Wilcoxon-Mann-Whitney test (non-parametric) - <code>"brunner_munzel"</code> : Brunner-Munzel test (non-parametric) You can specify just the initial letter (e.g., <code>"t"</code> for <code>"t.test"</code>). |
| <code>formula</code> | a formula of the form <code>lhs ~ rhs</code> where <code>lhs</code> is a numeric variable giving the data values and <code>rhs</code> either 1 for a one-sample or paired test or a factor with two levels giving the corresponding groups. If <code>lhs</code> is of class <code>"Pair"</code> and <code>rhs</code> is 1, a paired test is done. |
| <code>data</code> | an optional matrix or data frame (or similar: see <code>model.frame</code>) containing the variables in the formula <code>formula</code> . By default the variables are taken from <code>environment(formula)</code> . |
| <code>subset</code> | an optional vector specifying a subset of observations to be used. |
| <code>na.action</code> | a function which indicates what should happen when the data contain NAs. Defaults to <code>getOption("na.action")</code> . |

Details

This function provides a unified interface to several common hypothesis tests with expanded alternative hypotheses, particularly for equivalence testing and minimal effect testing.

When `alternative = "equivalence"`, the test evaluates whether the effect is contained within the bounds specified by `mu`. This corresponds to the alternative hypothesis that the true effect is between the specified bounds. The function performs two one-sided tests and returns the most conservative result (highest p-value).

When `alternative = "minimal.effect"`, the test evaluates whether the effect is outside the bounds specified by `mu`. This corresponds to the alternative hypothesis that the true effect is either less than the lower bound or greater than the upper bound. The function performs two one-sided tests and returns the most significant result (lowest p-value).

For standard alternatives (`"two.sided"`, `"less"`, `"greater"`), the function behaves similarly to the underlying test functions with some additional standardization in the output format.

The interpretation of `mu` depends on the test used:

- For `t.test` and `wilcox.test`: `mu` represents the difference in means/medians (default: 0)
- For `brunner_munzel`: `mu` represents the probability that a randomly selected value from the first sample exceeds a randomly selected value from the second sample (default: 0.5)

If `mu` is a single value for equivalence or minimal effect alternatives, symmetric bounds will be created automatically:

- For `t.test` and `wilcox.test`: bounds become `c(mu, -mu)`
- For `brunner_munzel`: bounds become `c(mu, abs(mu-1))`

Value

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

- `statistic`: the value of the test statistic.
- `parameter`: the parameter(s) for the test statistic (e.g., degrees of freedom for t-tests).
- `p.value`: the p-value for the test.
- `conf.int`: a confidence interval appropriate to the specified alternative hypothesis.
- `estimate`: the estimated effect (e.g., mean difference for t-tests, probability estimate for Brunner-Munzel).
- `null.value`: the specified hypothesized value(s). For equivalence and minimal effect tests, this will be two values.
- `stderr`: the standard error of the estimate (for t-tests).
- `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.

Purpose

Use this function when:

- You need a unified interface for different types of hypothesis tests
- You want to perform equivalence testing or minimal effect testing with non-parametric methods
- You need more flexibility in hypothesis testing than provided by standard functions
- You want to easily switch between parametric and non-parametric methods

See Also

Other TOST: [boot_log_TOST\(\)](#), [boot_t_TOST\(\)](#), [t_TOST\(\)](#), [tsum_TOST\(\)](#), [wilcox_TOST\(\)](#)

Other hstest: [as_hstest\(\)](#), [hstest-helpers](#)

Other hstest: [as_hstest\(\)](#), [hstest-helpers](#)

Examples

```
# Example 1: Basic t-test with equivalence alternative
# Testing if the difference in mpg between automatic and manual transmission cars
# is equivalent within ±3 units
data(mtcars)
simple_hstest(mpg ~ am, data = mtcars, alternative = "equivalence", mu = 3)

# Example 2: Using a non-parametric test with minimal effect alternative
# Testing if the effect of transmission type on mpg is meaningfully large
# (either less than -2 or greater than 2)
simple_hstest(mpg ~ am, data = mtcars,
             test = "wilcox",
             alternative = "minimal.effect",
```

```

mu = c(-2, 2))

# Example 3: Paired samples test
# Using the sleep dataset to test if drug has an effect on sleep
data(sleep)
with(sleep, simple_hstest(x = extra[group == 1],
                          y = extra[group == 2],
                          paired = TRUE,
                          alternative = "greater"))

# Example 4: Brunner-Munzel test
# Testing if values in one group tend to exceed values in another
set.seed(123)
group1 <- rnorm(20, mean = 5, sd = 1)
group2 <- rnorm(20, mean = 6, sd = 2)
simple_hstest(x = group1, y = group2,
             test = "brunner_munzel",
             alternative = "less")

```

smd_calc

Standardized Mean Difference (SMD) Calculation

Description

[Stable]

Calculates standardized mean difference (SMD) effect sizes and their confidence intervals from raw data. This function focuses solely on effect size estimation without performing hypothesis tests.

Usage

```

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")
)

```

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"),
...
)

## 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. |
| 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 | An 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 | 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"). |

Details

This function calculates standardized mean differences (SMD) for various study designs:

- One-sample design: Standardizes the difference between the sample mean and zero (or other specified value)
- Two-sample independent design: Standardizes the difference between two group means
- Paired samples design: Standardizes the mean difference between paired observations

The function supports multiple SMD variants:

- Cohen's d: Classic standardized mean difference (`bias_correction = FALSE`)
- Hedges' g: Bias-corrected version of Cohen's d (`bias_correction = TRUE`)
- Glass's delta: Uses only one group's standard deviation as the denominator (`glass = "glass1"` or `"glass2"`)
- Repeated measures d: Accounts for correlation in paired designs (`rm_correction = TRUE`)

Different confidence interval calculation methods are available:

- "nct": Uses the noncentral t-distribution (most accurate in most cases)
- "goulet": Uses the Goulet-Pelletier method
- "t": Uses the central t-distribution
- "z": Uses the normal distribution

Note that unlike the `t_TOST` and related functions, `smd_calc` only calculates effect sizes and their confidence intervals without performing hypothesis tests.

For detailed information on calculation methods, see `vignette("SMD_calcs")`.

Value

A data frame containing the following information:

- `estimate`: The standardized mean difference estimate (Cohen's d, Hedges' g, or Glass's delta)
- `SE`: Standard error of the estimate
- `lower.ci`: Lower bound of the confidence interval
- `upper.ci`: Upper bound of the confidence interval
- `conf.level`: Confidence level (1-alpha)

Purpose

Use this function when:

- You need to calculate standardized effect sizes (Cohen's d, Hedges' g, Glass's delta)
- You want confidence intervals for your effect size estimates
- You need effect sizes for meta-analysis or reporting
- You want to compare effect sizes across different studies or measures
- You don't need the hypothesis testing components of the TOST functions

See Also

Other effect sizes: [boot_ses_calc\(\)](#), [boot_smd_calc\(\)](#), [ses_calc\(\)](#)

Examples

```
# Example 1: Independent groups comparison (Cohen's d)
set.seed(123)
group1 <- rnorm(30, mean = 100, sd = 15)
group2 <- rnorm(30, mean = 110, sd = 18)
smd_calc(x = group1, y = group2, bias_correction = FALSE)

# Example 2: Independent groups with formula notation (Hedges' g)
df <- data.frame(
  value = c(group1, group2),
  group = factor(rep(c("A", "B"), each = 30))
)
smd_calc(formula = value ~ group, data = df)

# Example 3: Paired samples with repeated measures correction
before <- c(5.1, 4.8, 6.2, 5.7, 6.0, 5.5, 4.9, 5.8)
after <- c(5.6, 5.2, 6.7, 6.1, 6.5, 5.8, 5.3, 6.2)
smd_calc(x = before, y = after, paired = TRUE, rm_correction = TRUE)

# Example 4: Glass's delta (using only first group's SD)
smd_calc(x = group1, y = group2, glass = "glass1")
```

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

| | |
|----------------|---|
| 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 confidence interval TOST

References

Rogers, J. L., Howard, K. I., & Vessey, J. T. (1993). Using significance tests to evaluate equivalence 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 for TOSTnp objects

Description

Methods defined for objects returned from the wilcox_TOST function.

Usage

```
## 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.
[TOSTnp-methods](#).

Value

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

Examples

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

# PRINT
print(res1)

# DESCRIPTION
describe(res1)
```

TOSTone

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

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

| | |
|----------------|---|
| 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 |
| 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 |

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

```
## 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)
```

`TOSTpaired`*TOST function for a dependent t-test (Cohen's dz)*

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

```
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

| | |
|------------------|-------------------------------|
| <code>n</code> | sample size (pairs) |
| <code>m1</code> | mean of group 1 |
| <code>m2</code> | mean of group 2 |
| <code>sd1</code> | 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

Mara, C. A., & Cribbie, R. A. (2012). Paired-Samples Tests of Equivalence. *Communications in Statistics - Simulation and Computation*, 41(10), 1928-1943. <https://doi.org/10.1080/03610918.2011.626545>, formula page 1932. Note there is a typo in the formula: n-1 should be n (personal communication, 31-8-2016)

Examples

```
## 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 correlations*

Description

[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

```
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

Goertzen, J. R., & Cribbie, R. A. (2010). Detecting a lack of association: An equivalence testing approach. *British Journal of Mathematical and Statistical Psychology*, 63(3), 527-537. <https://doi.org/10.1348/000711009X475> formula page 531.

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

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

## S3 method for class 'TOSTt'
plot(
  x,
```

```

    type = c("simple", "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 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 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

Berger, R. L., & Hsu, J. C. (1996). Bioequivalence Trials, Intersection-Union Tests and Equivalence Confidence Sets. *Statistical Science*, 11(4), 283-302.

Gruman, J. A., Cribbie, R. A., & Arpin-Cribbie, C. A. (2007). The effects of heteroscedasticity on tests of equivalence. *Journal of Modern Applied Statistical Methods*, 6(1), 133-140, formula for Welch's t-test on page 135

Examples

```
## 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

[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

```
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

Tunes da Silva, G., Logan, B. R., & Klein, J. P. (2008). Methods for Equivalence and Noninferiority Testing. *Biology of Blood Marrow Transplant*, 15(1 Suppl), 120-127.

Yin, G. (2012). *Clinical Trial Design: Bayesian and Frequentist Adaptive Methods*. Hoboken, New Jersey: John Wiley & Sons, Inc.

Examples

```
## 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)
```

tsum_TOST

TOST with t-tests from Summary Statistics

Description

[Stable]

Performs equivalence testing using the Two One-Sided Tests (TOST) procedure with t-tests based on summary statistics rather than raw data. This function allows TOST analysis when only descriptive statistics are available from published studies or reports.

Usage

```
tsum_TOST(
  m1,
  sd1,
  n1,
  m2 = NULL,
```

```

sd2 = NULL,
n2 = NULL,
r12 = NULL,
hypothesis = c("EQU", "MET"),
paired = FALSE,
var.equal = FALSE,
eqb,
low_eqbound,
high_eqbound,
mu = 0,
eqbound_type = c("raw", "SMD"),
alpha = 0.05,
bias_correction = TRUE,
rm_correction = FALSE,
glass = NULL,
smd_ci = c("nct", "goulet", "t", "z")
)

```

Arguments

| | |
|--------------|--|
| m1 | mean of group 1. |
| sd1 | standard deviation of group 1. |
| n1 | sample size in group 1. |
| m2 | mean of group 2 (not required for one-sample tests). |
| sd2 | standard deviation of group 2 (not required for one-sample tests). |
| n2 | sample size in group 2 (not required for one-sample tests). |
| r12 | correlation between measurements for paired designs. Required when paired = TRUE. |
| hypothesis | 'EQU' for equivalence (default), or 'MET' for minimal effects test. |
| 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 (symmetric bound, 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 instead). |
| high_eqbound | upper equivalence bounds (deprecated, use eqb instead). |
| 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). |
| eqbound_type | Type of equivalence bound. Can be '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 | An 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

This function performs TOST equivalence testing using summary statistics instead of raw data. It is particularly useful when analyzing published results or conducting meta-analyses where only summary statistics are available.

The function supports three types of tests:

- One-sample test: Provide $m1$, $sd1$, and $n1$ only
- Two-sample independent test: Provide all parameters except $r12$, with `paired = FALSE`
- Paired samples test: Provide all parameters including $r12$, with `paired = TRUE`

For two-sample tests, the test is of $m1 - m2$ (mean of group 1 minus mean of group 2). For paired samples, the test is of the difference scores, wherein $z = m1 - m2$, 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).

The function calculates both raw mean differences and standardized effect sizes (Cohen's d or Hedges' g), along with their confidence intervals.

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

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), d_{low} (lower CI bound), d_{high} (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.

Purpose

Use this function when:

- You only have access to summary statistics (means, standard deviations, sample sizes)
- You want to perform meta-analyses using published results
- You're conducting power analyses based on previous studies
- You need to reanalyze published results within an equivalence testing framework

See Also

Other TOST: [boot_log_TOST\(\)](#), [boot_t_TOST\(\)](#), [simple_hstest\(\)](#), [t_TOST\(\)](#), [wilcox_TOST\(\)](#)

Examples

```
# Example 1: One-sample test
# Testing if a sample with mean 0.55 and SD 4 (n=18) is equivalent to zero within ±2 units
tsum_TOST(m1 = 0.55, n1 = 18, sd1 = 4, eqb = 2)

# Example 2: Two-sample independent test
# Testing if two groups with different means are equivalent within ±3 units
tsum_TOST(m1 = 15.2, sd1 = 5.3, n1 = 30,
          m2 = 13.8, sd2 = 4.9, n2 = 28,
          eqb = 3)

# Example 3: Paired samples test
# Testing if pre-post difference is equivalent to zero within ±2.5 units
# with correlation between measurements of 0.7
tsum_TOST(m1 = 24.5, sd1 = 6.2, n1 = 25,
          m2 = 26.1, sd2 = 5.8, n2 = 25,
          r12 = 0.7, paired = TRUE,
          eqb = 2.5)

# Example 4: Two-sample test using standardized effect size bounds
# Testing if the standardized mean difference is within ±0.5 SD
tsum_TOST(m1 = 100, sd1 = 15, n1 = 40,
          m2 = 104, sd2 = 16, n2 = 42,
          eqb = 0.5, eqbound_type = "SMD")
```

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: <ul style="list-style-type: none"> • "two.sided": correlation is not equal to null (default) • "greater": correlation is greater than null • "less": correlation is less than null • "equivalence": correlation is within the equivalence bounds (TOST) • "minimal.effect": correlation is outside the equivalence bounds (TOST) <p>You can specify just the initial letter.</p> |
| 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:

$$\phi = p_1/p_2$$

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 ϕ_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:

$$\phi_{lower} = \phi \cdot e^{-z_{\alpha} \cdot se(\ln(\phi))}$$

$$\phi_{upper} = \phi \cdot e^{z_{\alpha} \cdot se(\ln(\phi))}$$

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:

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

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

$$z = \frac{\ln(OR) - \ln(OR_0)}{se(\ln(OR))}$$

The confidence interval can then be calculated as the following:

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

Value

An S3 object of the class `htest`.

References

- Gart, J. J., & Nam, J. M. (1988). Approximate interval estimation of the ratio of binomial parameters: a review and corrections for skewness. *Biometrics*, 323-338.
- Tunes da Silva, G., Logan, B. R., & Klein, J. P. (2008). Methods for Equivalence and Noninferiority Testing. *Biology of Blood Marrow Transplant*, 15(1 Suppl), 120-127.
- Yin, G. (2012). *Clinical Trial Design: Bayesian and Frequentist Adaptive Methods*. Hoboken, New Jersey: John Wiley & Sons, Inc.

t_TOST

Two One-Sided T-tests (TOST) for Equivalence Testing

Description

[Stable]

Performs equivalence testing using the Two One-Sided Tests (TOST) procedure with t-tests. This function supports one-sample, two-sample (independent), and paired t-tests, providing a comprehensive framework for testing equivalence or minimal effects hypotheses.

The TOST procedure is designed for situations where you want to demonstrate that an effect falls within specified bounds (equivalence testing) or exceeds specified bounds (minimal effects testing).

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'
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. |
| 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 (symmetric bound, 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 instead). |
| high_eqbound | upper equivalence bounds (deprecated, use eqb instead). |
| eqbound_type | Type of equivalence bound. Can be '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 | An 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 | 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"). |

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).

The output combines three statistical tests:

1. A traditional two-tailed t-test (null hypothesis: difference = mu)
2. Lower bound test (one-tailed t-test against the lower equivalence bound)
3. Upper bound test (one-tailed t-test against the upper equivalence bound)

For equivalence testing (hypothesis = "EQU"):

- **Significant TOST:** Both one-sided tests are significant ($p < \alpha$), indicating the effect is significantly within the equivalence bounds

For minimal effects testing (hypothesis = "MET"):

- **Significant TOST:** At least one one-sided test is significant ($p < \alpha$), indicating the effect is significantly outside at least one of the bounds

Notes:

- For equivalence testing, the equivalence bounds represent the smallest effect sizes considered meaningful.
- When using `eqbound_type = "SMD"`, be aware that this can produce biased results and raw bounds are generally recommended.
- The function provides standardized effect sizes (Cohen's *d* or Hedges' *g*) along with their confidence intervals.
- For paired/repeated measures designs, setting `rm_correction = TRUE` adjusts the standardized effect size calculation to account for the correlation between measures.

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), *d*_{low} (lower CI bound), *d*_{high} (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.

Purpose

Use this function when:

- You want to show that two groups are practically equivalent
- You need to demonstrate that an effect is at least as large as a meaningful threshold
- You want to test if an observed effect is too small to be of practical importance

See Also

Other TOST: [boot_log_TOST\(\)](#), [boot_t_TOST\(\)](#), [simple_hstest\(\)](#), [tsum_TOST\(\)](#), [wilcox_TOST\(\)](#)

Examples

```
# Example 1: Basic Two-Sample Test
data(mtcars)
# Testing if the difference in mpg between automatic and manual
# transmission cars falls within ±3 mpg
result <- t_TOST(mpg ~ am, data = mtcars, eqb = 3)

# Example 2: Paired Sample Test with Specific Bounds
```

```

data(sleep)
result <- t_TOST(x = sleep$extra[sleep$group == 1],
                y = sleep$extra[sleep$group == 2],
                paired = TRUE,
                eqb = c(-0.5, 2)) # Asymmetric bounds

# Example 3: One Sample Equivalence Test
result <- t_TOST(x = rnorm(30, mean = 0.1, sd = 1),
                eqb = 1)

# Example 4: Minimal Effects Test
result <- t_TOST(mpg ~ am,
                data = mtcars,
                eqb = 1.5,
                hypothesis = "MET")

```

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

```

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

```

Default S3 method:

```

wilcox_TOST(
  x,
  y = NULL,
  hypothesis = "EQU",
  paired = FALSE,
  eqb,

```

```

    low_eqbound,
    high_eqbound,
    ses = c("rb", "odds", "logodds", "cstat"),
    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. |
| paired | a logical indicating whether you want a paired t-test. |
| eqb | Equivalence bound. Can provide 1 value (symmetric bound, 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 instead). |
| high_eqbound | upper equivalence bounds (deprecated, use eqb instead). |
| 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 <code>model.frame</code>) containing the variables in the formula formula. By default the variables are taken from <code>environment(formula)</code> . |
| 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 <code>getOption("na.action")</code> . |

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 μ 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.

References

David F. Bauer (1972). Constructing confidence sets using rank statistics. *Journal of the American Statistical Association* 67, 687–690. doi: 10.1080/01621459.1972.10481279.

Myles Hollander and Douglas A. Wolfe (1973). *Nonparametric Statistical Methods*. New York: John Wiley & Sons. Pages 27–33 (one-sample), 68–75 (two-sample). Or second edition (1999).

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_hstest\(\)](#), [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 and perform equivalence testing.

Usage

```
z_cor_test(
  x,
  y,
  alternative = c("two.sided", "less", "greater", "equivalence", "minimal.effect"),
  method = c("pearson", "kendall", "spearman"),
  alpha = 0.05,
  null = 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: <ul style="list-style-type: none"> • "two.sided": correlation is not equal to null (default) • "greater": correlation is greater than null • "less": correlation is less than null • "equivalence": correlation is within the equivalence bounds (TOST) • "minimal.effect": correlation is outside the equivalence bounds (TOST) <p>You can specify just the initial letter.</p> |
| 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 or vector indicating the null hypothesis value(s): <ul style="list-style-type: none"> • For standard tests: a single value (default = 0) • For equivalence/minimal effect tests: either a single value (symmetric bounds \pmvalue will be created) or a vector of two values representing the lower and upper bounds |

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 .

The function supports both standard hypothesis testing and equivalence/minimal effect testing:

- For standard tests (two.sided, less, greater), the function tests whether the correlation differs from the null value (typically 0).
- For equivalence testing ("equivalence"), it determines whether the correlation falls within the specified bounds, which can be set asymmetrically.
- For minimal effect testing ("minimal.effect"), it determines whether the correlation falls outside the specified bounds.

When performing equivalence or minimal effect testing:

- If a single value is provided for null, symmetric bounds \pm value will be used
- If two values are provided for null, they will be used as the lower and upper bounds

See vignette("correlations") for more details.

Value

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

- **p.value**: the p-value of the test.
- **statistic**: the value of the test statistic with a name describing it.
- **parameter**: the degrees of freedom or number of observations.
- **conf.int**: a confidence interval for the measure of association appropriate to the specified alternative hypothesis.
- **estimate**: the estimated measure of association, with name "cor", "tau", or "rho" corresponding to the method employed.
- **stderr**: the standard error of the test statistic.
- **null.value**: the value of the association measure under the null hypothesis.
- **alternative**: character string indicating the alternative hypothesis.
- **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 approach. *British Journal of Mathematical and Statistical Psychology*, 63(3), 527-537. <https://doi.org/10.1348/000711009X475> formula page 531.

See Also

Other Correlations: [boot_cor_test\(\)](#), [corsum_test\(\)](#), [plot_cor\(\)](#), [power_z_cor\(\)](#)

Examples

```
# Example 1: Standard significance test
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)
z_cor_test(x, y, method = "kendall", alternative = "t", null = 0)

# Example 2: Minimal effect test
# Testing if correlation is meaningfully different from  $\pm 0.2$ 
z_cor_test(x, y, method = "kendall", alternative = "min", null = 0.2)

# Example 3: Equivalence test with Pearson correlation
# Testing if correlation is equivalent to zero within  $\pm 0.3$ 
z_cor_test(x, y, method = "pearson", alternative = "equivalence", null = 0.3)

# Example 4: Using asymmetric bounds
# Testing if correlation is within bounds of  $-0.1$  and  $0.4$ 
z_cor_test(x, y, method = "spearman",
           alternative = "equivalence", null = c(-0.1, 0.4))
```

Index

- * **Correlations**
 - boot_cor_test, 10
 - corsum_test, 40
 - plot_cor, 59
 - power_z_cor, 76
 - z_cor_test, 112
- * **Robust tests**
 - boot_log_TOST, 14
 - boot_t_test, 24
 - boot_t_TOST, 28
 - brunner_munzel, 31
 - log_TOST, 57
 - wilcox_TOST, 109
- * **TOST**
 - boot_log_TOST, 14
 - boot_t_TOST, 28
 - simple_hstest, 82
 - t_TOST, 105
 - tsum_TOST, 99
 - wilcox_TOST, 109
- * **compare studies**
 - boot_compare_cor, 4
 - boot_compare_smd, 7
 - compare_cor, 34
 - compare_smd, 37
- * **datasets**
 - hawthorne, 54
- * **effect sizes**
 - boot_ses_calc, 17
 - boot_smd_calc, 20
 - ses_calc, 79
 - smd_calc, 85
- * **f-test**
 - equ_anova, 50
 - equ_ftest, 52
- * **hstest**
 - as_hstest, 3
 - hstest-helpers, 55
 - simple_hstest, 82
- * **plotting functions**
 - plot_cor, 59
 - plot_pes, 61
 - plot_smd, 63
- * **power**
 - power_eq_f, 71
 - power_t_TOST, 75
 - power_z_cor, 76
- as.data.frame, 43, 45, 47, 48, 50
- as_hstest, 3, 56, 84
- boot_compare_cor, 4, 10, 36, 39
- boot_compare_smd, 6, 7, 36, 39
- boot_cor_test, 10, 42, 60, 77, 113
- boot_log_TOST, 14, 27, 31, 34, 59, 84, 102, 108, 111
- boot_ses_calc, 17, 23, 81, 88
- boot_smd_calc, 19, 20, 81, 88
- boot_t_test, 16, 24, 31, 34, 59, 111
- boot_t_TOST, 16, 27, 28, 34, 59, 84, 102, 108, 111
- brunner_munzel, 16, 27, 31, 31, 59, 111
- compare_cor, 6, 10, 34, 39
- compare_smd, 6, 10, 36, 37
- corsum_test, 13, 40, 60, 77, 93, 113
- dataTOSTone, 42
- dataTOSTpaired, 44
- dataTOSTr, 46
- dataTOSTtwo, 47
- datatosttwoprop, 49
- describe (TOSTt-methods), 94
- describe.TOSTnp (TOSTnp-methods), 89
- describe_hstest (hstest-helpers), 55
- df_hstest (hstest-helpers), 55
- equ_anova, 50, 53
- equ_ftest, 51, 52
- extract_r_paired, 54

hawthorne, [54](#)
hstest-helpers, [55](#)

log_TOST, [16](#), [27](#), [31](#), [34](#), [57](#), [111](#)

np_ses (rbs), [77](#)

plot.TOSTt (TOSTt-methods), [94](#)
plot_cor, [13](#), [42](#), [59](#), [63](#), [65](#), [77](#), [113](#)
plot_pes, [60](#), [61](#), [65](#)
plot_smd, [60](#), [63](#), [63](#)
power_eq_f, [71](#), [76](#), [77](#)
power_t_TOST, [66](#), [67](#), [69](#), [72](#), [75](#), [77](#)
power_twoprop, [72](#)
power_z_cor, [13](#), [42](#), [60](#), [72](#), [76](#), [76](#), [113](#)
powerTOSTone, [66](#)
powerTOSTpaired, [67](#)
powerTOSTr (power_z_cor), [76](#)
powerTOSTtwo, [69](#)
powerTOSTtwo.prop (power_twoprop), [72](#)
print.TOSTnp (TOSTnp-methods), [89](#)
print.TOSTt (TOSTt-methods), [94](#)
prop.test, [102](#), [103](#)

rbs, [77](#)

ses_calc, [19](#), [23](#), [77](#), [79](#), [88](#)
simple_hstest, [4](#), [16](#), [31](#), [56](#), [82](#), [102](#), [108](#), [111](#)
smd_calc, [19](#), [23](#), [81](#), [85](#)
stats::wilcox.test, [78](#), [110](#)

t.test, [32](#)
t_TOST, [16](#), [31](#), [84](#), [102](#), [105](#), [111](#)
TOSTmeta, [88](#)
TOSTnp-methods, [89](#), [90](#)
TOSTone, [90](#)
TOSTpaired, [92](#)
TOSTr, [93](#)
TOSTt-methods, [94](#)
TOSTtwo, [96](#)
TOSTtwo.prop, [98](#)
tsum_TOST, [16](#), [31](#), [54](#), [84](#), [90](#), [92](#), [96](#), [99](#), [108](#),
[111](#)
twoprop_test, [98](#), [102](#)

wilcox_TOST, [16](#), [27](#), [31](#), [34](#), [59](#), [84](#), [102](#), [108](#),
[109](#)

z_cor_test, [13](#), [42](#), [60](#), [77](#), [112](#)