Package 'MOTE'

December 15, 2025

Version 1.2.2

Date 2025-12-05

Title Effect Size and Confidence Interval Calculator

Depends R (>= 3.1.0)

Imports stats

Suggests car, testthat (>= 3.0.0)

Description Measure of the Effect ('MOTE') is an effect size calculator, including a wide variety of effect sizes in the mean differences family (all versions of d) and the variance overlap family (eta, omega, epsilon, r). 'MOTE' provides non-central confidence intervals for each effect size, relevant test statistics, and output for reporting in APA Style (American Psychological Association, 2010, <ISBN:1433805618>) with 'LaTeX'. In research, an over-reliance on p-values may conceal the fact that a study is under-powered (Halsey, Curran-Everett, Vowler, & Drummond, 2015 <doi:10.1038/nmeth.3288>). A test may be statistically significant, yet practically inconsequential (Fritz, Scherndl, & Kühberger, 2012 <doi:10.1177/0959354312436870>). Although the American Psychological Association has long advocated for the inclusion of effect sizes (Wilkinson & American Psychological Association Task Force on Statistical Inference, 1999 <doi:10.1037/0003-066X.54.8.594>), the vast majority of peer-reviewed, published academic studies stop short of reporting effect sizes and confidence intervals (Cumming, 2013, <doi:10.1177/0956797613504966>). 'MOTE' simplifies the use and interpretation of effect sizes and confidence intervals.

License LGPL-3

Encoding UTF-8

URL `https://github.com/doomlab/MOTE`

BugReports `https://github.com/doomlab/MOTE/issues`

LazyData true

RoxygenNote 7.3.3

Config/testthat/edition 3

Language en-US

NeedsCompilation no

Author Erin M. Buchanan [aut, cre] (ORCID:
<https://orcid.org/0000-0002-9689-4189>),
Amber M Gillenwaters [aut] (ORCID:
<https://orcid.org/0000-0002-7580-3591>),
John E. Scofield [aut] (ORCID: <https://orcid.org/0000-0001-6345-1181>),
K. D. Valentine [aut] (ORCID: <https://orcid.org/0000-0001-6349-5395>)

Maintainer Erin M. Buchanan <buchananlab@gmail.com>

Repository CRAN

Date/Publication 2025-12-15 06:50:46 UTC

Contents

`apa` *Format numbers for APA-style reporting*

Description

Create "pretty" character representations of numeric values with a fixed number of decimal places, optionally keeping or omitting the leading zero for values between -1 and 1.

Usage

```
apa(value, decimals = 3, leading = TRUE)
```

Arguments

`value`	Numeric input: a single number, vector, matrix, or a data frame with all-numeric columns. Non-numeric inputs will error.
`decimals`	A single non-negative integer giving the number of decimal places to keep in the output.
`leading`	Logical: ‘TRUE‘ to keep leading zeros on decimals (e.g., ‘0.25‘), ‘FALSE‘ to drop them (e.g., ‘.25‘). Default is ‘TRUE‘.

Details

This function formats numbers for inclusion in manuscripts and reports. - When ‘leading = TRUE‘, numbers are rounded and padded to ‘decimals‘ places, keeping the leading zero for values with absolute value < 1. - When ‘leading = FALSE‘, the leading zero before the decimal point is removed for values with absolute value < 1. If ‘value‘ is a data frame, all columns must be numeric; otherwise an error is thrown.

Value

A character vector/array (matching the shape of ‘value‘) containing the formatted numbers.

Examples

```
apa(0.54674, decimals = 3, leading = TRUE)   # "0.547"
apa(c(0.2, 1.2345, -0.04), decimals = 2)     # "0.20" "1.23" "-0.04"
apa(matrix(c(0.12, -0.9, 2.3, 10.5), 2), decimals = 1, leading = FALSE)
# returns a character matrix with ".1", "-.9", "2.3", "10.5"
```

`bn1_data` *Between-Subjects One-Way ANOVA Example Data*

Description

Ratings of close interpersonal attachments for 45-year-old participants, categorized by self-reported health status: excellent, fair, or poor. This dataset is designed for use with functions such as `eta.F`, `eta.full.SS`, `omega.F`, `omega.full.SS`, and `epsilon.full.SS`.

Usage

```
data(bn1_data)
```

Format

A data frame with *n* rows and 2 variables:

group Factor with levels `"poor"`, `"fair"`, and `"excellent"`.

friends Numeric rating of close interpersonal attachments.

Source

Simulated data inspired by Nolan & Heinzen (4th ed.), *Statistics for the Behavioral Sciences*. Generated for instructional examples in the MOTE package.

References

Nolan, S. A., & Heinzen, T. E. (*4th ed.*). *Statistics for the Behavioral Sciences*. Macmillan Learning.

`bn2_data` *Between-Subjects Two-Way ANOVA Example Data*

Description

Example data for a between-subjects two-way ANOVA examining whether athletic spending differs by sport type and coach experience. This dataset contains simulated athletic budgets (in thousands of dollars) for baseball, basketball, football, soccer, and volleyball teams, with either a new or old coach. Designed for use with `omega.partial.SS.bn`, `eta.partial.SS`, and other between-subjects ANOVA designs.

Usage

```
data(bn2_data)
```

Format

A data frame with 3 variables:

coach Factor with levels `"old"` and `"new"` indicating coach experience.

type Factor indicating sport type: `"baseball"`, `"basketball"`, `"football"`, `"soccer"`, or `"volleyball"`.

money Numeric. Athletic spending in thousands of dollars.

Source

Simulated data generated for instructional examples in the MOTE package.

`chisq_data` *Chi-Square Test Example Data*

Description

Example data for a chi-square test of independence. Individuals were polled and asked to report their number of friends (low, medium, high) and their number of children (1, 2, 3 or more). The analysis examines whether there is an association between friend group size and number of children. It was hypothesized that those with more children may have less time for friendship-maintaining activities.

Usage

```
data(chisq_data)
```

Format

A data frame with 2 variables:

friends Factor with levels `"low"`, `"medium"`, and `"high"` indicating self-reported number of friends.

kids Factor with levels `"1"`, `"2"`, and `"3+"` indicating number of children.

Source

Simulated data inspired by Nolan & Heinzen (4th ed.), *Statistics for the Behavioral Sciences*. Generated for instructional examples in the MOTE package.

References

Nolan, S. A., & Heinzen, T. E. (*4th ed.*). *Statistics for the Behavioral Sciences*. Macmillan Learning.

`ci_r2` *Confidence interval for R^2 (exported helper)*

Description

Compute a confidence interval for the coefficient of determination (R^2). This implementation follows MBESS (Ken Kelley) and is exported here to avoid importing many dependencies. It supports cases with random or fixed predictors and can be parameterized via either degrees of freedom or sample size (n) and number of predictors (p/k).

Usage

```
ci_r2(
  r2 = NULL,
  df1 = NULL,
  df2 = NULL,
  conf_level = 0.95,
  random_predictors = TRUE,
  random_regressors = random_predictors,
  f_value = NULL,
  n = NULL,
  p = NULL,
  k = NULL,
  alpha_lower = NULL,
  alpha_upper = NULL,
  tol = 1e-09
)
```

Arguments

`r2`	Numeric. The observed R^2 (may be ‘NULL‘ if ‘f_value‘ is supplied).
`df1`	Integer. Numerator degrees of freedom from F.
`df2`	Integer. Denominator degrees of freedom from F.
`conf_level`	Numeric in (0, 1). Two-sided confidence level for a symmetric confidence interval. Default is ‘0.95‘. Cannot be used with ‘alpha_lower‘ or ‘alpha_upper‘.
`random_predictors`	Logical. If ‘TRUE‘ (default), compute limits for random predictors; if ‘FALSE‘, compute limits for fixed predictors.
`random_regressors`	Logical. Backwards-compatible alias for ‘random_predictors‘. If supplied, it overrides ‘random_predictors‘.
`f_value`	Numeric. The observed F statistic from the study.
`n`	Integer. Sample size.
`p`	Integer. Number of predictors.

`k`	Integer. Alias for ‘p‘ (number of predictors). If supplied along with ‘p‘, they must be equal.
`alpha_lower`	Numeric. Lower-tail noncoverage probability (cannot be used with ‘conf_level‘).
`alpha_upper`	Numeric. Upper-tail noncoverage probability (cannot be used with ‘conf_level‘).
`tol`	Numeric. Tolerance for the iterative method determining critical values. Default is ‘1e-9‘.

Details

If ‘n‘ and ‘p‘ (or ‘k‘) are provided, ‘df1‘ and ‘df2‘ are derived as ‘df1 = p‘ and ‘df2 = n - p - 1‘. Conversely, if ‘df1‘ and ‘df2‘ are provided, ‘n = df1 + df2 + 1‘ and ‘p = df1‘.

Value

A named list with the following elements:

`lower_conf_limit_r2`	The lower confidence limit for R^2.
`prob_less_lower`	Probability associated with values less than the lower limit.
`upper_conf_limit_r2`	The upper confidence limit for R^2.
`prob_greater_upper`	Probability associated with values greater than the upper limit.

References

Kelley, K. (2007). Methods for the behavioral, educational, and social sciences: An R package (MBESS).

`delta_ind_t` *d_δ for Between Subjects with Control Group SD Denominator*

Description

This function displays d_δ for between subjects data and the non-central confidence interval using the control group standard deviation as the denominator.

Usage

```
delta_ind_t(m1, m2, sd1, sd2, n1, n2, a = 0.05)

delta.ind.t(m1, m2, sd1, sd2, n1, n2, a = 0.05)
```

Arguments

`m1`	mean from control group
`m2`	mean from experimental group
`sd1`	standard deviation from control group
`sd2`	standard deviation from experimental group
`n1`	sample size from control group
`n2`	sample size from experimental group
`a`	significance level

Details

To calculate d_δ, the mean of the experimental group is subtracted from the mean of the control group, which is divided by the standard deviation of the control group.

$$d_\delta = \frac{m_1 - m_2}{sd_1}$$

Learn more on our example page.

Value

Provides the effect size (Cohen's d) with associated confidence intervals, the t-statistic, the confidence intervals associated with the means of each group, as well as the standard deviations and standard errors of the means for each group.

`d`	d-delta effect size
`dlow`	lower level confidence interval of d-delta value
`dhigh`	upper level confidence interval of d-delta value
`M1`	mean of group one
`sd1`	standard deviation of group one mean
`se1`	standard error of group one mean
`M1low`	lower level confidence interval of group one mean
`M1high`	upper level confidence interval of group one mean
`M2`	mean of group two
`sd2`	standard deviation of group two mean
`se2`	standard error of group two mean
`M2low`	lower level confidence interval of group two mean
`M2high`	upper level confidence interval of group two mean
`spooled`	pooled standard deviation
`sepooled`	pooled standard error
`n1`	sample size of group one
`n2`	sample size of group two

`df`	degrees of freedom (n1 - 1 + n2 - 1)
`t`	t-statistic
`p`	p-value
`estimate`	the d statistic and confidence interval in APA style for markdown printing
`statistic`	the t-statistic in APA style for markdown printing

Examples

```
# The following example is derived from the "indt_data"
# dataset, included in the MOTE library.

# A forensic psychologist conducted a study to examine whether
# being hypnotized during recall affects how well a witness
# can remember facts about an event. Eight participants
# watched a short film of a mock robbery, after which
# each participant was questioned about what he or she had
# seen. The four participants in the experimental group
# were questioned while they were hypnotized. The four
# participants in the control group received the same
# questioning without hypnosis.

    hyp <- t.test(correctq ~ group, data = indt_data)

# You can type in the numbers directly, or refer to the dataset,
# as shown below.

    delta_ind_t(m1 = 17.75, m2 = 23,
               sd1 = 3.30, sd2 = 2.16,
                n1 = 4, n2 = 4, a = .05)

    delta_ind_t(17.75, 23, 3.30, 2.16, 4, 4, .05)

    delta_ind_t(mean(indt_data$correctq[indt_data$group == 1]),
            mean(indt_data$correctq[indt_data$group == 2]),
            sd(indt_data$correctq[indt_data$group == 1]),
            sd(indt_data$correctq[indt_data$group == 2]),
            length(indt_data$correctq[indt_data$group == 1]),
            length(indt_data$correctq[indt_data$group == 2]),
            .05)

# Contrary to the hypothesized result, the group that underwent hypnosis were
# significantly less accurate while reporting facts than the control group
# with a large effect size, t(6) = -2.66, p = .038, d_delta = 1.59.
```

`dept_data` *Dependent t Example Data*

Description

Dataset for use in `d_dep_t_diff`, `d_dep_t_diff_t`, `d_dep_t_avg`, and `d_dep_t_rm` exploring the before and after effects of scifi movies on supernatural beliefs.

Usage

```
data(dept_data)
```

Format

A data frame of before and after scores for rating supernatural beliefs.

before: scores rated before watching a scifi movie after: scores rated after watching a scifi movie

References

Nolan and Heizen Statistics for the Behavioral Sciences

`d_dep_t_avg` *Cohen's d for Paired t Using the Average SD Denominator*

Description

Note on function names: This function now uses the snake_case name ‘d_dep_t_avg()‘ to follow modern R style guidelines and CRAN recommendations. The dotted version ‘d.dep.t.avg()‘ is still included as a wrapper for backward compatibility, so older code will continue to work. Both functions produce identical results, but new code should use ‘d_dep_t_avg()‘. The output function also provides backwards compatibility and new snake case variable names.

Usage

```
d_dep_t_avg(m1, m2, sd1, sd2, n, a = 0.05)

d.dep.t.avg(m1, m2, sd1, sd2, n, a = 0.05)
```

Arguments

`m1`	Mean from the first level/occasion.
`m2`	Mean from the second level/occasion.
`sd1`	Standard deviation from the first level/occasion.
`sd2`	Standard deviation from the second level/occasion.
`n`	Sample size (number of paired observations).
`a`	Significance level (alpha) for the confidence interval. Must be in (0, 1).

Details

Compute Cohen's d_{av} and a noncentral-t confidence interval for repeated-measures (paired-samples) designs using the **average of the two standard deviations** as the denominator.

The effect size is defined as the mean difference divided by the average SD:

$$d_{av} = \frac{m_1 - m_2}{(s_1 + s_2)/2}.$$

The test statistic used for the noncentral-t confidence interval is based on the average of the two standard errors, $se_i = s_i/\sqrt{n}$:

$$t = \frac{m_1 - m_2}{\left(\frac{s_1}{\sqrt{n}} + \frac{s_2}{\sqrt{n}}\right)/2}.$$

See the online example for additional context: Learn more on our example page.

Value

A list with the following elements:

d Cohen's d_{av}.

dlow Lower limit of the $(1 - \alpha)$ confidence interval for d_{av}.

dhigh Upper limit of the $(1 - \alpha)$ confidence interval for d_{av}.

M1, M2 Group means.

M1low, M1high, M2low, M2high Confidence interval bounds for each mean.

sd1, sd2 Standard deviations.

se1, se2 Standard errors of the means.

n Sample size.

df Degrees of freedom $(n - 1)$.

estimate APA-style formatted string for reporting d_{av} and its CI.

Examples

```
# The following example is derived from the "dept_data" dataset included
# in the MOTE package.

# Suppose seven people completed a measure of belief in the supernatural
# before and after watching a sci-fi movie.
# Higher scores indicate stronger belief.

    t.test(dept_data$before, dept_data$after, paired = TRUE)

# You can type in the numbers directly, or refer to the
# dataset, as shown below.

    d_dep_t_avg(m1 = 5.57, m2 = 4.43, sd1 = 1.99,
                sd2 = 2.88, n = 7, a = .05)
```

```
d_dep_t_avg(5.57, 4.43, 1.99, 2.88, 7, .05)

d_dep_t_avg(mean(dept_data$before), mean(dept_data$after),
            sd(dept_data$before), sd(dept_data$after),
            length(dept_data$before), .05)
```

`d_dep_t_diff` *Cohen's d for Paired t Using the SD of Difference Scores*

Description

Note on function and output names: This effect size is now implemented with the snake_case function name ‘d_dep_t_diff()‘ to follow modern R style guidelines. The original dotted version ‘d.dep.t.diff()‘ is still available as a wrapper for backward compatibility, and both functions return the same list. The returned object includes both the original element names (e.g., ‘mdiff‘, ‘Mlow‘, ‘Mhigh‘, ‘sddiff‘) and newer snake_case aliases (e.g., ‘m_diff‘, ‘m_diff_lower_limit‘, ‘m_diff_upper_limit‘, ‘sd_diff‘). New code should prefer ‘d_dep_t_diff()‘ and the snake_case output names, but existing code using the older names will continue to work.

Usage

```
d_dep_t_diff(mdiff, sddiff, n, a = 0.05)

d.dep.t.diff(mdiff, sddiff, n, a = 0.05)
```

Arguments

`mdiff`	Mean of the difference scores.
`sddiff`	Standard deviation of the difference scores.
`n`	Sample size (number of paired observations).
`a`	Significance level (alpha) for the confidence interval. Must be in (0, 1).

Details

Compute Cohen's d_z and a noncentral-t confidence interval for repeated-measures (paired-samples) designs using the **standard deviation of the difference scores** as the denominator.

The effect size is defined as:

$$d_z = \frac{\bar{X}_D}{s_D}$$

where $\bar{X}_D$ is the mean of the difference scores and s_D is the standard deviation of the difference scores.

The corresponding t statistic for the paired-samples t-test is:

$$t = \frac{\bar{X}_D}{s_D/\sqrt{n}}$$

See the online example for additional context: Learn more on our example page.

Value

A list with the following elements:

d Cohen's d_z.

dlow Lower limit of the $(1 - \alpha)$ confidence interval for d_z.

dhigh Upper limit of the $(1 - \alpha)$ confidence interval for d_z.

mdiff Mean difference score.

Mlow, Mhigh Confidence interval bounds for the mean difference.

sddiff Standard deviation of the difference scores.

se Standard error of the difference scores.

n Sample size.

df Degrees of freedom $(n - 1)$.

t t-statistic.

p p-value.

estimate APA-style formatted string for reporting d_z and its CI.

statistic APA-style formatted string for reporting the t-statistic and p-value.

Examples

```
# Example derived from the "dept_data" dataset included in MOTE

# Suppose seven people completed a measure of belief in the supernatural
# before and after watching a sci-fi movie.
# Higher scores indicate stronger belief.

t.test(dept_data$before, dept_data$after, paired = TRUE)

# Direct entry of summary statistics:
d_dep_t_diff(mdiff = 1.14, sddiff = 2.12, n = 7, a = .05)

# Equivalent shorthand:
d_dep_t_diff(1.14, 2.12, 7, .05)

# Using raw data from the dataset:
d_dep_t_diff(mdiff = mean(dept_data$before - dept_data$after),
             sddiff = sd(dept_data$before - dept_data$after),
             n = length(dept_data$before),
             a = .05)
```

`d_dep_t_diff_t` *Cohen's d from t for Paired Samples Using the SD of Difference Scores*

Description

Compute Cohen's d_z from a paired-samples t-statistic and provide a noncentral-t confidence interval, using the **standard deviation of the difference scores** as the denominator.

Usage

```
d_dep_t_diff_t(t_value, t = NULL, n, a = 0.05)
```

Arguments

`t_value`	t-statistic from a paired-samples t-test.
`t`	for backwards compatibility, you can also give t.
`n`	Sample size (number of paired observations).
`a`	Significance level (alpha) for the confidence interval. Must be in (0, 1).

Details

For paired designs, d_z can be obtained directly from the t-statistic:

$$d_z = \frac{t}{\sqrt{n}},$$

where n is the number of paired observations (df = $n - 1$). The $(1 - \alpha)$ confidence interval for d_z is derived from the noncentral t distribution for the observed t and df.

See the online example for additional context: Learn more on our example page.

Value

A list with the following elements:

d Cohen's d_z.

dlow Lower limit of the $(1 - \alpha)$ confidence interval for d_z.

dhigh Upper limit of the $(1 - \alpha)$ confidence interval for d_z.

n Sample size.

df Degrees of freedom $(n - 1)$.

t t-statistic.

p p-value.

estimate APA-style formatted string for reporting d_z and its CI.

statistic APA-style formatted string for reporting the t-statistic and p-value.

Examples

```
# Example derived from the "dept_data" dataset included in MOTE

# Suppose seven people completed a measure before and after an intervention.
# Higher scores indicate stronger endorsement.

    scifi <- t.test(dept_data$before, dept_data$after, paired = TRUE)

# The t-test value was 1.43. You can type in the numbers directly,
# or refer to the dataset, as shown below.

    d_dep_t_diff_t(t_value = 1.43, n = 7, a = .05)

    d_dep_t_diff_t(t_value = scifi$statistic,
        n = length(dept_data$before), a = .05)
```

`d_dep_t_rm` *Cohen's d for Paired t Controlling for Correlation (Repeated Measures)*

Description

Compute Cohen's d_{rm} and a noncentral-t confidence interval for repeated-measures (paired-samples) designs **controlling for the correlation between occasions**. The denominator uses the SDs and their correlation.

Usage

```
d_dep_t_rm(m1, m2, sd1, sd2, r, n, a = 0.05)
```

Arguments

`m1`	Mean from the first level/occasion.
`m2`	Mean from the second level/occasion.
`sd1`	Standard deviation from the first level/occasion.
`sd2`	Standard deviation from the second level/occasion.
`r`	Correlation between the two levels/occasions.
`n`	Sample size (number of paired observations).
`a`	Significance level (alpha) for the confidence interval. Must be in (0, 1).

Details

The effect size is defined as:

$$d_{rm} = \frac{m_1 - m_2}{\sqrt{s_1^2 + s_2^2 - 2 r s_1 s_2}} \sqrt{2(1-r)}.$$

The test statistic used for the noncentral-t confidence interval is:

$$t = \frac{m_1 - m_2}{\sqrt{\frac{s_1^2 + s_2^2 - 2rs_1s_2}{n}}} \sqrt{2(1-r)}.$$

See the online example for additional context: Learn more on our example page.

Value

A list with the following elements:

d Cohen's d_{rm}.

dlow Lower limit of the $(1-\alpha)$ confidence interval for d_{rm}.

dhigh Upper limit of the $(1-\alpha)$ confidence interval for d_{rm}.

M1, M2 Group means.

M1low, M1high, M2low, M2high Confidence interval bounds for each mean.

sd1, sd2 Standard deviations.

se1, se2 Standard errors of the means.

r Correlation between occasions.

n Sample size.

df Degrees of freedom $(n-1)$.

estimate APA-style formatted string for reporting d_{rm} and its CI.

Examples

```
# Example derived from the "dept_data" dataset included in MOTE

    t.test(dept_data$before, dept_data$after, paired = TRUE)

    scifi_cor <- cor(dept_data$before, dept_data$after, method = "pearson",
                     use = "pairwise.complete.obs")

# Direct entry of summary statistics, or refer to the dataset as shown below.

    d_dep_t_rm(m1 = 5.57, m2 = 4.43, sd1 = 1.99,
               sd2 = 2.88, r = .68, n = 7, a = .05)

    d_dep_t_rm(5.57, 4.43, 1.99, 2.88, .68, 7, .05)

    d_dep_t_rm(mean(dept_data$before), mean(dept_data$after),
               sd(dept_data$before), sd(dept_data$after),
               scifi_cor, length(dept_data$before), .05)
```

`d_effect` *General interface for Cohen's d*

Description

'd_effect()' is a convenience wrapper that will route to the appropriate Cohen's *d* helper function based on the arguments supplied. This allows users to call a single function for different study designs while maintaining backward compatibility with the more specific helpers.

Usage

```
d_effect(
  m1 = NULL,
  m2 = NULL,
  sd1 = NULL,
  sd2 = NULL,
  u = NULL,
  sig = NULL,
  r = NULL,
  mdiff = NULL,
  sddiff = NULL,
  t_value = NULL,
  z_value = NULL,
  p1 = NULL,
  p2 = NULL,
  n1 = NULL,
  n2 = NULL,
  n = NULL,
  a = 0.05,
  design,
  ...
)
```

Arguments

`m1`	Means of the two conditions or measurements.
`m2`	Means of the two conditions or measurements.
`sd1`	Standard deviations for the two conditions or measurements.
`sd2`	Standard deviations for the two conditions or measurements.
`u`	Population or comparison mean for one-sample t-designs, used when 'design = "single_t"'.
`sig`	Population standard deviation for z-based designs, used when 'design = "z_mean"'.
`r`	Correlation between the paired measurements (used for repeated-measures designs such as '"dep_t_rm"').
`mdiff`	Mean difference between paired observations.

`sddiff`	Standard deviation of the difference scores.
`t_value`	t statistic value for the test. Used in designs where the effect size is derived directly from a reported t-value (e.g., ‘"dep_t_diff_t"‘, ‘"ind_t_t"‘, or ‘"single_t_t"‘).
`z_value`	z statistic value for the test. Used in designs where the effect size is derived directly from a reported z-value (e.g., ‘"z_z"‘).
`p1`	Proportion for group one (between 0 and 1), used in the ‘"prop"‘ design.
`p2`	Proportion for group two (between 0 and 1), used in the ‘"prop"‘ design.
`n1`	Sample sizes for the two independent groups (used for independent-groups designs such as ‘"ind_t"‘).
`n2`	Sample sizes for the two independent groups (used for independent-groups designs such as ‘"ind_t"‘).
`n`	Sample size (number of paired observations).
`a`	Significance level used when computing confidence intervals. Defaults to ‘0.05‘.
`design`	Character string specifying the study design.
`...`	Reserved for future arguments and passed on to the underlying helper functions when appropriate.

Details

- ‘"delta_ind_t"‘ — independent-groups t-test using the delta effect size, where the SD of group 1 is used as the denominator. Supply ‘m1‘, ‘m2‘, ‘sd1‘, ‘sd2‘, ‘n1‘, and ‘n2‘. In this case, ‘d_effect()‘ will call [delta.ind.t()] with the same arguments.

- ‘"g_ind_t"‘ — independent-groups t-test using Hedges’ g, which applies a small-sample correction to the standardized mean difference. Supply ‘m1‘, ‘m2‘, ‘sd1‘, ‘sd2‘, ‘n1‘, and ‘n2‘. In this case, ‘d_effect()‘ will call [g_ind_t()] with the same arguments.

- ‘"z_z"‘ — one-sample z-test effect size where the *z* value is supplied directly along with the sample size ‘n‘. Supply ‘z_value‘ and ‘n‘. You may optionally supply ‘sig‘ (population SD) for descriptive reporting. In this case, ‘d_effect()‘ will call [d_z_z()] with the same arguments.

Value

A list with the same structure as returned by the underlying helper function. For the current paired-means case, this is the output of [d_dep_t_avg()], which includes:

- ‘d‘ – Cohen’s d using the average SD denominator.
- ‘dlow‘, ‘dhigh‘ – lower and upper confidence limits for ‘d‘.
- Snake_case aliases such as ‘d_lower_limit‘ and ‘d_upper_limit‘.
- Descriptive statistics (means, SDs, SEs, and their confidence limits) for each group.

Supported designs

- ‘"dep_t_avg"‘ — paired/dependent t-test with average SD denominator. Supply ‘m1‘, ‘m2‘, ‘sd1‘, ‘sd2‘, and ‘n‘. In this case, ‘d()‘ will call [d_dep_t_avg()] with the same arguments.

- ‘"dep_t_diff"‘ — paired/dependent t-test using the **SD of the difference scores**. Supply ‘mdiff‘, ‘sddiff‘, and ‘n‘. In this case, ‘d()‘ will call [d_dep_t_diff()] with the same arguments.

- ‘"dep_t_diff_t"‘ — paired/dependent t-test where the *t* value is supplied directly. Supply ‘t_value‘ and ‘n‘. In this case, ‘d()‘ will call [d_dep_t_diff_t()] with the same arguments.

- ‘"dep_t_rm"‘ — paired/dependent t-test using the repeated-measures effect size d_{rm}, which adjusts for the correlation between measurements. Supply ‘m1‘, ‘m2‘, ‘sd1‘, ‘sd2‘, ‘r‘, and ‘n‘. In this case, ‘d()‘ will call [d_dep_t_rm()] with the same arguments.

- ‘"ind_t"‘ — independent-groups t-test using the pooled SD (\(d_s\)). Supply ‘m1‘, ‘m2‘, ‘sd1‘, ‘sd2‘, ‘n1‘, and ‘n2‘. In this case, ‘d()‘ will call [d_ind_t()] with the same arguments.

- ‘"ind_t_t"‘ — independent-groups t-test where the *t* value is supplied directly. Supply ‘t_value‘, ‘n1‘, and ‘n2‘. In this case, ‘d()‘ will call [d_ind_t_t()] with the same arguments.

- ‘"g_ind_t"‘ — independent-groups t-test using Hedges’ g, which applies a small-sample correction to the standardized mean difference. Supply ‘m1‘, ‘m2‘, ‘sd1‘, ‘sd2‘, ‘n1‘, and ‘n2‘. In this case, ‘d_effect()‘ will call [g_ind_t()] with the same arguments.

- ‘"single_t"‘ — one-sample t-test effect size using the sample mean, population mean, sample SD, and sample size. Supply ‘m1‘ (sample mean), ‘u‘ (population mean), ‘sd1‘, and ‘n‘. In this case, ‘d()‘ will call [d_single_t()] with the same arguments.

- ‘"single_t_t"‘ — one-sample t-test effect size where the *t* value is supplied directly along with the sample size ‘n‘. In this case, ‘d()‘ will call [d_single_t_t()] with the same arguments.

- ‘"prop"‘ — independent proportions (binary outcome) using a standardized mean difference (SMD) that treats each proportion as the mean of a Bernoulli variable with pooled Bernoulli SD. Supply ‘p1‘, ‘p2‘, ‘n1‘, and ‘n2‘. In this case, ‘d()‘ will call [d_prop()] with the same arguments.

- ‘"prop_h"‘ — independent proportions (binary outcome) using Cohen’s \(h\) based on the arcsine-transformed difference between proportions. Supply ‘p1‘, ‘p2‘, ‘n1‘, and ‘n2‘. In this case, ‘d()‘ will call [h_prop()] with the same arguments.

- ‘"z_mean"‘ — one-sample z-test effect size using a known population standard deviation. Supply ‘m1‘ (sample mean), ‘u‘ (population mean), ‘sd1‘ (sample SD, used for descriptive CIs), ‘sig‘ (population SD), and ‘n‘. In this case, ‘d_effect()‘ will call [d_z_mean()] with the same arguments.

Examples

```
# Paired/dependent t-test using average SD denominator
# These arguments will route d() to d_dep_t_avg()
d_effect(
  m1 = 5.57, m2 = 4.43,
  sd1 = 1.99, sd2 = 2.88,
  n = 7, a = .05,
  design = "dep_t_avg"
)

# You can also call the helper directly
d_dep_t_avg(
  m1 = 5.57, m2 = 4.43,
```

```
    sd1 = 1.99, sd2 = 2.88,
    n = 7, a = .05
  )
```

`d_ind_t` *Cohen's d for Independent Samples Using the Pooled SD*

Description

Compute Cohen's d_s for between-subjects designs and a noncentral-t confidence interval using the **pooled standard deviation** as the denominator.

Usage

```
d_ind_t(m1, m2, sd1, sd2, n1, n2, a = 0.05)
```

Arguments

`m1`	Mean of group one.
`m2`	Mean of group two.
`sd1`	Standard deviation of group one.
`sd2`	Standard deviation of group two.
`n1`	Sample size of group one.
`n2`	Sample size of group two.
`a`	Significance level (alpha) for the confidence interval. Must be in (0, 1).

Details

The pooled standard deviation is:

$$s_{pooled} = \sqrt{\frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}}$$

Cohen's d_s is then:

$$d_s = \frac{m_1 - m_2}{s_{pooled}}$$

The corresponding t-statistic is:

$$t = \frac{m_1 - m_2}{\sqrt{s_{pooled}^2/n_1 + s_{pooled}^2/n_2}}$$

See the online example for additional context: Learn more on our example page.

Value

A list with the following elements:

d Cohen's d_s.

dlow Lower limit of the $(1-\alpha)$ confidence interval for d_s.

dhigh Upper limit of the $(1-\alpha)$ confidence interval for d_s.

M1, M2 Group means.

sd1, sd2 Standard deviations for each group.

se1, se2 Standard errors for each group mean.

M1low, M1high, M2low, M2high Confidence interval bounds for each group mean.

spooled Pooled standard deviation.

sepooled Pooled standard error.

n1, n2 Group sample sizes.

df Degrees of freedom ($n_1 - 1 + n_2 - 1$).

t t-statistic.

p p-value.

estimate APA-style formatted string for reporting d_s and its CI.

statistic APA-style formatted string for reporting the t-statistic and p-value.

Examples

```
# The following example is derived from the "indt_data" dataset
# included in MOTE.

# A forensic psychologist examined whether being hypnotized during recall
# affects how well a witness remembers facts about an event.

t.test(correctq ~ group, data = indt_data)

# Direct entry of summary statistics:
d_ind_t(m1 = 17.75, m2 = 23, sd1 = 3.30,
        sd2 = 2.16, n1 = 4, n2 = 4, a = .05)

# Equivalent shorthand:
d_ind_t(17.75, 23, 3.30, 2.16, 4, 4, .05)

# Using raw data from the dataset:
d_ind_t(mean(indt_data$correctq[indt_data$group == 1]),
        mean(indt_data$correctq[indt_data$group == 2]),
        sd(indt_data$correctq[indt_data$group == 1]),
        sd(indt_data$correctq[indt_data$group == 2]),
        length(indt_data$correctq[indt_data$group == 1]),
        length(indt_data$correctq[indt_data$group == 2]),
        .05)
```

`d_ind_t_t` *Cohen's d from t for Independent Samples (Pooled SD)*

Description

Note on function and output names: This effect size is now implemented with the snake_case function name ‘d_ind_t_t()‘ to follow modern R style guidelines. The original dotted version ‘d.ind.t.t()‘ is still available as a wrapper for backward compatibility, and both functions return the same list. The returned object includes both the original element names (e.g., ‘d‘, ‘dlow‘, ‘dhigh‘, ‘n1‘, ‘n2‘, ‘df‘, ‘t‘, ‘p‘, ‘estimate‘, ‘statistic‘) and newer snake_case aliases (e.g., ‘d_lower_limit‘, ‘d_upper_limit‘, ‘sample_size_1‘, ‘sample_size_2‘, ‘degrees_freedom‘, ‘t‘, ‘p_value‘). New code should prefer ‘d_ind_t_t()‘ and the snake_case output names, but existing code using the older names will continue to work.

Usage

```
d_ind_t_t(t_value, t = NULL, n1, n2, a = 0.05)

d.ind.t.t(t, n1, n2, a = 0.05)
```

Arguments

`t_value`	t-statistic from an independent-samples t-test.
`t`	t-statistic from an independent-samples t-test. Used for backwards compatibility.
`n1`	Sample size for group one.
`n2`	Sample size for group two.
`a`	Significance level (alpha) for the confidence interval. Must be in (0, 1).

Details

Compute Cohen's d_s from an independent-samples t-statistic and provide a noncentral-t confidence interval, assuming equal variances (pooled SD).

For between-subjects designs with pooled SD, d_s can be obtained directly from the t-statistic:

$$d_s = \frac{2t}{\sqrt{n_1 + n_2 - 2}},$$

where n_1 and n_2 are the group sample sizes (df = $n_1 + n_2 - 2$). The $(1 - \alpha)$ confidence interval for d_s is derived from the noncentral t distribution for the observed t and df.

See the online example for additional context: Learn more on our example page.

Value

A list with the following elements:

d Cohen's d_s.

dlow Lower limit of the $(1-\alpha)$ confidence interval for d_s.

dhigh Upper limit of the $(1-\alpha)$ confidence interval for d_s.

n1, n2 Group sample sizes.

df Degrees of freedom ($n_1 + n_2 - 2$).

t t-statistic.

p p-value.

estimate APA-style formatted string for reporting d_s and its CI.

statistic APA-style formatted string for reporting the t-statistic and p-value.

Examples

```
# The following example is derived from the "indt_data" dataset in MOTE.
    hyp <- t.test(correctq ~ group, data = indt_data)

# Direct entry of the t-statistic and sample sizes:
    d_ind_t_t(t = -2.6599, n1 = 4, n2 = 4, a = .05)

# Using the t-statistic from the model object:
    d_ind_t_t(hyp$statistic, length(indt_data$group[indt_data$group == 1]),
              length(indt_data$group[indt_data$group == 2]), .05)
```

`d_prop` *Cohen's d (SMD) for Independent Proportions (Binary Outcomes)*

Description

This function computes a standardized mean difference effect size for two independent proportions by treating each as the mean of a Bernoulli (0/1) variable and computing a standardized mean difference (SMD) directly using the pooled Bernoulli standard deviation. This follows the same logic as Cohen's d for continuous variables, but applied to binary outcomes:

Usage

```
d_prop(p1, p2, n1, n2, a = 0.05)

d.prop(p1, p2, n1, n2, a = 0.05)
```

Arguments

`p1`	Proportion for group one (between 0 and 1).
`p2`	Proportion for group two (between 0 and 1).
`n1`	Sample size for group one.
`n2`	Sample size for group two.
`a`	Significance level used for confidence intervals. Defaults to 0.05.

Details

$$d = \frac{p_1 - p_2}{s_{\text{pooled}}}$$

where

$$s_{\text{pooled}} = \sqrt{\frac{(n_1 - 1)p_1(1 - p_1) + (n_2 - 1)p_2(1 - p_2)}{n_1 + n_2 - 2}}$$

This replaces the original z-based formulation used in older versions of MOTE. The SMD effect size is directly comparable to all other d-type effect sizes in the package.

Value

A list with the same structure as [d_ind_t()], containing the standardized mean difference and its confidence interval, along with auxiliary statistics. The list is augmented with explicit entries ‘p1‘, ‘p2‘, ‘p1_value‘, and ‘p2_value‘ to emphasize that the original inputs were proportions.

Examples

```
d_prop(p1 = .25, p2 = .35, n1 = 100, n2 = 100, a = .05)
```

`d_single_t` *Cohen’s d for One-Sample t from Summary Stats*

Description

Note on function and output names: This effect size is now implemented with the snake_case function name ‘d_single_t()‘ to follow modern R style guidelines. The original dotted version ‘d.single.t()‘ is still available as a wrapper for backward compatibility, and both functions return the same list. The returned object includes both the original element names (e.g., ‘d‘, ‘dlow‘, ‘dhigh‘, ‘m‘, ‘sd‘, ‘se‘, ‘Mlow‘, ‘Mhigh‘, ‘u‘, ‘n‘, ‘df‘, ‘t‘, ‘p‘, ‘estimate‘, ‘statistic‘) and newer snake_case aliases (e.g., ‘d_lower_limit‘, ‘d_upper_limit‘, ‘mean_value‘, ‘sd_value‘, ‘se_value‘, ‘mean_lower_limit‘, ‘mean_upper_limit‘, ‘population_mean‘, ‘sample_size‘, ‘degrees_freedom‘, ‘t_value‘, ‘p_value‘). New code should prefer ‘d_single_t()‘ and the snake_case output names, but existing code using the older names will continue to work.

Usage

```
d_single_t(m, u, sd, n, a = 0.05)

d.single.t(m, u, sd, n, a = 0.05)
```

Arguments

`m`	Sample mean.
`u`	Population (reference) mean μ.
`sd`	Sample standard deviation s.
`n`	Sample size n.
`a`	Significance level (alpha) for the confidence interval. Must be in (0, 1).

Details

Compute Cohen's d and a noncentral-t confidence interval for a one-sample (single) t-test using summary statistics.

The effect size is defined as the standardized mean difference between the sample mean and the population/reference mean:

$$d = \frac{m - \mu}{s}.$$

The corresponding t-statistic is:

$$t = \frac{m - \mu}{s/\sqrt{n}}.$$

See the online example for additional context: Learn more on our example page.

Value

A list with the following elements:

d Cohen's d.

dlow Lower limit of the $(1 - \alpha)$ confidence interval for d.

dhigh Upper limit of the $(1 - \alpha)$ confidence interval for d.

m Sample mean.

sd Sample standard deviation.

se Standard error of the mean.

Mlow, Mhigh Confidence interval bounds for the mean.

u Population (reference) mean.

n Sample size.

df Degrees of freedom $(n - 1)$.

t t-statistic.

p p-value.

estimate APA-style formatted string for reporting d and its CI.

statistic APA-style formatted string for reporting the t-statistic and p-value.

Examples

```
# Example derived from the "singt_data" dataset included in MOTE.

# A school claims their gifted/honors program outperforms the national
# average (1080). Their students' SAT scores (sample) have mean 1370 and
# SD 112.7.

    gift <- t.test(singt_data$SATscore, mu = 1080, alternative = "two.sided")

# Direct entry of summary statistics:
    d_single_t(m = 1370, u = 1080, sd = 112.7, n = 14, a = .05)

# Equivalent shorthand:
    d_single_t(1370, 1080, 112.7, 14, .05)

# Using values from the t-test object and dataset:
    d_single_t(gift$estimate, gift$null.value,
               sd(singt_data$SATscore), length(singt_data$SATscore), .05)
```

`d_single_t_t` *Cohen's d from t for One-Sample t-Test*

Description

Compute Cohen's d and a noncentral-t confidence interval for a one-sample (single) t-test using the observed t-statistic.

Usage

```
d_single_t_t(t, n, a = 0.05)

d.single.t.t(t, n, a = 0.05)
```

Arguments

`t`	t-test value.
`n`	Sample size.
`a`	Significance level (alpha) for the confidence interval. Must be in (0, 1).

Details

The effect size is calculated as:

$$d = \frac{t}{\sqrt{n}},$$

where t is the one-sample t-statistic and n is the sample size.

The corresponding $(1 - \alpha)$ confidence interval for d is derived from the noncentral t distribution.

See the online example for additional context: Learn more on our example page.

Value

A list with the following elements:

d Cohen's d.

dlow Lower limit of the $(1-\alpha)$ confidence interval for d.

dhigh Upper limit of the $(1-\alpha)$ confidence interval for d.

n Sample size.

df Degrees of freedom ($n-1$).

t t-statistic.

p p-value.

estimate APA-style formatted string for reporting d and its CI.

statistic APA-style formatted string for reporting the t-statistic and p-value.

Examples

```
# A school has a gifted/honors program that they claim is
# significantly better than others in the country. The gifted/honors
# students in this school scored an average of 1370 on the SAT,
# with a standard deviation of 112.7, while the national average
# for gifted programs is a SAT score of 1080.

    gift <- t.test(singt_data$SATscore, mu = 1080, alternative = "two.sided")

# Direct entry of t-statistic and sample size:
    d_single_t_t(9.968, 15, .05)

# Equivalent shorthand:
    d_single_t_t(9.968, 15, .05)

# Using values from a t-test object and dataset:
    d_single_t_t(gift$statistic, length(singt_data$SATscore), .05)
```

`d_to_r` *r and Coefficient of Determination (R2) from d*

Description

Note on function and output names: This effect size translation is now implemented with the snake_case function name ‘d_to_r()‘ to follow modern R style guidelines. The original dotted version ‘d.to.r()‘ is still available as a wrapper for backward compatibility, and both functions return the same list. The returned object includes both the original element names (e.g., ‘r‘, ‘rlow‘, ‘rhigh‘, ‘R2‘, ‘R2low‘, ‘R2high‘, ‘se‘, ‘n‘, ‘dfm‘, ‘dfe‘, ‘t‘, ‘F‘, ‘p‘, ‘estimate‘, ‘estimateR2‘, ‘statistic‘) and newer snake_case aliases (e.g., ‘r_lower_limit‘, ‘r_upper_limit‘, ‘r2_value‘, ‘r2_lower_limit‘, ‘r2_upper_limit‘, ‘se_value‘, ‘sample_size‘, ‘degrees_freedom_model‘, ‘degrees_freedom_error‘, ‘t_value‘, ‘f_value‘, ‘p_value‘, ‘estimate_r‘, ‘estimate_r2‘). New code should prefer ‘d_to_r()‘ and the snake_case output names, but existing code using the older names will continue to work.

Usage

```
d_to_r(d, n1, n2, a = 0.05)

d.to.r(d, n1, n2, a = 0.05)
```

Arguments

`d`	Effect size statistic.
`n1`	Sample size for group one.
`n2`	Sample size for group two.
`a`	Significance level.

Details

Calculates r from d and then translates r to r2 to calculate the non-central confidence interval for r2 using the F distribution.

The correlation coefficient (r) is calculated by dividing Cohen's d by the square root of the total sample size squared, divided by the product of the sample sizes of group one and group two.

$$r = \frac{d}{\sqrt{d^2 + \frac{(n_1+n_2)^2}{n_1 n_2}}}$$

Learn more on our example page.

Value

Provides the effect size (correlation coefficient) with associated confidence intervals, the t-statistic, F-statistic, and other estimates appropriate for d to r translation. Note this CI is not based on the traditional r-to-z transformation but rather non-central F using the ci.R function from MBESS.

r Correlation coefficient.

rlow Lower level confidence interval for r.

rhigh Upper level confidence interval for r.

R2 Coefficient of determination.

R2low Lower level confidence interval of R2.

R2high Upper level confidence interval of R2.

se Standard error.

n Sample size.

dfm Degrees of freedom of mean.

dfe Degrees of freedom error.

t t-statistic.

F F-statistic.

p p-value.

estimate The r statistic and confidence interval in APA style for markdown printing.

estimateR2 The R^2 statistic and confidence interval in APA style for markdown printing.

statistic The t-statistic in APA style for markdown printing.

Examples

```
# The following example is derived from the "indt_data"
# dataset, included in the MOTE library.

# A forensic psychologist conducted a study to examine whether
# being hypnotized during recall affects how well a witness
# can remember facts about an event. Eight participants
# watched a short film of a mock robbery, after which
# each participant was questioned about what he or she had
# seen. The four participants in the experimental group
# were questioned while they were hypnotized. The four
# participants in the control group received the same
# questioning without hypnosis.

# Contrary to the hypothesized result, the group that underwent
# hypnosis were significantly less accurate while reporting
# facts than the control group with a large effect size, t(6) = -2.66,
# p = .038, d_s = -1.88.

     d_to_r(d = -1.88, n1 = 4, n2 = 4, a = .05)
```

`d_z_mean` *Cohen's d for Z-test from Population Mean and SD*

Description

Computes Cohen's d for a Z-test using the sample mean, population mean, and population standard deviation. The function also provides a normal-theory confidence interval for d, and returns relevant statistics including the z-statistic and its p-value.

Usage

```
d_z_mean(mu, m1, sig, sd1, n, a = 0.05)

d.z.mean(mu, m1, sig, sd1, n, a = 0.05)
```

Arguments

`mu`	The population mean.
`m1`	The sample study mean.
`sig`	The population standard deviation.
`sd1`	The standard deviation from the study.
`n`	The sample size.
`a`	The significance level.

Details

The effect size is computed as:

$$d = \frac{m_1 - \mu}{\sigma}$$

where m_1 is the sample mean, μ is the population mean, and σ is the population standard deviation.

The z-statistic is:

$$z = \frac{m_1 - \mu}{\sigma/\sqrt{n}}$$

where n is the sample size.

Learn more on our example page.

Value

A list with the following components:

d	Effect size (Cohen's d).
dlow	Lower level confidence interval d value.
dhigh	Upper level confidence interval d value.
M1	Mean of sample.
sd1	Standard deviation of sample.
se1	Standard error of sample.
M1low	Lower level confidence interval of the mean.
M1high	Upper level confidence interval of the mean.
Mu	Population mean.
Sigma	Standard deviation of population.
se2	Standard error of population.
z	Z-statistic.
p	P-value.
n	Sample size.
estimate	The d statistic and confidence interval in APA style for markdown printing.
statistic	The Z-statistic in APA style for markdown printing.

Examples

```
# The average quiz test taking time for a 10 item test is 22.5
# minutes, with a standard deviation of 10 minutes. My class of
# 25 students took 19 minutes on the test with a standard deviation of 5.

d_z_mean(mu = 22.5, m1 = 19, sig = 10, sd1 = 5, n = 25, a = .05)
```

`d_z_z` *Cohen's d from z-statistic for Z-test*

Description

Compute Cohen's d from a z-statistic for a Z-test.

Usage

```
d_z_z(z, n, a = 0.05, sig = NA)

d.z.z(z, sig = NA, n, a = 0.05)
```

Arguments

`z`	z-statistic from a Z-test.
`n`	Sample size.
`a`	Significance level (alpha) for the confidence interval. Must be in (0, 1).
`sig`	Population standard deviation (σ). This value is retained for descriptive purposes but is not required to compute the confidence interval for d.

Details

The effect size is computed as:

$$d = \frac{z}{\sqrt{n}},$$

where n is the sample size.

The confidence interval bounds assume a normal-theory standard error for d of $1/\sqrt{n}$ (given that $d = z/\sqrt{n}$). Thus:

$$d_{\text{low}} = d - z_{\alpha/2} \cdot 1/\sqrt{n}$$

$$d_{\text{high}} = d + z_{\alpha/2} \cdot 1/\sqrt{n}$$

where $z_{\alpha/2}$ is the critical value from the standard normal distribution.

The population standard deviation (σ) is retained for descriptive purposes but is not required for computing confidence intervals for d.

See the online example for additional context: Learn more on our example page.

Value

A list with the following elements:

d Effect size.

dlow Lower confidence interval bound for d.

dhigh Upper confidence interval bound for d.

sigma Population standard deviation (σ).

z z-statistic.

p Two-tailed p-value.

n Sample size.

estimate The d statistic and confidence interval in APA style for markdown printing.

statistic The Z-statistic in APA style for markdown printing.

Examples

```
# A recent study suggested that students (N = 100) learning
# statistics improved their test scores with the use of
# visual aids (Z = 2.5). The population standard deviation is 4.

# You can type in the numbers directly as shown below,
# or refer to your dataset within the function.

    d_z_z(z = 2.5, sig = 4, n = 100, a = .05)

    d_z_z(z = 2.5, n = 100, a = .05)

    d.z.z(2.5, 4, 100, .05)
```

`epsilon_full_ss` *$\epsilon\hat{}2$ for ANOVA from F and Sum of Squares*

Description

Note on function and output names: This effect size is now implemented with the snake_case function name ‘epsilon_full_ss()‘ to follow modern R style guidelines. The original dotted version ‘epsilon.full.SS()‘ is still available as a wrapper for backward compatibility, and both functions return the same list. The returned object includes both the original element names (e.g., ‘epsilon‘, ‘epsilonlow‘, ‘epsilonhigh‘, ‘dfm‘, ‘dfe‘, ‘F‘, ‘p‘, ‘estimate‘, ‘statistic‘) and newer snake_case aliases (e.g., ‘epsilon_value‘, ‘epsilon_lower_limit‘, ‘epsilon_upper_limit‘, ‘df_model‘, ‘df_error‘, ‘f_value‘, ‘p_value‘). New code should prefer ‘epsilon_full_ss()‘ and the snake_case output names, but existing code using the older names will continue to work.

Usage

```
epsilon_full_ss(dfm, dfe, msm, mse, sst, a = 0.05)

epsilon.full.SS(dfm, dfe, msm, mse, sst, a = 0.05)
```

Arguments

`dfm`	degrees of freedom for the model/IV/between
`dfe`	degrees of freedom for the error/residual/within
`msm`	mean square for the model/IV/between

`mse`	mean square for the error/residual/within
`sst`	sum of squares total
`a`	significance level

Details

This function displays ϵ^2 from ANOVA analyses and its non-central confidence interval based on the F distribution. This formula works for one way and multi way designs with careful focus on the sum of squares total calculation.

To calculate ϵ^2, first, the mean square for the error is is multiplied by the degrees of freedom for the model. The product is divided by the sum of squares total.

$$\epsilon^2 = \frac{df_m(ms_m - ms_e)}{SS_T}$$

Learn more on our example page.

Value

Provides the effect size (ϵ^2) with associated confidence intervals from the F-statistic.

epsilon effect size

epsilonlow lower level confidence interval of epsilon

epsilonhigh upper level confidence interval of epsilon

dfm degrees of freedom for the model/IV/between

dfe degrees of freedom for the error/residual/within

F F-statistic

p p-value

estimate the ϵ^2 statistic and confidence interval in APA style for markdown printing

statistic the F-statistic in APA style for markdown printing

Examples

```
# The following example is derived from the "bn1_data"
# dataset, included in the MOTE library.

# A health psychologist recorded the number of close inter-personal
# attachments of 45-year-olds who were in excellent, fair, or poor
# health. People in the Excellent Health group had 4, 3, 2, and 3
# close attachments; people in the Fair Health group had 3, 5,
# and 8 close attachments; and people in the Poor Health group
# had 3, 1, 0, and 2 close attachments.

anova_model <- lm(formula = friends ~ group, data = bn1_data)
summary.aov(anova_model)

epsilon_full_ss(dfm = 2, dfe = 8, msm = 12.621,
                mse = 2.458, sst = (25.24 + 19.67), a = .05)
```

```
# Backwards-compatible dotted name (deprecated)
epsilon.full.SS(dfm = 2, dfe = 8, msm = 12.621,
                mse = 2.458, sst = (25.24 + 19.67), a = .05)
```

---

eta_f *η^2 and Coefficient of Determination (R^2) for ANOVA from F*

Description

Note on function and output names: This effect size is now implemented with the snake_case function name ‘eta_f()‘ to follow modern R style guidelines. The original dotted version ‘eta.F()‘ is still available as a wrapper for backward compatibility, and both functions return the same list. The returned object includes both the original element names (e.g., ‘eta‘, ‘etalow‘, ‘etahigh‘, ‘dfm‘, ‘dfe‘, ‘F‘, ‘p‘, ‘estimate‘, ‘statistic‘) and newer snake_case aliases (e.g., ‘eta_value‘, ‘eta_lower_limit‘, ‘eta_upper_limit‘, ‘df_model‘, ‘df_error‘, ‘f_value‘, ‘p_value‘). New code should prefer ‘eta_f()‘ and the snake_case output names, but existing code using the older names will continue to work.

Usage

```
eta_f(dfm, dfe, f_value, a = 0.05, Fvalue)

eta.F(dfm, dfe, Fvalue, a = 0.05)
```

Arguments

`dfm`	degrees of freedom for the model/IV/between
`dfe`	degrees of freedom for the error/residual/within
`f_value`	F statistic
`a`	significance level
`Fvalue`	F statistic only for older function

Details

This function displays η^2 from ANOVA analyses and their non-central confidence interval based on the F distribution. These values are calculated directly from F statistics and can be used for between subjects and repeated measures designs. Remember if you have two or more IVs, these values are partial eta squared.

Eta is calculated by multiplying the degrees of freedom of the model by the F-statistic. This is divided by the product of degrees of freedom of the model, the F-statistic, and the degrees of freedom for the error or residual.

$$\eta^2 = \frac{df_m \cdot F}{df_m \cdot F + df_e}$$

Learn more on our example page.

Value

Provides the effect size (η^2) with associated confidence intervals and relevant statistics.

eta η^2 effect size

etalow lower level confidence interval of η^2

etahigh upper level confidence interval of η^2

dfm degrees of freedom for the model/IV/between

dfe degrees of freedom for the error/residual/within

F F-statistic

p p-value

estimate the η^2 statistic and confidence interval in APA style for markdown printing

statistic the F-statistic in APA style for markdown printing

Examples

```
# The following example is derived from the "bn1_data"
# dataset, included in the MOTE library.

# A health psychologist recorded the number of close inter-personal
# attachments of 45-year-olds who were in excellent, fair, or poor
# health. People in the Excellent Health group had 4, 3, 2, and 3
# close attachments; people in the Fair Health group had 3, 5,
# and 8 close attachments; and people in the Poor Health group
# had 3, 1, 0, and 2 close attachments.

anova_model <- lm(formula = friends ~ group, data = bn1_data)
summary.aov(anova_model)

eta_f(dfm = 2, dfe = 8,
      Fvalue = 5.134, a = .05)

# Backwards-compatible dotted name (deprecated)
eta.F(dfm = 2, dfe = 8,
      Fvalue = 5.134, a = .05)
```

`eta_full_ss` *η^2 for ANOVA from F and Sum of Squares*

Description

This function displays η^2 from ANOVA analyses and its non-central confidence interval based on the F distribution. This formula works for one way and multi way designs with careful focus on the sum of squares total.

Usage

```
eta_full_ss(dfm, dfe, ssm, sst, f_value, a = 0.05, Fvalue)

eta.full.SS(dfm, dfe, ssm, sst, Fvalue, a = 0.05)
```

Arguments

`dfm`	degrees of freedom for the model/IV/between
`dfe`	degrees of freedom for the error/residual/within
`ssm`	sum of squares for the model/IV/between
`sst`	sum of squares total
`f_value`	F statistic
`a`	significance level
`Fvalue`	Backward-compatible argument for the F statistic (deprecated; use ‘f_value‘ instead). If supplied, it overrides ‘f_value‘. Included for users of the legacy ‘eta.full.SS()‘.

Details

Eta squared is calculated by dividing the sum of squares for the model by the sum of squares total.

$$\eta^2 = \frac{SS_M}{SS_T}$$

Note on function and output names: This effect size is now implemented with the snake_case function name ‘eta_full_ss()‘ to follow modern R style guidelines. The original dotted version ‘eta.full.SS()‘ is still available as a wrapper for backward compatibility, and both functions return the same list. The returned object includes both the original element names (e.g., ‘eta‘, ‘etalow‘, ‘etahigh‘, ‘dfm‘, ‘dfe‘, ‘F‘, ‘p‘, ‘estimate‘, ‘statistic‘) and newer snake_case aliases (e.g., ‘eta_value‘, ‘eta_lower_limit‘, ‘eta_upper_limit‘, ‘df_model‘, ‘df_error‘, ‘f_value‘, ‘p_value‘). New code should prefer ‘eta_full_ss()‘ and the snake_case output names, but existing code using the older names will continue to work.

Learn more on our example page.

Value

Provides the effect size (η^2) with associated confidence intervals and relevant statistics.

eta η^2 effect size

etalow lower level confidence interval of η^2

etahigh upper level confidence interval of η^2

dfm degrees of freedom for the model/IV/between

dfe degrees of freedom for the error/residual/within

F F-statistic

p p-value

estimate the η^2 statistic and confidence interval in APA style for markdown printing

statistic the F-statistic in APA style for markdown printing

Examples

```
# The following example is derived from the "bn1_data"
# dataset, included in the MOTE library.

# A health psychologist recorded the number of close inter-personal
# attachments of 45-year-olds who were in excellent, fair, or poor
# health. People in the Excellent Health group had 4, 3, 2, and 3
# close attachments; people in the Fair Health group had 3, 5,
# and 8 close attachments; and people in the Poor Health group
# had 3, 1, 0, and 2 close attachments.

anova_model <- lm(formula = friends ~ group, data = bn1_data)
summary.aov(anova_model)

eta_full_ss(dfm = 2, dfe = 8, ssm = 25.24,
            sst = (25.24 + 19.67), f_value = 5.134, a = .05)

# Backwards-compatible dotted name (deprecated)
eta.full.SS(dfm = 2, dfe = 8, ssm = 25.24,
            sst = (25.24 + 19.67), Fvalue = 5.134, a = .05)
```

`eta_partial_ss` *η^2_p for ANOVA from F and Sum of Squares*

Description

This function displays η_p^2 from ANOVA analyses and its non-central confidence interval based on the F distribution. This formula works for one way and multi way designs.

Usage

```
eta_partial_ss(dfm, dfe, ssm, sse, f_value, a = 0.05, Fvalue)

eta.partial.SS(dfm, dfe, ssm, sse, Fvalue, a = 0.05)
```

Arguments

`dfm`	degrees of freedom for the model/IV/between
`dfe`	degrees of freedom for the error/residual/within
`ssm`	sum of squares for the model/IV/between
`sse`	sum of squares for the error/residual/within
`f_value`	F statistic
`a`	significance level
`Fvalue`	Backward-compatible argument for the F statistic (deprecated; use ‘f_value‘ instead). If supplied, it overrides ‘f_value‘. Included for users of the legacy ‘eta.partial.SS()‘.

Details

η_p^2 is calculated by dividing the sum of squares of the model by the sum of the sum of squares of the model and sum of squares of the error.

$$\eta_p^2 = \frac{SS_M}{SS_M + SS_E}$$

Learn more on our example page.

Note on function and output names: This effect size is now implemented with the snake_case function name ‘eta_partial_ss()‘ to follow modern R style guidelines. The original dotted version ‘eta.partial.SS()‘ is still available as a wrapper for backward compatibility, and both functions return the same list. The returned object includes both the original element names (e.g., ‘eta‘, ‘etalow‘, ‘etahigh‘, ‘dfm‘, ‘dfe‘, ‘F‘, ‘p‘, ‘estimate‘, ‘statistic‘) and newer snake_case aliases (e.g., ‘eta_value‘, ‘eta_lower_limit‘, ‘eta_upper_limit‘, ‘df_model‘, ‘df_error‘, ‘f_value‘, ‘p_value‘). New code should prefer ‘eta_partial_ss()‘ and the snake_case output names, but existing code using the older names will continue to work.

Value

Provides the effect size (η_p^2) with associated confidence intervals and relevant statistics.

eta η_p^2 effect size

etalow lower level confidence interval of η_p^2

etahigh upper level confidence interval of η_p^2

dfm degrees of freedom for the model/IV/between

dfe degrees of freedom for the error/residual/within

F F-statistic

p p-value

estimate the η_p^2 statistic and confidence interval in APA style for markdown printing

statistic the F-statistic in APA style for markdown printing

Examples

```
# The following example is derived from the "bn2_data"
# dataset, included in the MOTE library.

# Is there a difference in athletic spending budget for different sports?
# Does that spending interact with the change in coaching staff?
# This data includes (fake) athletic budgets for baseball, basketball,
# football, soccer, and volleyball teams with new and old coaches
# to determine if there are differences in
# spending across coaches and sports.

# Example using reported ANOVA table values directly
eta_partial_ss(dfm = 4, dfe = 990,
               ssm = 338057.9, sse = 32833499,
               f_value = 2.548, a = .05)
```

```
# Example computing Type III SS with code (requires the "car" package)
if (requireNamespace("car", quietly = TRUE)) {

  # Fit the model using stats::lm
  mod <- stats::lm(money ~ coach * type, data = bn2_data)

  # Type III table for the effects
  aov_type3 <- car::Anova(mod, type = 3)

  # Extract DF, SS, and F for the interaction (coach:type)
  dfm_int <- aov_type3["coach:type", "Df"]
  ssm_int <- aov_type3["coach:type", "Sum Sq"]
  F_int   <- aov_type3["coach:type", "F value"]

  # Residual DF and SS from the standard ANOVA table
  aov_type1 <- stats::anova(mod)
  dfe <- aov_type1["Residuals", "Df"]
  sse <- aov_type1["Residuals", "Sum Sq"]

  # Calculate partial eta-squared for the interaction using Type III SS
  eta_partial_ss(dfm = dfm_int, dfe = dfe,
                 ssm = ssm_int, sse = sse,
                 f_value = F_int, a = .05)
#'
# Backwards-compatible dotted name (deprecated)
eta.partial.SS(dfm = 4, dfe = 990,
               ssm = 338057.9, sse = 32833499,
               Fvalue = 2.548, a = .05)
}
```

`ges_partial_ss_mix` *η^2_G (Partial Generalized Eta-Squared) for Mixed Design ANOVA from F*

Description

This function displays partial generalized eta-squared (η^2_G) from ANOVA analyses and its non-central confidence interval based on the F distribution. This formula works for mixed designs.

Usage

```
ges_partial_ss_mix(dfm, dfe, ssm, sss, sse, f_value, a = 0.05, Fvalue)

ges.partial.SS.mix(dfm, dfe, ssm, sss, sse, Fvalue, a = 0.05)
```

Arguments

`dfm`	degrees of freedom for the model/IV/between

`dfe`	degrees of freedom for the error/residual/within
`ssm`	sum of squares for the model/IV/between
`sss`	sum of squares subject variance
`sse`	sum of squares for the error/residual/within
`f_value`	F statistic
`a`	significance level
`Fvalue`	Backward-compatible argument for the F statistic (deprecated; use ‘f_value‘ instead). If supplied, it overrides ‘f_value‘. Included for users of the legacy ‘ges.partial.SS.mix()‘ API.

Details

To calculate partial generalized eta squared, first, the sum of squares of the model, sum of squares of the subject variance, sum of squares for the subject variance, and the sum of squares for the error/residual/within are added together.

$$\eta^2_G = \frac{SS_M}{SS_M + SS_S + SS_E}$$

Learn more on our example page.

Note on function and output names: This effect size is now implemented with the snake_case function name ‘ges_partial_ss_mix()‘ to follow modern R style guidelines. The original dotted version ‘ges.partial.SS.mix()‘ is still available as a wrapper for backward compatibility, and both functions return the same list. The returned object includes both the original element names (e.g., ‘ges‘, ‘geslow‘, ‘geshigh‘, ‘dfm‘, ‘dfe‘, ‘F‘, ‘p‘, ‘estimate‘, ‘statistic‘) and newer snake_case aliases (e.g., ‘ges_value‘, ‘ges_lower_limit‘, ‘ges_upper_limit‘, ‘df_model‘, ‘df_error‘, ‘f_value‘, ‘p_value‘). New code should prefer ‘ges_partial_ss_mix()‘ and the snake_case output names, but existing code using the older names will continue to work.

Value

ges η^2_G effect size

geslow lower level confidence interval for η^2_G

geshigh upper level confidence interval for η^2_G

dfm degrees of freedom for the model/IV/between

dfe degrees of freedom for the error/residual/within

F F-statistic

p p-value

estimate the η^2_G statistic and confidence interval in APA style for markdown printing

statistic the F-statistic in APA style for markdown printing

Examples

```
# The following example is derived from the
# "mix2_data" dataset, included in the MOTE library.

# Given previous research, we know that backward strength in free
# association tends to increase the ratings participants give when
# you ask them how many people out of 100 would say a word in
# response to a target word (like Family Feud). This result is
# tied to people's overestimation of how well they think they know
# something, which is bad for studying. So, we gave people instructions
# on how to ignore the BSG.  Did it help? Is there an interaction
# between BSG and instructions given?

# You would calculate one partial GES value for each F-statistic.
# Here's an example for the interaction using reported ANOVA values.
ges_partial_ss_mix(dfm = 1, dfe = 156,
                   ssm = 71.07608,
                   sss = 30936.498,
                   sse = 8657.094,
                   f_value = 1.280784, a = .05)

# Backwards-compatible dotted name (deprecated)
ges.partial.SS.mix(dfm = 1, dfe = 156,
                   ssm = 71.07608,
                   sss = 30936.498,
                   sse = 8657.094,
                   Fvalue = 1.280784, a = .05)
```

`ges_partial_ss_rm` *η^2_G (Partial Generalized Eta-Squared) for Repeated-Measures ANOVA from F*

Description

This function displays partial generalized eta-squared (η^2_G) from ANOVA analyses and its non-central confidence interval based on the F distribution. This formula works for multi-way repeated measures designs.

Usage

```
ges_partial_ss_rm(
  dfm,
  dfe,
  ssm,
  sss,
  sse1,
  sse2,
  sse3,
```

```
  f_value,
  a = 0.05,
  Fvalue
)

ges.partial.SS.rm(dfm, dfe, ssm, sss, sse1, sse2, sse3, Fvalue, a = 0.05)
```

Arguments

`dfm`	degrees of freedom for the model/IV/between
`dfe`	degrees of freedom for the error/residual/within
`ssm`	sum of squares for the model/IV/between
`sss`	sum of squares subject variance
`sse1`	sum of squares for the error/residual/within for the first IV
`sse2`	sum of squares for the error/residual/within for the second IV
`sse3`	sum of squares for the error/residual/within for the interaction
`f_value`	F statistic
`a`	significance level
`Fvalue`	Backward-compatible argument for the F statistic (deprecated; use ‘f_value‘ instead). If supplied, it overrides ‘f_value‘. Included for users of the legacy ‘ges.partial.SS.rm()‘ API.

Details

To calculate partial generalized eta squared, first, the sum of squares of the model, sum of squares of the subject variance, sum of squares for the first and second independent variables, and the sum of squares for the interaction are added together. The sum of squares of the model is divided by this value.

$$\eta^2_G = \frac{SS_M}{SS_M + SS_S + SS_{E1} + SS_{E2} + SS_{E3}}$$

Learn more on our example page.

Note on function and output names: This effect size is now implemented with the snake_case function name ‘ges_partial_ss_rm()‘ to follow modern R style guidelines. The original dotted version ‘ges.partial.SS.rm()‘ is still available as a wrapper for backward compatibility, and both functions return the same list. The returned object includes both the original element names (e.g., ‘ges‘, ‘geslow‘, ‘geshigh‘, ‘dfm‘, ‘dfe‘, ‘F‘, ‘p‘, ‘estimate‘, ‘statistic‘) and newer snake_case aliases (e.g., ‘ges_value‘, ‘ges_lower_limit‘, ‘ges_upper_limit‘, ‘df_model‘, ‘df_error‘, ‘f_value‘, ‘p_value‘). New code should prefer ‘ges_partial_ss_rm()‘ and the snake_case output names, but existing code using the older names will continue to work.

Value

ges η^2_G effect size

geslow lower level confidence interval for η^2_G

geshigh upper level confidence interval for η^2_G

dfm degrees of freedom for the model/IV/between

dfe degrees of freedom for the error/residual/within

F F-statistic

p p-value

estimate the η^2_G statistic and confidence interval in APA style for markdown printing

statistic the F-statistic in APA style for markdown printing

Examples

```
# The following example is derived from the "rm2_data" dataset, included
# in the MOTE library.

# In this experiment people were given word pairs to rate based on
# their "relatedness". How many people out of a 100 would put LOST-FOUND
# together? Participants were given pairs of words and asked to rate them
# on how often they thought 100 people would give the second word if shown
# the first word.  The strength of the word pairs was manipulated through
# the actual rating (forward strength: FSG) and the strength of the reverse
# rating (backward strength: BSG). Is there an interaction between FSG and
# BSG when participants are estimating the relation between word pairs?

# You would calculate one partial GES value for each F-statistic.
# Here's an example for the interaction with typing in numbers.
ges_partial_ss_rm(dfm = 1, dfe = 157,
                  ssm = 2442.948, sss = 76988.13,
                  sse1 = 5402.567, sse2 = 8318.75, sse3 = 6074.417,
                  f_value = 70.9927, a = .05)

# Backwards-compatible dotted name (deprecated)
ges.partial.SS.rm(dfm = 1, dfe = 157,
                  ssm = 2442.948, sss = 76988.13,
                  sse1 = 5402.567, sse2 = 8318.75, sse3 = 6074.417,
                  Fvalue = 70.9927, a = .05)
```

`g_ind_t` *d_g Corrected for Independent t*

Description

This function displays d_g (Hedges' g) corrected and the non-central confidence interval for independent t.

Usage

```
g_ind_t(m1, m2, sd1, sd2, n1, n2, a = 0.05)

g.ind.t(m1, m2, sd1, sd2, n1, n2, a = 0.05)
```

Arguments

`m1`	mean group one
`m2`	mean group two
`sd1`	standard deviation group one
`sd2`	standard deviation group two
`n1`	sample size group one
`n2`	sample size group two
`a`	significance level

Details

The small-sample correction factor is:

$$\text{correction} = 1 - \frac{3}{4(n_1 + n_2) - 9}$$

d_g is computed as the standardized mean difference multiplied by the correction:

$$d_g = \frac{m_1 - m_2}{s_{\text{pooled}}} \times \text{correction}$$

Learn more on our example page.

Value

d d_g corrected effect size

dlow lower level confidence interval for d_g

dhigh upper level confidence interval for d_g

M1 mean of group one

sd1 standard deviation of group one

se1 standard error of group one

M1low lower level confidence interval of mean one

M1high upper level confidence interval of mean one

M2 mean of group two

sd2 standard deviation of group two

se2 standard error of group two

M2low lower level confidence interval of mean two

M2high upper level confidence interval of mean two

spooled pooled standard deviation

sepooled pooled standard error

correction Hedges' small-sample correction factor

n1 sample size of group one

n2 sample size of group two

df degrees of freedom ($n_1 - 1 + n_2 - 1$)

t t-statistic

p p-value

estimate the d_g statistic and confidence interval in APA style for markdown printing

statistic the t-statistic in APA style for markdown printing

Examples

```
# The following example is derived from the "indt_data"
# dataset, included in the MOTE library.

# A forensic psychologist conducted a study to examine whether
# being hypnotized during recall affects how well a witness
# can remember facts about an event. Eight participants
# watched a short film of a mock robbery, after which
# each participant was questioned about what he or she had
# seen. The four participants in the experimental group
# were questioned while they were hypnotized. The four
# participants in the control group received the same
# questioning without hypnosis.

   t.test(correctq ~ group, data = indt_data)

# You can type in the numbers directly, or refer to the dataset,
# as shown below.

    g_ind_t(m1 = 17.75, m2 = 23, sd1 = 3.30,
           sd2 = 2.16, n1 = 4, n2 = 4, a = .05)

    g_ind_t(17.75, 23, 3.30, 2.16, 4, 4, .05)

    g_ind_t(mean(indt_data$correctq[indt_data$group == 1]),
            mean(indt_data$correctq[indt_data$group == 2]),
            sd(indt_data$correctq[indt_data$group == 1]),
            sd(indt_data$correctq[indt_data$group == 2]),
            length(indt_data$correctq[indt_data$group == 1]),
            length(indt_data$correctq[indt_data$group == 2]),
            .05)

# Contrary to the hypothesized result, the group that underwent hypnosis were
# significantly less accurate while reporting facts than the control group
# with a large effect size, t(6) = -2.66, p = .038, d_g = 1.64.
```

`h_prop` *Cohen's h for Independent Proportions*

Description

This function computes Cohen's h effect size for the difference between two independent proportions. Cohen's h is defined as a difference between arcsine-transformed proportions:

Usage

```
h_prop(p1, p2, n1, n2, a = 0.05)

h.prop(p1, p2, n1, n2, a = 0.05)
```

Arguments

`p1`	Proportion for group one (between 0 and 1).
`p2`	Proportion for group two (between 0 and 1).
`n1`	Sample size for group one.
`n2`	Sample size for group two.
`a`	Significance level used for confidence intervals. Defaults to 0.05.

Details

$$h = 2 \arcsin \sqrt{p_1} - 2 \arcsin \sqrt{p_2}$$

where p_1 and p_2 are proportions for groups 1 and 2, respectively.

Using a simple large-sample approximation (via the delta method), the standard error of h can be taken as:

$$\mathrm{SE}(h) \approx \sqrt{1/n_1 + 1/n_2}$$

,

which leads to a $(1 - \alpha)$ confidence interval for h:

$$h \pm z_{1-\alpha/2}\, \mathrm{SE}(h).$$

This effect size is commonly recommended for differences in proportions (Cohen, 1988) and is particularly useful for power analysis and meta-analysis when working directly with proportions.

Value

A list containing Cohen's h effect size and related statistics:

- 'h' – Cohen's h.
- 'hlow', 'hhigh' – lower and upper confidence interval limits.
- 'h_lower_limit', 'h_upper_limit' – snake_case aliases for the confidence limits.
- 'p1', 'p2' – input proportions for each group.
- 'n1', 'n2' – sample sizes for each group, with snake_case aliases 'sample_size_1', 'sample_size_2'.

- ‘z‘, ‘p‘ – z statistic and p value for the difference in proportions using a pooled-proportion standard error.
- ‘z_value‘, ‘p_value‘ – snake_case aliases for the z statistic and p value.
- ‘estimate‘ – APA-style formatted string for Cohen’s h and its confidence interval.
- ‘statistic‘ – APA-style formatted string for the z test of the difference in proportions.

Examples

```
h_prop(p1 = .25, p2 = .35, n1 = 100, n2 = 100, a = .05)
```

`indt_data` *Independent-Samples t-Test Example Data*

Description

Example data for an independent-samples t-test examining whether a hypnotism intervention affects recall accuracy after witnessing a crime. Designed for use with functions such as `d_ind_t`, `d_ind_t_t`, and `delta_ind_t`.

Usage

```
data(indt_data)
```

Format

A data frame with 2 variables:

correctq Numeric recall score/accuracy.

group Factor indicating condition with levels `"control"` and `"hypnotism"`.

`mix2_data` *Mixed Two-Way ANOVA Example Data*

Description

Example data for a mixed two-way ANOVA examining whether instructions to ignore backward strength in free association (BSG) reduce overestimation in response probabilities. Participants provided estimates of how many out of 100 people would say a given response to a target word ("Family Feud"-style), under low vs. high BSG conditions, after receiving either regular or debiasing instructions.

Usage

```
data(mix2_data)
```

Format

A data frame with 3 variables:

group Factor indicating instruction condition with levels `"Regular JAM Task"` and `"Debiasing JAM task"`.

bsglo Numeric. Estimated response percentage in the Low BSG condition.

bsghi Numeric. Estimated response percentage in the High BSG condition.

`odds_ratio` *Odds Ratio from 2x2 Table*

Description

This function displays odds ratios and their normal confidence intervals. This statistic is calculated as (level 1.1/level 1.2) / (level 2.1/level 2.2), which can be considered the odds of level 1.1 given level1 overall versus level2.1 given level2 overall.

Usage

```
odds_ratio(n11, n12, n21, n22, a = 0.05)

odds(n11, n12, n21, n22, a = 0.05)
```

Arguments

`n11`	sample size for level 1.1
`n12`	sample size for level 1.2
`n21`	sample size for level 2.1
`n22`	sample size for level 2.2
`a`	significance level

Details

Note on function and output names: This effect size is now implemented with the snake_case function name ‘odds_ratio()‘ to follow modern R style guidelines. The original name ‘odds()‘ is still available as a wrapper for backward compatibility, and both functions return the same list. The returned object includes both the original element names (e.g., ‘odds‘, ‘olow‘, ‘ohigh‘, ‘se‘) and newer snake_case aliases (e.g., ‘odds_value‘, ‘odds_lower_limit‘, ‘odds_upper_limit‘, ‘standard_error‘). New code should prefer ‘odds_ratio()‘ and the snake_case output names, but existing code using the older names will continue to work.

Value

odds	odds ratio statistic (legacy name; see also ‘odds_value‘)
olow	lower-level confidence interval of odds ratio (legacy name; see also ‘odds_lower_limit‘)
ohigh	upper-level confidence interval of odds ratio (legacy name; see also ‘odds_upper_limit‘)
se	standard error (legacy name; see also ‘standard_error‘)
odds_value	odds ratio statistic (snake_case alias of ‘odds‘)
odds_lower_limit	lower-level confidence interval of odds ratio (alias of ‘olow‘)
odds_upper_limit	upper-level confidence interval of odds ratio (alias of ‘ohigh‘)
standard_error	standard error (alias of ‘se‘)

Examples

```
# A health psychologist was interested in the rates of anxiety in
# first generation and regular college students. They polled campus
# and found the following data:

  # |              | First Generation | Regular |
  # |--------------|------------------|---------|
  # | Low Anxiety  | 10               | 50      |
  # | High Anxiety | 20               | 15      |

# What are the odds for the first generation students to have anxiety?

odds_ratio(n11 = 10, n12 = 50, n21 = 20, n22 = 15, a = .05)

# Backwards-compatible wrapper (deprecated name)
odds(n11 = 10, n12 = 50, n21 = 20, n22 = 15, a = .05)
```

`omega_f` *ωˆ2 for ANOVA from F*

Description

This function displays ω^2 from ANOVA analyses and its non-central confidence interval based on the F distribution. These values are calculated directly from F statistics and can be used for between subjects and repeated measures designs. Remember if you have two or more IVs, these values are partial omega squared.

Usage

```
omega_f(dfm, dfe, f_value, n, a = 0.05, Fvalue)

omega.F(dfm, dfe, Fvalue, n, a = 0.05)
```

Arguments

`dfm`	degrees of freedom for the model/IV/between
`dfe`	degrees of freedom for the error/residual/within
`f_value`	F statistic
`n`	full sample size
`a`	significance level
`Fvalue`	Backward-compatible argument for the F statistic (deprecated; use ‘f_value‘ instead). If supplied, it overrides ‘f_value‘. Included for users of the legacy ‘omega.F()‘.

Details

Omega squared or partial omega squared is calculated by subtracting one from the F-statistic and multiplying it by degrees of freedom of the model. This is divided by the same value after adding the number of valid responses. This value will be omega squared for one-way ANOVA designs, and will be partial omega squared for multi-way ANOVA designs (i.e. with more than one IV).

$$\omega^2 = \frac{df_m(F-1)}{df_m(F-1)+n}$$

Learn more on our example page.

Note on function and output names: This effect size is now implemented with the snake_case function name ‘omega_f()‘ to follow modern R style guidelines. The original dotted version ‘omega.F()‘ is still available as a wrapper for backward compatibility, and both functions return the same list. The returned object includes both the original element names (e.g., ‘omega‘, ‘omegalow‘, ‘omegahigh‘, ‘dfm‘, ‘dfe‘, ‘F‘, ‘p‘, ‘estimate‘, ‘statistic‘) and newer snake_case aliases (e.g., ‘omega_value‘, ‘omega_lower_limit‘, ‘omega_upper_limit‘, ‘df_model‘, ‘df_error‘, ‘f_value‘, ‘p_value‘). New code should prefer ‘omega_f()‘ and the snake_case output names, but existing code using the older names will continue to work.

Value

omega ω^2 effect size (legacy name; see also ‘omega_value‘)

omegalow lower-level confidence interval of ω^2 (legacy name; see also ‘omega_lower_limit‘)

omegahigh upper-level confidence interval of ω^2 (legacy name; see also ‘omega_upper_limit‘)

dfm degrees of freedom for the model/IV/between (legacy name; see also ‘df_model‘)

dfe degrees of freedom for the error/residual/within (legacy name; see also ‘df_error‘)

F F-statistic (legacy name; see also ‘f_value‘)

p p-value (legacy name; see also ‘p_value‘)

estimate the ω^2 statistic and confidence interval in APA style for markdown printing

statistic the F-statistic in APA style for markdown printing

omega_value ω^2 effect size (snake_case alias of ‘omega‘)

omega_lower_limit lower-level confidence interval of ω^2 (alias of ‘omegalow‘)

omega_upper_limit upper-level confidence interval of ω^2 (alias of ‘omegahigh‘)

df_model degrees of freedom for the model/IV/between (alias of ‘dfm‘)

df_error degrees of freedom for the error/residual/within (alias of ‘dfe‘)

f_value F-statistic (alias of ‘F‘)

p_value p-value (alias of ‘p‘)

Examples

```
# The following example is derived from
# the "bn1_data" dataset, included in the MOTE library.

# A health psychologist recorded the number of close inter-personal
# attachments of 45-year-olds who were in excellent, fair, or poor
# health. People in the Excellent Health group had 4, 3, 2, and 3
# close attachments; people in the Fair Health group had 3, 5,
# and 8 close attachments; and people in the Poor Health group
# had 3, 1, 0, and 2 close attachments.

anova_model <- lm(formula = friends ~ group, data = bn1_data)
summary.aov(anova_model)

omega_f(dfm = 2, dfe = 8,
        f_value = 5.134, n = 11, a = .05)

# Backwards-compatible dotted name (deprecated)
omega.F(dfm = 2, dfe = 8,
        Fvalue = 5.134, n = 11, a = .05)
```

---

omega_full_ss *omega^2 for One-Way and Multi-Way ANOVA from F*

Description

This function displays ω^2 from ANOVA analyses and its non-central confidence interval based on the F distribution. This formula works for one way and multi way designs with careful focus on which error term you are using for the calculation.

Usage

```
omega_full_ss(dfm, dfe, msm, mse, sst, a = 0.05)

omega.full.SS(dfm, dfe, msm, mse, sst, a = 0.05)
```

Arguments

`dfm`	degrees of freedom for the model/IV/between
`dfe`	degrees of freedom for the error/residual/within
`msm`	mean square for the model/IV/between
`mse`	mean square for the error/residual/within
`sst`	sum of squares total
`a`	significance level

Details

Omega squared is calculated by deducting the mean square of the error from the mean square of the model and multiplying by the degrees of freedom for the model. This is divided by the sum of the sum of squares total and the mean square of the error.

$$\omega^2 = \frac{df_m(ms_m - ms_e)}{SS_T + ms_e}$$

Learn more on our example page.

Note on function and output names: This effect size is now implemented with the snake_case function name ‘omega_full_ss()‘ to follow modern R style guidelines. The original dotted version ‘omega.full.SS()‘ is still available as a wrapper for backward compatibility, and both functions return the same list. The returned object includes both the original element names (e.g., ‘omega‘, ‘omegalow‘, ‘omegahigh‘, ‘dfm‘, ‘dfe‘, ‘F‘, ‘p‘, ‘estimate‘, ‘statistic‘) and newer snake_case aliases (e.g., ‘omega_value‘, ‘omega_lower_limit‘, ‘omega_upper_limit‘, ‘df_model‘, ‘df_error‘, ‘f_value‘, ‘p_value‘). New code should prefer ‘omega_full_ss()‘ and the snake_case output names, but existing code using the older names will continue to work.

Value

omega ω^2 effect size (legacy name; see also ‘omega_value‘)

omegalow lower-level confidence interval of ω^2 (legacy name; see also ‘omega_lower_limit‘)

omegahigh upper-level confidence interval of ω^2 (legacy name; see also ‘omega_upper_limit‘)

dfm degrees of freedom for the model/IV/between (legacy name; see also ‘df_model‘)

dfe degrees of freedom for the error/residual/within (legacy name; see also ‘df_error‘)

F F-statistic (legacy name; see also ‘f_value‘)

p p-value (legacy name; see also ‘p_value‘)

estimate the ω^2 statistic and confidence interval in APA style for markdown printing

statistic the F-statistic in APA style for markdown printing

omega_value ω^2 effect size (snake_case alias of ‘omega‘)

omega_lower_limit lower-level confidence interval of ω^2 (alias of ‘omegalow‘)

omega_upper_limit upper-level confidence interval of ω^2 (alias of ‘omegahigh‘)

df_model degrees of freedom for the model/IV/between (alias of ‘dfm‘)

df_error degrees of freedom for the error/residual/within (alias of ‘dfe‘)

f_value F-statistic (alias of ‘F‘)

p_value p-value (alias of ‘p‘)

Examples

```
# The following example is derived from the "bn1_data"
# dataset, included in the MOTE library.

# A health psychologist recorded the number of close inter-personal
# attachments of 45-year-olds who were in excellent, fair, or poor
# health. People in the Excellent Health group had 4, 3, 2, and 3
# close attachments; people in the Fair Health group had 3, 5,
# and 8 close attachments; and people in the Poor Health group
# had 3, 1, 0, and 2 close attachments.

anova_model <- lm(formula = friends ~ group, data = bn1_data)
summary.aov(anova_model)

omega_full_ss(dfm = 2, dfe = 8,
              msm = 12.621, mse = 2.548,
              sst = (25.54 + 19.67), a = .05)

# Backwards-compatible dotted name (deprecated)
omega.full.SS(dfm = 2, dfe = 8,
              msm = 12.621, mse = 2.548,
              sst = (25.54 + 19.67), a = .05)
```

`omega_g_ss_rm` *omega^2_G (Generalized Omega Squared) for Multi-Way and Mixed ANOVA from F*

Description

This function displays ω^2_G (generalized omega squared) from ANOVA analyses and its non-central confidence interval based on the F distribution. This formula is appropriate for multi-way repeated-measures designs and mixed-level designs.

Usage

```
omega_g_ss_rm(dfm, dfe, ssm, ssm2, sst, mss, j, f_value, a = 0.05, Fvalue)

omega.gen.SS.rm(dfm, dfe, ssm, ssm2, sst, mss, j, Fvalue, a = 0.05)
```

Arguments

`dfm`	degrees of freedom for the model/IV/between
`dfe`	degrees of freedom for the error/residual/within
`ssm`	sum of squares for the MAIN model/IV/between
`ssm2`	sum of squares for the OTHER model/IV/between
`sst`	sum of squares total across the whole ANOVA
`mss`	mean square for the subject variance

`j`	number of levels in the OTHER IV
`f_value`	F statistic from the output for your IV
`a`	significance level
`Fvalue`	Backward-compatible argument for the F statistic (deprecated; use ‘f_value‘ instead). This argument is only used by the wrapper function ‘omega.gen.SS.rm()‘, which forwards ‘Fvalue‘ to the ‘f_value‘ argument of ‘omega_g_ss_rm()‘.

Details

Omega squared is calculated by subtracting the product of the degrees of freedom of the model and the mean square of the subject variance from the sum of squares for the model.

This is divided by the value obtained after combining the sum of squares total, sum of squares for the other independent variable, and the mean square of the subject variance multiplied by the number of levels in the other model/IV/between.

$$\omega_G^2 = \frac{SS_M - (df_m \times MS_S)}{SS_T + SS_{M2} + j \times MS_S}$$

Learn more on our example page.

Note on function and output names: This effect size is now implemented with the snake_case function name ‘omega_g_ss_rm()‘ to follow modern R style guidelines. The original dotted version ‘omega.gen.SS.rm()‘ is still available as a wrapper for backward compatibility, and both functions return the same list. The returned object includes both the original element names (e.g., ‘omega‘, ‘omegalow‘, ‘omegahigh‘, ‘dfm‘, ‘dfe‘, ‘F‘, ‘p‘, ‘estimate‘, ‘statistic‘) and newer snake_case aliases (e.g., ‘omega_value‘, ‘omega_lower_limit‘, ‘omega_upper_limit‘, ‘df_model‘, ‘df_error‘, ‘f_value‘, ‘p_value‘). New code should prefer ‘omega_g_ss_rm()‘ and the snake_case output names, but existing code using the older names will continue to work.

Value

omega ω_G^2 effect size (legacy name; see also ‘omega_value‘)

omegalow lower-level confidence interval of ω_G^2 (legacy name; see also ‘omega_lower_limit‘)

omegahigh upper-level confidence interval of ω_G^2 (legacy name; see also ‘omega_upper_limit‘)

dfm degrees of freedom for the model/IV/between (legacy name; see also ‘df_model‘)

dfe degrees of freedom for the error/residual/within (legacy name; see also ‘df_error‘)

F F-statistic (legacy name; see also ‘f_value‘)

p p-value (legacy name; see also ‘p_value‘)

estimate the ω_G^2 statistic and confidence interval in APA style for markdown printing

statistic the F-statistic in APA style for markdown printing

omega_value ω_G^2 effect size (snake_case alias of ‘omega‘)

omega_lower_limit lower-level confidence interval of ω_G^2 (alias of ‘omegalow‘)

omega_upper_limit upper-level confidence interval of ω_G^2 (alias of ‘omegahigh‘)

df_model degrees of freedom for the model/IV/between (alias of ‘dfm‘)

df_error degrees of freedom for the error/residual/within (alias of ‘dfe‘)

f_value F-statistic (alias of ‘F‘)

p_value p-value (alias of ‘p‘)

Examples

```
# The following example is derived from the "mix2_data"
# dataset, included in the MOTE library.

# Given previous research, we know that backward strength in free
# association tends to increase the ratings participants give when
# you ask them how many people out of 100 would say a word in
# response to a target word (like Family Feud). This result is
# tied to people’s overestimation of how well they think they know
# something, which is bad for studying. So, we gave people instructions
# on how to ignore the BSG.  Did it help? Is there an interaction
# between BSG and instructions given?

# You would calculate one partial GOS value for each F-statistic.
# Here's an example for the main effect 1 with typing in numbers.
omega_g_ss_rm(dfm = 1, dfe = 156,
          ssm = 6842.46829,
          ssm2 = 14336.07886,
          sst = sum(c(30936.498, 6842.46829,
                      14336.07886, 8657.094, 71.07608)),
          mss = 30936.498 / 156,
          j = 2, f_value = 34.503746, a = .05)

# Backwards-compatible dotted name (deprecated)
omega.gen.SS.rm(dfm = 1, dfe = 156,
                ssm = 6842.46829,
                ssm2 = 14336.07886,
                sst = sum(c(30936.498, 6842.46829,
                            14336.07886, 8657.094, 71.07608)),
                mss = 30936.498 / 156,
                j = 2, Fvalue = 34.503746, a = .05)
```

`omega_partial_ss_bn` *omega^2_p (Partial Omega Squared) for Between-Subjects ANOVA from F*

Description

This function displays ω_p^2 from ANOVA analyses and its non-central confidence interval based on the F distribution. This formula is appropriate for multi-way between-subjects designs.

Usage

```
omega_partial_ss_bn(dfm, dfe, msm, mse, ssm, n, a = 0.05)

omega.partial.SS.bn(dfm, dfe, msm, mse, ssm, n, a = 0.05)
```

Arguments

`dfm`	degrees of freedom for the model/IV/between
`dfe`	degrees of freedom for the error/residual/within
`msm`	mean square for the model/IV/between
`mse`	mean square for the error/residual/within
`ssm`	sum of squares for the model/IV/between
`n`	total sample size
`a`	significance level

Details

Partial omega squared is calculated by subtracting the mean square for the error from the mean square of the model, which is multiplied by degrees of freedom of the model. This is divided by the product of the degrees of freedom for the model are deducted from the sample size, multiplied by the mean square of the error, plus the sum of squares for the model.

$$\omega_p^2 = \frac{df_m(MS_M - MS_E)}{SS_M + (n - df_m) \times MS_E}$$

Learn more on our example page.

Note on function and output names: This effect size is now implemented with the snake_case function name ‘omega_partial_ss_bn()‘ to follow modern R style guidelines. The original dotted version ‘omega.partial.SS.bn()‘ is still available as a wrapper for backward compatibility, and both functions return the same list. The returned object includes both the original element names (e.g., ‘omega‘, ‘omegalow‘, ‘omegahigh‘, ‘dfm‘, ‘dfe‘, ‘F‘, ‘p‘, ‘estimate‘, ‘statistic‘) and newer snake_case aliases (e.g., ‘omega_value‘, ‘omega_lower_limit‘, ‘omega_upper_limit‘, ‘df_model‘, ‘df_error‘, ‘f_value‘, ‘p_value‘). New code should prefer ‘omega_partial_ss_bn()‘ and the snake_case output names, but existing code using the older names will continue to work.

Value

omega ω_p^2 effect size (legacy name; see also ‘omega_value‘)

omegalow lower level confidence interval of ω_p^2 (legacy name; see also ‘omega_lower_limit‘)

omegahigh upper level confidence interval of ω_p^2 (legacy name; see also ‘omega_upper_limit‘)

dfm degrees of freedom for the model/IV/between (legacy name; see also ‘df_model‘)

dfe degrees of freedom for the error/residual/within (legacy name; see also ‘df_error‘)

F F-statistic (legacy name; see also ‘f_value‘)

p p-value (legacy name; see also ‘p_value‘)

estimate the ω_p^2 statistic and confidence interval in APA style for markdown printing

statistic the F-statistic in APA style for markdown printing

omega_value ω_p^2 effect size (snake_case alias of ‘omega‘)

omega_lower_limit lower level confidence interval of ω_p^2 (alias of ‘omegalow‘)

omega_upper_limit upper level confidence interval of ω_p^2 (alias of ‘omegahigh‘)

df_model degrees of freedom for the model/IV/between (alias of ‘dfm‘)

df_error degrees of freedom for the error/residual/within (alias of ‘dfe‘)

f_value F-statistic (alias of ‘F‘)

p_value p-value (alias of ‘p‘)

Examples

```
# The following example is derived from the "bn2_data"
# dataset, included in the MOTE library.

# Is there a difference in athletic spending budget for different sports?
# Does that spending interact with the change in coaching staff?
# This data includes (fake) athletic budgets for baseball,
# basketball, football, soccer, and volleyball teams
# with new and old coaches to determine if there are differences in
# spending across coaches and sports.

# You would calculate one omega value for each F-statistic.
# Here's an example for the interaction using reported ANOVA values.
omega_partial_ss_bn(dfm = 4, dfe = 990,
                    msm = 338057.9 / 4,
                    mse = 32833499 / 990,
                    ssm = 338057.9,
                    n = 1000, a = .05)

# Backwards-compatible dotted name (deprecated)
omega.partial.SS.bn(dfm = 4, dfe = 990,
                    msm = 338057.9 / 4,
                    mse = 32833499 / 990,
                    ssm = 338057.9,
                    n = 1000, a = .05)

# The same analysis can be fit with stats::lm and car::Anova(type = 3).
# This example shows how to obtain the ANOVA table and plug its values
# into omega.partial.SS.bn without relying on ezANOVA.
if (requireNamespace("car", quietly = TRUE)) {

  mod <- stats::lm(money ~ coach * type, data = bn2_data)

  # Type I table (for residual SS and df)
  aov_type1 <- stats::anova(mod)

  # Type III SS table for the effects
  aov_type3 <- car::Anova(mod, type = 3)

  # Extract dfs and sums of squares for the interaction coach:type
  dfm_int <- aov_type3["coach:type", "Df"]
  ssm_int <- aov_type3["coach:type", "Sum Sq"]
  msm_int <- ssm_int / dfm_int

  dfe <- aov_type1["Residuals", "Df"]
  sse <- aov_type1["Residuals", "Sum Sq"]
```

```
  mse <- sse / dfe

  omega_partial_ss_bn(dfm = dfm_int,
                      dfe = dfe,
                      msm = msm_int,
                      mse = mse,
                      ssm = ssm_int,
                      n = nrow(bn2_data),
                      a = .05)
}
```

`omega_partial_ss_rm` *omega^2_p (Partial Omega Squared) for Repeated Measures ANOVA from F*

Description

This function displays ω_p^2 from ANOVA analyses and its non-central confidence interval based on the F distribution. This formula is appropriate for multi-way repeated measures designs and mixed-level designs.

Usage

```
omega_partial_ss_rm(dfm, dfe, msm, mse, mss, ssm, sse, sss, a = 0.05)

omega.partial.SS.rm(dfm, dfe, msm, mse, mss, ssm, sse, sss, a = 0.05)
```

Arguments

`dfm`	degrees of freedom for the model/IV/between
`dfe`	degrees of freedom for the error/residual/within
`msm`	mean square for the model/IV/between
`mse`	mean square for the error/residual/within
`mss`	mean square for the subject variance
`ssm`	sum of squares for the model/IV/between
`sse`	sum of squares for the error/residual/within
`sss`	sum of squares for the subject variance
`a`	significance level

Details

Partial omega squared is calculated by subtracting the mean square for the error from the mean square of the model, which is multiplied by degrees of freedom of the model. This is divided by the sum of the sum of squares for the model, sum of squares for the error, sum of squares for the subject, and the mean square of the subject.

$$\omega_p^2 = \frac{df_m(MS_M - MS_E)}{SS_M + SS_E + SS_S + MS_S}$$

The F-statistic is calculated by dividing the mean square of the model by the mean square of the error.

F = msm / mse

Learn more on our example page.

Note on function and output names: This effect size is now implemented with the snake_case function name ‘omega_partial_ss_rm()‘ to follow modern R style guidelines. The original dotted version ‘omega.partial.SS.rm()‘ is still available as a wrapper for backward compatibility, and both functions return the same list. The returned object includes both the original element names (e.g., ‘omega‘, ‘omegalow‘, ‘omegahigh‘, ‘dfm‘, ‘dfe‘, ‘F‘, ‘p‘, ‘estimate‘, ‘statistic‘) and newer snake_case aliases (e.g., ‘omega_value‘, ‘omega_lower_limit‘, ‘omega_upper_limit‘, ‘df_model‘, ‘df_error‘, ‘f_value‘, ‘p_value‘). New code should prefer ‘omega_partial_ss_rm()‘ and the snake_case output names, but existing code using the older names will continue to work.

Value

omega	ω_p^2 effect size (legacy name; see also ‘omega_value‘)
omegalow	lower-level confidence interval of ω_p^2 (legacy name; see also ‘omega_lower_limit‘)
omegahigh	upper-level confidence interval of ω_p^2 (legacy name; see also ‘omega_upper_limit‘)
dfm	degrees of freedom for the model/IV/between (legacy name; see also ‘df_model‘)
dfe	degrees of freedom for the error/residual/within (legacy name; see also ‘df_error‘)
F	F-statistic (legacy name; see also ‘f_value‘)
p	p-value (legacy name; see also ‘p_value‘)
estimate	the ω_p^2 statistic and confidence interval in APA style for markdown printing
statistic	the F-statistic in APA style for markdown printing
omega_value	ω_p^2 effect size (snake_case alias of ‘omega‘)
omega_lower_limit	lower-level confidence interval of ω_p^2 (alias of ‘omegalow‘)
omega_upper_limit	upper-level confidence interval of ω_p^2 (alias of ‘omegahigh‘)
df_model	degrees of freedom for the model/IV/between (alias of ‘dfm‘)
df_error	degrees of freedom for the error/residual/within (alias of ‘dfe‘)
f_value	F-statistic (alias of ‘F‘)
p_value	p-value (alias of ‘p‘)

Examples

```
# The following example is derived from the "rm2_data" dataset,
# included in the MOTE library.

# In this experiment people were given word pairs to rate based on
# their "relatedness". How many people out of a 100 would put LOST-FOUND
# together? Participants were given pairs of words and asked to rate them
```

```
# on how often they thought 100 people would give the second word if shown
# the first word.  The strength of the word pairs was manipulated through
# the actual rating (forward strength: FSG) and the strength of the reverse
# rating (backward strength: BSG). Is there an interaction between FSG and
# BSG when participants are estimating the relation between word pairs?

# You would calculate one partial GOS value for each F-statistic.
# You can leave out the MS options if you include all the SS options.
# Here's an example for the interaction with typing in numbers.
omega_partial_ss_rm(dfm = 1, dfe = 157,
                    msm = 2442.948 / 1,
                    mse = 5402.567 / 157,
                    mss = 76988.130 / 157,
                    ssm = 2442.948, sss = 76988.13,
                    sse = 5402.567, a = .05)

# Backwards-compatible dotted name (deprecated)
omega.partial.SS.rm(dfm = 1, dfe = 157,
                    msm = 2442.948 / 1,
                    mse = 5402.567 / 157,
                    mss = 76988.130 / 157,
                    ssm = 2442.948, sss = 76988.13,
                    sse = 5402.567, a = .05)
```

`rm1_data` *Repeated Measures One-Way ANOVA Example Data*

Description

Example data for a repeated measures one-way ANOVA examining whether pulse rate differs across stimulus types. Participants were exposed to three categories of images: neutral (e.g., household objects like a toaster), positive (e.g., puppies, babies), and negative (e.g., mutilated faces, scenes of war). Pulse rate was measured for each participant under each condition. Designed for use with `omega.F`.

Usage

```
data(rm1_data)
```

Format

A data frame with 3 variables:

neutral Numeric. Pulse rate during exposure to neutral stimuli.

positive Numeric. Pulse rate during exposure to positive stimuli.

negative Numeric. Pulse rate during exposure to negative stimuli.

`rm2_data` *Repeated Measures Two-Way ANOVA Example Data*

Description

Example data for a mixed repeated-measures two-way ANOVA examining the effect of instruction type and forward/backward strength in word associations. Designed for use with `omega.partial.SS.rm` and other repeated measures ANOVA designs.

The dataset contains a between-subjects variable for instruction type, a subject identifier, and four repeated-measures conditions: - FSG (forward strength): e.g., "cheddar" → "cheese" - BSG (backward strength): e.g., "cheese" → "cheddar" Forward and backward strength were manipulated to measure overestimation of association strength.

Usage

```
data(rm2_data)
```

Format

A data frame with 6 variables:

group Factor. Between-subjects variable indicating the type of instructions given.

subject Integer or factor. Subject identifier.

fsglobsglo Numeric. Low FSG, low BSG condition.

fsghibsglo Numeric. High FSG, low BSG condition.

fsglobsghi Numeric. Low FSG, high BSG condition.

fsghibsghi Numeric. High FSG, high BSG condition.

`r_correl` *r to Coefficient of Determination (R^2) from F*

Description

This function displays the transformation from r to R^2 to calculate the non-central confidence interval for R^2 using the F distribution.

Usage

```
r_correl(r, n, a = 0.05)

r.correl(r, n, a = 0.05)
```

Arguments

`r`	correlation coefficient
`n`	sample size
`a`	significance level

Details

The t-statistic is calculated by:

$$t = \frac{r}{\sqrt{\frac{1-r^2}{n-2}}}$$

The F-statistic is the t-statistic squared:

$$F = t^2$$

Learn more on our example page.

Note on function and output names: This effect size is now implemented with the snake_case function name ‘r_correl()‘ to follow modern R style guidelines. The original dotted version ‘r.correl()‘ is still available as a wrapper for backward compatibility, and both functions return the same list. The returned object includes both the original element names (e.g., ‘r‘, ‘rlow‘, ‘rhigh‘, ‘R2‘, ‘R2low‘, ‘R2high‘, ‘se‘, ‘n‘, ‘dfm‘, ‘dfe‘, ‘t‘, ‘F‘, ‘p‘, ‘estimate‘, ‘estimateR2‘, ‘statistic‘) and newer snake_case aliases (e.g., ‘r_value‘, ‘r_lower_limit‘, ‘r_upper_limit‘, ‘r2_value‘, ‘r2_lower_limit‘, ‘r2_upper_limit‘, ‘standard_error‘, ‘sample_size‘, ‘df_model‘, ‘df_error‘, ‘t_value‘, ‘f_value‘, ‘p_value‘). New code should prefer ‘r_correl()‘ and the snake_case output names, but existing code using the older names will continue to work.

Value

r correlation coefficient

rlow lower level confidence interval for r

rhigh upper level confidence interval for r

R2 coefficient of determination

R2low lower level confidence interval of R^2

R2high upper level confidence interval of R^2

se standard error

n sample size

dfm degrees of freedom of mean

dfe degrees of freedom of error

t t-statistic

F F-statistic

p p-value

estimate the r statistic and confidence interval in APA style for markdown printing

estimateR2 the R^2 statistic and confidence interval in APA style for markdown printing

statistic the t-statistic in APA style for markdown printing

Examples

```
# This example is derived from the mtcars dataset provided in R.

# What is the correlation between miles per gallon and car weight?

cor.test(mtcars$mpg, mtcars$wt)

r_correl(r = -0.8676594, n = 32, a = .05)

# Backwards-compatible dotted name (deprecated)
r.correl(r = -0.8676594, n = 32, a = .05)
```

`r_effect` *r-family effect size wrapper*

Description

This function provides a unified interface for computing r- and variance-based effect sizes (e.g., correlations and coefficients of determination) from different input summaries. It is analogous to the `d_effect()` wrapper for standardized mean difference effect sizes.

Usage

```
r_effect(
  d = NULL,
  n1 = NULL,
  n2 = NULL,
  r = NULL,
  n = NULL,
  x2 = NULL,
  c = NULL,
  dfm = NULL,
  dfe = NULL,
  msm = NULL,
  mse = NULL,
  mss = NULL,
  sst = NULL,
  ssm = NULL,
  ssm2 = NULL,
  sss = NULL,
  sse = NULL,
  sse1 = NULL,
  sse2 = NULL,
  sse3 = NULL,
  j = NULL,
  f_value = NULL,
  a = 0.05,
```

```
  design,
  ...
)
```

Arguments

`d`	Cohen’s d value for the contrast of interest (used when ‘design = "d_to_r"‘).
`n1`	Sample size for group one (used when ‘design = "d_to_r"‘).
`n2`	Sample size for group two (used when ‘design = "d_to_r"‘).
`r`	Sample Pearson correlation coefficient (used when ‘design = "r_correl"‘), or the number of rows in the contingency table (used when ‘design = "v_chi_sq"‘).
`n`	Sample size for the correlation (used when ‘design = "r_correl"‘), the total sample size for the chi-square test (used when ‘design = "v_chi_sq"‘), or the total sample size for the ANOVA (used when ‘design = "omega_f"‘ or ‘design = "omega_partial_ss_bn"‘).
`x2`	Chi-square test statistic for the contingency table (used when ‘design = "v_chi_sq"‘).
`c`	Number of columns in the contingency table (used when ‘design = "v_chi_sq"‘).
`dfm`	Degrees of freedom for the model term (used when ‘design = "epsilon_full_ss"‘, ‘design = "eta_f"‘, ‘design = "omega_f"‘, ‘design = "omega_full_ss"‘, ‘design = "omega_partial_ss_bn"‘, ‘design = "eta_full_ss"‘, ‘design = "eta_partial_ss"‘, ‘design = "ges_partial_ss_mix"‘, ‘design = "ges_partial_ss_rm"‘, ‘design = "omega_partial_ss_rm"‘, or ‘design = "omega_g_ss_rm"‘).
`dfe`	Degrees of freedom for the error term (used when ‘design = "epsilon_full_ss"‘, ‘design = "eta_f"‘, ‘design = "omega_f"‘, ‘design = "omega_full_ss"‘, ‘design = "omega_partial_ss_bn"‘, ‘design = "eta_full_ss"‘, ‘design = "eta_partial_ss"‘, ‘design = "ges_partial_ss_mix"‘, ‘design = "ges_partial_ss_rm"‘, ‘design = "omega_partial_ss_rm"‘, or ‘design = "omega_g_ss_rm"‘).
`msm`	Mean square for the model (used when ‘design = "epsilon_full_ss"‘, ‘design = "omega_full_ss"‘, ‘design = "omega_partial_ss_bn"‘, or ‘design = "omega_partial_ss_rm"‘).
`mse`	Mean square for the error (used when ‘design = "epsilon_full_ss"‘, ‘design = "omega_full_ss"‘, ‘design = "omega_partial_ss_bn"‘, or ‘design = "omega_partial_ss_rm"‘).
`mss`	Mean square for the subject or between-subjects term (used when ‘design = "omega_partial_ss_rm"‘).
`sst`	Total sum of squares for the outcome (used when ‘design = "epsilon_full_ss"‘, ‘design = "omega_full_ss"‘, or ‘design = "omega_g_ss_rm"‘).
`ssm`	Sum of squares for the model term (used when ‘design = "eta_full_ss"‘, ‘design = "eta_partial_ss"‘, ‘design = "ges_partial_ss_mix"‘, ‘design = "ges_partial_ss_rm"‘, ‘design = "omega_partial_ss_bn"‘, ‘design = "omega_partial_ss_rm"‘, or ‘design = "omega_g_ss_rm"‘).
`ssm2`	Sum of squares for a second model or component term (used when ‘design = "omega_g_ss_rm"‘).
`sss`	Sum of squares for the subject or between-subjects term (used when ‘design = "ges_partial_ss_mix"‘, ‘design = "ges_partial_ss_rm"‘, or ‘design = "omega_partial_ss_rm"‘).

`sse`	Sum of squares for the error term (used when ‘design = "eta_partial_ss"‘, ‘design = "ges_partial_ss_mix"‘, or ‘design = "omega_partial_ss_rm"‘).
`sse1`	Sum of squares for the first error term (used when ‘design = "ges_partial_ss_rm"‘).
`sse2`	Sum of squares for the second error term (used when ‘design = "ges_partial_ss_rm"‘).
`sse3`	Sum of squares for the third error term (used when ‘design = "ges_partial_ss_rm"‘).
`j`	Number of levels for the factor (used when ‘design = "omega_g_ss_rm"‘).
`f_value`	F statistic for the model term (used when ‘design = "eta_f"‘, ‘design = "eta_full_ss"‘, ‘design = "eta_partial_ss"‘, ‘design = "ges_partial_ss_mix"‘, ‘design = "ges_partial_ss_rm"‘, ‘design = "omega_f"‘, or ‘design = "omega_g_ss_rm"‘).
`a`	Significance level used for confidence intervals. Defaults to 0.05.
`design`	Character string indicating which r-family effect size design to use. See **Supported designs**.
`...`	Additional arguments for future methods (currently unused).

Details

Currently, ‘r_effect()‘ supports effect sizes derived from Cohen’s d, from correlations, and from ANOVA summaries via several designs (see **Supported designs**). These designs call lower-level functions as [d_to_r()], [r_correl()], [epsilon_full_ss()], [eta_f()], [omega_f()], [omega_full_ss()], [eta_full_ss()], [eta_partial_ss()], [ges_partial_ss_mix()], [ges_partial_ss_rm()], [omega_partial_ss_rm()], and [omega_g_ss_rm()] with the appropriate arguments.

Value

A list whose structure depends on the selected design. For ‘design = "d_to_r"‘, the returned object is the same as from [d_to_r()].

Supported designs

- ‘"d_to_r"‘ — correlation and R^2 from Cohen’s d for independent groups. Supply ‘d‘, ‘n1‘, and ‘n2‘. In this case, ‘r_effect()‘ will call [d_to_r()] with the same arguments.

- ‘"r_correl"‘ — correlation and R^2 from a sample Pearson correlation. Supply ‘r‘ and ‘n‘. In this case, ‘r_effect()‘ will call [r_correl()] with the same arguments.

- ‘"v_chi_sq"‘ — Cramer’s V from a chi-square test of association for an r x c contingency table. Supply ‘x2‘, ‘n‘, ‘r‘, and ‘c‘. In this case, ‘r_effect()‘ will call [v_chi_sq()] with the same arguments.

- ‘"epsilon_full_ss"‘ — epsilon-squared (ϵ^2) from an ANOVA table using model and error mean squares and the total sum of squares. Supply ‘dfm‘, ‘dfe‘, ‘msm‘, ‘mse‘, and ‘sst‘. In this case, ‘r_effect()‘ will call [epsilon_full_ss()] with the same arguments.

- ‘"eta_f"‘ — eta-squared (η^2) from an ANOVA F statistic and its associated degrees of freedom. Supply ‘dfm‘, ‘dfe‘, and ‘f_value‘. In this case, ‘r_effect()‘ will call [eta_f()] with the same arguments.

- ‘"omega_f"‘ — omega-squared (ω^2) from an ANOVA F statistic, its associated degrees of freedom, and the total sample size. Supply ‘dfm‘, ‘dfe‘, ‘n‘, and ‘f_value‘. In this case, ‘r_effect()‘ will call [omega_f()] with the same arguments.

- ‘"omega_full_ss"‘ — omega-squared (ω^2) from ANOVA sums of squares, using the model mean square, error mean square, and total sum of squares along with the model and error degrees of freedom. Supply ‘dfm‘, ‘dfe‘, ‘msm‘, ‘mse‘, and ‘sst‘. In this case, ‘r_effect()‘ will call [omega_full_ss()] with the same arguments.

- ‘"omega_partial_ss_bn"‘ — partial omega-squared (ω_p^2) for between-subjects designs, using the model mean square, error mean square, model sum of squares, and total sample size along with the model and error degrees of freedom. Supply ‘dfm‘, ‘dfe‘, ‘msm‘, ‘mse‘, ‘ssm‘, and ‘n‘. In this case, ‘r_effect()‘ will call [omega_partial_ss_bn()] with the same arguments.

- ‘"eta_full_ss"‘ — eta-squared (η^2) from ANOVA sums of squares, using the model sum of squares and total sum of squares along with the model and error degrees of freedom. Supply ‘dfm‘, ‘dfe‘, ‘ssm‘, ‘sst‘, and ‘f_value‘. In this case, ‘r_effect()‘ will call [eta_full_ss()] with the same arguments.

- ‘"eta_partial_ss"‘ — partial eta-squared (η_p^2) from ANOVA sums of squares, using the model sum of squares and error sum of squares along with the model and error degrees of freedom. Supply ‘dfm‘, ‘dfe‘, ‘ssm‘, ‘sse‘, and ‘f_value‘. In this case, ‘r_effect()‘ will call [eta_partial_ss()] with the same arguments.

- ‘"ges_partial_ss_mix"‘ — partial generalized eta-squared (η_G^2) for mixed designs, using the model sum of squares, between-subjects sum of squares, and error sum of squares along with the model and error degrees of freedom. Supply ‘dfm‘, ‘dfe‘, ‘ssm‘, ‘sss‘, ‘sse‘, and ‘f_value‘. In this case, ‘r_effect()‘ will call [ges_partial_ss_mix()] with the same arguments.

- ‘"ges_partial_ss_rm"‘ — partial generalized eta-squared (η_G^2) for repeated-measures designs, using the model sum of squares, between-subjects sum of squares, and multiple error sums of squares (e.g., for each level or effect) along with the model and error degrees of freedom. Supply ‘dfm‘, ‘dfe‘, ‘ssm‘, ‘sss‘, ‘sse1‘, ‘sse2‘, ‘sse3‘, and ‘f_value‘. In this case, ‘r_effect()‘ will call [ges_partial_ss_rm()] with the same arguments.

- ‘"omega_partial_ss_rm"‘ — partial omega-squared (ω_p^2) for repeated-measures designs, using the model, subject, and error sums of squares and their associated mean squares along with the model and error degrees of freedom. Supply ‘dfm‘, ‘dfe‘, ‘msm‘, ‘mse‘, ‘mss‘, ‘ssm‘, ‘sse‘, and ‘sss‘. In this case, ‘r_effect()‘ will call [omega_partial_ss_rm()] with the same arguments.

- ‘"omega_g_ss_rm"‘ — generalized omega-squared (ω_G^2) for repeated-measures or mixed designs, using sums of squares for the model, an additional model/component term, and the total sum of squares, along with the mean square for the subject term and the number of levels for the factor. Supply ‘dfm‘, ‘dfe‘, ‘ssm‘, ‘ssm2‘, ‘sst‘, ‘mss‘, ‘j‘, and ‘f_value‘. In this case, ‘r_effect()‘ will call [omega_g_ss_rm()] with the same arguments.

Examples

```
# From Cohen's d for independent groups to r and R^2
r_effect(d = -1.88, n1 = 4, n2 = 4, a = .05, design = "d_to_r")
# From a sample correlation to r and R^2
r_effect(r = -0.8676594, n = 32, a = .05, design = "r_correl")
# From a chi-square test of association to Cramer's V
r_effect(x2 = 2.0496, n = 60, r = 3, c = 3, a = .05, design = "v_chi_sq")
# From F and degrees of freedom to eta^2
r_effect(dfm = 2, dfe = 8, f_value = 5.134, a = .05, design = "eta_f")
# From F, degrees of freedom, and N to omega^2
r_effect(dfm = 2, dfe = 8, n = 11, f_value = 5.134,
a = .05, design = "omega_f")
```

```
# From sums of squares to omega^2
r_effect(
  dfm   = 2,
  dfe   = 8,
  msm   = 12.621,
  mse   = 2.548,
  sst   = (25.54 + 19.67),
  a     = .05,
  design = "omega_full_ss"
)
# From sums of squares to partial eta^2
r_effect(
  dfm    = 4,
  dfe    = 990,
  ssm    = 338057.9,
  sse    = 32833499,
  f_value = 2.548,
  a      = .05,
  design = "eta_partial_ss"
)
# From mixed-design sums of squares to partial generalized eta^2
r_effect(
  dfm     = 1,
  dfe     = 156,
  ssm     = 71.07608,
  sss     = 30936.498,
  sse     = 8657.094,
  f_value = 1.280784,
  a       = .05,
  design  = "ges_partial_ss_mix"
)

# From repeated-measures sums of squares to partial generalized eta^2
r_effect(
  dfm     = 1,
  dfe     = 157,
  ssm     = 2442.948,
  sss     = 76988.13,
  sse1    = 5402.567,
  sse2    = 8318.75,
  sse3    = 6074.417,
  f_value = 70.9927,
  a       = .05,
  design  = "ges_partial_ss_rm"
)

# From repeated-measures sums of squares to partial omega^2_p
r_effect(
  dfm   = 1,
  dfe   = 157,
  msm   = 2442.948 / 1,
  mse   = 5402.567 / 157,
  mss   = 76988.130 / 157,
```

```
  ssm   = 2442.948,
  sss   = 76988.13,
  sse   = 5402.567,
  a     = .05,
  design = "omega_partial_ss_rm"
)

# From repeated-measures sums of squares to generalized omega^2_G
r_effect(
  dfm     = 1,
  dfe     = 156,
  ssm     = 6842.46829,
  ssm2    = 14336.07886,
  sst     = sum(c(30936.498, 6842.46829,
                14336.07886, 8657.094, 71.07608)),
  mss     = 30936.498 / 156,
  j       = 2,
  f_value = 34.503746,
  a       = .05,
  design  = "omega_g_ss_rm"
)
```

`singt_data` *One-Sample t-Test Example Data*

Description

Simulated dataset of SAT scores from gifted/honors students at a specific school, intended for comparison to the national average SAT score (1080) for gifted/honors students nationwide. Designed for use with functions such as `d.single.t` and `d.single.t.t`.

Usage

```
data(singt_data)
```

Format

A data frame with 1 variable:

SATscore Numeric. SAT scores of gifted/honors program students at one school.

`v_chi_sq` *V for Chi-Square*

Description

This function displays V and its non-central confidence interval for the specified χ^2 statistic.

Usage

```
v_chi_sq(x2, n, r, c, a = 0.05)

v.chi.sq(x2, n, r, c, a = 0.05)
```

Arguments

`x2`	chi-square statistic
`n`	sample size
`r`	number of rows in the contingency table
`c`	number of columns in the contingency table
`a`	significance level

Details

V is calculated by finding the square root of χ^2 divided by the product of the sample size and the smaller of the two degrees of freedom.

$$V = \sqrt{\frac{\chi^2}{n \times df_{\text{small}}}}$$

Learn more on our example page.

Note on function and output names: This effect size is now implemented with the snake_case function name ‘v_chi_sq()‘ to follow modern R style guidelines. The original dotted version ‘v.chi.sq()‘ is still available as a wrapper for backward compatibility, and both functions return the same list. The returned object includes both the original element names (e.g., ‘v‘, ‘vlow‘, ‘vhigh‘, ‘n‘, ‘df‘, ‘x2‘, ‘p‘, ‘estimate‘, ‘statistic‘) and newer snake_case aliases (e.g., ‘v_value‘, ‘v_lower_limit‘, ‘v_upper_limit‘, ‘sample_size‘, ‘df_total‘, ‘chi_square‘, ‘p_value‘). New code should prefer ‘v_chi_sq()‘ and the snake_case output names, but existing code using the older names will continue to work.

Value

v V statistic

vlow lower level confidence interval of V

vhigh upper level confidence interval of V

n sample size

df degrees of freedom

x2 χ^2 statistic

p p-value

estimate the V statistic and confidence interval in APA style for markdown printing

statistic the χ^2 statistic in APA style for markdown printing

Examples

```
# The following example is derived from the "chisq_data"
# dataset, included in the MOTE library.

# Individuals were polled about their number of friends (low, medium, high)
# and their number of kids (1, 2, 3+) to determine if there was a
# relationship between friend groups and number of children, as we
# might expect that those with more children may have less time for
# friendship maintaining activities.

chisq.test(chisq_data$kids, chisq_data$friends)

v_chi_sq(x2 = 2.0496, n = 60, r = 3, c = 3, a = .05)

# Backwards-compatible dotted name (deprecated)
v.chi.sq(x2 = 2.0496, n = 60, r = 3, c = 3, a = .05)
```

Index