Package ‘ViSe’

April 22, 2024

Type Package

Title Visualizing Sensitivity

Version 0.1.2

Depends R (>= 3.1.0)

Imports stats, ggplot2, methods, dplyr, tidyr, scales, cowplot, shiny

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

Description Designed to help the user to determine the sensitivity of an proposed causal effect to unconsidered common causes. Users can create visualizations of sensitivity, effect sizes, and determine which pattern of effects would support a causal claim for between group differences. Number needed to treat formula from Kraemer H.C. & Kupfer D.J. (2006) <doi:10.1016/j.biopsych.2005.09.014>.

License LGPL-3

Encoding UTF-8

RoxygenNote 7.3.1

URL http://www.aggieerin.com/ViSe/

Suggests knitr, rmarkdown, plotly

VignetteBuilder knitr

Collate 'globals.R' 'adjusted_coef.R' 'apa.R' 'big_donut.R' 'calculate_d.R' 'd_to_f2.R' 'd_to_nnt.R' 'd_to_r.R' 'estimate_d.R' 'estimate_r.R' 'noncentral_t.R' 'other_to_d.R' 'probability_superiority.R' 'proportion_overlap.R' 'runExample.R' 'visualize_c.R' 'visualize_c_map.R' 'visualize_effects.R'

NeedsCompilation no

Author Erin M. Buchanan [aut, cre] (<https://orcid.org/0000-0002-9689-4189>)

Repository CRAN

Date/Publication 2024-04-22 14:32:33 UTC

R topics documented:

`adjusted_coef` *Adjust coefficient for confounders*

Description

This function calculates the adjusted effect after controlling for confounding effects. You can use d values or standardized regression coefficients.

Usage

```
adjusted_coef(effect_xz, effect_uxz, effect_d)
```

Arguments

`effect_xz`	Effect of x on y given z
`effect_uxz`	Effect of u on y given x and z
`effect_d`	Effect size difference of interest

Value

Adjusted effect size of x on y given u and z

Examples

```
adjusted_coef(effect_xz = .2,
  effect_uxz = .4,
  effect_d = .12)
```

apa — *APA Format*

Description

A function that formats decimals and leading zeroes for creating reports in scientific style.

Usage

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

Arguments

`value`	A set of numeric values, either a single number, vector, or set of columns.
`decimals`	The number of decimal points desired in the output.
`leading`	Logical value: `TRUE` for leading zeroes on decimals and `FALSE` for no leading zeroes on decimals. The default is `TRUE`.

Details

This function creates "pretty" character vectors from numeric variables for printing as part of a report. The value can take a single number, matrix, vector, or multiple columns from a data frame, as long as they are numeric. The values will be coerced into numeric if they are characters or logical values, but this process may result in an error if values are truly alphabetical.

Value

Returns a nicely formatted character vector for numbers for reporting purposes.

Examples

```
apa(value = 0.54674, decimals = 3, leading = TRUE)
```

calculate_d — *d_s for Between Subjects with Pooled SD Denominator*

Description

This function displays d for two between subjects groups and gives the central and non-central confidence interval using the pooled standard deviation as the denominator.

Usage

```
calculate_d(
  m1 = NULL,
  m2 = NULL,
  sd1 = NULL,
  sd2 = NULL,
  n1 = NULL,
  n2 = NULL,
  t = NULL,
  model = NULL,
  df = NULL,
  x_col = NULL,
  y_col = NULL,
  d = NULL,
  a = 0.05,
  lower = TRUE
)
```

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
`t`	optional, calculate d from independent t, you must include n1 and n2 for degrees of freedom
`model`	optional, calculate d from t.test for independent t, you must still include n1 and n2
`df`	optional dataframe that includes the x_col and y_col
`x_col`	name of the column that contains the factor levels OR a numeric vector of group 1 scores
`y_col`	name of the column that contains the dependent score OR a numeric vector of group 2 scores
`d`	a previously calculated d value from a study
`a`	significance level
`lower`	Use this to indicate if you want the lower or upper bound of d for one sided confidence intervals. If d is positive, you generally want `lower = TRUE`, while negative d values should enter `lower = FALSE` for the upper bound that is closer to zero.

Details

To calculate d_s, mean two is subtracted from mean one and divided by the pooled standard deviation.

$$d_s = \frac{M_1 - M_2}{S_{pooled}}$$

You should provide one combination of the following:

1: m1 through n2

2: t, n1, n2

3: model, n1, n2

4: df, "x_col", "y_col"

5: x_col, y_col as numeric vectors

6: d, n1, n2

You must provide alpha and lower to ensure the right confidence interval is provided for you.

Value

Provides the effect size (Cohen's *d*) with associated central and non-central 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. The one-tailed confidence interval is also included for sensitivity analyses.

`d`	effect size
`dlow`	noncentral lower level confidence interval of d value
`dhigh`	noncentral upper level confidence interval of d value
`dlow_central`	central lower level confidence interval of d value
`dhigh_central`	central upper level confidence interval of d value
`done_low`	noncentral lower bound of one tailed confidence interval
`done_low_central`	central lower bound of one tailed confidence interval
`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

```
calculate_d(m1 = 14.37, # neglect mean
   sd1 = 10.716, # neglect sd
   n1 = 71, # neglect n
   m2 = 10.69, # none mean
   sd2 = 8.219, # none sd
   n2 = 3653, # none n
   a = .05, # alpha/confidence interval
   lower = TRUE) # lower or upper bound
```

---

d_to_f2 *Convert d to Cohen's f*

Description

This function allows you to convert d to Cohen's f and f^2 statistics.

Usage

```
d_to_f2(d)
```

Arguments

d	the effect size to convert

Value

Both Cohen's f and f^2 statistics

f	d values translated into f
f2	d values translated into f^2

Examples

```
d_to_f2(.25)
```

d_to_nnt — *Convert d to Number Needed to Treat*

Description

This function calculates the number needed to treat from continuous measures (Cohen's d) using Kraemer and Kupfer (2006) formula.

Usage

```
d_to_nnt(d = NULL)
```

Arguments

d	the effect size

Value

nnt values from d

References

Kraemer H.C., Kupfer D.J. (2006) Size of treatment effects and their importance to clinical research and practice. *Biolological Psychiatry, 59*, 990–996. https://doi.org/10.1016/j.biopsych.2005.09.014

Examples

```
d_to_nnt(d = .25)
```

d_to_r — *Convert d to correlation coefficient*

Description

This function allows you to convert d to Pearson's correlation coefficient.

Usage

```
d_to_r(d)
```

Arguments

d	the effect size to convert

Value

correlation coefficient

Examples

```
d_to_r(.25)
```

`estimate_d` *Visualization for Estimating* d_s

Description

This function displays a visualization of effect sizes.

Usage

```
estimate_d(
  m1 = NULL,
  m2 = NULL,
  sd1 = NULL,
  sd2 = NULL,
  n1 = NULL,
  n2 = NULL,
  d = NULL
)
```

Arguments

`m1`	mean from first group
`m2`	mean from second group
`sd1`	standard deviation from first group
`sd2`	standard deviation from second group
`n1`	sample size for first group
`n2`	sample size for the second group
`d`	estimate of the effect size

Value

Returns a pretty graph

`d`	effect size
`graph`	A graph of the distributions of the effect size

Examples

```
estimate_d(d = .25)

estimate_d(m1 = 10, m2 = 8, sd1 = 5, sd2 = 4,
 n1 = 100, n2 = 75)
```

`estimate_r` *Visualization for Estimating r*

Description

This function displays a visualization of effect sizes.

Usage

```
estimate_r(r = NULL)
```

Arguments

`r`	a correlation to visualize

Value

Returns a pretty graph

`graph`	A graph of the effect size

Examples

```
estimate_r(r = .4)
```

`other_to_d` *Convert other statistics to d*

Description

This function allows you to convert other effect sizes to d including f, f squared, number needed to treat, correlation coefficient, probability of superiority, proportion overlap (u1, u2, u3, and proportion distribution overlap). Please note these are approximations.

Usage

```
other_to_d(
  f = NULL,
  f2 = NULL,
  nnt = NULL,
  r = NULL,
  prob = NULL,
  prop_u1 = NULL,
  prop_u2 = NULL,
  prop_u3 = NULL,
  prop_overlap = NULL
)
```

Arguments

`f`	Cohen's f
`f2`	Cohen's f squared
`nnt`	Number needed to treat
`r`	Correlation coefficient
`prob`	Probability superiority
`prop_u1`	Proportion Overlap U1
`prop_u2`	Proportion Overlap U2
`prop_u3`	Proportion Overlap U3
`prop_overlap`	Proportion Overlap of Distributions

Value

d effect size

Examples

```
other_to_d(f = .1)
```

`probability_superiority` *Probability of Superiority Calculation*

Description

This function calculates the probability of superiority from independent samples Cohen's d calculation.

Usage

```
probability_superiority(
  d = NULL,
  m1 = NULL,
  m2 = NULL,
  sd1 = NULL,
  sd2 = NULL,
  n1 = NULL,
  n2 = NULL,
  a = 0.05,
  t = NULL,
  model = NULL,
  df = NULL,
  x_col = NULL,
  y_col = NULL
)
```

Arguments

`d`	the effect size
`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
`t`	optional, calculate d from independent t, you must include n1 and n2 for degrees of freedom
`model`	optional, calculate d from t.test for independent t, you must still include n1 and n2
`df`	optional dataframe that includes the x_col and y_col
`x_col`	name of the column that contains the factor levels OR a numeric vector of group 1 scores
`y_col`	name of the column that contains the dependent score OR a numeric vector of group 2 scores

Details

You should provide one combination of the following:

1: d

2: m1 through n2

3: t, n1, n2

4: model, n1, n2

5: df, "x_col", "y_col"

6: x_col, y_col as numeric vectors

Value

The probability of superiority.

Examples

```
probability_superiority(d = .25)
```

---

proportion_overlap *Proportion Overlap Calculations for Cohen's d*

Description

This function calculates the proportion overlap from two independent group d effect size calculations. Cohen's u1, u2, u3 and proportion overlap are provided.

Usage

```
proportion_overlap(
  model = NULL,
  x_col = NULL,
  y_col = NULL,
  df = NULL,
  d = NULL
)
```

Arguments

`model`	a saved independent t-test model
`x_col`	name of the column that contains the factor levels OR a numeric vector of group 1 scores
`y_col`	name of the column that contains the dependent score OR a numeric vector of group 2 scores
`df`	optional dataframe that includes the x_col and y_col
`d`	previously calculated d value

Value

A list of the following:

`u1`	Proportion of non-overlap across both distributions
`u2`	Proportion that one group is more than the same proportion in the other group
`u3`	Proportion of one group that is smaller than the median of the other group
`p_o`	Proportional overlap of distributions

Examples

```
proportion_overlap(d = .25)
```

`runExample` *Run Shiny App*

Description

This function is a convenience function to help you easily run the shiny app for the package.

Usage

```
runExample()
```

Value

Opens the shiny app version of the package to use interactively.

`visualize_c` *Visualization for Estimating c Bias*

Description

This function displays a visualization of the possible bias c that allows for a non-zero effect in sensitivity.

Usage

```
visualize_c(dlow, lower = TRUE)
```

Arguments

`dlow`	The lower limit of the possible effect size
`lower`	Use this to indicate if you want the lower or upper bound of d for one sided confidence intervals. If d is positive, you generally want `lower = TRUE`, while negative d values should enter `lower = FALSE` for the upper bound that is closer to zero.

Value

Returns a pretty graph

`graph`	The graph of possible values for c

Examples

```
visualize_c(dlow = .25, lower = TRUE)
```

---

visualize_c_map   *Visualization for Estimating c Bias + Estimates*

Description

This function displays a visualization of the possible bias c that allows for a non-zero effect in sensitivity. This function includes the ability to add values of effect size and correlation to see how they map onto the proposed c value.

Usage

```
visualize_c_map(
  dlow,
  r_values,
  d_values = NULL,
  f_values = NULL,
  f2_values = NULL,
  nnt_values = NULL,
  prob_values = NULL,
  prop_u1_values = NULL,
  prop_u2_values = NULL,
  prop_u3_values = NULL,
  prop_overlap_values = NULL,
  lower = TRUE
)
```

Arguments

`dlow`	The lower limit of the possible effect size (required).
`r_values`	A vector of correlation values that are possible (required).
`d_values`	A vector of effect size values that are possible.
`f_values`	A vector of f effect size values that are possible.
`f2_values`	A vector of f2 effect size values that are possible.
`nnt_values`	A vector of number needed to treat effect size values that are possible.
`prob_values`	A vector of probability of superiority effect size values that are possible.
`prop_u1_values`	A vector of proportion of overlap u1 effect size values that are possible.
`prop_u2_values`	A vector of proportion of overlap u2 effect size values that are possible.
`prop_u3_values`	A vector of proportion of overlap u3 effect size values that are possible.

`prop_overlap_values`
: A vector of proportion of distribution overlap effect size values that are possible.

`lower`
: Use this to indicate if you want the lower or upper bound of d for one sided confidence intervals. If d is positive, you generally want `lower = TRUE`, while negative d values should enter `lower = FALSE` for the upper bound that is closer to zero (required).

Value

Returns a pretty graph of the possible effect size and correlation combinations with the region of effect colored in. Note that all effect sizes are converted to d for the graph.

`graph`
: The graph of possible values for c

Examples

```
visualize_c_map(dlow = .25,
  d_values = c(.2, .3, .8),
  r_values = c(.1, .4, .3),
  lower = TRUE)
```

`visualize_effects` *Visualization for Conversions of Effect Sizes*

Description

This function displays a visualization the same effect in various effect sizes including d, f, f^2, proportion overlap, correlation, number needed to treat, and more.

Usage

```
visualize_effects(d)
```

Arguments

`d`
: d effect size to convert to other numbers

Value

Returns a pretty graph of all the effects

`graph`
: ggplot object of converted effect sizes

Examples

```
visualize_effects(d = .25)
```

Index