Package ‘InteractionPoweR’

March 24, 2026

Title Power Analyses for Interaction Effects in Cross-Sectional Regressions

Version 0.2.4

Description Power analysis for regression models which test the interaction of two or three independent variables on a single dependent variable. Includes options for correlated interacting variables and specifying variable reliability. Two-way interactions can include continuous, binary, or ordinal variables. Power analyses can be done either analytically or via simulation. Includes tools for simulating single data sets and visualizing power analysis results. The primary functions are power_interaction_r2() and power_interaction() for two-way interactions, and power_interaction_3way_r2() for three-way interactions. The function run_pos_power_search() provides a stability analysis for two-way interactions. Please cite as: Baranger DAA, Finsaas MC, Goldstein BL, Vize CE, Lynam DR, Olino TM (2023). ``Tutorial: Power analyses for interaction effects in cross-sectional regressions.'' <doi:10.1177/25152459231187531>. If you use the stability analyses, please cite: Castillo A, Miller JD, Vize C, Baranger DAA, Lynam DR. ``When Do Interaction/Moderation Effects Stabilize in Linear Regression?''<doi:10.1177/25152459251407860>.

Maintainer David Baranger <dbaranger@gmail.com>

URL https://dbaranger.github.io/InteractionPoweR/,
https://doi.org/10.1177/25152459231187531,
https://doi.org/10.1177/25152459251407860

BugReports https://github.com/dbaranger/InteractionPoweR/issues

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

Depends R (>= 3.5.0)

Imports dplyr, parallel, doParallel, foreach, ggplot2, polynom, chngpt, rlang, tidyr, stats, ggbeeswarm, Matrix

NeedsCompilation no

Author David Baranger [aut, cre],
Andrew Castillo [aut],
Brandon Goldstein [ctb],
Megan Finsaas [ctb],
Thomas Olino [ctb],
Colin Vize [ctb],
Don Lynam [ctb]

Repository CRAN

Date/Publication 2026-03-24 06:10:45 UTC

Contents

compute_adjustment 2
cor.mat.3way 3
generate.interaction.cov.input 4
generate_interaction 5
name_key 7
norm2ordinal 7
plot_interaction 8
plot_power_curve 8
plot_simple_slope 9
power_estimate 10
power_interaction 10
power_interaction_3way_r2 13
power_interaction_r2 15
power_interaction_r2_covs 16
run_pos_power_search 17
simple.slopes.3way 19
test_interaction 20

Index **21**

`compute_adjustment` *compute_adjustment*

Description

Computes how much variable correlations need to be adjusted so that they have the desired correlation structure after transformation. Intended for internal use only.

Usage

```
compute_adjustment(
  r.x1.y,
  r.x2.y,
  r.x1x2.y,
  r.x1.x2,
  N.adjustment = 1e+06,
  tol = 0.005,
  iter = 10,
  k.x1,
  k.x2,
  k.y
)
```

Arguments

`r.x1.y`	Internal use only
`r.x2.y`	Internal use only
`r.x1x2.y`	Internal use only
`r.x1.x2`	Internal use only
`N.adjustment`	Internal use only
`tol`	Internal use only
`iter`	Internal use only
`k.x1`	Internal use only
`k.x2`	Internal use only
`k.y`	Internal use only

Value

Correlation adjustments.

Examples

```
compute_adjustment(r.x1.y = .2,r.x2.y = .2,r.x1x2.y = .1,r.x1.x2 = .2,
k.x1 = 0,k.x2=0,k.y=2)
```

`cor.mat.3way` *See the correlation matrix for a 3-way interaction*

Description

Prints or plots the correlation matrix for a 3-way interaction

Usage

```
cor.mat.3way(power.results, row.num = 1, return.plot = FALSE)
```

Arguments

`power.results` Data frame of results from power_interaction_3way_r2().

`row.num` Which row to show? Can only be a single number. Default is 1.

`return.plot` Return a matrix (FALSE, default), or a plot (TRUE)?

Value

A matrix or a ggplot2 object

Examples

```
power_analysis = power_interaction_3way_r2(detailed_results = TRUE,N = c(1000),
r.x1.y = .2,r.x2.y = .3,r.x3.y = .1,r.x1x2.y = .01,r.x1x3.y = .05,r.x2x3.y = .1,
b.x1x2x3 = 0.1,r.x1.x2 = .1,r.x1.x3 = .1,r.x2.x3 = .1,
rel.x1 = 1,rel.x2 = 1,rel.x3 = 1,rel.y = 1 )
cor.mat.3way(power_analysis)
```

`generate.interaction.cov.input` *Creates the input to* `power_interaction_r2_covs()`

Description

Companion function to 'power_interaction_r2_covs()'. Generates a formatted list for users to specify the analysis parameters.

Usage

```
generate.interaction.cov.input(c.num)
```

Arguments

`c.num` Number of covariates in the model.

Value

A list to be used with the 'power_interaction_r2_covs()' function.

Examples

```
ex1 = generate.interaction.cov.input(c.num=2)
ex1$correlations$r.y.x1x2 = c(0.1,0.2,0.3)
```

generate_interaction *Generate interaction data set*

Description

Simulate a single data set with an interaction (y ~ x1 + x2 + x1*x2). All values other than 'N' are population-level effects - the values within any single simulated data set will vary around the defined values.

Usage

```
generate_interaction(
  N,
  r.x1.y,
  r.x2.y,
  r.x1x2.y,
  r.x1.x2,
  rel.x1 = 1,
  rel.x2 = 1,
  rel.y = 1,
  k.x1 = 0,
  k.x2 = 0,
  k.y = 0,
  adjust.correlations = TRUE,
  tol = 0.005,
  iter = 10,
  N.adjustment = 1e+06,
  r.x1.y.adjust = NULL,
  r.x2.y.adjust = NULL,
  r.x1.x2.adjust = NULL,
  r.x1x2.y.adjust = NULL,
  internal.adjust = FALSE,
  skew.x1 = NA,
  skew.x2 = NA,
  skew.y = NA
)
```

Arguments

`N` Sample size. Must be a positive integer. Has no default value.

`r.x1.y` Pearson's correlation between x1 and y. Must be between -1 and 1. Has no default value.

`r.x2.y` Pearson's correlation between x2 and y. Must be between -1 and 1. Assumed to be the 'moderator' in some functions. Has no default value.

`r.x1x2.y` Pearson's correlation between the interaction term x1x2 (x1 * x2) and y. Must be between -1 and 1. Has no default value.

`r.x1.x2` Pearson’s correlation between x1 and x2. Must be between -1 and 1. Has no default value.

`rel.x1` Reliability of x1 (e.g. test-retest reliability, ICC, Cronbach’s alpha). Default is 1 (perfect reliability). Must be greater than 0 and less than or equal to 1.

`rel.x2` Reliability of x2 (e.g. test-retest reliability, ICC, Cronbach’s alpha). Default is 1 (perfect reliability). Must be greater than 0 and less than or equal to 1.

`rel.y` Reliability of xy (e.g. test-retest reliability, ICC, Cronbach’s alpha). Default is 1 (perfect reliability). Must be greater than 0 and less than or equal to 1.

`k.x1` Number of discrete values for x1. Can be used to make a variable binary or ordinal.

`k.x2` Number of discrete values for x2. Can be used to make a variable binary or ordinal.

`k.y` Number of discrete values for y.Can be used to make a variable binary or ordinal.

`adjust.correlations`
If variables are ordinal or binary, should correlations be adjusted so that output data has the specified correlation structure? Default is TRUE.

`tol` Correlation adjustment tolerance. When adjust.correlations = TRUE, correlations are adjusted so that the population correlation is within r=’tol’ of the target. Default = 0.005.

`iter` Max number of iterations to run the correlation adjustment for. Typically only a couple are needed. Default = 10.

`N.adjustment` Sample size to use when adjusting correlations. Default = 1000000.

`r.x1.y.adjust` Internal use only.

`r.x2.y.adjust` Internal use only.

`r.x1.x2.adjust` Internal use only.

`r.x1x2.y.adjust`
Internal use only.

`internal.adjust`
Internal use only.

`skew.x1` No longer supported.

`skew.x2` No longer supported.

`skew.y` No longer supported.

Value

A data frame containing variables ’x1’, ’x2’, ’y’, and ’x1x2’. ’x1x2’ is x1*x2. The correlations between these variables are drawn from the defined population-level values. Output variables are all z-scored (mean=0, sd=1).

Examples

```
dataset <- generate_interaction(N = 10,r.x1.y = 0,r.x2.y = .1,r.x1x2.y = -.2,r.x1.x2 = .3)
```

`name_key` *Name key for plotting*

Description

Expanded variable names so that plots look nicer.

Usage

```
data(name_key)
```

Format

A data frame with 25 rows and 2 variables

`norm2ordinal` *norm2ordinal*

Description

Transforms a vector with a normal distribution to a binomial distribution with two values.

Usage

```
norm2ordinal(x, k)
```

Arguments

`x`	Input vector
`k`	Number of discrete values (e.g., 2=binary, 5=likert scale)

Value

A ordinal or binary variable

Examples

```
norm2ordinal(x = rnorm(n = 100,mean = 0,sd = 1),k=2)
```

plot_interaction *Plot interaction*

Description

Plots a single simulated interaction data set

Usage

```
plot_interaction(data, q = 3)
```

Arguments

data	Output of generate_interaction().
q	Simple slope quantiles. Default is 2. X2 is the default moderator, unless X1 is already binary. Must be a positive integer > 1.

Value

A ggplot2 object

Examples

```
dataset <- generate_interaction(N = 250,r.x1.y = 0,r.x2.y = .1,r.x1x2.y = -.2,r.x1.x2 = .3)
plot_interaction(dataset,q=3)
```

plot_power_curve *Plot power curve*

Description

Plot the output of power_interaction().

Usage

```
plot_power_curve(
  power_data,
  x = NULL,
  group = NULL,
  facets = NULL,
  power_target = 0.8
)
```

Arguments

`power_data`	Data frame of results from power_interaction(). Can accept the raw results if up to 3 parameters were varied during simulation. Any more and data should be filtered first.
`x`	Optional, the x-axis of the plot. Default is the first variable after 'pwr'.
`group`	Optional, grouping variable for the line color. Default is the second variable after 'pwr', if present.
`facets`	Optional, grouping variable for plot facets. Default is the third variable after 'pwr' if present.
`power_target`	The target power. Default is 80%.

Value

A ggplot2 object

Examples

```
power_analysis <- power_interaction(n.iter = 10,N = seq(100,300,by=100),
r.x1.y = 0,r.x2.y = .1,r.x1x2.y = -.2,r.x1.x2 = .3,detailed_results = TRUE)
plot_power_curve(power_analysis)
```

`plot_simple_slope` *Simple slope plot*

Description

Plots the simple slope min and max estimates from power_interaction().

Usage

```
plot_simple_slope(power_data, x = NULL, facets = NULL)
```

Arguments

`power_data`	Data frame of results from power_interaction(). Can accept the raw results if up to 2 parameters were varied during simulation. Any more and data should be filtered first.
`x`	Optional, the x-axis of the plot. Default is the first variable after 'pwr'.
`facets`	Optional, grouping variable for plot facets. Default is the second variable after 'pwr' if present.

Value

A ggplot2 object

Examples

```
power_analysis <- power_interaction(n.iter = 10,N = seq(100,300,by=100),
r.x1.y = 0,r.x2.y = .1,r.x1x2.y = -.2,r.x1.x2 = .3,detailed_results = TRUE)
plot_simple_slope(power_analysis)
```

`power_estimate` *Power estimate*

Description

Uses regression to estimate the value needed to attain the target power, given a set of simulation results.

Usage

```
power_estimate(power_data, x, power_target)
```

Arguments

`power_data` Output of power_interaction().

`x` The name of the target variable as a character string.

`power_target` The desired power level. Must be between 0 and 1 (e.g., 0.8 for 80% power).

Value

A data frame containing the value of x that achieves the target power for each combination of settings. Will return NA if target power is outside the simulation data.

Examples

```
simulation_results = power_interaction_r2(N=seq(100,300,by=10),
r.x1.y=0.2, r.x2.y=.2,r.x1x2.y=0.2,r.x1.x2=.2)
power_estimate(power_data = simulation_results, x = "N", power_target = .8)
```

`power_interaction` *Power analysis for interactions*

Description

Power analysis for interaction models, by simulation. A set of n.iter simulations is run for each unique combination of model settings.

Usage

```
power_interaction(
  n.iter,
  N,
  r.x1.y,
  r.x2.y,
  r.x1x2.y,
  r.x1.x2,
  rel.x1 = 1,
  rel.x2 = 1,
  rel.y = 1,
  k.x1 = 0,
  k.x2 = 0,
  k.y = 0,
  adjust.correlations = TRUE,
  alpha = 0.05,
  q = 2,
  cl = NULL,
  ss.IQR = 1.5,
  N.adjustment = 1e+06,
  detailed_results = FALSE,
  full_simulation = FALSE,
  tol = 0.005,
  iter = 10,
  skew.x1 = NA,
  skew.x2 = NA,
  skew.y = NA
)
```

Arguments

`n.iter` Number of iterations. The number of simulations to run for each unique setting combination. Must be a positive integer.

`N` Sample size. Must be a positive integer. Has no default value. Can be a single value or a vector of values.

`r.x1.y` Pearson's correlation between x1 and y. Must be between -1 and 1.. Has no default value. Can be a single value or a vector of values.

`r.x2.y` Pearson's correlation between x2 and y. Must be between -1 and 1.. Assumed to be the 'moderator' in some functions. Has no default value. Can be a single value or a vector of values.

`r.x1x2.y` Pearson's correlation between the interaction term x1x2 (x1 * x2) and y. Must be between -1 and 1.. Has no default value. Can be a single value or a vector of values.

`r.x1.x2` Pearson's correlation between x1 and x2. Must be between -1 and 1.. Has no default value. Can be a single value or a vector of values.

`rel.x1` Reliability of x1 (e.g. test-retest reliability, ICC, Cronbach's alpha). Default is 1 (perfect reliability). Must be greater than 0 and less than or equal to 1.

`rel.x2` Reliability of x2 (e.g. test-retest reliability, ICC, Cronbach's alpha). Default is 1 (perfect reliability). Must be greater than 0 and less than or equal to 1.

`rel.y` Reliability of xy (e.g. test-retest reliability, ICC, Cronbach's alpha). Default is 1 (perfect reliability). Must be greater than 0 and less than or equal to 1.

`k.x1` Number of discrete values for x1. Can be used to make a variable binary or ordinal.

`k.x2` Number of discrete values for x2. Can be used to make a variable binary or ordinal.

`k.y` Number of discrete values for y. Can be used to make a variable binary or ordinal.

`adjust.correlations`
If variables are ordinal or binary, should correlations be adjusted so that output data has the specified correlation structure? Default is TRUE.

`alpha` The alpha. At what p-value is the interaction deemed significant? Default is 0.05.

`q` Simple slopes. How many quantiles should x2 be split into for simple slope testing? Default is 2. Simple slope testing returns the effect-size (slope) of y~x1 for the two most extreme quantiles of x2. If q=3 then the two slopes are y~x1 for the bottom 33% of x2, and the top 33% of x2.

`cl` Number of clusters to use for running simulations in parallel (recommended). Default is 1 (i.e. not in parallel).

`ss.IQR` Simple slope IQR. Multiplier when estimating the distribution of simple slopes within each simulation setting. Default is 1.5.

`N.adjustment` Sample size for simulations where correlation matrix is corrected to allow for binary/ordinal variables. Default is 1000000

`detailed_results`
Default is FALSE. Should detailed results be reported?

`full_simulation`
Default is FALSE. If TRUE, will return a list that includes the full per-simulation results.

`tol` Correlation adjustment tolerance. When adjust.correlations = TRUE, correlations are adjusted so that the population correlation is within r='tol' of the target. Default = 0.005.

`iter` Max number of iterations to run the correlation adjustment for. Typically only a couple are needed. Default = 10.

`skew.x1` No longer supported.

`skew.x2` No longer supported.

`skew.y` No longer supported.

Value

A data frame containing the power (% significant results) for each unique setting combination. If full_simulation = TRUE will return a list, with one data frame that includes power, and a second that includes raw simulation results.

Examples

```
power_interaction(n.iter=10, N=10,r.x1.y=0.2, r.x2.y=.2,r.x1x2.y=0.5,r.x1.x2=.2)
```

`power_interaction_3way_r2`

Analytic power analysis for 3-way interactions

Description

Power analysis for 3-way interaction models, computed via change in R2. Valid for interactions with continuous, normally distributed, variables. `b.x1x2x3` is used to specify the magnitude of the interaction effect size.

Usage

```
power_interaction_3way_r2(
  N,
  b.x1x2x3,
  r.x1.y,
  r.x2.y,
  r.x3.y,
  r.x1x2.y,
  r.x1x3.y,
  r.x2x3.y,
  r.x1.x2,
  r.x1.x3,
  r.x2.x3,
  rel.x1 = 1,
  rel.x2 = 1,
  rel.x3 = 1,
  rel.y = 1,
  alpha = 0.05,
  detailed_results = FALSE,
  cl = NULL
)
```

Arguments

`N` Sample size. Must be a positive integer. Has no default value. Can be a single value or a vector of values.

`b.x1x2x3` Regression coefficient of the 3-way interaction term x1x2x3.Must be between -1 and 1. Default is NULL. Can be a single value or a vector of values.

`r.x1.y` Pearson's correlation between x1 and y. Must be between -1 and 1. Has no default value. Can be a single value or a vector of values.

`r.x2.y` Pearson's correlation between x2 and y. Must be between -1 and 1. Assumed to be the 'moderator' in some functions. Has no default value. Can be a single value or a vector of values.

`r.x3.y` Pearson's correlation between x3 and y. Must be between -1 and 1. Assumed to be the 'moderator' in some functions. Has no default value. Can be a single value or a vector of values.

`r.x1x2.y` Pearson's correlation between the interaction term x1x2 (x1 * x2) and y. Must be between -1 and 1. Has no default value. Can be a single value or a vector of values.

`r.x1x3.y` Pearson's correlation between the interaction term x1x2 (x1 * x3) and y. Must be between -1 and 1. Has no default value. Can be a single value or a vector of values.

`r.x2x3.y` Pearson's correlation between the interaction term x1x2 (x2 * x3) and y. Must be between -1 and 1. Has no default value. Can be a single value or a vector of values.

`r.x1.x2` Pearson's correlation between x1 and x2. Must be between -1 and 1. Has no default value. Can be a single value or a vector of values.

`r.x1.x3` Pearson's correlation between x1 and x3. Must be between -1 and 1. Has no default value. Can be a single value or a vector of values.

`r.x2.x3` Pearson's correlation between x2 and x3. Must be between -1 and 1. Has no default value. Can be a single value or a vector of values.

`rel.x1` Reliability of x1 (e.g. test-retest reliability, ICC, Cronbach's alpha). Default is 1 (perfect reliability). Must be greater than 0 and less than or equal to 1.

`rel.x2` Reliability of x2 (e.g. test-retest reliability, ICC, Cronbach's alpha). Default is 1 (perfect reliability). Must be greater than 0 and less than or equal to 1.

`rel.x3` Reliability of x3 (e.g. test-retest reliability, ICC, Cronbach's alpha). Default is 1 (perfect reliability). Must be greater than 0 and less than or equal to 1.

`rel.y` Reliability of xy (e.g. test-retest reliability, ICC, Cronbach's alpha). Default is 1 (perfect reliability). Must be greater than 0 and less than or equal to 1.

`alpha` The alpha. At what p-value is the interaction deemed significant? Default is 0.05.

`detailed_results`
Default is FALSE. Should detailed results be reported? Returns regression slopes, f2, r2, and the full correlation matrix.

`cl` Number of clusters to use for running simulations in parallel. Default is NULL (i.e. not in parallel). Useful when running several thousand analyses at once.

Value

A data frame containing the power for each unique setting combination.

Examples

```
power_interaction_3way_r2(N=1000,r.x1.y = .1,r.x2.y = .2,r.x3.y = .3,
r.x1x2.y =  .05,r.x1x3.y =  .07,r.x2x3.y =  .09,b.x1x2x3 =0.01,
r.x1.x2 = .2,r.x1.x3 = .4,r.x2.x3 = .3)
```

`power_interaction_r2` *Analytic power analysis for interactions*

Description

Power analysis for interaction models, computed via change in R2. Valid for interactions with continuous, normally distributed, variables.

Usage

```
power_interaction_r2(
  N,
  r.x1.y,
  r.x2.y,
  r.x1x2.y,
  r.x1.x2,
  rel.x1 = 1,
  rel.x2 = 1,
  rel.y = 1,
  alpha = 0.05,
  detailed_results = FALSE
)
```

Arguments

`N` Sample size. Must be a positive integer. Has no default value. Can be a single value or a vector of values.

`r.x1.y` Pearson's correlation between x1 and y. Must be between -1 and 1. Has no default value. Can be a single value or a vector of values.

`r.x2.y` Pearson's correlation between x2 and y. Must be between -1 and 1. Assumed to be the 'moderator' in some functions. Has no default value. Can be a single value or a vector of values.

`r.x1x2.y` Pearson's correlation between the interaction term x1x2 (x1 * x2) and y. Must be between -1 and 1. Has no default value. Can be a single value or a vector of values.

`r.x1.x2` Pearson's correlation between x1 and x2. Must be between -1 and 1. Has no default value. Can be a single value or a vector of values.

`rel.x1` Reliability of x1 (e.g. test-retest reliability, ICC, Cronbach's alpha). Default is 1 (perfect reliability). Must be greater than 0 and less than or equal to 1.

`rel.x2` Reliability of x2 (e.g. test-retest reliability, ICC, Cronbach's alpha). Default is 1 (perfect reliability). Must be greater than 0 and less than or equal to 1.

`rel.y` Reliability of xy (e.g. test-retest reliability, ICC, Cronbach's alpha). Default is 1 (perfect reliability). Must be greater than 0 and less than or equal to 1.

`alpha` The alpha. At what p-value is the interaction deemed significant? Default is 0.05.

detailed_results
: Default is FALSE. Should detailed results be reported?

Value

A data frame containing the power for each unique setting combination.

Examples

```
power_interaction_r2(N=seq(100,300,by=10),r.x1.y=0.2, r.x2.y=.2,r.x1x2.y=0.2,r.x1.x2=.2)
```

---

power_interaction_r2_covs

Analytic interaction power analysis with covariates

Description

Analytic power analysis of an interaction model with covariates. Additional covariate x main effect interaction terms are additionally added.

Usage

```
power_interaction_r2_covs(
  cov.input,
  N,
  alpha = 0.05,
  detailed_results = FALSE,
  cl = NULL
)
```

Arguments

cov.input
: Output of 'power_interaction_r2_covs()'. Variable correlations and reliabilities are set by first modifying this list.

N
: Sample size. Must be a positive integer. Has no default value. Can be a single value or a vector of values.

alpha
: The alpha. At what p-value is the interaction deemed significant? Default is 0.05.

detailed_results
: Default is FALSE. Should detailed results be reported?

cl
: Number of clusters to use for running simulations in parallel. Default is NULL (i.e. not in parallel). Useful when running several thousand analyses at once.

Value

A data frame containing the analytic power for each unique setting combination.

Examples

```
ex1 = generate.interaction.cov.input(c.num=2)
ex1$correlations$r.y.x1x2 = c(0.1,0.2,0.3)
power_interaction_r2_covs(cov.input = ex1,N=100)
```

---

run_pos_power_search *Calculate the point of stability for a two-way interaction estimate*

Description

Calculate the point of stability for a two-way interaction estimate

Usage

```
run_pos_power_search(
  r.x1.y,
  r.x2.y,
  r.x1x2.y = NULL,
  r.x1.x2,
  rel.x1,
  rel.x2,
  rel.y = 1,
  single.N = NULL,
  start.power = 0.8,
  step = NULL,
  n.datasets = 1000,
  lower.bound = NULL,
  upper.bound = NULL,
  cos.width = 0.5,
  pos.percent = 0.8,
  n.cores = 1
)
```

Arguments

`r.x1.y`	Pearson's correlation between x1 and y. Must be between -1 and 1. Has no default value.
`r.x2.y`	Pearson's correlation between x2 and y. Must be between -1 and 1. Assumed to be the 'moderator' in some functions. Has no default value.
`r.x1x2.y`	Pearson's correlation between the interaction term x1x2 (x1 * x2) and y. Must be between -1 and 1. If NULL or unspecified, it is automatically assigned as the average of the main effects divided by 2, considering reliability. The default value is NULL.
`r.x1.x2`	Pearson's correlation between x1 and x2. Must be between -1 and 1. Has no default value.

`rel.x1` Reliability of x1 (e.g. test-retest reliability, ICC, Cronbach’s alpha). Default is 1 (perfect reliability). Must be greater than 0 and less than or equal to 1.

`rel.x2` Reliability of x2 (e.g. test-retest reliability, ICC, Cronbach’s alpha). Default is 1 (perfect reliability). Must be greater than 0 and less than or equal to 1.

`rel.y` Reliability of xy (e.g. test-retest reliability, ICC, Cronbach’s alpha). Default is 1 (perfect reliability). Must be greater than 0 and less than or equal to 1.

`single.N` Numeric or NULL. A specific sample size at which the stability statistics (POS, COS) are computed. A numeric value overrides the search for a POS. Must be a whole number greater than 0 or NULL. Default is NULL.

`start.power` Numeric or NULL. Determines the starting sample size of the search by locating the sample size with power equal to start.power. If it is non-null, it must be greater than 0 and less than 1. Note that a non-null start.power overrides user inputs for the parameters lower.bound and upper.bound so both are NULL. The default is 0.8.

`step` Numeric or NULL. Determines the rounding (degree of precision) in the estimate of the POS. When set to NULL, dynamic adjustments are made with 1/100th the of the interval between the upper.bound and lower.bound on each iteration of the search algorithm. If it is non-null, it must be a whole natural number. Default is NULL.

`n.datasets` Numeric. Number of simulations conducted at each sample size identified by the search process. Larger values result in greater computational demands. Default is 10000. Must be greater than 1.

`lower.bound` Numeric or NULL. The lower bound for the search. It is automatically assigned when set to NULL The user can manually input one. There is an automatic override to NULL if start.power is non-null. The default is NULL.

`upper.bound` Numeric or NULL. The upper bound for the search. If it is null, an upper bound is determined automatically; there is a ceiling of 1e7. Default is NULL.

`cos.width` Numeric. The width of the corridor of stability COS within which we require some number of estimates to fall to identify the POS. Must be greater than 0. Values above 1 are unlikely to be informative. Default is 0.5.

`pos.percent` Numeric. The minimum proportion of simulated estimates we require to fall inside the COS to consider the sample size to produce stable estimates. Default is 0.8.

`n.cores` Numeric. Number of cores to be used in parallel processing. Default is 1.

Value

A data frame containing the variables ’rel.x1’, ’rel.x2’, ’rel.x1x2’, ’rel.y’, ’r.x1x2.y’, ’r.x1.y’, ’r.x2.y’, ’obs.r.x1x2.y’, ’COS_pos.percents’, ’COS_interval’, ’POS_samplesize’, and ’POS_power’ OR a data frame containing the variables ’single.N’, ’rel.x1’, ’rel.x2’, ’rel.x1x2’, ’rel.y’, ’r.x1x2.y’, ’r.x1.y’, ’r.x2.y’, ’obs.r.x1x2.y’, ’COS_pos.percents’, ’within_COS_interval’, ’POS_determined_COS’, ’within_POS_determined_C ’POS_determined_COS.percent’, sign_error_rate’, and ’power’ If the user inputs single.N as NULL, the first dataframe is returned with the search results for the POS. If the user inputs single.N as a numeric whole number greater than 3, the second dataframe is returned with stability information at that specific sample size.

```
COS_pos.percents: A string representing the COS width and target POS percentage.
within_COS_interval: The proportion of estimates within the inputted COS at the sample size single.N.
POS_determined_COS: The COS necessary to achieve the inputted pos.percent at the sample size single.N
within_POS_determined_COS: The proportion of estimates within the POS_determined_COS at the sample s
POS_determined_COS.percent: The percent deviation from the expected value defining POS_determined_C
sign_error_rate: The number of estimates signed incorrectly at the sample size single.N.
power: The statistical power, calculated analytically, at the sample size single.N.
r.x1x2.y_to_be_stable_at_single.N: The minimum interaction effect required for stability at the samp
power_to_be_stable: The statistical power associated with the stable_effect_size at sample size sing
```

Examples

```
## Not run: run_pos_power_search(r.x1.y = 0.2, r.x2.y = 0.2, r.x1x2.y = 0.15, r.x1.x2 = 0.1,
rel.x1 = 0.8, rel.x2 = 0.8,start.power = 0.8, step = NULL, n.datasets = 1000,
lower.bound = NULL, upper.bound = 500, cos.width = 0.5, pos.percent = 0.8)
## End(Not run)
```

---

simple.slopes.3way *See the simple slopes for a 3-way interaction*

Description

Prints or plots the simple slopes for a 3-way interaction

Usage

```
simple.slopes.3way(power.results, row.num = 1, return.plot = FALSE)
```

Arguments

power.results	Data frame of results from power_interaction_3way_r2().
row.num	Which row to show? Can only be a single number. Default is 1.
return.plot	Return a matrix (FALSE, default), or a plot (TRUE)?

Value

A matrix or a ggplot2 object

Examples

```
power_analysis = power_interaction_3way_r2(detailed_results = TRUE,N = c(1000),
r.x1.y = .2,r.x2.y = .3,r.x3.y = .1,r.x1x2.y = .01,r.x1x3.y = .05,r.x2x3.y = .1,
b.x1x2x3 = 0.1,r.x1.x2 = .1,r.x1.x3 = .1,r.x2.x3 = .1,
rel.x1 = 1,rel.x2 = 1,rel.x3 = 1,rel.y = 1 )
simple.slopes.3way(power_analysis)
```

test_interaction *Test interaction*

Description

Test the interaction from a single simulated data set.

Usage

```
test_interaction(
  data,
  alpha = 0.05,
  detailed_results = FALSE,
  q = 2,
  simple = FALSE
)
```

Arguments

data	Simulated data set. Output of 'generate_interaction()'.
alpha	The alpha. At what p-value is the interaction deemed significant? Default is 0.05.
detailed_results	Should results beyond the linear model (change in R2, simple slopes, correlations, and confidence intervals) be returned? Default is FALSE.
q	Simple slopes. How many quantiles should x2 be split into for simple slope testing? Default is 2. Simple slope testing returns the effect-size (slope) of y~x1 for the two most extreme quantiles of x2. If q=3 then the two slopes are y~x1 for the bottom 33% of x2, and the top 33% of x2.
simple	For internal use. Default is FALSE.

Value

Either a named list or a data frame containing the results of the regression y~x1+x2+x1*x2, the pearson's correlation between y, x1,x2, and x1x2, and the slopes of the simple slopes.

Examples

```
dataset <- generate_interaction(N = 250,r.x1.y = 0,r.x2.y = .1,r.x1x2.y = -.2,r.x1.x2 = .3)
test_interaction(data = dataset, alpha=0.05, q=2)
```

Index

compute_adjustment, 2
cor.mat.3way, 3

generate.interaction.cov.input, 4
generate_interaction, 5

name_key, 7
norm2ordinal, 7

plot_interaction, 8
plot_power_curve, 8
plot_simple_slope, 9
power_estimate, 10
power_interaction, 10
power_interaction_3way_r2, 13
power_interaction_r2, 15
power_interaction_r2_covs, 16

run_pos_power_search, 17

simple.slopes.3way, 19

test_interaction, 20