Package 'InteractionPoweR'

August 24, 2022

Title Power Analyses for Interaction Effects in Cross-Sectional Regressions

Version 0.1.1

Description Power analysis for regression models which test the interaction of two independent variables on a single dependent variable. Includes options for continuous, binary, and/or skewed variables, as well as correlated interacting variables. Also includes options to specify variable reliability. 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(). Please cite as: Baranger DAA, Finsaas MC, Goldstein BL, Vize CE, Lynam DR, Olino TM (2022). ``Tutorial: Power analyses for interaction effects in cross-sectional regressions." <doi:10.31234/osf.io/5ptd7>.

Maintainer David Baranger <dbaranger@gmail.com>

URL https://dbaranger.github.io/InteractionPoweR/, https://doi.org/10.31234/osf.io/5ptd7

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

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.1.2

Depends R (>= 3.5.0)

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

NeedsCompilation no

Author David Baranger [aut, cre] (<https://orcid.org/0000-0002-6659-357X>, davidbaranger.com),
Brandon Goldstein [ctb],
Megan Finsaas [ctb],
Thomas Olino [ctb],
Colin Vize [ctb],
Don Lynam [ctb]

Repository CRAN

Date/Publication 2022-08-24 15:12:35 UTC

R topics documented:

`binary.p2skew` *binary.p2skew*

Description

Converts the probability parameter of a binomial distribution to the skew, assuming n=1.

Usage

```
binary.p2skew(p)
```

Arguments

`p`	The binomial probability

Value

Skew

Examples

```
binary.p2skew(p=.5)
```

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,
  skew.x1,
  skew.x2,
  skew.y,
  transform.x1,
  transform.x2,
  transform.y,
  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
skew.x1	Internal use only
skew.x2	Internal use only
skew.y	Internal use only
transform.x1	Internal use only
transform.x2	Internal use only

transform.y	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,
skew.x1 = 0,skew.x2=0,skew.y=0,k.x1 = 0,k.x2=0,k.y=2,transform.x1 = "default",
transform.x2 = "default",transform.y = "binary")
```

---

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,
  skew.x1 = 0,
  skew.x2 = 0,
  skew.y = 0,
  k.x1 = 0,
  k.x2 = 0,
  k.y = 0,
  transform.x1 = "default",
  transform.x2 = "default",
  transform.y = "default",
  adjust.correlations = TRUE,
```

```
  r.x1.y.adjust = NULL,
  r.x2.y.adjust = NULL,
  r.x1x2.y.adjust = NULL,
  r.x1.x2.adjust = NULL,
  tol = 0.005,
  iter = 10
)
```

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.
skew.x1	Skew of x1. Default is 0 (normally distributed).
skew.x2	Skew of x2. Default is 0 (normally distributed).
skew.y	Skew of y. Default is 0 (normally distributed).
k.x1	Number of discrete values for x1. k.x1 = 2 is equivalent to transform.x1 = "binary". Performs best with k<= 5 if variable is skewed. Otherwise, up to k=20. Values less than 2 result in a continuous variable.
k.x2	Number of discrete values for x2. k.x2 = 2 is equivalent to transform.x2 = "binary". Performs best with k<= 5 if variable is skewed. Otherwise, up to k=20. Values less than 2 result in a continuous variable.
k.y	Number of discrete values for y. k.y = 2 is equivalent to transform.y = "binary". Performs best with k<= 5 if variable is skewed. Otherwise, up to k=20. Values less than 2 result in a continuous variable.
transform.x1	Transform x1? Options are "default", "binary", or "gamma". "binary" will cause variable to be binarized - 2 unique values. Default ("default") will pick "gamma" if variable is skewed.
transform.x2	Transform x2? Options are "default", "binary", or "gamma". "binary" will cause variable to be binarized - 2 unique values. Default ("default") will pick "gamma" if variable is skewed.

transform.y	Transform y? Options are "default", "binary", or "gamma". "binary" will cause variable to be binarized - 2 unique values. Default ("default") will pick "gamma"if variable is skewed.
adjust.correlations	If variables are skewed or binary, should correlations be adjusted so that output data has the specified correlation structure? Default is TRUE.
r.x1.y.adjust	Internal use only
r.x2.y.adjust	Internal use only
r.x1x2.y.adjust	Internal use only
r.x1.x2.adjust	Internal use only
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.

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.

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

norm2binary *norm2binary*

Description

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

Usage

```
norm2binary(x, skew)
```

Arguments

x	Input vector
skew	Desired output skew

Value

A binary variable

Examples

```
norm2binary(x = rnorm(n = 100,mean = 0,sd = 1), skew = 1)
```

norm2gamma *norm2gamma*

Description

Transforms a vector with a normal distribution to a gamma distribution.

Usage

```
norm2gamma(x, skew)
```

Arguments

x	Input vector
skew	Desired skew

Value

A vector with a (skewed) gamma distribution

Examples

```
norm2gamma(x = rnorm(n = 100,mean = 0,sd = 1), skew = 1)
```

norm2ordinal *norm2ordinal*

Description

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

Usage

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

Arguments

x	Input vector
skew	Desired output skew
k	Number of discrete values (e.g., 2=binary, 5=ordinal scale)

Value

A ordinal or binary variable

Examples

```
norm2ordinal(x = rnorm(n = 100,mean = 0,sd = 1), skew = 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,
  skew.x1 = 0,
  skew.x2 = 0,
  skew.y = 0,
  k.x1 = 0,
  k.x2 = 0,
  k.y = 0,
  transform.x1 = "default",
  transform.x2 = "default",
  transform.y = "default",
  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
)
```

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.
skew.x1	Skew of x1. Default is 0 (normally distributed).
skew.x2	Skew of x2. Default is 0 (normally distributed).
skew.y	Skew of y. Default is 0 (normally distributed).
k.x1	Number of discrete values for x1. k.x1 = 2 is equivalent to transform.x1 = "binary". Performs best with k<= 5 if variable is skewed. Otherwise, up to k=20. Values less than 2 result in a continuous variable.
k.x2	Number of discrete values for x2. k.x2 = 2 is equivalent to transform.x2 = "binary". Performs best with k<= 5 if variable is skewed. Otherwise, up to k=20. Values less than 2 result in a continuous variable.
k.y	Number of discrete values for y. k.y = 2 is equivalent to transform.y = "binary". Performs best with k<= 5 if variable is skewed. Otherwise, up to k=20. Values less than 2 result in a continuous variable.
transform.x1	Transform x1? Options are "default", "binary", or "gamma". "binary" will cause variable to be binarized - 2 unique values. Default ("default") will pick "gamma" if variables are skewed.
transform.x2	Transform x2? Options are "default", "binary", or "gamma". "binary" will cause variable to be binarized - 2 unique values. Default ("default") will pick "gamma"if variables are skewed.

`transform.y` Transform y? Options are "default", "binary", or "gamma". "binary" will cause variable to be binarized - 2 unique values. Default ("default") will pick "gamma" if variables are skewed.

`adjust.correlations` If variables are skewed 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 skew. Default is 1,000,000

`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.

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_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)
```

---

test_interaction *Test interaction*

Description

Test the interaction from a single simulated data set.

Usage

```
test_interaction(data, alpha = 0.05, 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.
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