Package ‘anomo’

May 7, 2026

Type Package

Title Analysis of Moderation, Statistical Power, and Optimal Design for Studies Detecting Difference and Equivalence

Version 1.3.2

Date 2026-2-4

Description Analysis of moderation (ANOMO) method conceptualizes the difference and equivalence tests as a moderation problem to test the difference and equivalence of two estimates (e.g., two means or two effects).

Language en-US

Depends R (>= 4.0.0), graphics (>= 4.0.0), base(>= 4.0.0)

License GPL-3

Encoding UTF-8

Suggests rmarkdown, knitr, markdown, odr, grDevices

VignetteBuilder rmarkdown, knitr, markdown

RoxygenNote 7.3.3

NeedsCompilation no

Author Zuchao Shen [aut, cre] (ORCID: <`https://orcid.org/0000-0003-3483-0451`>)

Maintainer Zuchao Shen <`zuchao.shen@gmail.com`>

Repository CRAN

Date/Publication 2026-02-04 16:40:02 UTC

Contents

---

mcci *Compute Monte Carlo Confidence Intervals*

Description

Compute Monte Carlo confidence intervals (MCCIs) for the difference and equivalence tests.

Usage

```
mcci(
  d = NULL,
  se = NULL,
  mediation = FALSE,
  n.mcci = 10000,
  sig.level = 0.05,
  sig.adjusted = TRUE,
  two.tailed = TRUE,
  seed = 123,
  eq.bd = NULL,
  xlim = NULL,
  xlab = NULL,
  ylab = NULL,
  dashed.lines = TRUE,
  verbose = TRUE
)
```

Arguments

`d`	The estimated effect(s), it has a length of one, two, and four. (1) When the length is one, it is an estimated main or moderation effect, the MCCI compute the CI for this estimate; (2) When the length is two, they represent two estimated effects. These two estimated effects are main or moderation effects when `mediation` is FALSE, the MCCI compute the CI for the difference of the two estimates; These two estimated effects are the treatment-mediator and mediator-outcome path estimates for a mediation effect when `mediation` is TRUE; the MCCI compute the CI for the mediation effects. (3) When the length is four, they represent the mediation effects in two studies in the following order: the treatment-mediator and mediator-outcome path estimates in studies (groups) 1 and 2.
`se`	The corresponding standard error(s) for parameter `d`.
`mediation`	Logical; `d` and `se` represent parameters for a mediation effect if TRUE; `d` and `se` represent parameters for a main or moderation effect if FALSE; default value is FALSE.
`n.mcci`	The number of draws for the MCCI method. Default is 10,000.
`sig.level`	The significance level. Default is .05.

`sig.adjusted`	Logical; use Bonferroni correction (i.e., dividing the original significance level by the number of tests) if TRUE, otherwise not; default value is TRUE.
`two.tailed`	Logical of two tailed test for difference test. Default is TRUE.
`seed`	Random seed for replication, default is 123.
`eq.bd`	The limit of the equivalence bounds for an equivalence test. Default is the MCCI for the equivalence test. It can be specified in the arguments as eq.bd = a positive number or eq.bd = c(lower bound #, upper bound #).
`xlim`	The limits set for the x-axis in the plot. Default is the MCCI for the difference test. It can be specified in the arguments as xlim = c(lower #, higher #).
`xlab`	The label for the x-axis in the plot. Default is "Differences in Effects".
`ylab`	The label for the y-axis in the plot. Default is NULL.
`dashed.lines`	Logical of whether dashed lines of equivalence bounds and zero should be added in the plot. Default is TRUE.
`verbose`	Logical; print the process if TRUE, otherwise not; default value is TRUE.

Value

The results of moderation analysis and equivalence tests using the MCCI method. It will also provide a plot for the MCCIs.

Examples

```
   library(anomo)
# 1. Compute MCCIs for main or moderation effects-----
   # 1.1. Compute MCCIs for one main or moderation effect from one study
   myci <- mcci(d = .1, se = .02); myci$out
   # 1.2 Compute MCCIs for differences in two main (or moderation) effects
   myci <- mcci(d = c(0.1, 0.15), se = c(.02, 0.01)); myci$out
   # 1.3 Compute MCCIs for differences across five main (or moderation) effects
   myci <- mcci(d = c(0.10, 0.15, 0.20, 0.25, 0.30),
                se = c(0.01, 0.01, 0.02, 0.02, 0.03))
   myci$out

# 2. Compute MCCIs for mediation effects
   # 2.1. Compute MCCIs for an estimated mediation effect
   myci <- mcci(d = c(.1, 0.15), se = c(.02, 0.01), mediation = TRUE)
   myci$out
   # 2.1. Compute MCCIs for differences in two mediation effects
   myci <- mcci(d = c(0.30, 0.50, 0.33, 0.55),
                se = c(0.02, 0.01, 0.02, 0.03), mediation = TRUE)
                myci$out
# 3. Explicitly specify other parameters
   myci <- mcci(d = .05, se = .02, eq.bd = 0.1) # equivalence bounds
   myci <- mcci(d = .05, se = .02, xlim = c(-0.15, 0.15)) # Range of x-axis
```

`od.1.eq` *Optimal sample allocation calculation for equivalence test of two-group means*

Description

The optimal design of single-level experiments detecting equivalence of two-group means is to choose the optimal sample allocation that minimizes the variance of a treatment effect under a fixed budget, which is approximately the optimal sample allocation that maximizes statistical power under a fixed budget. The optimal design parameter is the proportion of individuals to be assigned to treatment (`p`).

Usage

```
od.1.eq(
  p = NULL,
  r12 = NULL,
  c1 = NULL,
  c1t = NULL,
  m = NULL,
  plots = TRUE,
  plim = NULL,
  varlim = NULL,
  plab = NULL,
  varlab = NULL,
  vartitle = NULL,
  verbose = TRUE
)
```

Arguments

`p`	The proportion of individuals to be assigned to treatment.
`r12`	The proportion of outcome variance explained by covariates.
`c1`	The cost of sampling one unit in the control condition.
`c1t`	The cost of sampling one unit in the treated condition.
`m`	Total budget, default value is the total costs of sampling 600 individuals across treatment conditions.
`plots`	Logical, provide variance plots if TRUE, otherwise not; default value is TRUE.
`plim`	The plot range for p, default value is c(0, 1).
`varlim`	The plot range for variance, default value is c(0, 0.05).
`plab`	The plot label for `p`, default value is "Proportion of Units in Treatment: p".
`varlab`	The plot label for variance, default value is "Variance".
`vartitle`	The title of variance plot, default value is NULL.
`verbose`	Logical; print the value of `p` if TRUE, otherwise not; default value is TRUE.

Value

Unconstrained or constrained optimal sample allocation (p). The function also returns function name, design type, and parameters used in the calculation.

Examples

```
# Unconstrained optimal design #---------
  myod <- od.1.eq(r12 = 0.5, c1 = 1, c1t = 50)
  myod$out # output
```

`plot.power.eq` *Plot statistical power curves under a fixed budget across optimal design parameters for equivalence testing*

Description

This function plots statistical power curves (for equivalence testing) under a fixed budget across optimal design parameters.

Usage

```
## S3 method for class 'power.eq'
plot(
  expr = NULL,
  nlim = c(2, 300),
  plim = c(0.01, 0.99),
  Jlim = c(3, 300),
  powerlim = c(0, 1),
  plot.title = NULL,
  m = NULL,
  d = NULL,
  q = 1,
  power = 0.8,
  eq.dis = NULL,
  by = c("n", "p", "J"),
  legend = TRUE,
  nlab = "Level-One Sample Size (n)",
  plab = "Proportion (p)",
  Jlab = "Level-Two Sample Size (J)",
  powerlab = "Statistical Power"
)
```

Arguments

`expr`	Returned objects from an od function (e.g., od.1.eq).

`nlim`	The limits of the level-1 sample size (n) for calculating and plotting power curves.
`plim`	The limits of the proportion to the treated (p) for calculating and plotting power curves.
`Jlim`	The limits of the level-2 sample size (J) for calculating and plotting power curves.
`powerlim`	The power limits for plotting power curves.
`plot.title`	The title of the plot (e.g., plot.title = "Power Curves"). The default is NULL.
`m`	Total budget, default value is the total costs of sampling 600 individuals across treatment conditions.
`d`	The estimated difference in two-group means.
`q`	The number of predictors in the combined linear regression model. Default is 1.
`power`	Statistical power.
`eq.dis`	A positive number to specify the distance from equivalence bounds to `d`. The equivalence bounds are c(-abs(d)-eq.dis, abs(d)+eq.dis).
`by`	Dimensions to plot power curves by the optimal design parameters. The default value is by all optimal design parameters for a type of design. For example, default values are by = "p" for single-level designs, by = c("n", "p") for two-level designs, and by = c("n", "p", "J") for three-level designs.
`legend`	Logical; present plot legend if TRUE. The default is TRUE.
`nlab`	Label for the x-axis when the plot is by the optimal design parameter "n".
`plab`	Label for the x-axis when the plot is by the optimal design parameter "p".
`Jlab`	Label for the x-axis when the plot is by the optimal design parameter "J".
`powerlab`	The label for the statistical power.

Examples

```
# Optimal sample allocation identification
od <- od.1.eq(r12 = 0.5, c1 = 1, c1t = 10)
# plot the power curve
plot.power.eq(expr = od, d = 0.1, eq.dis = 0.1)
```

`power.1.eq` *Statistical Power Analysis for Equivalence Tests of Two-Group Means*

Description

Statistical power analysis for equivalence test of two-group means.

Usage

```
power.1.eq(
  cost.model = FALSE,
  expr = NULL,
  constraint = NULL,
  d = NULL,
  eq.dis = NULL,
  m = NULL,
  c1 = NULL,
  c1t = NULL,
  n = NULL,
  p = NULL,
  q = 1,
  sig.level = 0.05,
  r12 = NULL,
  power = NULL,
  powerlim = NULL,
  nlim = NULL,
  mlim = NULL,
  eq.dislim = NULL,
  verbose = TRUE
)
```

Arguments

`cost.model`	Logical; power analyses accommodating costs and budget (e.g., required budget for desired power, power, minimum detectable eq.dis under a fixed budget) if TRUE. Otherwise, conventional power analysis is performed (e.g., required sample size, power, or minimum detectable eq.dis calculation); default value is FALSE, and it will be changed to TRUE if expr is not NULL.
`expr`	Returned object from function `od.1.eq`; default value is NULL; if `expr` is specified, parameter values of `r12`, `c1`, `c1t`, and `p` used or solved in function `od.1.eq` will be passed to the current function; only the value of `p` that specified or solved in function `od.1.eq` can be overwritten if `constraint` is specified.
`constraint`	Specify the constrained value of `p` in list format to overwrite that from `expr`; default value is NULL.
`d`	The estimated difference in two-group means.
`eq.dis`	A positive number to specify the distance from equivalence bounds to `d`. The equivalence bounds are c(-abs(d)-eq.dis, abs(d)+eq.dis).
`m`	Total budget.
`c1`	The cost of sampling one unit in the control condition.
`c1t`	The cost of sampling one unit in the treated condition.
`n`	The total sample size across groups.
`p`	The proportion of individuals in the intervention group or group 1.
`q`	The number of predictors in the combined linear regression model. Default is 1.

`sig.level`	The significance level. Default is .05.
`r12`	The proportion of variance explained by covariates if any.
`power`	Statistical power.
`powerlim`	The range for solving the root of power (power) numerically, default value is c(1e-10, 1 - 1e-10).
`nlim`	The range for searching the root of sample size (n) numerically, default value is c(10, 10e10).
`mlim`	The range for searching the root of budget (`m`) numerically, default value is the costs sampling `nlim` units across treatment conditions or c(4 * ncost, 10e10 * ncost) with ncost = ((1 - p) * c1 + p * c1t).
`eq.dislim`	The range for solving the root of equivalence difference with the effect size (d) numerically, default value is c(0, 10).
`verbose`	Logical; print the process if TRUE, otherwise not; default value is TRUE.

Value

Required budget (and/or required sample size), statistical power, or minimum detectable eq.dis depending on the specification of parameters. The function also returns the function name, design type, and parameters used in the calculation.

Examples

```
   library(anomo)
# 1. Conventional Power Analyses from Difference Perspectives
# Calculate the required sample size to achieve certain level of power
mysample <- power.1.eq(d = .1, eq.dis = 0.1,  p =.5,
                            r12 = .5, q = 1, power = .8)
mysample$out

# Calculate power provided by a sample size allocation
mypower <- power.1.eq(d = 1, eq.dis = .1, n = 1238, p =.5,
                           r12 = .5, q = 1)
mypower$out

# Calculate the minimum detectable distance a given sample size allocation
# can achieve
myeq.dis <- power.1.eq(d = .1, n = 1238, p =.5,
                           r12 = .5, q = 1, power = .8)
myeq.dis$out

# 2. Power Analyses Using Optimal Sample Allocation
myod <- od.1.eq(r12 = 0.5, c1 = 1, c1t = 10)
budget <- power.1.eq(expr = myod, d = .1, eq.dis = 0.1,
                          q = 1, power = .8)
budget.balanced <- power.1.eq(expr = myod, d = .1, eq.dis = 0.1,
                                   q = 1, power = .8,
                                   constraint = list(p = .50))
(budget.balanced$out$m-budget$out$m)/budget$out$m *100
# 27% more budget required from the balanced design with p = 0.50.
```

`re` *Relative efficiency (RE) calculation*

Description

Calculate the relative efficiency (RE) between two designs using the re function from the R package odr.

Usage

```
re(od, subod, rounded = TRUE, verbose = TRUE)
```

Arguments

`od`	Returned object of first design (e.g., unconstrained optimal design) from function `od.1.eq`.
`subod`	Returned object of second design (e.g., constrained optimal design) from function `od.1.eq`.
`rounded`	Logical; round the values of `p` to two decimal places if TRUE. No rounding if FALSE; default is TRUE.
`verbose`	Logical; print the value of relative efficiency if TRUE, otherwise not; default is TRUE.

Value

Relative efficiency value.

Examples

```
# Unconstrained optimal design #----------
  myod1 <- od.1.eq(r12 = 0.5, c1 = 1, c1t = 20)
# Constrained optimal design with p = .50
  myod2 <- od.1.eq(r12 = 0.5, c1 = 1, c1t = 20, p = .50)
# Relative efficiency (RE)
  myre <- re(od = myod1, subod= myod2)
  myre$re # RE = 0.71
```

Index