Package ‘mc.heterogeneity’

May 20, 2019

Type Package

Title A Monte Carlo Based Heterogeneity Test for Meta-Analysis

Version 0.1.0

Author Han Du <hdu@psych.ucla.edu>, Zijun Ke <keziyun@mail.sysu.edu.cn>, Ge Jiang <gejiang2@illinois.edu>

Maintainer Ge Jiang <gejiang2@illinois.edu>

Description Implements a Monte Carlo Based Heterogeneity Test for standardized mean differences (d), Fisher-transformed Pearson's correlations (r), and natural-logarithm-transformed odds ratio (OR) in Meta-Analysis Studies. Depending on the presence of moderators, this Monte Carlo Based Test can be implemented in the random or mixed-effects model. This package uses rma() function from the R package 'metafor' to obtain parameter estimates and likelihood, so installation of R package 'metafor' is required.

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Depends R (>= 3.1.0)

Imports stats, metafor, utils

Suggests base, HSAUR2, roxygen2, knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2019-05-20 08:50:26 UTC

R topics documented:

mc.d ................................................................. 2
mc.fcor ............................................................ 3
mc.lnOR ............................................................ 5

Index 7
mc.d

Standardized Mean Differences (d): Monte Carlo Based Heterogeneity Test for Between-study Heterogeneity in Random- or Mixed-Effects Model

Description

mc.d returns the Monte Carlo based tests of the residual heterogeneity in random- or mixed-effects model of standardized mean differences (d).

Usage

mc.d(n1, n2, est, model = "random", adjust = FALSE, mods = NULL, nrep = 10^4, p_cut = 0.05, mc.include = FALSE)

Arguments

n1
a vector of sample sizes from group 1 in each of the included studies.
n2
a vector of sample sizes from group 2 in each of the included studies.
est
a vector of unbiased estimates of standardized mean differences.
model
choice of random- or mixed-effects models. Can only be set to "random", or "mixed".
adjust
if biased estimates (i.e., g values) are provided, adjust must be set to TRUE to compensate for small sample bias. By default, adjust is set to FALSE.
mods
optional argument to include one or more moderators in the model. mods is NULL for random-effects model and a dataframe for mixed-effects model. A single moderator can be given as a vector of length k specifying the values of the moderator. Multiple moderators are specified by giving a matrix with k rows and as many columns as there are moderator variables. See rma for more details.
nrep
number of replications used in Monte Carlo Simulations. Default to 10^4.
p_cut
cutoff for p-values, which is the alpha level. Default to 0.05.
mc.include
if true, Monte Carlo simulation results are included in the output (e.g., Monte Carlo critical values).

Details

For standardized mean difference, if the biased estimates (i.e., g values) are provided, adjust=TRUE can be specified to obtain the corresponding unbiased estimates.

This function returns the test statistics as well as their p-value and significances using (1) Q-test, (2) Monte Carlo Based Heterogeneity Test with Maximum Likelihood (ML), and (3) Monte Carlo Based Heterogeneity Test with Restricted Maximum Likelihood (REML).

The results of significances are classified as "sig" or "n.s" based on the cutoff p-value (i.e., alpha level). "sig" means that the between-study heterogeneity is significantly different from zero whereas "n.s" means the between-study heterogeneity is not significantly different from zero. The default alpha level is 0.05.
References


Examples

# A meta-analysis of 18 studies in which the effect of open versus traditional education on students’ self-concept was studied (Hedges et al., 1981).
selfconcept <- mc.heterogeneity::selfconcept
# n1 and n2 are lists of samples sizes in two groups
n1 <- selfconcept$n1
n2 <- selfconcept$n2
# g is a list of biased estimates of standardized mean differences in the meta-analytical study
g <- selfconcept$g
cm <- (1-3/(4*(n1+n2-2)-1)) #correct factor to compensate for small sample bias (Hedges, 1981)
d <- cm*g
## not run
mc.run <- mc.d(n1, n2, est = d, model = 'random', p_cut = 0.05)
# is equivalent to:
mc.run2 <- mc.d(n1, n2, est = g, model = 'random', adjust = TRUE, p_cut = 0.05)
## end (not run)

# A hypothetical meta-analysis of 18 studies with 2 moderators.
hypo_moder <- mc.heterogeneity::hypo_moder
## not run
mc.run3 <- mc.d(n1 = hypo_moder$n1, n2 = hypo_moder$n2, est = hypo_moder$d, model = 'mixed',
mods = cbind(hypo_moder$cov.z1, hypo_moder$cov.z2, hypo_moder$cov.z3), p_cut = 0.05)
## end (not run)

mc.fcor

Fisher-transformed Pearson’s correlation: Monte Carlo Based Heterogeneity Test for Between-study Heterogeneity in Random- or Mixed- Effects Model

Description

mc.fcor returns the Monte Carlo based tests of the residual heterogeneity in random- or mixed-effects model of Pearson’s correlation coefficients transformed with Fisher’s r-to-z transformation (z scores).
Usage

mc.fcor(n, z, model = "random", mods = NULL, nrep = 10^4, p_cut = 0.05, mc.include = FALSE)

Arguments

n  a vector of sample sizes in each of the included studies.

z  a vector of Fisher-transformed Pearson’s correlations.

model  choice of random- or mixed- effects models. Can only be set to "random", or "mixed".

mods  optional argument to include one or more moderators in the model. mods is NULL for random-effects model and a dataframe for mixed-effects model. A single moderator can be given as a vector of length k specifying the values of the moderator. Multiple moderators are specified by giving a matrix with k rows and as many columns as there are moderator variables. See rma for more details.

nrep  number of replications used in Monte Carlo Simulations. Default to 10^4.

p_cut  cutoff for p-values, which is the alpha level. Default to 0.05.

mc.include  if true, Monte Carlo simulation results are included in the output (e.g., Monte Carlo critical values).

Details

This function returns the test statistics as well as their p-value and significances using (1) Q-test, (2) Monte Carlo Based Heterogeneity Test with Maximum Likelihood (ML), and (3) Monte Carlo Based Heterogeneity Test with Restricted Maximum Likelihood (REML).

The results of significances are classified as "sig" or "n.s" based on the cutoff p-value (i.e., alpha level). "sig" means that the between-study heterogeneity is significantly different from zero whereas "n.s" means the between-study heterogeneity is not significantly different from zero. The default alpha level is 0.05.

References


Examples

# A meta-analysis of 13 studies studying the correlation between sensation
# seeking scores and levels of monoamine oxidase (Zuckerman, 1994).
sensation <- mc.heterogeneity::sensation
# n is a list of samples sizes
n <- sensation$n
# Pearson’s correlation
r <- sensation$r
# Fisher’s Transformation
Natural-Logarithm-Transformed Odds Ratio (lnOR): Monte Carlo Based Heterogeneity Test for Between-study Heterogeneity in Random- or Mixed-Effects Model

Description

mc.lnOR returns the Monte Carlo based tests of the residual heterogeneity in random- or mixed-effects model of natural-logarithm-transformed observed odds ratio (lnOR).

Usage

mc.lnOR(n_00, n_01, n_10, n_11, model = "random", mods = NULL, nrep = 10^4, p_cut = 0.05, mc.include = FALSE)

Arguments

- **n_00**: a vector of number of participants who score negatively on both Y1 and Y2 (e.g., mortality cases in the control group).
- **n_01**: a vector of number of participants who score negatively on Y1 and positively on Y2 (e.g., recovery cases in the control group).
- **n_10**: a vector of number of participants who score positively on Y1 and negatively on Y2 (e.g., mortality cases in the experimental group).
- **n_11**: a vector of number of participants who score positively on both Y1 and Y2 (e.g., recovery cases in the experimental group).
- **model**: choice of random- or mixed-effects models. Can only be set to "random", or "mixed".
- **mods**: optional argument to include one or more moderators in the model. mods is NULL for random-effects model and a dataframe for mixed-effects model. A single moderator can be given as a vector of length k specifying the values of the moderator. Multiple moderators are specified by giving a matrix with k rows and as many columns as there are moderator variables. See rma for more details.
- **nrep**: number of replications used in Monte Carlo Simulations. Default to 10^4.
- **p_cut**: cutoff for p-values, which is the alpha level. Default to 0.05.
- **mc.include**: if true, Monte Carlo simulation results are included in the output (e.g., Monte Carlo critical values).
- **lnOR**: a vector of natural-logarithm-transformed odds ratio in the included studies, which is calculated as ln(n11*n00/n01/n10)
Details

For odds ratio, its standard error will be infinite if any one of the four cells in the contingency tables is zero. In this case, Haldane and Anscombe correction is used by adding 0.5 to each cell value (Anscombe, 1956; Haldane, 1940). This function returns the test statistics as well as their p-value and significances using (1) Q-test, (2) Monte Carlo Based Heterogeneity Test with Maximum Likelihood (ML), and (3) Monte Carlo Based Heterogeneity Test with Restricted Maximum Likelihood (REML).

The results of significances are classified as "sig" or "n.s" based on the cutoff p-value (i.e., alpha level). "sig" means that the between-study heterogeneity is significantly different from zero whereas "n.s" means the between-study heterogeneity is not significantly different from zero. The default alpha level is 0.05.

Source


References


Examples

# A meta-analysis consists of 26 studies on nicotine replacement therapy for smoking cessation
library(HSAUR2)
data(smoking)
# Y1: receive treatment; Y2: stop smoking
n_00 <- smoking$tc - smoking$qc  # not receive treatment yet not stop smoking
n_01 <- smoking$qc  # not receive treatment but stop smoking
n_10 <- smoking$tt - smoking$qt  # receive treatment but not stop smoking
n_11 <- smoking$qt  # receive treatment and stop smoking
lnOR <- log(n_11*n_00/n_01/n_10)
## Not run:
mc.run <- mc-lnOR(n_00, n_01, n_10, n_11, model = 'random', p_cut = 0.05)
## End(Not run)
Index

mc.d, 2
mc.fcOR, 3
mc.lnOR, 5
rma, 2, 4, 5