Package ‘moderate.mediation’

July 23, 2025

Version 0.0.11

Date 2025-04-3

Title Causal Moderated Mediation Analysis

Maintainer Xu Qin <`xuqin@pitt.edu`>

Depends R (>= 3.5.0)

Imports stats, mvtnorm, cowplot, distr, graphics, ggplot2, reshape2, doSNOW, scales, foreach, earth

Description Causal moderated mediation analysis using the methods proposed by Qin and Wang (2023) <doi:10.3758/s13428-023-02095-4>. Causal moderated mediation analysis is crucial for investigating how, for whom, and where a treatment is effective by assessing the heterogeneity of mediation mechanism across individuals and contexts. This package enables researchers to estimate and test the conditional and moderated mediation effects, assess their sensitivity to unmeasured pre-treatment confounding, and visualize the results. The package is built based on the quasi-Bayesian Monte Carlo method, because it has relatively better performance at small sample sizes, and its running speed is the fastest. The package is applicable to a treatment of any scale, a binary or continuous mediator, a binary or continuous outcome, and one or more moderators of any scale.

License GPL-2

RoxygenNote 7.3.2

NeedsCompilation no

Author Xu Qin [aut, cre, cph],
Lijuan Wang [ctb]

Repository CRAN

Date/Publication 2025-04-03 23:50:02 UTC

Contents

`modmed` *Causal Moderated Mediation Analysis*

Description

'modmed' is used to fit mediator and outcome models and estimate and test causal effects for causal moderated mediation analysis. It is applicable to a treatment of any scale, a binary or continuous mediator and outcome, one or more moderators of any scale, and a wide range of scenarios of moderated mediation.

Usage

```
modmed(
  data,
  treatment,
  mediator,
  outcome,
  covariates.disc = NULL,
  covariates.cont = NULL,
  moderators.disc = NULL,
  moderators.cont = NULL,
  m.model,
  y.model,
  comp.treatment.value = 1,
  ref.treatment.value = 0,
  comp.mod.disc.values = NULL,
  ref.mod.disc.values = NULL,
  comp.mod.cont.values = NULL,
  ref.mod.cont.values = NULL,
  m.scale = "continuous",
  y.scale = "continuous",
  method = "mc",
  nmc = 1000,
  nboot = 1000,
  conf.level = 0.95,
  seed = NULL,
  object = NULL,
  is.U = FALSE,
  sens.effect = NULL,
  b.m = NULL,
  b.y = NULL,
  plot.effect = NULL
)
```

Arguments

`data`	A data frame containing the variables in the analysis. Users need to impute any missing values in the data before running the function.
`treatment`	A character string indicating the name of the treatment variable. If a treatment is binary, please code it as 0 and 1.
`mediator`	A character string indicating the name of the mediator variable.
`outcome`	A character string indicating the name of the outcome variable.
`covariates.disc`	A name vector of the discrete pretreatment covariates that are not moderators (string). The default is NULL. Users do not need to reformat discrete variables. If treatment is randomized, it should contain confounders of the mediator-outcome relationship. If treatment is not randomized, it should also contain confounders of the treatment-mediator and treatment-outcome relationships. Please do not include moderators here. The main effects of moderators are accounted for by including the moderators in the model of the main model intercept.
`covariates.cont`	A name vector of the continuous pretreatment covariates that are not moderators (string). The default is NULL. If treatment is randomized, it should contain confounders of the mediator-outcome relationship. If treatment is not randomized, it should also contain confounders of the treatment-mediator and treatment-outcome relationships. Please do not include moderators here. The main effects of moderators are accounted for by including the moderators in the model of the main model intercept.
`moderators.disc`	A name vector of all the discrete moderators under study (string). The default is NULL.
`moderators.cont`	A name vector of all the continuous moderators under study (string). The default is NULL.
`m.model`	A list. The name of each element in the list is a predictor in the main mediator model. The names must include intercept, treatment, covariates.disc, and covariates.cont. Each element of the list is a vector of the names of the moderators (string) of the coefficient of the main model predictor as represented by the name of the element. Each moderator specified in moderators.disc and moderators.cont must moderate at least one slope in either the mediator model or the outcome model. The moderators of the intercept must include all the moderators in the mediator model. If a main model coefficient is not moderated, then the corresponding vector should be specified as NULL. The set of moderators in m.model and that in y.model are not necessarily the same. In other words, for a moderator that moderates coefficient(s) in one model, it is possible that it does not occur in the other model. It is also possible that it only has its main effect in the other model, in which case it will be only included in the vector of the intercept in the list of the other model. The union of the two sets of moderators should be contained in the union of moderators.disc, and moderators.cont. If no moderators are specified, only the population average mediation effects are estimated and tested. Check examples for details.

`y.model`	A list. The name of each element in the list is a predictor in the main outcome model. The names must include intercept, treatment, mediator, covariates.disc, and covariates.cont. If the treatment is assumed to interact with the mediator when affecting the outcome, an additional element should be added to y.model, named as "tm". Each element of the list is a vector of the names of the moderators (string) of the coefficient of the main model predictor as represented by the name of the element. Each moderator specified in moderators.disc and moderators.cont must moderate at least one slope in either the main mediator model or the main outcome model. The moderators of the intercept must include all the moderators in the mediator model. If a main model coefficient is not moderated, then the corresponding vector should be specified as NULL. The set of moderators in m.model and that in y.model are not necessarily the same. In other words, for a moderator that moderates coefficient(s) in one model, it is possible that it does not occur in the other model. It is also possible that it only has its main effect in the other model, in which case it will be only included in the vector of the intercept in the list of the other model. The union of the two sets of moderators should be contained in the union of moderators.disc, and moderators.cont. If no moderators are specified, only the population average mediation effects are estimated and tested. Check examples for details.
`comp.treatment.value`	If the total treatment effect for each individual is defined as Y(t) - Y(t'), comp.treatment.value refers to t. The default is 1.
`ref.treatment.value`	If the total treatment effect for each individual is defined as Y(t) - Y(t'), ref.treatment.value refers to t'. The default is 0.
`comp.mod.disc.values`	A vector of compare values of the discrete moderators given which the conditional causal effects are estimated. The default is NULL. The length and order of comp.mod.disc.values should be the same as moderators.disc. If one does not want to condition some moderators on specific values, one may specify their values to be NA. If the discrete moderators take the same compare value, comp.moderator.disc.values can be specified as a single value. If not NULL, results contrasting the causal effects given compare moderator values with those given reference moderator values will be reported.
`ref.mod.disc.values`	A vector of reference values of the discrete moderators given which the conditional causal effects are estimated. The default is NULL. The length and order of ref.mod.disc.values should be the same as moderators.disc. If one does not want to condition some moderators on specific values, one may specify their values to be NA. If the discrete moderators take the same reference value, ref.moderator.disc.values can be specified as a single value. The moderators whose reference values are specified must be included in either the mediator model or the outcome model or both.
`comp.mod.cont.values`	A vector of compare values of the continuous moderators given which the conditional causal effects are estimated. The default is NULL. The length and order of comp.mod.cont.values should be the same as moderators.cont. If one does not want to condition some moderators on specific values, one may specify their

	values to be NA. If the continuous moderators take the same compare value, comp.moderator.cont.values can be specified as a single value. If not NULL, results contrasting the causal effects given compare moderator values with those given reference moderator values will be reported.
`ref.mod.cont.values`	A vector of reference values of the continuous moderators given which the conditional causal effects are estimated. The default is NULL. The length and order of ref.mod.cont.values should be the same as moderators.cont. If one does not want to condition some moderators on specific values, one may specify their values to be NA. If the continuous moderators take the same reference value, ref.moderator.cont.values can be specified as a single value. The moderators whose reference values are specified must be included in either the mediator model or the outcome model or both.
`m.scale`	A character string indicating the scale of the mediator. "continuous" if the mediator is continuous. "binary" if the mediator is binary. Probit link is used in this case. The default is "continuous".
`y.scale`	A character string indicating the scale of the outcome. "continuous" if the outcome is continuous. "binary" if the outcome is binary. Probit link is used in this case. The default is "continuous".
`method`	Estimation and inference method. if 'mc', the Monte Carlo method is used; if 'boot', bootstrap is used. Default is 'mc'. When sample size is relatively small and the mediator or binary is binary, 'boot' is preferred. Otherwise, 'mc' is more recommended, mainly because its running speed is much faster.
`nmc`	Number of simulations involved in the Monte Carlo algorithm. Used if method = 'mc'. The default value is 1000.
`nboot`	Number of bootstrapped samples involved in the bootstrapping algorithm. Used if method = 'boot'. The default value is 1000.
`conf.level`	Level of the returned two-sided confidence intervals. The default is 0.95, which returns the 2.5 and 97.5 percentiles.
`seed`	The seed for the random number generator. The default value is NULL.
`object`	Output from the original analysis only if is.U = TRUE. NULL in the original analysis. The default is NULL.
`is.U`	A logical value. 'FALSE' in the original analysis. 'TRUE' in the sensitivity analysis which adjusts for a simulated unmeasured pretreatment confounder U. The default is FALSE.
`sens.effect`	The default is NULL. It is used only when modmed.sens is run.
`b.m`	Sensitivity parameter that represents the conditional association between the unobserved pretreatment confounder and the mediator. It is NULL in the original analysis. The default is NULL. It is used only when modmed.sens is run.
`b.y`	Sensitivity parameter that represents the conditional association between the unobserved pretreatment confounder and the outcome. It is NULL in the original analysis. The default is NULL. It is used only when modmed.sens is run.
`plot.effect`	The effect to be plotted. The default is NULL. It is used only when modmed.plot is run.

Value

The output of the function modmed contains a list containing the following elements. The function 'summary' or 'summary_modmed' can be used to obtain a table of the results.

`effects`	Estimation results of the causal effects. "TE" indicates the total treatment effect. "TIE" indicates the total indirect effect, "PDE" indicates the pure direct effect, and "INT" indicates the natural treatment-by-mediator interaction effect. "TE.ref", "TIE.ref", "PDE.ref", and "INT.ref" indicate the corresponding effects when the moderators take the reference values. "TE.dif", TIE.dif", "PDE.dif", and "INT.dif" each indicates the difference in the corresponding effect between the compare levels and the reference levels of the moderators.
`m.model`	Estimation results of the mediator model
`y.model`	Estimation results of the outcome model
`results`	1000 draws from the sampling distribution of the causal effects.
`args`	A list of the arguments specified in the function of modmed except for the default ones
`args.full`	The full list of all the arugments specified in the function of modmed
`l.m`	The mediator model
`l.y`	The outcome model
`formula.m`	The formula for the mediator model
`formula.y`	The formula for the outcome model
`m.predictors`	All the original predictors in the main mediator model
`y.predictors`	All the original predictors in the main outcome model
`m.moderators`	All the original moderators in the mediator model
`y.moderators`	All the original moderators in the outcome model
`m.predictors.new`	All the predictors in the main mediator model, in which the discrete confounders are transformed into multiple indicator codes.
`y.predictors.new`	All the predictors in the main outcome model, in which the discrete confounders are transformed into multiple indicator codes.
`m.moderators.new`	All the moderators in the mediator model, in which the discrete confounders are transformed into multiple indicator codes.
`y.moderators.new`	All the moderators in the outcome model, in which the discrete confounders are transformed into multiple indicator codes.
`predict.m.data.ref`	Data for prediction of the conditional potential mediator at the reference level of the moderators.
`predict.m.data.comp`	Data for prediction of the conditional potential mediator at the compare level of the moderators.

`predict.m.data`	Data for prediction of the marginal potential mediator
`predict.y.data.ref`	Data for prediction of the conditional potential outcome at the reference level of the moderators.
`predict.y.data.comp`	Data for prediction of the conditional potential outcome at the compare level of the moderators.
`predict.y.data`	Data for prediction of the marginal potential outcome
`data`	The data used in the analysis

Author(s)

Xu Qin and Lijuan Wang

References

Qin, X., & Wang, L. (2023). Causal moderated mediation analysis: Methods and software

Examples

```
data(newws)
modmed.results = modmed(data = newws, treatment = "treat",
    mediator = "emp", outcome = "depression", covariates.disc = NULL,
    covariates.cont = NULL, moderators.disc = "CHCNT", moderators.cont = NULL,
    m.model = list(intercept = "CHCNT", treatment = "CHCNT"),
    y.model = list(intercept = "CHCNT", treatment = "CHCNT",
        mediator = "CHCNT", tm = "CHCNT"), comp.mod.disc.values = 3,
    ref.mod.disc.values = 2, comp.mod.cont.values = NULL, ref.mod.cont.values = NULL,
    m.scale = "binary", y.scale = "continuous", seed = 1)

modmed.results = modmed(data = newws, treatment = "treat",
    mediator = "emp", outcome = "depression", covariates.disc = c("emp_prior",
        "nevmar", "hispanic", "nohsdip"), covariates.cont = c("workpref",
        "attitude", "depress_prior"), moderators.disc = "CHCNT",
    moderators.cont = "ADCPC", m.model = list(intercept = c("ADCPC",
        "CHCNT"), treatment = c("ADCPC", "CHCNT"), emp_prior = NULL,
        nevmar = NULL, hispanic = NULL, nohsdip = NULL,
        workpref = NULL, attitude = NULL, depress_prior = NULL),
    y.model = list(intercept = c("ADCPC", "CHCNT"), treatment = c("ADCPC",
        "CHCNT"), mediator = c("ADCPC", "CHCNT"), tm = c("ADCPC",
        "CHCNT"), emp_prior = NULL, nevmar = NULL, hispanic = NULL,
        nohsdip = NULL, workpref = NULL, attitude = NULL,
        depress_prior = NULL), comp.mod.disc.values = 3,
    ref.mod.disc.values = 2, comp.mod.cont.values = 5050,
    ref.mod.cont.values = 5050, m.scale = "binary", y.scale = "continuous",
    seed = 1)
summary(modmed.results)
```

`modmed.plot` *Visual Representation of the Causal Moderated Mediation Analysis Results*

Description

'modmed.plot' is used to visualize results from `modmed` function. This applies only if moderators.disc or moderators.cont is not NULL. The plot consists of two parts. The top represents the sampling distribution of the specified causal effect as a function of the specified moderator within the given levels of the other moderators. The bottom represents the distribution of the specified moderator on the x axis.

Usage

```
modmed.plot(
  object,
  effect,
  moderator,
  other.mod.disc.values = NULL,
  other.mod.cont.values = NULL,
  is.dist.moderator = FALSE,
  probs = c(0.1, 0.25, 0.5, 0.75, 0.9),
  ncore = 2
)
```

Arguments

`object`	Output from the `modmed` function.
`effect`	A character string indicating which causal effect to be plotted. effect can be specified as "TE", "TIE", "PIE", "PDE", "TDE", "INT", "TE.ref", "TIE.ref", "PIE.ref", "PDE.ref", "TDE.ref", "INT.ref", "TE.dif", "TIE.dif", "PIE.dif", "PDE.dif", "TDE.dif", or "INT.dif".
`moderator`	A character string indicating which moderator to be plotted. It must be one of the moderators specified in the function of `modmed`.
`other.mod.disc.values`	A vector of values of the other discrete moderators given which the conditional effect at each value of the specified moderator is estimated. The order of other.mod.disc.values should be the same as moderators.disc specified in the function of `modmed`, with the specified moderator removed if the specified moderator is discrete. If one does not want to condition some moderators on specific values, one may specify their values to be NA. NULL if there are no other discrete moderators.
`other.mod.cont.values`	A vector of values of the other continuous moderators given which the conditional effect at each value of the specified moderator is estimated. The order of other.mod.cont.values should be the same as moderators.cont specified in the

	function of `modmed`, with the specified moderator removedif the specified moderator is continuous. If one does not want to condition some moderators on specific values, one may specify their values to be NA. NULL if there are no other continuous moderators.
`is.dist.moderator`	A logical value. "TRUE" if distribution of the specified moderator is to be plotted at the bottom, and "FALSE" if not. The default is "FALSE".
`probs`	A vector of percentiles to be plotted on the distribution of the moderator if the moderator is continuous. NULL if is.dist.moderator = FALSE. The default is c(0.1, 0.25, 0.5, 0.75, 0.9).
`ncore`	The number of cores for parallel computing. The default is the number of CPU cores on the current host minus 1. One core is saved for users to run other programs on the computer while running the R function.

Value

`modmed` returns causal moderated mediation analysis results. The `plot.modmed` function plots the results.

Author(s)

Xu Qin and Lijuan Wang

References

Qin, X., & Wang, L. (2023). Causal moderated mediation analysis: Methods and software

Examples

```
data(newws)
modmed.results = modmed(data = newws, treatment = "treat", mediator = "emp",
    outcome = "depression", covariates.disc = c("emp_prior", "nevmar",
        "hispanic", "nohsdip"), covariates.cont = c("workpref", "attitude",
        "depress_prior"), moderators.disc = "CHCNT", moderators.cont = "ADCPC",
    m.model = list(intercept = c("ADCPC", "CHCNT"), treatment = c("ADCPC",
        "CHCNT"), emp_prior = NULL, nevmar = NULL, hispanic = NULL,
        nohsdip = NULL, workpref = NULL, attitude = NULL, depress_prior = NULL),
    y.model = list(intercept = c("ADCPC", "CHCNT"), treatment = c("ADCPC",
        "CHCNT"), mediator = c("ADCPC", "CHCNT"), tm = c("ADCPC",
        "CHCNT"), emp_prior = NULL, nevmar = NULL, hispanic = NULL,
        nohsdip = NULL, workpref = NULL, attitude = NULL, depress_prior = NULL),
    comp.mod.disc.values = 3, ref.mod.disc.values = 2, comp.mod.cont.values = 5050,
    ref.mod.cont.values = 5050, m.scale = "binary", y.scale = "continuous",
    seed = 1)
modmed.plot(modmed.results, effect = "TIE", moderator = "ADCPC",
    other.mod.disc.values = 1, is.dist.moderator = TRUE, ncore = 1)
```

`modmed.sens` *Simulation-Based Sensitivity Analysis Table for Causal Moderated Mediation Analysis*

Description

modmed.sens' is used to evaluate the sensitivity of the estimated causal effects obtained from `modmed` function to potential violations of the ignorability assumptions from the frequentist perspective. It estimates the causal effects after adjusting for an unmeasured pretreatment confounder, U, with a specified degree of confounding. In a randomized experiment, the degree of confounding is evaluated via two sensitivity parameters, the coefficient of U in the mediator model and that in the outcome model, given the specified prior distribution of U. When the treatment is not randomized, an additional sensitivity parameter is introduced – the coefficient of U in the treatment model. The treatment, mediator, outcome, and unmeasured pretreatment confounder could be either binary or continuous.

Usage

```
modmed.sens(
  object,
 sens.effect = rownames(object$effects)[-which(rownames(object$effects) %in% c("TE",
    "TE.ref", "TE.dif"))],
  range.b.m = NULL,
  range.b.y = NULL,
  grid.b.m = 10,
  grid.b.y = 10,
  U.scale = "binary",
  p.u = 0.5,
  sigma.u = 1,
  t.rand = TRUE,
  t.model = NULL,
  t.scale = "binary",
  b.t = NULL,
  iter = 10,
  nsim = 5,
  ncore = 2
)
```

Arguments

`object`	Output from the `modmed` function.
`sens.effect`	A vector of effects whose sensitivity will be assessed (string). It is a vector of all the population average, conditional, and moderated mediation effects in the output of modmed function by default (i.e., if sens.effect is not specified). It can also be specified as a subvector of the default. For example, if a researcher is mainly interested in the sensitivity of moderated indirect effects, sens.effect can be specified as c("TIE.dif", "PIE.dif").

`range.b.m`	The range of the sensitivity parameter that represents the slope of the unmeasured pretreatment confounder in the standardized mediator model, in which both the independent and dependent variables are standardized. If the dependent variable is binary, its latent index is standardized instead. E.g., it can be specified as c(-2, 2). If NULL, the upper bound of the range is determined by 2 times the maximum magnitude of the standardized conditional effect of the existing pretreatment confounders with the mediator. The lower bound is the negative of the upper bound. The default is NULL.
`range.b.y`	The range of the sensitivity parameter that represents the slope of the unmeasured pretreatment confounder in the standardized outcome model, in which both the independent and dependent variables are standardized. If the dependent variable is binary, its latent index is standardized instead. E.g., it can be specified as c(-2, 2). If NULL, the upper bound of the range is determined by 2 times the maximum magnitude of the standardized conditional effect of the existing pretreatment confounders with the outcome. The lower bound is the negative of the upper bound. The default is NULL.
`grid.b.m`	The horizontal dimension of the grid. The default is 10. Increase the number for more smooth curves.
`grid.b.y`	The vertical dimension of the grid. The default is 10. Increase the number for more smooth curves.
`U.scale`	The scale of the unobserved pretreatment confounder (string). Can be "continuous" or "binary". The default is "binary".
`p.u`	This needs to be specified only if U.scale = "binary". The prior probability of the unobserved pretreatment confounder if it is binary. The default is 0.5.
`sigma.u`	This needs to be specified only if U.scale = "continuous". The standard deviation of the prior distribution of the unobserved pretreatment confounder if it is continuous. The default is 1.
`t.rand`	TRUE if the treatment is randomized, and FALSE if not. The default is TRUE.
`t.model`	A list. The default is NULL. This needs to be specified only if t.rand = FALSE because a treatment model is required for the derivation of the conditional distribution of the unmeasured pretreatment confounder. The names of the elements in t.model must include intercept and the pretreatment covariates that predict the treatment and are contained in the union of covariates.disc, and covariates.cont. Each element of the list is a vector of the names of the moderators (string) of the coefficient of the main model predictor as represented by the name of the element. The moderators of the intercept must include all the moderators in the treatment model. If a main model coefficient is not moderated, then the corresponding vector should be specified as NULL. The moderators should be contained in the union of moderators.disc, and moderators.cont. A moderator in the mediator and outcome models does not necessarily moderate any coefficient in the treatment model. If it only has its main effect in the treatment model, it will be included in the vector of the intercept in t.model.
`t.scale`	The scale of the treatment (string). Can be "continuous" or "binary". The default is "binary".
`b.t`	This needs to be specified only if t.rand = FALSE. The value of the sensitivity parameter that represents the slope of the unmeasured pretreatment confounder

	in the standardized treatment model, in which both the independent and dependent variables are standardized. Please make sure that 1 - b.t^2 is not negative. If the dependent variable is binary, its latent index is standardized instead. When t.rand = TRUE, b.t is NULL, because when the treatment is randomized, there is no need to specify this sensitivity parameter. Hence, the default of b.t is NULL.
`iter`	The number of iterations in the stochastic EM algorithm for generating the unobserved pretreatment confounder. The default is 10.
`nsim`	The number of random draws of the unobserved pretreatment confounder in each cell. The default is 5. Increase the number for more smooth curves.
`ncore`	The number of cores for parallel computing. The default is the number of CPU cores on the current host minus 1. One core is saved for users to run other programs on the computer while running the R function.

Value

A list containing

`X.coef.plot`	the sopes of the observed pretreatment confounders in the standardized mediator and outcome models (which are used to calibrate the strength of the sensitivity parameters in the sensitivity plots).
`b.t`	same as specified in the `modmed.sens` function if b.t is not NULL.
`range.b.m`, `range.b.y`	the value range of each sensitivity parameter.
`b.m.all`, `b.y.all`	all the specific values of the sensitivity parameters for constructing the grids.
`results.new`	tables for the causal effect estimates, standard errors, and p values in all the grids, after the adjustment of the unobserved pretreatment confounder.

Author(s)

Xu Qin and Fan Yang

References

Qin, X., & Yang, F. (2020). Simulation-Based Sensitivity Analysis for Causal Mediation Studies.

Examples

```
data(newws)
modmed.results = modmed(data = newws, treatment = "treat", mediator = "emp",
    outcome = "depression", covariates.disc = c("emp_prior", "nevmar",
        "hispanic", "nohsdip"), covariates.cont = c("workpref", "attitude",
        "depress_prior"), moderators.disc = "CHCNT", moderators.cont = "ADCPC",
    m.model = list(intercept = c("ADCPC", "CHCNT"), treatment = c("ADCPC",
        "CHCNT"), emp_prior = NULL, nevmar = NULL, hispanic = NULL,
        nohsdip = NULL, workpref = NULL, attitude = NULL, depress_prior = NULL),
    y.model = list(intercept = c("ADCPC", "CHCNT"), treatment = c("ADCPC",
        "CHCNT"), mediator = c("ADCPC", "CHCNT"), tm = c("ADCPC",
        "CHCNT"), emp_prior = NULL, nevmar = NULL, hispanic = NULL,
```

```
        nohsdip = NULL, workpref = NULL, attitude = NULL, depress_prior = NULL),
    comp.mod.disc.values = 3, ref.mod.disc.values = 2, comp.mod.cont.values = 5050,
    ref.mod.cont.values = 5050, m.scale = "binary", y.scale = "continuous")
sens.results = modmed.sens(modmed.results, sens.effect = "TIE.ref",
    U.scale = "binary", grid.b.m = 2, grid.b.y = 2, iter = 2, nsim = 2,
    ncore = 1)
sens.results = modmed.sens(modmed.results, sens.effect = "TIE.ref",
    U.scale = "binary", t.model = list(intercept = c("ADCPC", "CHCNT"),
        emp_prior = "ADCPC", nevmar = "CHCNT", hispanic = NULL, nohsdip = NULL,
        workpref = NULL, attitude = NULL, depress_prior = NULL), grid.b.m = 2,
    grid.b.y = 2, iter = 2, nsim = 2, ncore = 1)
```

`newws` *NEWWS Riverside data*

Description

The National Evaluation of Welfare-to-Work Strategies (NEWWS) was conducted before the nationwide welfare reform in the mid-1990s. Participants of the study consist of applicants to the Aid to Families with Dependent Children (AFDC) program and current AFDC recipients who were not working for 30 or more hours per week. They were randomly assigned to the labor force attachment (LFA) program, which aimed at moving low-income parents from welfare to work by providing employment-focused incentives and services, and the control group, which received aid from AFDC without requirement for working. The data is from the Riverside sample. It includes 694 participants with preschool-age children, 208 randomly assigned to the LFA program and 486 randomly assigned to the control group. The National Evaluation of Welfare-to-Work Strategies (NEWWS) Riverside study.

Value

A list containing

`treat`	Treatment is a binary variable, taking the value of 1 if a participant was assigned to LFA and 0 otherwise
`emp`	Mediator is a binary variable, taking the value of 1 if one was ever employed during the two-year period after randomization and 0 if not.
`depression`	Outcome, maternal depression at the end of the second year after randomization, is a summary score of 12 items measuring depressive symptoms during the past week on a 0-3 scale.
`nevmar`	1 if never married and 0 otherwise.
`emp_prior`	1 if employed and 0 otherwise.
`hispanic`	1 if Hispanic and 0 otherwise.
`CHCNT`	1 if had 1 child, 2 if had 2 children, and 3 if had 3 or more children before randomization.
`ADCPC`	welfare amount in the year before randomization

`attitude`	a composite score of two attitude items - so many family problems that I cannot work at a full time or part time job; so much to do during the day that I cannot go to a school or job training program - measured on the scale of 1-4.
`depress_prior`	a composite score of three depressive symptom items - sad, depressed, blues, and lonely - in the week before randomization measured on the scale of 1-4.
`workpref`	one's level of preference for taking care of family full time than working on the scale of 1-4.
`nohsdip`	1 if had never obtained a high school diploma or a General Educational Development certificate and 0 otherwise.

`sens.plot` *Simulation-Based Sensitivity Analysis Plot for Causal Moderated Mediation Analysis*

Description

Simulation-Based Sensitivity Analysis Plot for Causal Moderated Mediation Analysis

Usage

```
sens.plot(object, sens.results, effect)
```

Arguments

`object`	Output from the `modmed` function.
`sens.results`	An output from the `modmed.sens` function.
`effect`	The name of the effect whose sensitivity analysis results are to be plotted (string). Only one effect is plotted at a time. It can be specified as "TIE", "PIE", "PDE", "TDE", "INT", "TIE.ref", "PIE.ref", "PDE.ref", "TDE.ref", "INT.ref", "TIE.dif", "PIE.dif", "PDE.dif", "TDE.dif", or "INT.dif". It must be included in sens.effect when running the `modmed.sens` function.

Value

Sensitivity analysis plots for the causal effects in the causal moderated mediation analysis.

Author(s)

Xu Qin and Fan Yang

References

Qin, X., & Yang, F. (2020). Simulation-Based Sensitivity Analysis for Causal Mediation Studies.

Examples

```
data(newws)
modmed.results = modmed(data = newws, treatment = "treat", mediator = "emp",
    outcome = "depression", covariates.disc = c("emp_prior", "nevmar",
        "hispanic", "nohsdip"), covariates.cont = c("workpref", "attitude",
        "depress_prior"), moderators.disc = "CHCNT", moderators.cont = "ADCPC",
    m.model = list(intercept = c("ADCPC", "CHCNT"), treatment = c("ADCPC",
        "CHCNT"), emp_prior = NULL, nevmar = NULL, hispanic = NULL,
        nohsdip = NULL, workpref = NULL, attitude = NULL, depress_prior = NULL),
    y.model = list(intercept = c("ADCPC", "CHCNT"), treatment = c("ADCPC",
        "CHCNT"), mediator = c("ADCPC", "CHCNT"), tm = c("ADCPC",
        "CHCNT"), emp_prior = NULL, nevmar = NULL, hispanic = NULL,
        nohsdip = NULL, workpref = NULL, attitude = NULL, depress_prior = NULL),
    comp.mod.disc.values = 3, ref.mod.disc.values = 2, comp.mod.cont.values = 5050,
    ref.mod.cont.values = 5050, m.scale = "binary", y.scale = "continuous",
    seed = 1)
sens.results = modmed.sens(modmed.results, sens.effect = "TIE.ref",
    U.scale = "binary", grid.b.m = 2, grid.b.y = 2, iter = 2, nsim = 2,
    ncore = 1)
sens.plot(modmed.results, sens.results, "TIE.ref")
```

`summary_modmed` *Summarizing Output for Causal Moderated Mediation Analysis*

Description

'summary_modmed' is used to report from causal moderated mediation analysis. Alternatively, 'summary' can also be used to generate the same output.

Usage

```
summary_modmed(object, ...)
```

Arguments

`object`	output from `modmed` function
`...`	additional arguments affecting the summary produced.

Value

`modmed` returns causal moderated mediation analysis results. The `summary_modmed` function provides summary tables of the results.

Author(s)

Xu Qin and Lijuan Wang

References

Qin, X., & Wang, L. (2023). Causal moderated mediation analysis: Methods and software

Examples

```
data(newws)
modmed.results = modmed(data = newws, treatment = "treat", mediator = "emp",
    outcome = "depression", covariates.disc = c("emp_prior", "nevmar",
        "hispanic", "nohsdip"), covariates.cont = c("workpref", "attitude",
        "depress_prior"), moderators.disc = "CHCNT", moderators.cont = "ADCPC",
    m.model = list(intercept = c("ADCPC", "CHCNT"), treatment = c("ADCPC",
        "CHCNT"), emp_prior = NULL, nevmar = NULL, hispanic = NULL,
        nohsdip = NULL, workpref = NULL, attitude = NULL, depress_prior = NULL),
    y.model = list(intercept = c("ADCPC", "CHCNT"), treatment = c("ADCPC",
        "CHCNT"), mediator = c("ADCPC", "CHCNT"), tm = c("ADCPC",
        "CHCNT"), emp_prior = NULL, nevmar = NULL, hispanic = NULL,
        nohsdip = NULL, workpref = NULL, attitude = NULL, depress_prior = NULL),
    comp.mod.disc.values = 3, ref.mod.disc.values = 2, comp.mod.cont.values = 5050,
    ref.mod.cont.values = 5050, m.scale = "binary", y.scale = "continuous",
    seed = 1)
summary_modmed(modmed.results)
summary(modmed.results)
```

Index