

Package ‘mediation’

September 18, 2009

Version 2.1

Date 2009-09-14

Title R Package for Causal Mediation Analysis

Author Luke Keele <keele.4@polisci.osu.edu>, Dustin Tingley <dtingley@princeton.edu>, Teppei Yamamoto <tyamamot@princeton.edu>, Kosuke Imai <kimai@princeton.edu>

Maintainer Luke Keele <keele.4@polisci.osu.edu>

Depends R (>= 2.7), MASS, Matrix

Suggests mgcv, quantreg

Description mediation is a publicly available R package that allows both parametric and nonparametric causal mediation analysis. It implements the methods and suggestions in Imai, Keele, and Yamamoto (2008) and Imai, Keele, Tingley (2009). In addition to the estimation of causal mediation effects, the software also allows researchers to conduct sensitivity analysis for certain parametric models.

LazyLoad yes

LazyData yes

License GPL (>= 2)

Repository CRAN

Date/Publication 2009-09-18 20:25:25

R topics documented:

jobs	2
mediate	2
medsens	5
plot.medsens	8
summary.mediate	10
summary.medsens	12

Index 14

jobs

JOB II data

Description

Job Search Intervention Study (JOBS II). JOBS II is a randomized field experiment that investigates the efficacy of a job training intervention on unemployed workers. The program is designed to not only increase reemployment among the unemployed but also enhance the mental health of the job seekers. In the JOBS II field experiment, 1,801 unemployed workers received a pre-screening questionnaire and were then randomly assigned to treatment and control groups. Those in the treatment group participated in job-skills workshops. In the workshops, respondents learned job-search skills and coping strategies for dealing with setbacks in the job-search process. Those in the control condition received a booklet describing job-search tips. In follow-up interviews, the two key outcome variables were measured; a continuous measure of depressive symptoms based on the Hopkins Symptom Checklist, and a binary variable, representing whether the respondent had become employed.

Usage

data

Format

A data matrix containing no missing values and hence is provided only for illustrative purposes and not for inference about program efficacy.

Source

The complete JOBS II data is available from the data archives at www.icpsr.umich.edu/

References

Vinokur, A. and Schul, Y. (1997). Mastery and inoculation against setbacks as active ingredients in the jobs intervention for the unemployed. *Journal of Consulting and Clinical Psychology* 65, 5.

mediate

Causal Mediation Analysis

Description

Function to estimate causal mediation effects.

Usage

```
mediate(model.m, model.y, sims=1000, boot=FALSE, INT=FALSE, treat="treat.name",
mediator="med.name", control=NULL)
```

Arguments

<code>model.m</code>	R model object for mediator. Can be of class <code>lm</code> , <code>polr</code> , <code>glm</code> , or <code>gam</code> .
<code>model.y</code>	R model object for outcome. Can be of class <code>lm</code> , <code>glm</code> , <code>gam</code> , or <code>rq</code> .
<code>sims</code>	Number of Monte Carlo draws for nonparametric bootstrap or quasi-Bayesian approximation.
<code>boot</code>	If <code>FALSE</code> a quasi-Bayesian approximation is used for confidence interval, if <code>TRUE</code> nonparametric bootstrap will be used.
<code>INT</code>	If <code>TRUE</code> this indicates that treatment is interacted with mediator in <code>model.y</code> object. Default is <code>FALSE</code> .
<code>treat</code>	Name of binary treatment indicator.
<code>mediator</code>	Name of mediator variable.
<code>control</code>	Name of binary treatment indicator for control. Only necessary for <code>gam()</code> object with interaction.

Details

This is the workhorse function for estimating mediation effects for a variety of data types. For a continuous mediator variable and a continuous outcome variable, the results will be identical to the usual Baron and Kenny method. The function can, however, accomodate other data types including binary outcomes and mediators and discrete mediators. Continuous variables can also be modeled nonparametrically or semiparametrically.

In fitting the `model.y` object with an interaction between treatment and mediator, the user must input the interaction in the format `treatment:mediator`. Furthermore, the user must make sure to set the `INT` option to `TRUE` since the current function does not sense the existence of the interaction term automatically.

Users should note that use of the nonparametric bootstrap requires several minutes of computing time.

Value

`mediate` returns an object of class `"mediate"`. The function `summary` is used to obtain a table of the results. The object `mediate` is a list that contains the following components. Some of these elements are not available depending on whether `INT` is specified as `TRUE` or `FALSE` by the user.

<code>d0</code>	point estimate for mediation effect under control.
<code>d1</code>	point estimate for mediation effect under treatment.
<code>d0.ci</code>	confidence interval for mediation effect under control. For all confidence intervals, the confidence level is set to 95 percent.

<code>d1.ci</code>	confidence interval for mediation effect under control.
<code>tau.coef</code>	point estimate for total effect.
<code>tau.ci</code>	confidence interval for total effect.
<code>z0</code>	point estimate for direct effect under control.
<code>z1</code>	point estimate for direct effect under treatment.
<code>z0.ci</code>	confidence interval for direct effect under control.
<code>z1.ci</code>	confidence interval for direct effect under control.
<code>pct.coef</code>	percentage of total effect due to mediation.
<code>pct.ci</code>	confidence interval for percentage of total effect due to mediation.
<code>boot</code>	the boot argument used.
<code>INT</code>	the INT argument used.
<code>treat</code>	name of the 'treat' variable used.
<code>mediator</code>	name of the 'mediator' variable used.
<code>call.m</code>	call of the mediator model used.
<code>call.y</code>	call of the outcome model used.
<code>env.m</code>	environment in which the mediator model was fitted.
<code>env.y</code>	environment in which the outcome model was fitted.

Warning

These functions assume that all missing values have been removed from the data set. This can be done using the `na.omit()` command before the outcome and mediation models are fitted.

Author(s)

Luke Keele, Ohio State University, keele.4@osu.edu , Dustin Tingley, Princeton University, dtingley@princeton.edu, Teppei Yamamoto, Princeton University, tyamamot@princeton.edu, Kosuke Imai, Princeton University, kimai@princeton.edu

References

Imai, Kosuke, Luke Keele and Dustin Tingley (2009) A General Approach to Causal Mediation Analysis. Imai, Kosuke, Luke Keele and Teppei Yamamoto (2009) Identification, Inference, and Sensitivity Analysis for Causal Mediation Effects. Imai, Kosuke, Luke Keele, Dustin Tingley, and Teppei Yamamoto. (2009) "Causal Mediation Analysis Using R" in Advances in Social Science Research Using R, ed. H. D. Vinod New York: Springer. Forthcoming.

See Also

See also [medsens](#)

Examples

```
#Example with JOBS II Field experiment

##**For illustration purposes simulations set to low number**

data(jobs)

#####
#Continuous Outcome and Continuous Mediator
#####

b <- lm(job_seek ~ treat + econ_hard + sex + age, data=jobs)
c <- lm(depress2 ~ treat + job_seek + depress1 + econ_hard + sex + age, data=jobs)

#Calculates quantities using quasi-Bayesian approximation
#The general format of the function is to record two model objects
#The first for the mediator the second for the outcome variable
continuous <- mediate(b, c, sims=10, treat="treat", mediator="job_seek")
summary(continuous)

#Calculates quantities using the non-parametric bootstrap - This takes several minutes
continuous_boot <- mediate(b, c, boot=TRUE, sims=10, treat="treat", mediator="job_seek")
summary(continuous_boot)

#Interaction term between treatment and mediator is included in the model
b <- lm(job_seek ~ treat + depress1, data=jobs)
d <- lm(depress2 ~ treat + job_seek + treat:job_seek + depress1, data=jobs)
cont.int <- mediate(b, d, sims=10, INT=TRUE, treat="treat", mediator="job_seek")
summary(cont.int)

#####
#Binary Outcome
#####

b <- lm(job_seek ~ treat + depress1, data=jobs)
c <- glm(work1 ~ treat + job_seek + depress1, family=binomial(link="probit"), data=jobs)
binary <- mediate(b, c, sims=10, treat="treat", mediator="job_seek")
summary(binary)
```

Description

Function to perform sensitivity analysis on mediation effect for violations of sequential ignorability assumption. This allows for a correlation between the error terms of the outcome model and the mediator model. Sensitivity analysis is possible with 1) continuous mediator and continuous outcome, 2) binary outcome and continuous mediator, and 3) continuous outcome and binary mediator.

Output from the function can be passed through `summary` or `plot` functions which display estimated mediation effects for given values of `rho`.

Usage

```
medsens(x, rho.by=.1, sims=1000, eps=.Machine$double.eps)
```

Arguments

<code>x</code>	Output from <code>mediate</code> function
<code>rho.by</code>	Specifies the grid for <code>rho</code> (sensitivity parameter) on which estimation is calculated for. Decreasing this will result in more precise sensitivity estimates but significantly more computational time.
<code>sims</code>	Number of Monte Carlo draws for calculating 95 percent confidence intervals. This argument is used only in cases where either binary mediator or binary outcome is involved.
<code>eps</code>	Convergence parameter for FGLS estimation. This argument is used only in the continuous-continuous case.

Details

This is the workhorse function for estimating sensitivity analyses for mediation effects. In fitting models for a binary variable a probit function must be used.

Users should note that computation time is several minutes for these functions. Setting `rho.by` to smaller numbers significantly increases computational time, as does increasing `eps` or the number of simulations.

Value

`medsens` returns an object of class "medsens". Some of the below elements are not available depending on whether `INT` is specified as `TRUE` or `FALSE` by the user or depending on the type of model fit. The function `summary` is used to obtain a table of the results. The function `plot` is used to plot the results.

<code>d0</code>	vector of point estimates for mediation effect under control.
<code>d1</code>	vector of point estimates for mediation effect under treatment.
<code>upper.d0</code>	vector of upper confidence intervals for mediation effect under control.
<code>lower.d0</code>	vector of lower confidence intervals for mediation effect under control.
<code>upper.d1</code>	vector of upper confidence intervals for mediation effect under treatment.
<code>lower.d1</code>	vector of lower confidence intervals for mediation effect under treatment.
<code>tau</code>	vector of point estimates for total effect.
<code>upper.tau</code>	vector of upper confidence intervals for total effect.
<code>lower.tau</code>	vector of lower confidence intervals for total effect.
<code>nu</code>	vector of estimated proportions of total effect mediated.

<code>upper.nu</code>	vector of upper confidence intervals for proportion mediated.
<code>lower.nu</code>	vector of lower confidence intervals for proportion mediated.
<code>rho</code>	vector of sensitivity parameter rho used.
<code>rho.by</code>	increment of rho used.
<code>err.cr</code>	value of rho corresponding to the mediation effect of zero.
<code>R2star.prod</code>	value of the product of R square terms for the proportion of additional variance left unexplained, corresponding to value of rho.
<code>R2tilda.prod</code>	value of the product of R square terms for the proportion of total variance left unexplained, corresponding to value of rho.
<code>R2star.thresh</code>	value of R square for the proportion of additional variance left unexplained which corresponds to the mediation effect of zero.
<code>R2tilda.thresh</code>	value of R square for the proportion of total variance left unexplained which corresponds to the mediation effect of zero.
<code>r.square.y</code>	R square for the outcome model.
<code>r.square.m</code>	R square for the mediator model.
<code>INT</code>	the INT argument used in the original mediate object.
<code>type</code>	type of the mediator and outcome models used. Currently either 'ct', 'bm' or 'bo'.

Warning

These functions assume that all missing values have been removed from the data set. This can be done using the `na.omit()` command before the outcome and mediation models are fitted.

Author(s)

Luke Keele, Ohio State University, keele.4@osu.edu , Dustin Tingley, Princeton University, dtingley@princeton.edu, Teppei Yamamoto, Princeton University, tyamamot@princeton.edu, Kosuke Imai, Princeton University, kimai@princeton.edu

References

Imai, Kosuke, Luke Keele and Dustin Tingley (2009) A General Approach to Causal Mediation Analysis. Imai, Kosuke, Luke Keele and Teppei Yamamoto (2009) Identification, Inference, and Sensitivity Analysis for Causal Mediation Effects. Imai, Kosuke, Luke Keele, Dustin Tingley, and Teppei Yamamoto. (2009) "Causal Mediation Analysis Using R" in Advances in Social Science Research Using R, ed. H. D. Vinod New York: Springer. Forthcoming.

See Also

See also [mediate](#)

Examples

```
#Example with JOBS II Field experiment

##**For illustration purposes simulations set to low number**

data(jobs)

#####
#continuous mediator and continuous outcome
#####

#Fit parametric models
model.m <- lm(job_seek ~ treat + depress1, data=jobs)
model.y <- lm(depress2 ~ treat + job_seek + depress1, data=jobs)

#Pass model objects through mediate function
med.cont <- mediate(model.m, model.y, treat="treat", mediator="job_seek", sims=10)

#Pass mediate output through medsens function
sens.cont <- medsens(med.cont, rho.by=.2, eps=.01)

#Use summary function to display values of rho and R^2 where 95
summary(sens.cont)

#Plot mediation effect and 95
plot(sens.cont, main="JOBS", ylim=c(-.2,.2))

#Plot sensitivity analysis using R^2 method. See plot.medsens for additional detail
plot(sens.cont, sens.par="R2", r.type=2, sign.prod=1)
```

plot.medsens

Plotting results from sensitivity analysis of mediation effects.

Description

Plots results from medsens function. Y axis plots mediation effect and x-axis plots the error correlation rho. Standard options for plot function available.

Usage

```
## S3 method for class 'medsens':
plot(x, sens.par="rho", r.type=1, sign.prod=1, pr.plot=FALSE, smooth.effect=FALSE)

# ## S3 method for class 'plot.medsens':
print(z)
```

Arguments

x	output from medsens function.
sens.par	type of sensitivity interpretation to be plotted. Default plots mediation effects in terms of sensitivity parameter rho. If "R2" is specified then in terms of variances explained by an unobserved pretreatment confounder.
r.type	type of R square interpretation to be used. If 1 is specified then proportion of previously unexplained variance is used. If 2 is specified then proportion of total unexplained variance is used. Only relevant if sens.par is set to "R2".
sign.prod	whether the omitted variable affects the mediator and outcome variable in the same direction (1) or different directions (-1). Only relevant if sens.par is set to "R2".
pr.plot	if pr.plot=TRUE then proportion mediated will be plotted.
smooth.effect	whether the estimated mediation effects are smoothed via a lowess smoother before being plotted.
smooth.ci	whether the confidence bands are smoothed via a lowess smoother before being plotted.
levels	vector of levels at which to draw contour lines. Only relevant if sens.par is set to "R2".
xlab	x-axis label.
ylab	y-axis label.
xlim	range for x-axis.
ylim	range for y-axis.
main	main title for graph.
...	additional arguments to be passed.

Warning

The smooth.effect and smooth.ci options should be used with caution since the smoothing could affect substantive implications of the graphical analysis in a significant way.

Author(s)

Luke Keele, Ohio State University, <keele.4@osu.edu> , Dustin Tingley, Princeton University, <dtingley@princeton.edu>, Teppei Yamamoto, Princeton University, <tyamamot@princeton.edu>, Kosuke Imai, Princeton University, <kimai@princeton.edu>

References

Imai, Kosuke, Luke Keele and Dustin Tingley (2009) A General Approach to Causal Mediation Analysis. Imai, Kosuke, Luke Keele and Teppei Yamamoto (2009) Identification, Inference, and Sensitivity Analysis for Causal Mediation Effects. Imai, Kosuke, Luke Keele, Dustin Tingley, and Teppei Yamamoto. (2009) "Causal Mediation Analysis Using R" in Advances in Social Science Research Using R, ed. H. D. Vinod New York: Springer. Forthcoming.

See Also

See also [medsens](#)

Examples

```
#Example with JOBS II Field experiment

data(jobs)

## Not run:
#####
#Continuous mediator and continuous outcome
#####

#Fit parametric models
model.m <- lm(job_seek ~ treat + depress1, data=jobs)
model.y <- lm(depress2 ~ treat + job_seek + depress1, data=jobs)

#Pass model objects through mediate function
med.cont <- mediate(model.m, model.y, treat="treat", mediator="job_seek", sims=1000)

#Pass mediate output through medsens function
sens.cont <- medsens(med.cont, sims=1000, rho.by=.1)

#Use summary function to display values of rho where 95
summary(sens.cont)

#Plot mediation effect and 95
plot(sens.cont, main="JOBS", ylim=c(-.2, .2))

#Plot sensitivity analysis using R^2 method. See plot.medsens for additional detail
plot(sens.cont, sens.par="R2", r.type=2, sign.prod=1)
## End(Not run)
```

summary.mediate *Summarizing output from mediation analysis.*

Description

Function to report results from mediation analysis. Reported categories are mediation effect, direct effect, total effect, and proportion of total effect mediated. All quantities reported with 95 percent confidence intervals. If the `INT = TRUE` option is selected in the `mediate` function, which allows for interaction between treatment and mediator, then effects are reported separately for the treatment and control groups.

Usage

```
## S3 method for class 'mediate':
summary(object, ...)

## S3 method for class 'summary.mediate':
print(x, ...)
```

Arguments

object	Output from mediate function.
x	Output from mediate function.
...	Additional arguments to be passed.

Author(s)

Luke Keele, Ohio State University, <keele.4@osu.edu> , Dustin Tingley, Princeton University, <dtingley@princeton.edu>, Teppei Yamamoto, Princeton University, <tyamamot@princeton.edu>, Kosuke Imai, Princeton University, <kimai@princeton.edu>

References

Imai, Kosuke, Luke Keele and Dustin Tingley (2009) A General Approach to Causal Mediation Analysis. Imai, Kosuke, Luke Keele and Teppei Yamamoto (2009) Identification, Inference, and Sensitivity Analysis for Causal Mediation Effects. Imai, Kosuke, Luke Keele, Dustin Tingley, and Teppei Yamamoto. (2009) "Causal Mediation Analysis Using R" in Advances in Social Science Research Using R, ed. H. D. Vinod New York: Springer. Forthcoming.

See Also

See also [mediate](#)

Examples

```
#Example with JOBS II Field experiment

##*For illustration purposes simulations set to low number**

data(jobs)

#####
#continuous mediator and continuous outcome
#####

model.m <- lm(job_seek ~ treat + depress1 + econ_hard + sex + age + occp + marital + nonwhi
model.y <- lm(depress2 ~ treat + job_seek + depress1 + econ_hard + sex + age + occp + marita
continuous <- mediate(model.m, model.y, sims=10, treat="treat", mediator="job_seek")
summary(continuous)
```

summary.medsens *Summarizing results from sensitivity analysis of mediation effects.*

Description

Function to perform sensitivity analysis on mediation effect for violations of sequential ignorability assumption. The procedure estimates the mediation effect allowing for a correlation between the error terms of the outcome model and the mediator model. The extent of this correlation is expressed in terms of the parameter ρ . Sensitivity analysis is possible with continuous mediator and continuous outcome and binary outcome and continuous mediator. Future versions of the function will also permit sensitivity analysis for a continuous outcome and binary mediator. Output from the function can be passed through summary or plot functions which display estimated mediation effects for given values of ρ .

Usage

```
## S3 method for class 'medsens':  
summary(object, ...)  
  
## S3 method for class 'summary.medsens':  
print(x, ...)
```

Arguments

x	Output from medsens function.
object	Output from medsens function.
...	Additional arguments to be passed.

Author(s)

Luke Keele, Ohio State University, <keele.4@osu.edu> , Dustin Tingley, Princeton University, <dtingley@princeton.edu>, Teppei Yamamoto, Princeton University, <tyamamot@princeton.edu>, Kosuke Imai, Princeton University, <kimai@princeton.edu>

References

Imai, Kosuke, Luke Keele and Dustin Tingley (2009) A General Approach to Causal Mediation Analysis. Imai, Kosuke, Luke Keele and Teppei Yamamoto (2009) Identification, Inference, and Sensitivity Analysis for Causal Mediation Effects. Imai, Kosuke, Luke Keele, Dustin Tingley, and Teppei Yamamoto. (2009) "Causal Mediation Analysis Using R" in Advances in Social Science Research Using R, ed. H. D. Vinod New York: Springer. Forthcoming.

See Also

See also [medsens](#)

Examples

```
## Not run:
#####
#Continuous mediator and continuous outcome
#####

#Fit parametric models
model.m <- lm(job_seek ~ treat + depress1, data=jobs)
model.y <- lm(depress2 ~ treat + job_seek + depress1, data=jobs)

#Pass model objects through mediate function
med.cont <- mediate(model.m, model.y, treat="treat", mediator="job_seek", sims=1000)

#Pass mediate output through medsens function
sens.cont <- medsens(med.cont, sims=1000, rho.by=.1)

#Use summary function to display values of rho where 95
summary(sens.cont)

#####
#binary outcome and continuous mediator
#####

model.m <- lm(job_seek ~ treat + depress1 + econ_hard + sex + age + occp + marital + nonwhite)
model.y <- glm(work1 ~ treat + job_seek + depress1 + econ_hard + sex + age + occp + marital)
med.bin <- mediate(model.m, model.y, treat="treat", mediator="job_seek", sims=1000)
sens.bin <- medsens(med.bin, sims=1000, rho.by=.1)
summary(sens.bin)

## End(Not run)
```

Index

*Topic **datasets**

jobs, [2](#)

jobs, [2](#)

mediate, [2](#), [7](#), [11](#)

medsens, [4](#), [5](#), [10](#), [13](#)

plot.medsens, [8](#)

print.summary.mediate
(*summary.mediate*), [10](#)

print.summary.medsens
(*summary.medsens*), [12](#)

summary.mediate, [10](#)

summary.medsens, [12](#)