

Package: mediation (via r-universe)

September 6, 2024

Version 4.5.0

Date 2019-6-10

Title Causal Mediation Analysis

Author Dustin Tingley <dtingley@gov.harvard.edu>, Teppei Yamamoto <teppeii@mit.edu>, Kentaro Hirose <hirose@princeton.edu>, Luke Keele <ljk20@psu.edu>, Kosuke Imai <kimai@princeton.edu>, Minh Trinh <mdtrinh@mit.edu>, Weihuang Wong <wwong@mit.edu>

Maintainer Teppei Yamamoto <teppeii@mit.edu>

Depends R (>= 2.15.1), MASS, Matrix, mvtnorm, sandwich

Imports IpSolve, Hmisc, lme4 (>= 1.0), stats, graphics, grDevices, utils, boot

Suggests mgcv, quantreg, VGAM, SuppDists, survival, parallel, testthat, speedglm

Description We implement parametric and non parametric mediation analysis. This package performs the methods and suggestions in Imai, Keele and Yamamoto (2010) <DOI:10.1214/10-STS321>, Imai, Keele and Tingley (2010) <DOI:10.1037/a0020761>, Imai, Tingley and Yamamoto (2013) <DOI:10.1111/j.1467-985X.2012.01032.x>, Imai and Yamamoto (2013) <DOI:10.1093/pan/mps040> and Yamamoto (2013) <<http://web.mit.edu/teppeii/www/research/IVmediate.pdf>>. 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)

URL <https://imai.princeton.edu/projects/mechanisms.html>

RoxygenNote 6.1.1

Repository <https://teppeiyamamoto.r-universe.dev>

RemoteUrl <https://github.com/teppeiyamamoto/mediation>

RemoteRef HEAD

RemoteSha e5310d5b2a5b75c5e26b38f564007464c4d3fb9d

Contents

amelidiate	2
boundsdata	4
CEDdata	5
framing	6
ivmediate	7
jobs	10
mediate	11
mediate.ced	21
mediate.pd	23
mediate.ped	25
mediate.sed	27
mediate_tsls	29
mediation	31
mediations	31
medsens	34
multimed	38
plot.ivmediate	42
plot.mediate	43
plot.mediate.mer	45
plot.mediations	46
plot.medsens	47
plot.multimed	50
school	52
student	53
summary.ivmediate	54
summary.mediate	55
summary.mediate.design	56
summary.mediate.mer	57
summary.mediations	59
summary.medsens	60
summary.multimed	61
test.modmed	62
test.TMint	64
Index	66

amelidiate	<i>Function for combining outputs from mediations function and calculating quantities of interest. For use with multiple imputation procedures.</i>
------------	---

Description

'amelidiate' takes the output from `mediations` and stacks the different vectors. Next it outputs these stacked vectors in the format of a `mediate` object.

Usage

```
amelidiate(g)
```

Arguments

g output from `mediations` that used the same models and variables but run on different datasets.

Details

`amelidiate` is designed to help users process multiple datasets where missing values have been imputed. First create multiple datasets using your preferred imputation software.

Next pass the data sets, as shown in the example below, to the `mediations` function. Finally pass the output of `mediations` through the `amelidiate` function. This will output an object that can then be passed through the standard summary and plot commands.

This function is not completely developed. It does not support models for ordered outcomes, inherits the limitations of the `mediations` function, and does not pass the information required for calculation of p-values.

Value

An object of class "mediate".

Author(s)

Dustin Tingley, Harvard University, <dtingley@gov.harvard.edu>

See Also

[mediate](#), [mediations](#).

Examples

```
## Not run:
# Hypothetical example

## To use mediations, must make list of multiple datasets. Then,
## must also repeat the treatment assignment list as many times
## as you have data sets.
# datasets <- list(D1=D1, D2=D2) # list of multiply imputed data sets
# mediators <- c("M1")
# outcome <- c("Ycont1")
# treatment <- c("T1","T1") # note how the treatment indicator is repeated
# covariates <- c("X1+X2")
# olsols <- mediations(datasets, treatment, mediators, outcome, covariates,
#                       families=c("gaussian","gaussian"), interaction=FALSE,
#                       conf.level=.90, sims=1000)
# output <- amelidiate(olsols)
# summary(output)
# plot(output)
```

```
## End(Not run)
```

boundsdata

Example Data for the Design Functions

Description

A random subsample of the simulated data used in Imai, Tingley, and Yamamoto (2012). The data contains 1000 rows and 7 columns with no missing values.

Usage

```
boundsdata
```

Format

A data frame containing the following variables, which are interpreted as results from a hypothetical randomized trial. See the source for a full description.

out: The binary outcome variable under the parallel design.

out.enc: The binary outcome variable under the parallel encouragement design.

med: The binary mediator under the parallel design.

med.enc: The binary mediator under the parallel encouragement design.

ttt: The binary treatment variable.

manip: The design indicator, or the variable indicating whether the mediator is manipulated under the parallel design.

enc: The trichotomous encouragement variable under the parallel encouragement design. Equals 0 if subject received no encouragement; 1 if encouraged for the mediator value of 1; and -1 if encouraged for the mediator value of 0.

Details

Conditioning on `'manip' = 0` will simulate a randomized trial under the single experiment design, where `'out'` and `'med'` equal observed outcome and mediator values, respectively.

Unconditionally, using `'out'`, `'med'`, `'ttt'` and `'manip'` will simulate an experiment under the parallel design.

The `'out.enc'` and `'med.enc'` variables represent the outcome and mediator values observed when subjects received the encouragement indicated in `'enc'`. Therefore, using `'out.enc'`, `'med.enc'`, `'ttt'` and `'enc'` will simulate an experiment under the parallel encouragement design.

Note that all the observed responses are generated from an underlying distribution of potential outcomes and mediators (not shown in this dataset) satisfying the assumptions described in Imai, Tingley and Yamamoto (2012). The full simulation code is available as a companion replication archive for the article.

Source

Imai, K., Tingley, D. and Yamamoto, T. (2012) Experimental Designs for Identifying Causal Mechanisms. *Journal of the Royal Statistical Society, Series A (Statistics in Society)*.

CEDdata

Example Data for the Crossover Encouragement Design

Description

A randomly generated dataset containing 2000 rows and 7 columns with no missing values.

Usage

CEDdata

Format

A data frame containing the following variables, which are interpreted as results from a hypothetical randomized trial employing the crossover encouragement design.

T1: The binary treatment indicator in the first stage.

M1: The binary mediator variable recorded in the first stage.

Y1: The binary outcome variable recorded in the first stage.

T2: The binary treatment in the second stage. Equal to $1 - T1$ by design.

Z: The binary encouragement indicator for the second stage.

M2: The binary mediator recorded in the second stage.

Y2: The binary outcome recorded in the second stage.

Details

Note that all the observed responses are generated from an underlying distribution of potential outcomes and mediators (not shown in this dataset) satisfying the assumptions described in Imai, Tingley and Yamamoto (2012).

Source

Imai, K., Tingley, D. and Yamamoto, T. (2012) Experimental Designs for Identifying Causal Mechanisms. *Journal of the Royal Statistical Society, Series A (Statistics in Society)*.

framing

*Brader, Valentino and Suhay (2008) Framing Experiment Data***Description**

The framing data contains 265 rows and 15 columns of data from a framing experiment conducted by Brader, Valentino and Suhay (2008).

Usage

framing

Format

A data frame containing the following variables:

immigr: A four-point scale measuring subjects' attitudes toward increased immigration. Larger values indicate more negative attitudes.

english: A four-point scale indicating whether subjects favor or oppose a law making English the official language of the U.S.

cong_mesg: Whether subjects requested sending an anti-immigration message to Congress on their behalf.

anti_info: Whether subjects wanted to receive information from anti-immigration organizations.

tone: 1st treatment; whether the news story is framed positively or negatively.

eth: 2nd treatment; whether the news story features a Latino or European immigrant.

cond: Four level measure recording joint treatment status of tone and eth.

treat: Product of the two treatment variables. In the original study the authors only find this cell to be significant.

emo: Measure of subjects' negative feeling during the experiment. A numeric scale ranging between 3 and 12 where 3 indicates the most negative feeling.

anx: A four-point scale measuring subjects' anxiety about increased immigration.

p_harm: Subjects' perceived harm caused by increased immigration. A numeric scale between 2 and 8.

age: Subjects' age.

educ: Subjects' highest educational attainments.

gender: Subjects' gender.

income: Subjects' income, measured as a 19-point scale.

Source

Brader, T., Valentino, N. and Suhay, E. (2008). What triggers public opposition to immigration? Anxiety, group cues, and immigration threat. *American Journal of Political Science* 52, 4, 959–978.

Description

'ivmediate' is used to estimate local average causal mediation effects, local average natural direct effects and local average treatment effects for compliers using the method of Yamamoto (2013).

Usage

```
ivmediate(model.t, model.m, model.y, ci = TRUE, sims = 1000,
  boot = TRUE, enc = "enc.name", treat = "treat.name",
  mediator = "med.name", conf.level = 0.95, long = TRUE,
  dropobs = FALSE, multicore = FALSE,
  mc.cores = getOption("mc.cores", 2L))
```

Arguments

model.t	a fitted model object for actual treatment. Can be of class 'lm' or 'glm'.
model.m	a fitted model object for mediator. Can be of class 'lm', or 'glm'.
model.y	a fitted model object for outcome. Can be of class 'lm', or 'glm'.
ci	a logical value. if 'TRUE' both point estimates and confidence intervals are calculated; if 'FALSE' only point estimates are produced. Default is 'TRUE'.
sims	number of Monte Carlo draws for nonparametric bootstrap or quasi-Bayesian approximation.
boot	a logical value. if 'FALSE' a quasi-Bayesian approximation is used for confidence intervals; if 'TRUE' nonparametric bootstrap will be used. Default is 'TRUE'.
enc	a character string indicating the name of the encouragement variable used in the models. Must be a binary variable.
treat	a character string indicating the name of the actual treatment variable used in the models. Must be a binary variable.
mediator	a character string indicating the name of the mediator variable used in the models.
conf.level	a numeric vector indicating the levels of the returned two-sided confidence intervals. Default is to return the 2.5 and 97.5 percentiles of the simulated quantities.
long	a logical value. If 'TRUE', the output will contain the entire sets of simulation draws of the estimated effects. Default is 'TRUE'.
dropobs	a logical value indicating the behavior when the model frames of 'model.t', 'model.m' and 'model.y' are composed of different observations. If 'TRUE', models will be re-fitted using common data rows. If 'FALSE', error is returned. Default is 'FALSE'.
multicore	a logical value indicating whether to parallelize simulations into multiple cores. Default is 'FALSE'. Note that this option is currently unavailable on Windows.

`mc.cores` number of cores to be used in the parallelization of bootstrap or Monte Carlo simulations. Default is the value in the `'mc.cores'` option (see `mclapply`).

Details

This is the workhorse function for estimating local causal mediation effects for compliers using the approach of Yamamoto (2013).

Although the method is quite general and accommodate various types of mediator and outcome variables, the current function can only handle binary variables (modeled via `glm` with `family=binomial`) and continuous variables (modeled via `lm`). In addition, when the mediator is continuous and its model contains any predictor other than the encouragement and actual treatment, computation of confidence intervals is extremely slow because it requires numerical integration via `integrate` for each observation in each simulation iteration. Users are advised to use a powerful computer (preferably equipped with multiple CPU cores and a non-Windows operating system) for such calculation, especially if the data contain many rows and/or the desired number of simulations is large.

Value

`ivmediate` returns an object of class `'ivmediate'`, a list that contains the components listed below. Some of these elements are not available depending on the values of the `'ci'` and `'long'` options.

The function `summary` (i.e., `summary.ivmediate`) can be used to obtain a table of the results. The function `plot` (i.e., `plot.ivmediate`) can be used to produce a plot of the estimated effects along with their confidence intervals.

<code>dc0, dc1</code>	point estimates for the local average causal mediation effects under the control and treatment conditions.
<code>dc0.ci, dc1.ci</code>	confidence intervals for the local average causal mediation effects. The confidence levels are set at the value specified in <code>'conf.level'</code> .
<code>dc0.issue, dc1.issue</code>	number of observations for which the numerical integration via <code>integrate</code> encountered computational problems when calculating <code>dc0</code> and <code>dc1</code> , respectively.
<code>dc0.inf, dc1.inf</code>	number of observations for which the numerical integration produced non-finite values when calculating <code>dc0</code> and <code>dc1</code> , respectively. (Such values are excluded from the calculation of average effects.)
<code>dc0.sims, dc1.sims</code>	vectors of length <code>'sims'</code> containing simulation draws of local average causal mediation effects.
<code>zc0, zc1</code>	point estimates for the local average natural direct effects under the control and treatment conditions.
<code>zc0.ci, zc1.ci</code>	confidence intervals for the local average natural direct effects.
<code>zc0.issue, zc1.issue</code>	number of observations for which the numerical integration encountered computational problems when calculating <code>zc0</code> and <code>zc1</code> , respectively.
<code>zc0.inf, zc1.inf</code>	number of observations for which the numerical integration produced non-finite values for <code>zc0</code> and <code>zc1</code> .

<code>zc0.sims, zc1.sims</code>	vectors of length 'sims' containing simulation draws of local average natural direct effects.
<code>tauc</code>	point estimate for the local average treatment effect.
<code>tauc.ci</code>	confidence interval for the local average treatment effect.
<code>tauc.sims</code>	a vector of length 'sims' containing simulation draws of the local average treatment effect.
<code>boot</code>	logical, the 'boot' argument used.
<code>enc</code>	a character string indicating the name of the 'enc' variable used.
<code>treat</code>	a character string indicating the name of the 'treat' variable used.
<code>mediator</code>	a character string indicating the name of the 'mediator' variable used.
<code>conf.level</code>	the confidence levels used.
<code>nobs</code>	number of observations in the model frame for 'model.t', 'model.m' and 'model.y'. May differ from the numbers in the original models input to 'ivmediate' if 'dropobs' was 'TRUE'.
<code>sims</code>	number of simulation draws used.

Author(s)

Teppei Yamamoto, Massachusetts Institute of Technology, <tepei@mit.edu>.

References

Tingley, D., Yamamoto, T., Hirose, K., Imai, K. and Keele, L. (2014). "mediation: R package for Causal Mediation Analysis", *Journal of Statistical Software*, Vol. 59, No. 5, pp. 1-38.

Yamamoto, T. (2013). Identification and Estimation of Causal Mediation Effects with Treatment Noncompliance. Unpublished manuscript.

See Also

[plot.ivmediate](#), [summary.ivmediate](#)

Examples

```
# Examples with JOBS II Field Experiment

# ** For illustration purposes a small number of simulations are used **

require(parallel)
require(MASS)

data(jobs)

a <- lm(comply ~ treat + sex + age + marital + nonwhite + educ + income,
        data = jobs)
b <- glm(job_dich ~ comply + treat + sex + age + marital + nonwhite + educ + income,
        data = jobs, family = binomial)
```

```

c <- lm(depress2 ~ job_dich * (comply + treat) + sex + age + marital + nonwhite + educ + income,
      data = jobs)

out <- ivmediate(a, b, c, sims = 50, boot = FALSE,
               enc = "treat", treat = "comply", mediator = "job_dich")

summary(out)
plot(out)

# Using non-parametric bootstrap
out.boot <- ivmediate(a, b, c, sims = 50, boot = TRUE,
                    enc = "treat", treat = "comply", mediator = "job_dich")

summary(out.boot)
plot(out.boot)

```

 jobs

JOBS 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

```
jobs
```

Format

A data matrix with 899 rows and 17 columns, containing no missing values. The data are provided only for illustrative purposes and not for inference about program efficacy, for which the original data source should be consulted.

econ_hard: Level of economic hardship pre-treatment with values from 1 to 5.

depress1: Measure of depressive symptoms pre-treatment.

sex: Indicator variable for sex. 1 = female

- age:** Age in years.
- occp:** Factor with seven categories for various occupations.
- marital:** Factor with five categories for marital status.
- nonwhite:** Indicator variable for race. 1 = nonwhite.
- educ:** Factor with five categories for educational attainment.
- income:** Factor with five categories for level of income.
- job_seek:** A continuous scale measuring the level of job-search self-efficacy with values from 1 to 5. The mediator variable.
- depress2:** Measure of depressive symptoms post-treatment.
- work1:** Indicator variable for employment. 1 = employed.
- job_dich:** The job_seek measure recoded into two categories of high and low. 1 = high job search self-efficacy.
- job_disc:** The job_seek measure recoded into four categories from lowest to highest.
- treat:** Indicator variable for whether participant was randomly selected for the JOBS II training program. 1 = assignment to participation.
- comply:** Indicator variable for whether participant actually participated in the JOBS II program. 1 = participation.
- control:** Indicator variable for whether participant was randomly selected to not participate in the JOBS II training program. 1 = non-participation.

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):867-77.

Description

'mediate' is used to estimate various quantities for causal mediation analysis, including average causal mediation effects (indirect effect), average direct effects, proportions mediated, and total effect.

Usage

```
mediate(model.m, model.y, sims = 1000, boot = FALSE,
        boot.ci.type = "perc", treat = "treat.name", mediator = "med.name",
        covariates = NULL, outcome = NULL, control = NULL,
        conf.level = 0.95, control.value = 0, treat.value = 1,
        long = TRUE, dropobs = FALSE, robustSE = FALSE, cluster = NULL,
        group.out = NULL, use_speed = FALSE, ...)
```

Arguments

<code>model.m</code>	a fitted model object for mediator. Can be of class 'lm', 'polr', 'bayespolr', 'glm', 'bayesglm', 'gam', 'rq', 'survreg', or 'merMod'.
<code>model.y</code>	a fitted model object for outcome. Can be of class 'lm', 'polr', 'bayespolr', 'glm', 'bayesglm', 'gam', 'vglm', 'rq', 'survreg', or 'merMod'.
<code>sims</code>	number of Monte Carlo draws for nonparametric bootstrap or quasi-Bayesian approximation.
<code>boot</code>	a logical value. if 'FALSE' a quasi-Bayesian approximation is used for confidence intervals; if 'TRUE' nonparametric bootstrap will be used. Default is 'FALSE'.
<code>boot.ci.type</code>	a character string indicating the type of bootstrap confidence intervals. If "bca" and <code>boot = TRUE</code> , bias-corrected and accelerated (BCa) confidence intervals will be estimated. If "perc" and <code>boot = TRUE</code> , percentile confidence intervals will be estimated. Default is "perc".
<code>treat</code>	a character string indicating the name of the treatment variable used in the models. The treatment can be either binary (integer or a two-valued factor) or continuous (numeric).
<code>mediator</code>	a character string indicating the name of the mediator variable used in the models.
<code>covariates</code>	a list or data frame containing values for a subset of the pre-treatment covariates in 'model.m' and 'model.y'. If provided, the function will return the estimates conditional on those covariate values.
<code>outcome</code>	a character string indicating the name of the outcome variable in 'model.y'. Only necessary if 'model.y' is of class 'survreg'; otherwise ignored.
<code>control</code>	a character string indicating the name of the control group indicator. Only relevant if 'model.y' is of class 'gam'. If provided, 'd0', 'z0' and 'n0' are allowed to differ from 'd1', 'z1' and 'n1', respectively.
<code>conf.level</code>	level of the returned two-sided confidence intervals. Default is to return the 2.5 and 97.5 percentiles of the simulated quantities.
<code>control.value</code>	value of the treatment variable used as the control condition. Default is 0.
<code>treat.value</code>	value of the treatment variable used as the treatment condition. Default is 1.
<code>long</code>	a logical value. If 'TRUE', the output will contain the entire sets of simulation draws of the the average causal mediation effects, direct effects, proportions mediated, and total effect. Default is 'TRUE'.

dropobs	a logical value indicating the behavior when the model frames of 'model.m' and 'model.y' (and the 'cluster' variable if included) are composed of different observations. If 'TRUE', models will be re-fitted using common data rows. If 'FALSE', error is returned. Default is 'FALSE'.
robustSE	a logical value. If 'TRUE', heteroskedasticity-consistent standard errors will be used in quasi-Bayesian simulations. Ignored if 'boot' is 'TRUE' or neither 'model.m' nor 'model.y' has a method for vcovHC in the sandwich package. Default is 'FALSE'.
cluster	a variable indicating clusters for standard errors. Note that this should be a vector of cluster indicators itself, not a character string for the name of the variable.
group.out	a character string indicating the name of the lmer/glmer group on which the mediate output is based. Can be used even when a merMod function is applied to only one of the mediator or the outcome. If merMod functions are applied to both the mediator and the outcome, default is the group name used in the outcome model; if the mediator group and the outcome group are different and the user is interested in the mediate output based on the mediator group, then set group.out to the group name used in the mediator merMod model. If a merMod function is applied to only one of the mediator or the outcome, group.out is automatically set to the group name used in the merMod model.
use_speed	a logical value indicating whether, if nonparametric bootstrap is used, lm and glm models should be re-fit using functions from the speedglm package. Ignored if 'boot' is 'FALSE' or if neither 'model.m' nor 'model.y' is of class 'lm' or 'glm'. Default is 'FALSE'.
...	other arguments passed to vcovHC in the sandwich package: typically the 'type' argument, which is ignored if 'robustSE' is 'FALSE'. Arguments to the boot in the boot package may also be passed, e.g. 'parallel' and 'ncpus'.

Details

This is the workhorse function for estimating causal mediation effects for a variety of data types. The average causal mediation effect (ACME) represents the expected difference in the potential outcome when the mediator took the value that would realize under the treatment condition as opposed to the control condition, while the treatment status itself is held constant. That is,

$$\delta(t) = E\{Y(t, M(t_1)) - Y(t, M(t_0))\},$$

where t, t_1, t_0 are particular values of the treatment T such that $t_1 \neq t_0$, $M(t)$ is the potential mediator, and $Y(t, m)$ is the potential outcome variable. The average direct effect (ADE) is defined similarly as,

$$\zeta(t) = E\{Y(t_1, M(t)) - Y(t_0, M(t))\},$$

which represents the expected difference in the potential outcome when the treatment is changed but the mediator is held constant at the value that would realize if the treatment equals t . The two quantities on average add up to the total effect of the treatment on the outcome, τ . See the references for more details.

When both the mediator model ('model.m') and outcome model ('model.y') are normal linear regressions, the results will be identical to the usual LSEM method by Baron and Kenny (1986). The function can, however, accommodate other data types including binary, ordered and count outcomes

and mediators as well as censored outcomes. Variables can also be modeled nonparametrically, semiparametrically, or using quantile regression.

If it is desired that inference be made conditional on specific values of the pre-treatment covariates included in the model, the `'covariates'` argument can be used to set those values as a list or data frame. The list may contain either the entire set or any strict subset of the covariates in the model; in the latter case, the effects will be averaged over the other covariates. The `'covariates'` argument will be particularly useful when the models contain interactions between the covariates and the treatment and/or mediator (known as “moderated mediation”).

The prior weights in the mediator and outcome models are taken as sampling weights and the estimated effects will be weighted averages when non-NULL weights are used in fitting `'model.m'` and `'model.y'`. This will be useful when data does not come from a simple random sample, for example.

As of version 4.0, the mediator model can be of either `'lm'`, `'glm'` (or `'bayesglm'`), `'polr'` (or `'bayespolr'`), `'gam'`, `'rq'`, `'survreg'`, or `'merMod'` class, corresponding respectively to the linear regression models, generalized linear models, ordered response models, generalized additive models, quantile regression models, parametric duration models, or multilevel models.. For binary response models, the `'mediator'` must be a numeric variable with values 0 or 1 as opposed to a factor. Quasi-likelihood-based inferences are not allowed for the mediator model because the functional form must be exactly specified for the estimation algorithm to work. The `'binomial'` family can only be used for binary response mediators and cannot be used for multiple-trial responses. This is due to conflicts between how the latter type of models are implemented in `glm` and how `'mediate'` is currently written.

For the outcome model, the censored regression model fitted via package VGAM (of class `'vglm'` with `'family@vfamily'` equal to `"tobit"`) can be used in addition to the models listed above for the mediator. The `'mediate'` function is not compatible with censored regression models fitted via other packages. When the quantile regression is used for the outcome model (`'rq'`), the estimated quantities are quantile causal mediation effects, quantile direct effects and etc., instead of the average effects. If the outcome model is of class `'survreg'`, the name of the outcome variable must be explicitly supplied in the `'outcome'` argument. This is due to the fact that `'survreg'` objects do not contain that information in an easily extractable form. It should also be noted that for `survreg` models, the `Surv` function must be directly used in the model formula in the call to the `survreg` function, and that censoring types requiring more than two arguments to `Surv` (e.g., interval censoring) are not currently supported by `'mediate'`.

The quasi-Bayesian approximation (King et al. 2000) cannot be used if `'model.m'` is of class `'rq'` or `'gam'`, or if `'model.y'` is of class `'gam'`, `'polr'` or `'bayespolr'`. In these cases, either an error message is returned or use of the nonparametric bootstrap is forced. Users should note that use of the nonparametric bootstrap often requires significant computing time, especially when `'sims'` is set to a large value.

The `'control'` argument must be provided when `'gam'` is used for the outcome model and user wants to allow ACME and ADE to vary as functions of the treatment (i.e., to relax the “no interaction” assumption). Note that the outcome model must be fitted via package `mgcv` with appropriate formula using `s` constructs (see Imai et al. 2009 in the references). For other model types, the interaction can be allowed by including an interaction term between T and M in the linear predictor of the outcome model. As of version 3.0, the `'INT'` argument is deprecated and the existence of the interaction term is automatically detected (except for `'gam'` outcome models).

When the treatment variable is continuous or a factor with multiple levels, user must specify the

values of t_1 and t_0 using the 'treat.value' and 'control.value' arguments, respectively. The value of t in the above expressions is set to t_0 for 'd0', 'z0', etc. and to t_1 for 'd1', 'z1', etc.

Value

mediate returns an object of class "mediate", "mediate.order" if the outcome model used is 'polr' or 'bayespolr', or "mediate.mer" if 'lmer' or 'glmer' is used for the outcome or the mediator model, a list that contains the components listed below. Some of these elements are not available if 'long' is set to 'FALSE' by the user.

The function summary (i.e., summary.mediate, summary.mediate.order, or summary.mediate.mer) can be used to obtain a table of the results. The function plot (i.e., plot.mediate, plot.mediate.order, or plot.mediate.mer) can be used to produce a plot of the estimated average causal mediation, average direct, and total effects along with their confidence intervals.

d0, d1	point estimates for average causal mediation effects under the control and treatment conditions.
d0.ci, d1.ci	confidence intervals for average causal mediation effects. The confidence level is set at the value specified in 'conf.level'.
d0.p, d1.p	two-sided p-values for average causal mediation effects.
d0.sims, d1.sims	vectors of length 'sims' containing simulation draws of average causal mediation effects.
z0, z1	point estimates for average direct effect under the control and treatment conditions.
z0.ci, z1.ci	confidence intervals for average direct effects.
z0.p, z1.p	two-sided p-values for average causal direct effects.
z0.sims, z1.sims	vectors of length 'sims' containing simulation draws of average direct effects.
n0, n1	the "proportions mediated", or the size of the average causal mediation effects relative to the total effect.
n0.ci, n1.ci	confidence intervals for the proportions mediated.
n0.p, n1.p	two-sided p-values for proportions mediated.
n0.sims, n1.sims	vectors of length 'sims' containing simulation draws of the proportions mediated.
tau.coef	point estimate for total effect.
tau.ci	confidence interval for total effect.
tau.p	two-sided p-values for total effect.
tau.sims	a vector of length 'sims' containing simulation draws of the total effect.
d.avg, z.avg, n.avg	simple averages of d0 and d1, z0 and z1, n0 and n1, respectively, which users may want to use as summary values when those quantities differ.
d.avg.ci, z.avg.ci, n.avg.ci	confidence intervals for the above.

`d.avg.p`, `z.avg.p`, `n.avg.p`
 two-sided p-values for the above.

`d.avg.sims`, `z.avg.sims`, `n.avg.sims`
 vectors of length 'sims' containing simulation draws of `d.avg`, `z.avg` and `n.avg`, respectively.

`d0.group`, `d1.group`
 group-specific point estimates for average causal mediation effects under the control and treatment conditions.

`d0.ci.group`, `d1.ci.group`
 group-specific confidence intervals for average causal mediation effects. The confidence level is set at the value specified in 'conf.level'.

`d0.p.group`, `d1.p.group`
 group-specific two-sided p-values for average causal mediation effects.

`d0.sims.group`, `d1.sims.group`
 group-specific vectors of length 'sims' containing simulation draws of average causal mediation effects.

`z0.group`, `z1.group`
 group-specific point estimates for average direct effect under the control and treatment conditions.

`z0.ci.group`, `z1.ci.group`
 group-specific confidence intervals for average direct effects.

`z0.p.group`, `z1.p.group`
 group-specific two-sided p-values for average causal direct effects.

`z0.sims.group`, `z1.sims.group`
 group-specific vectors of length 'sims' containing simulation draws of average direct effects.

`n0.group`, `n1.group`
 the group-specific "proportions mediated", or the size of the group-specific average causal mediation effects relative to the total effect.

`n0.ci.group`, `n1.ci.group`
 group-specific confidence intervals for the proportions mediated.

`n0.p.group`, `n1.p.group`
 group-specific two-sided p-values for proportions mediated.

`n0.sims.group`, `n1.sims.group`
 group-specific vectors of length 'sims' containing simulation draws of the proportions mediated.

`tau.coef.group` group-specific point estimate for total effect.

`tau.ci.group` group-specific confidence interval for total effect.

`tau.p.group` group-specific two-sided p-values for total effect.

`tau.sims.group` a group-specific vector of length 'sims' containing simulation draws of the total effect.

`d.avg.group`, `z.avg.group`, `n.avg.group`
 group-specific simple averages of `d0` and `d1`, `z0` and `z1`, `n0` and `n1`, respectively, which users may want to use as summary values when those quantities differ.

`d.avg.ci.group`, `z.avg.ci.group`, `n.avg.ci.group`
 group-specific confidence intervals for the above.

d.avg.p.group, z.avg.p.group, n.avg.p.group	group-specific two-sided p-values for the above.
d.avg.sims.group, z.avg.sims.group, n.avg.sims.group	group-specific vectors of length 'sims' containing simulation draws of d.avg, z.avg and n.avg, respectively.
boot	logical, the 'boot' argument used.
treat	a character string indicating the name of the 'treat' variable used.
mediator	a character string indicating the name of the 'mediator' variable used.
INT	a logical value indicating whether the model specification allows the effects to differ between the treatment and control conditions.
conf.level	the confidence level used.
model.y	the outcome model used.
model.m	the mediator model used.
group.m	the name of the mediator group used.
group.y	the name of the outcome group used.
group.name	the name of the group on which the output is based.
group.id.m	the data on the mediator group.
group.id.y	the data on the outcome group.
group.id	the data on the group on which the output is based.
control.value	value of the treatment variable used as the control condition.
treat.value	value of the treatment variable used as the treatment condition.
nobs	number of observations in the model frame for 'model.m' and 'model.y'. May differ from the numbers in the original models input to 'mediate' if 'dropobs' was 'TRUE'.
robustSE	'TRUE' or 'FALSE'.
cluster	the clusters used.

Author(s)

Dustin Tingley, Harvard University, <dtingley@gov.harvard.edu>; Teppei Yamamoto, Massachusetts Institute of Technology, <teppey@mit.edu>; Luke Keele, Penn State University, <ljk20@psu.edu>; Kosuke Imai, Princeton University, <kimai@princeton.edu>; Kentaro Hirose, Princeton University, <hirose@princeton.edu>.

References

- Tingley, D., Yamamoto, T., Hirose, K., Imai, K. and Keele, L. (2014). "mediation: R package for Causal Mediation Analysis", *Journal of Statistical Software*, Vol. 59, No. 5, pp. 1-38.
- Imai, K., Keele, L., Tingley, D. and Yamamoto, T. (2011). Unpacking the Black Box of Causality: Learning about Causal Mechanisms from Experimental and Observational Studies, *American Political Science Review*, Vol. 105, No. 4 (November), pp. 765-789.
- Imai, K., Keele, L. and Tingley, D. (2010) A General Approach to Causal Mediation Analysis, *Psychological Methods*, Vol. 15, No. 4 (December), pp. 309-334.

Imai, K., Keele, L. and Yamamoto, T. (2010) Identification, Inference, and Sensitivity Analysis for Causal Mediation Effects, *Statistical Science*, Vol. 25, No. 1 (February), pp. 51-71.

Imai, K., Keele, L., Tingley, D. and Yamamoto, T. (2009) "Causal Mediation Analysis Using R" in *Advances in Social Science Research Using R*, ed. H. D. Vinod New York: Springer.

See Also

[medsens](#), [plot.mediate](#), [summary.mediate](#), [summary.mediate.mer](#), [plot.mediate.mer](#), [mediations](#), [vcovHC](#)

Examples

```
# Examples with JOBS II Field Experiment

# **For illustration purposes a small number of simulations are used**

data(jobs)

#####
# Example 1: Linear Outcome and Mediator Models
#####
b <- lm(job_seek ~ treat + econ_hard + sex + age, data=jobs)
c <- lm(depress2 ~ treat + job_seek + econ_hard + sex + age, data=jobs)

# Estimation via quasi-Bayesian approximation
contcont <- mediate(b, c, sims=50, treat="treat", mediator="job_seek")
summary(contcont)
plot(contcont)

## Not run:
# Estimation via nonparametric bootstrap
contcont.boot <- mediate(b, c, boot=TRUE, sims=50, treat="treat", mediator="job_seek")
summary(contcont.boot)

## End(Not run)

# Allowing treatment-mediator interaction
d <- lm(depress2 ~ treat + job_seek + treat:job_seek + econ_hard + sex + age, data=jobs)
contcont.int <- mediate(b, d, sims=50, treat="treat", mediator="job_seek")
summary(contcont.int)

# Allowing ``moderated mediation'' with respect to age
b.int <- lm(job_seek ~ treat*age + econ_hard + sex, data=jobs)
d.int <- lm(depress2 ~ treat*job_seek*age + econ_hard + sex, data=jobs)
contcont.age20 <- mediate(b.int, d.int, sims=50, treat="treat", mediator="job_seek",
  covariates = list(age = 20))
contcont.age70 <- mediate(b.int, d.int, sims=50, treat="treat", mediator="job_seek",
  covariates = list(age = 70))
summary(contcont.age20)
summary(contcont.age70)

# Continuous treatment
```

```

jobs$treat_cont <- jobs$treat + rnorm(nrow(jobs)) # (hypothetical) continuous treatment
b.contT <- lm(job_seek ~ treat_cont + econ_hard + sex + age, data=jobs)
c.contT <- lm(depress2 ~ treat_cont + job_seek + econ_hard + sex + age, data=jobs)
contcont.cont <- mediate(b.contT, c.contT, sims=50,
                        treat="treat_cont", mediator="job_seek",
                        treat.value = 4, control.value = -2)
summary(contcont.cont)

# Categorical treatment
## Not run:
b <- lm(job_seek ~ educ + sex, data=jobs)
c <- lm(depress2 ~ educ + job_seek + sex, data=jobs)

# compare two categories of educ --- gradwk and somcol
model.cat <- mediate(b, c, treat="educ", mediator="job_seek", sims=50,
                    control.value = "gradwk", treat.value = "somcol")
summary(model.cat)

## End(Not run)

#####
# Example 2: Binary Outcome and Ordered Mediator
#####
## Not run:
jobs$job_disc <- as.factor(jobs$job_disc)
b.ord <- polr(job_disc ~ treat + econ_hard + sex + age, data=jobs,
             method="probit", Hess=TRUE)
d.bin <- glm(work1 ~ treat + job_disc + econ_hard + sex + age, data=jobs,
            family=binomial(link="probit"))
ordbin <- mediate(b.ord, d.bin, sims=50, treat="treat", mediator="job_disc")
summary(ordbin)

# Using heteroskedasticity-consistent standard errors
ordbin.rb <- mediate(b.ord, d.bin, sims=50, treat="treat", mediator="job_disc",
                   robustSE=TRUE)
summary(ordbin.rb)

# Using non-parametric bootstrap
ordbin.boot <- mediate(b.ord, d.bin, sims=50, treat="treat", mediator="job_disc",
                     boot=TRUE)
summary(ordbin.boot)

## End(Not run)

#####
# Example 3: Quantile Causal Mediation Effect
#####
require(quantreg)
c.quan <- rq(depress2 ~ treat + job_seek + econ_hard + sex + age, data=jobs,
            tau = 0.5) # median
contquan <- mediate(b, c.quan, sims=50, treat="treat", mediator="job_seek")
summary(contquan)

```

```
#####
# Example 4: GAM Outcome
#####
## Not run:
require(mgcv)
c.gam <- gam(depress2 ~ treat + s(job_seek, bs="cr") +
             econ_hard + sex + age, data=jobs)
contgam <- mediate(b, c.gam, sims=10, treat="treat",
                  mediator="job_seek", boot=TRUE)
summary(contgam)

# With interaction
d.gam <- gam(depress2 ~ treat + s(job_seek, by = treat) +
             s(job_seek, by = control) + econ_hard + sex + age, data=jobs)
contgam.int <- mediate(b, d.gam, sims=10, treat="treat", mediator="job_seek",
                      control = "control", boot=TRUE)
summary(contgam.int)

## End(Not run)
#####
# Example 5: Multilevel Outcome and Mediator Models
#####
## Not run:
require(lme4)

# educ: mediator group
# occp: outcome group

# Varying intercept for mediator
model.m <- glmer(job_dich ~ treat + econ_hard + (1 | educ),
                 family = binomial(link = "probit"), data = jobs)

# Varying intercept and slope for outcome
model.y <- glmer(work1 ~ treat + job_dich + econ_hard + (1 + treat | occp),
                 family = binomial(link = "probit"), data = jobs)

# Output based on mediator group ("educ")
multilevel <- mediate(model.m, model.y, treat = "treat",
                     mediator = "job_dich", sims=50, group.out="educ")

# Output based on outcome group ("occp")
# multilevel <- mediate(model.m, model.y, treat = "treat",
#                       mediator = "job_dich", sims=50)

# Group-average effects
summary(multilevel)

# Group-specific effects organized by effect
summary(multilevel, output="byeffect")
# plot(multilevel, group.plots=TRUE)
# See summary.mediate.mer and plot.mediate.mer for detailed explanations

# Group-specific effects organized by group
```

```
summary(multilevel, output="bygroup")
# See summary.mediate.mer for detailed explanations

## End(Not run)
```

mediate.ced	<i>Estimating Average Causal Mediation Effects under the Crossover Encouragement Design</i>
-------------	---

Description

'mediate.ced' estimates the average causal mediation effects for the crossover encouragement design.

Usage

```
mediate.ced(outcome, med.1, med.2, treat, encourage, data, sims = 1000,
  conf.level = 0.95)
```

Arguments

outcome	variable name in 'data' containing the outcome values observed in the second experiment. The variable must be binary (factor or numeric 0/1).
med.1	variable name in 'data' containing the mediator values observed in the first experiment. The variable must be binary (factor or numeric 0/1).
med.2	variable name in 'data' containing the mediator values observed in the second experiment.
treat	variable name in 'data' containing the treatment values in the first experiment. Must be binary (factor or numeric 0/1).
encourage	name of the encouragement indicator in 'data'. Must be binary (factor or numeric 0/1).
data	a data frame containing all the above variables.
sims	number of bootstrap simulations.
conf.level	level of the returned two-sided confidence intervals.

Details

This function estimates the average indirect effects for the pliable units under the crossover encouragement design. The design has two stages. In the first stage the treatment is randomized and the mediator and outcome variables are measured. In the second the treatment is set to the value opposite of first period and a randomly selected group of units receives encouragement to take on the mediator opposite to the values observed in the first stage. See Imai, Tingley and Yamamoto (2012) for a full description. The confidence intervals are calculated via the nonparametric bootstrap.

Note that outcome should be the observed responses in the *second* stage whereas treat should be the values in the *first* stage.

Value

mediate.ced returns an object of class "mediate.design", a list that contains the components listed below.

The summary function can be used to obtain a table of the results.

d0, d1	point estimates of the average indirect effects under the control and treatment conditions.
d0.ci, d1.ci	confidence intervals for the effects. The confidence level is set at the value specified in 'conf.level'.
conf.level	confidence level used.
sims	number of bootstrap simulations.
nobs	number of observations used.
design	indicates the design. Always equals "CED".

Author(s)

Dustin Tingley, Harvard University, <dtingley@gov.harvard.edu>; Teppei Yamamoto, Massachusetts Institute of Technology, <tepei@mit.edu>.

References

Tingley, D., Yamamoto, T., Hirose, K., Imai, K. and Keele, L. (2014). "mediation: R package for Causal Mediation Analysis", Journal of Statistical Software, Vol. 59, No. 5, pp. 1-38.

Imai, K., Tingley, D. and Yamamoto, T. (2012) Experimental Designs for Identifying Causal Mechanisms. Journal of the Royal Statistical Society, Series A (Statistics in Society)"

Imai, K., Keele, L., Tingley, D. and Yamamoto, T. (2011). Unpacking the Black Box of Causality: Learning about Causal Mechanisms from Experimental and Observational Studies, American Political Science Review, Vol. 105, No. 4 (November), pp. 765-789.

Imai, K., Keele, L. and Yamamoto, T. (2010) Identification, Inference, and Sensitivity Analysis for Causal Mediation Effects, Statistical Science, Vol. 25, No. 1 (February), pp. 51-71.

Imai, K., Keele, L., Tingley, D. and Yamamoto, T. (2009) Causal Mediation Analysis Using R" in Advances in Social Science Research Using R, ed. H. D. Vinod New York: Springer.

See Also

[mediate](#), [summary.mediate.design](#)

Examples

```
data(CEDdata)

res <- mediate.ced("Y2", "M1", "M2", "T1", "Z", CEDdata, sims = 100)
summary(res)
```

mediate.pd	<i>Estimating Average Causal Mediation Effects under the Parallel Design</i>
------------	--

Description

'mediate.pd' estimates the average causal mediation effects for the parallel design. If a treatment-mediator interaction is allowed then the nonparametric sharp bounds are calculated. If a treatment-mediator interaction is not allowed then the estimates of the (point-identified) effects are computed along with bootstrapped confidence intervals.

Usage

```
mediate.pd(outcome, mediator, treat, manipulated, data, NINT = TRUE,  
           sims = 1000, conf.level = 0.95)
```

Arguments

outcome	name of the outcome variable in 'data'.
mediator	name of the mediator in 'data'. The variable must be binary (factor or numeric 0/1).
treat	name of the treatment variable in 'data'. Must be binary (factor or numeric 0/1).
manipulated	name of the binary design indicator in 'data', indicating whether observation received mediator manipulation.
data	a data frame containing all the above variables.
NINT	whether the no interaction assumption is made.
sims	number of bootstrap simulations. Only relevant when 'NINT' is TRUE.
conf.level	level of the returned two-sided confidence intervals. Only relevant when 'NINT' is TRUE.

Details

This function calculates average causal mediation effects (ACME) for the parallel design. The design consists of two randomly separated experimental arms, indicated by 'manipulated'. In one the treatment is randomized and the mediator and outcome variables are measured. In the second arm, the treatment is randomized, the mediator is perfectly manipulated and the outcome variable is measured.

Under the parallel design, the ACME is identified when it is assumed that there is no interaction between the treatment and mediator. Without the assumption the nonparametric sharp bounds can be computed. See Imai, Tingley and Yamamoto (2012) for details.

Value

mediate.pd returns an object of class "mediate.design", a list that contains the components listed below.

The function summary (i.e., summary.mediate.design) can be used to obtain a table of the results.

d0, d1	point estimates or bounds for the average causal mediation effects under the control and treatment conditions, respectively.
d0.ci, d1.ci	confidence intervals for the effects based on the nonparametric bootstrap. The confidence level is set at the value specified in 'conf.level'. Only exists when 'NINT' is TRUE.
nobs	number of observations used.
conf.level	confidence level used. Only exists when 'NINT' is TRUE.
sims	number of bootstrap simulations used for confidence interval calculation. Only exists when 'NINT' is TRUE.
design	indicates the design. "PD.NINT" if no interaction assumed; "PD" if interaction allowed.

Author(s)

Dustin Tingley, Harvard University, <dtingley@gov.harvard.edu>; Teppei Yamamoto, Massachusetts Institute of Technology, <tepei@mit.edu>.

References

- Tingley, D., Yamamoto, T., Hirose, K., Imai, K. and Keele, L. (2014). "mediation: R package for Causal Mediation Analysis", Journal of Statistical Software, Vol. 59, No. 5, pp. 1-38.
- Imai, K., Tingley, D. and Yamamoto, T. (2012) Experimental Designs for Identifying Causal Mechanisms. Journal of the Royal Statistical Society, Series A (Statistics in Society)"
- Imai, K., Keele, L., Tingley, D. and Yamamoto, T. (2011). Unpacking the Black Box of Causality: Learning about Causal Mechanisms from Experimental and Observational Studies, American Political Science Review, Vol. 105, No. 4 (November), pp. 765-789.
- Imai, K., Keele, L. and Yamamoto, T. (2010) Identification, Inference, and Sensitivity Analysis for Causal Mediation Effects, Statistical Science, Vol. 25, No. 1 (February), pp. 51-71.
- Imai, K., Keele, L., Tingley, D. and Yamamoto, T. (2009) Causal Mediation Analysis Using R" in Advances in Social Science Research Using R, ed. H. D. Vinod New York: Springer.

See Also

[mediate](#), [summary.mediate.design](#)

Examples

```
data(boundsdata)

bound2 <- mediate.pd("out", "med", "ttt", "manip", boundsdata,
  NINT = TRUE, sims = 100, conf.level=.95)
```



```
summary(bound2)

bound2.1 <- mediate.pd("out", "med", "ttt", "manip", boundsdata, NINT = FALSE)
summary(bound2.1)
```

mediate.ped	<i>Computing Bounds on Average Causal Mediation Effects under the Parallel Encouragement Design</i>
-------------	---

Description

'mediate.ped' computes the nonparametric bounds on the average causal mediation effects for the parallel encouragement design.

Usage

```
mediate.ped(outcome, mediator, treat, encourage, data)
```

Arguments

outcome	name of the outcome variable in 'data'.
mediator	name of the mediator in 'data'. The variable must be binary (factor or numeric 0/1).
treat	name of the treatment variable in 'data'. Must be binary (factor or numeric 0/1).
encourage	name of the encouragement variable in 'data'. The variable must be a numeric vector taking on either -1, 0, or 1.
data	a data frame containing all the above variables.

Details

This function calculates average causal mediation effects (ACME) for the parallel encouragement design.

In the design two experimental arms are used. In one the treatment is randomized and the mediator and outcome variables are measured. In the second arm the treatment is randomized, the mediator is randomly encouraged either up or down, and the outcome variable is measured.

Two type of causal quantities are estimated: the population ACME and the complier ACME. The latter refers to the subpopulation of the units for whom the encouragement has its intended effect, and the width of its bounds are tighter than that of the population ACME. See Imai, Tingley and Yamamoto (2012) for details.

Value

mediate.pd returns an object of class "mediate.design", a list that contains the components listed below.

The function summary (i.e., summary.mediate.design) can be used to obtain a table of the results.

d0, d1	estimated nonparametric sharp bounds for the population ACME under the control and treatment conditions.
d0.p, d1.p	estimated nonparametric sharp bounds for the complier ACME under the control and treatment conditions.
nobs	number of observations used.
design	indicates the design. Always equals "PED".

Author(s)

Dustin Tingley, Harvard University, <dtingley@gov.harvard.edu>; Teppei Yamamoto, Massachusetts Institute of Technology, <tepei@mit.edu>.

References

Tingley, D., Yamamoto, T., Hirose, K., Imai, K. and Keele, L. (2014). "mediation: R package for Causal Mediation Analysis", Journal of Statistical Software, Vol. 59, No. 5, pp. 1-38.

Imai, K., Tingley, D. and Yamamoto, T. (2012) Experimental Designs for Identifying Causal Mechanisms. Journal of the Royal Statistical Society, Series A (Statistics in Society)"

Imai, K., Keele, L., Tingley, D. and Yamamoto, T. (2011). Unpacking the Black Box of Causality: Learning about Causal Mechanisms from Experimental and Observational Studies, American Political Science Review, Vol. 105, No. 4 (November), pp. 765-789.

Imai, K., Keele, L. and Tingley, D. (2010) A General Approach to Causal Mediation Analysis, Psychological Methods, Vol. 15, No. 4 (December), pp. 309-334.

Imai, K., Keele, L. and Yamamoto, T. (2010) Identification, Inference, and Sensitivity Analysis for Causal Mediation Effects, Statistical Science, Vol. 25, No. 1 (February), pp. 51-71.

Imai, K., Keele, L., Tingley, D. and Yamamoto, T. (2009) "Causal Mediation Analysis Using R" in Advances in Social Science Research Using R, ed. H. D. Vinod New York: Springer.

See Also

[mediate](#), [medsens](#), [plot.mediate](#), [summary.mediate](#), [mediations](#)

Examples

```
data(boundsdata)
```

```
bound3 <- mediate.ped("out.enc", "med.enc", "ttt", "enc", boundsdata)
summary(bound3)
```

mediate.sed	<i>Estimating Average Causal Mediation Effects under the Single Experiment Design</i>
-------------	---

Description

Estimating Average Causal Mediation Effects under the Single Experiment Design

Usage

```
mediate.sed(outcome, mediator, treat, data, SI = FALSE, sims = 1000,
  conf.level = 0.95, boot = FALSE)
```

Arguments

outcome	name of the outcome variable in 'data'. The variable must be binary (factor or numeric 0/1) if 'SI' is FALSE.
mediator	name of the mediator in 'data'. The variable must be binary (factor or numeric 0/1) if 'SI' is FALSE and discrete if TRUE.
treat	name of the treatment variable in 'data'. Must be binary (factor or numeric 0/1).
data	a data frame containing all the above variables.
SI	whether the sequential ignorability assumption is made.
sims	number of bootstrap simulations. Only relevant when 'SI' is TRUE.
conf.level	level of the returned two-sided confidence intervals. Only relevant when 'SI' is TRUE.
boot	a logical value. if 'FALSE' a large sample Delta method approximation is used for confidence intervals; if 'TRUE' nonparametric bootstrap will be used. Default is 'FALSE'. Only relevant if 'SI' is TRUE.

Details

'mediate.sed' estimates average causal mediation effects for the single experiment design. The two options are to use either the sequential ignorability (SI) assumption in which nonparametric estimates of the average causal mediation effect are produced, or, to relax the SI assumption and to calculate the nonparametric bounds on the average causal mediation effect.

This function calculates average causal mediation effects (ACME) for the single experiment design, where the treatment is randomized and the mediator/outcome variables are measured. The user specifies whether they want non-parametric point estimates based on the sequential ignorability (SI) assumption, or nonparametric bounds without the SI assumption.

Value

mediate.sed returns an object of class "mediate.design", a list that contains the components listed below.

The summary function can be used to obtain a table of the results.

d0, d1	point estimates or lower/upper bounds for causal mediation effects under the control and treatment conditions, respectively.
d0.ci, d1.ci	confidence intervals for average causal mediation effects for the nonparametric estimates. The confidence level is set at the value specified in 'conf.level'. The value exists only when 'SI' is TRUE.
boot	logical, the 'boot' argument used.
conf.level	the confidence level used.
sims	number of bootstrap simulations used for confidence interval calculation.
nobs	number of observations used.
design	indicates the design. Equals either "SED.NP.SI" or "SED.NP.NOSI".

Author(s)

Dustin Tingley, Harvard University, <dtingley@gov.harvard.edu>; Teppei Yamamoto, Massachusetts Institute of Technology, <teppeii@mit.edu>.

References

- Tingley, D., Yamamoto, T., Hirose, K., Imai, K. and Keele, L. (2014). "mediation: R package for Causal Mediation Analysis", Journal of Statistical Software, Vol. 59, No. 5, pp. 1-38.
- Imai, K., Tingley, D. and Yamamoto, T. (2012) Experimental Designs for Identifying Causal Mechanisms. Journal of the Royal Statistical Society, Series A (Statistics in Society)"
- Imai, K., Keele, L., Tingley, D. and Yamamoto, T. (2011). Unpacking the Black Box of Causality: Learning about Causal Mechanisms from Experimental and Observational Studies, American Political Science Review, Vol. 105, No. 4 (November), pp. 765-789.
- Imai, K., Keele, L. and Yamamoto, T. (2010) Identification, Inference, and Sensitivity Analysis for Causal Mediation Effects, Statistical Science, Vol. 25, No. 1 (February), pp. 51-71.
- Imai, K., Keele, L., Tingley, D. and Yamamoto, T. (2009) "Causal Mediation Analysis Using R" in Advances in Social Science Research Using R, ed. H. D. Vinod New York: Springer.

See Also

[mediate](#), [summary.mediate.design](#)

Examples

```
# Example 1: Bounds without SI assumption

data(boundsdata)

data.SED <- subset(boundsdata, manip == 0)
```

```

bound1 <- mediate.sed("out", "med", "ttt", data.SED, SI=FALSE)
summary(bound1)

# Example 2: Nonparametric estimate of ACME under SI assumption
# Example with JOBS II Field Experiment

data(jobs)

foo.1 <- mediate.sed("depress2", "job_disc", "treat", jobs, SI=TRUE)
summary(foo.1)

foo.2 <- mediate.sed("depress2", "job_disc", "treat", jobs, SI=TRUE, boot=TRUE)
summary(foo.2)

```

mediate_tsls	<i>Two-stage Least Squares Estimation of the Average Causal Mediation Effects</i>
--------------	---

Description

Estimate quantities for causal mediation analysis using an instrumental variable estimator.

Usage

```

mediate_tsls(model.m, model.y, treat = "treat.name", conf.level = 0.95,
  robustSE = FALSE, cluster = NULL, boot = FALSE, sims = 1000,
  est_se = TRUE, ...)

```

Arguments

model.m	a fitted model object for mediator, of class <code>lm</code> .
model.y	a fitted model object for outcome, of class <code>lm</code> .
treat	a character string indicating the name of the treatment variable used in the models.
conf.level	level of the returned two-sided confidence intervals. Default is to return the 2.5 and 97.5 percentiles of the simulated quantities.
robustSE	a logical value. If 'TRUE', heteroskedasticity-consistent standard errors will be used. Default is 'FALSE'.
cluster	a variable indicating clusters for standard errors. Note that this should be a vector of cluster indicators itself, not a character string for the name of the variable.
boot	a logical value. if FALSE analytic confidence intervals based on Aroian (1947) will be used; if TRUE nonparametric bootstrap will be used. Default is FALSE.
sims	number of Monte Carlo draws for nonparametric bootstrap.
est_se	estimate standard errors. Primarily for internal use. Default is TRUE.
...	other arguments passed to <code>vcovCL</code> in the <code>sandwich</code> package: typically the type argument for heteroskedasticity-consistent standard errors.

Value

mediate returns an object of class "mediate", "mediate.tsls", a list that contains the components listed below.

The function summary can be used to obtain a table of the results.

d1	point estimate for average causal mediation effects.
d1.ci	confidence intervals for average causal mediation effect. The confidence level is set at the value specified in 'conf.level'.
z0	point estimates for average direct effect.
z0.ci	confidence intervals for average direct effect.
z0.p	two-sided p-values for average causal direct effect.
n0	the "proportions mediated", or the size of the average causal mediation effect relative to the total effect.
n0.ci	confidence intervals for the proportion mediated.
n0.p	two-sided p-values for proportion mediated.
tau.coef	point estimate for total effect.
tau.ci	confidence interval for total effect.
tau.p	two-sided p-values for total effect.
boot	logical, the boot argument used.
treat	a character string indicating the name of the 'treat' variable used.
mediator	a character string indicating the name of the 'mediator' variable used.
INT	a logical value indicating whether the model specification allows the effects to differ between the treatment and control conditions.
conf.level	the confidence level used.
model.y	the outcome model used.
model.m	the mediator model used.
nobs	number of observations in the model frame for 'model.m' and 'model.y'. May differ from the numbers in the original models input to 'mediate' if 'dropobs' was 'TRUE'.
cluster	the clusters used.

References

Aroian, L. A. 1947. The probability function of the product of two normally distributed variables. *Annals of Mathematical Statistics,* 18, 265-271.

Examples

```
# Generate data. We use TSLS to address unobserved confounding (n).
set.seed(123)
sims <- 1000
dat <- data.frame(z = sample(0:1, sims, replace = TRUE),
                  t = sample(0:1, sims, replace = TRUE))
```

```

dat$n <- rnorm(sims, mean = 1)
dat$m <- rnorm(sims, mean = dat$z * 0.3 + dat$t * 0.2 + dat$n * 0.7, sd = 0.2)
dat$y <- rnorm(sims, mean = 5 + dat$t + dat$m * (-3) + dat$n, sd = 1)
model.m <- lm(m ~ t + z, data = dat)
model.y <- lm(y ~ t + m, data = dat)
cluster <- factor(sample(1:3, sims, replace = TRUE))
med <- mediate_tsIs(model.m, model.y, cluster = cluster, treat = "t")
summary(med)

```

mediation	<i>mediation: Parametric and non parametric mediation analysis.</i>
-----------	---

Description

mediation: Parametric and non parametric mediation analysis.

mediations	<i>Causal Mediation Analysis for Multiple Outcome/Treatment/Mediator Combinations</i>
------------	---

Description

'mediations' can be used to process a set of outcome/treatment/mediator combinations through the [mediate](#) function to produce a series of causal mediation analysis results.

Usage

```

mediations(datasets, treatment, mediators, outcome, covariates = NULL,
  families = c("gaussian", "gaussian"), tau.m = 0.5, tau.y = 0.5,
  LowerY = NULL, UpperY = NULL, interaction = FALSE,
  conf.level = 0.95, sims = 500, boot = FALSE, weights = NULL, ...)

```

Arguments

datasets	a named list of data frames. Each data frame has a separate treatment variable. The names of each data frame must begin with the exact name of the treatment variable that is contained in that dataset (see example below).
treatment	a vector of character strings indicating the names of the treatment variables, with length equal to the length of 'datasets'. Each treatment variable must be included in the data frame listed in the same position of list 'datasets' and its name must match the first part of the corresponding data frame.
mediators	a vector of character strings indicating the names of the mediators contained within each data frame. All of the mediators will be used with each treatment variable and hence must be included in each data frame of 'datasets'.

outcome	a vector of character strings indicating the names of the outcome variables contained within each data frame. All of the outcomes will be used with each treatment variable and must be in each data frame.
covariates	a character string representing the set of pre-treatment covariate names (as they appear in the data frame) to be included in each model. The value must take the form of standard model formula, with each additive component separated by "+", etc. (see example below). All covariates must be in each data frame. Default is 'NULL'.
families	a vector of length two specifying the types of the mediator and outcome models. Currently only supports "gaussian" (for linear regression), "binomial" (for binary probit), "oprobit" (for ordered probit) and "quantile" (for quantile regression, see 'tau'). For the outcome the tobit model ("tobit") is also available in addition to the mediator model options.
tau.m	a numeric value specifying the quantile to be used for a quantile regression for the mediator model. Only relevant if the first element of 'families' is "quantile". See <code>rq</code> .
tau.y	a numeric value specifying the quantile to be used for a quantile regression for the outcome model. Only relevant if the second element of 'families' is "quantile". See <code>rq</code> .
LowerY	a numeric value indicating the lower bound for the tobit outcome model. See <code>tobit</code> .
UpperY	a numeric value indicating the upper bound for the tobit outcome model. See <code>tobit</code> .
interaction	a logical value indicating whether the treatment and mediator variables should be interacted. This will apply to applications of <code>mediate</code> to all the treatment/mediator/outcome combinations.
conf.level	confidence level used in each application of the <code>mediate</code> function.
sims	an integer indicating the desired number of simulations for inference. This will apply to all applications of 'mediate' to all the treatment/mediator/outcome combinations.
boot	a logical value, indicating whether or not nonparametric bootstrap should be used in each <code>mediate</code> application.
weights	a single valued vector of a character string indicating a weight variable to be used in all model fitting.
...	other arguments passed to <code>mediate</code> , such as 'robustSE', 'dropobs', etc.

Details

This function processes multiple treatment/mediators/outcome variable combinations to produce a collected set of output ready for analysis or graphing. In principle, this is a function designed to facilitate running causal mediation analyses on multiple models that share the same basic specification (i.e. the types of parametric models and the set of pre-treatment covariates) except the treatment, mediator and outcome variables can differ across specifications. The function works by looping over a set of data frames that are pre-loaded into the workspace. Each one of these data frames has a specific treatment variable that is used for analysis with that data frame. Then the code

runs causal mediation analysis via `mediate` on every combination of the treatment, mediator, and outcomes specified in these arguments. This allows the users to explore whether different mediators transmit the effect of the treatment variable on a variety of outcome variables. A single set of pre-treatment control variables can be specified in `'covariates'`, which will be used throughout.

The `'mediations'` function can be used with either multiple mediators and a single outcome, a single mediator and multiple outcomes, or multiple mediators and outcomes. For example, with three different treatments, user will create three different data frames, each containing a treatment variable. In addition, if there are also four different mediators, each of these will be contained in each data frame, along with the outcome variable. The function will estimate all of the combinations of treatment variables and mediators instead of separate lines of code being written for each one.

Individual elements of the output list (see "Value") may be passed through `summary` and `plot` for tabular and graphical summaries of the results. Alternatively, the entire output may be directly passed to `summary` or `plot` for all results to be inspected.

The default value of `'covariates'` is `'NULL'` and no covariate will be included in either mediator or outcome models without a custom value. It should be noted that users typically should have pre-treatment covariates to make the sequential ignorability assumption more plausible.

There are several limitations to the code. First, it works only with a subset of the model types that will be accommodated if `'mediate'` is used individually (see the `'families'` argument above for details). Second, one cannot specify separate sets of covariates for different treatment/mediator/outcome combinations. Users should use `'mediate'` separately for individual models if more flexibility is required in their specific applications.

Value

An object of class "mediations" (or "mediations.order" if the outcome model is ordered probit), a list of "mediate" ("mediate.order") objects produced by applications of `mediate` for the specified treatment/mediator/outcome combinations. The elements are named based on the names of the outcome, treatment, and mediator variables, each separated by a "." (see example below).

Author(s)

Dustin Tingley, Harvard University, <dtingley@gov.harvard.edu>; Teppei Yamamoto, Massachusetts Institute of Technology, <tepei@mit.edu>.

See Also

`mediate`, `summary.mediations`, `plot.mediations`, `rq`, `tobit`.

Examples

```
## Not run:
# Hypothetical example

datasets <- list(T1 = T1, T2 = T2)
# List of data frames corresponding to the two different treatment variables
#"T1vsCont" and "T2vsCont".
# Each data set has its respective treatment variable.

mediators <- c("M1", "M2")
```

```

# Vector of mediator names, all included in each data frame.

outcome <- c("Ycont1", "Ycont2")
# Vector of outcome variable names, again all included in each data frame.

treatment <- c("T1vsCont", "T2vsCont")
# Vector of treatment variables names; must begin with identical strings with dataset
# names in 'datasets'.

covariates <- c("X1 + X2")
# Set of covariates (in each data set), entered using the standard model formula format.

x <- mediations(datasets, treatment, mediators, outcome, covariates,
  families=c("gaussian", "gaussian"), interaction=FALSE,
  conf.level=.90, sims=50)
# Runs 'mediate' iteratively for each variable combinations, with 'lm' on both mediator
# and outcome model.

summary(x) # tabular summary of results for all model combinations
plot(x) # graphical summary of results for all model combinations at once

plot(x$Ycont1.T1vsCont.M1)
# Individual 'mediate' outputs are stored as list elements and can be
# accessed using the usual "$" operator.

## End(Not run)

```

medsens

Sensitivity Analysis for Causal Mediation Effects

Description

'medsens' is used to perform sensitivity analysis on the average causal mediation effects and direct effects for violations of the sequential ignorability assumption. The function takes output from 'mediate' and calculates the true average causal mediation effects and direct effects for different values of the sensitivity parameter representing the degree of the sequential ignorability violation.

Usage

```

medsens(x, rho.by = 0.1, sims = 1000,
  eps = sqrt(.Machine$double.eps), effect.type = c("indirect",
  "direct", "both"))

```

Arguments

x an object of class 'mediate', typically an output from the mediate function.

rho.by a numeric value between 0 and 1 indicating the increment for the sensitivity parameter, rho.

<code>sims</code>	the number of Monte Carlo draws for the calculation of confidence intervals. Only used in cases where either the mediator or outcome variable is binary.
<code>eps</code>	convergence tolerance parameter for the iterative FGLS. Only used when both the mediator and outcome models are linear.
<code>effect.type</code>	a character string indicating which effect(s) to be analyzed. Default is "indirect".

Details

This is the workhorse function for sensitivity analyses for average causal mediation effects. The sensitivity analysis can be used to assess the robustness of the findings from `mediate` to the violation of sequential ignorability, the crucial identification assumption necessary for the estimates to be valid. The analysis proceeds by quantifying the degree of sequential ignorability violation as the correlation between the error terms of the mediator and outcome models, and then calculating the true values of the average causal mediation effect for given values of this sensitivity parameter, ρ . The original findings are deemed sensitive if the true effects are found to vary widely as function of ρ .

The sensitivity analysis is only implemented for the following three model combinations: linear mediator and outcome models (both of class 'lm'), binary probit mediator (fitted via 'glm' with family "binomial" and link "probit") and linear outcome models, and linear mediator and binary probit outcome models. In addition, the binary outcome model cannot include a treatment-mediator interaction term. An error is returned if the 'mediate' object in 'x' is based on other model combinations. As of version 3.0, the sensitivity analysis can also be conducted with respect to the average direct effect by setting 'effect.type' to "direct" (or "both" if results for the average causal mediation effect are also desired).

Users should note that computation can take significant time for `medsens`. Setting 'rho.by' to a larger number significantly decreases computational time, as does decreasing 'eps' (for the linear-linear case) or the number of simulations 'sims' (for the binary-linear and linear-binary cases).

Value

`medsens` returns an object of class "medsens", a list containing the following elements. Some of these elements are not available depending on the 'effect.type' argument specified by the user. The output can then be passed to the `summary` (i.e., `summary.medsens`) and `plot` (i.e., `plot.medsens`) functions to produce tabular and graphical summaries of the results.

<code>d0, d1</code>	vectors of point estimates for average causal mediation effects under the control and treatment conditions for each value of sensitivity parameter ρ .
<code>upper.d0, lower.d0, upper.d1, lower.d1</code>	vectors of upper and lower confidence limits for average causal mediation effect under the control and treatment conditions for each value of ρ .
<code>z0, z1</code>	vectors of point estimates for average direct effect under the control and treatment conditions for each value of sensitivity parameter ρ .
<code>upper.z0, lower.z0, upper.z1, lower.z1</code>	vectors of upper and lower confidence limits for average direct effect under the control and treatment conditions for each value of ρ .
<code>tau</code>	a vector of point estimates for total effect for each value of ρ . Only present when the outcome model is binary.

<code>upper.tau, lower.tau</code>	vectors of upper and lower confidence limits for total effect. Only present when the outcome model is binary.
<code>nu</code>	a vector of point estimates for the proportion mediated for each value of rho. Only present when the outcome model is binary.
<code>upper.nu, lower.nu</code>	vectors of upper and lower confidence limits for the proportion mediated. Only present when the outcome model is binary.
<code>rho</code>	a numeric vector containing the values of sensitivity parameter rho used.
<code>rho.by</code>	a numeric value indicating the increment of rho used.
<code>sims</code>	a numeric value indicating the number of Monte Carlo draws used.
<code>err.cr.d, err.cr.z</code>	the values of rho with which the average causal mediation and direct effects are zero. Vectors of length two if 'INT' is 'TRUE'; numeric values otherwise.
<code>ind.d0, ind.d1, ind.z0, ind.z1</code>	vectors of 0s/1s, indicating whether the confidence intervals of d0, d1, z0 and z1 do not cover zero for each value of rho.
<code>R2star.prod</code>	a numeric vector containing the values of the products of the two "R square stars", representing the proportions of residual variance in the mediator and outcome explained by the hypothesized unobserved confounder. The values correspond to those of rho. See plot.medsens for details.
<code>R2tilde.prod</code>	a numeric vector containing the values of the products of the two "R square tildes", representing the proportions of total variance in the mediator and outcome explained by the hypothesized unobserved confounder. The values correspond to those of rho. See plot.medsens for details.
<code>R2star.d.thresh, R2star.z.thresh</code>	the values of the product of "R square stars" for which the average causal mediation and direct effects are zero, respectively.
<code>R2tilde.d.thresh, R2tilde.z.thresh</code>	the values of the product of "R square tildes" for which the average causal mediation and direct effects are zero, respectively.
<code>r.square.y, r.square.m</code>	the usual R square statistics for the outcome and mediator models.
<code>INT</code>	a logical value indicating whether interaction between the treatment and mediator is allowed in the original mediate object.
<code>conf.level</code>	the confidence level used.
<code>effect.type</code>	the 'effect.type' argument used.
<code>type</code>	a character string indicating the type of the mediator and outcome models used. Currently either "ct" (linear mediator and outcome models), 'bm' (binary mediator and linear outcome models) or 'bo' (linear mediator and binary outcome models).
<code>robustSE</code>	'TRUE' or 'FALSE'.
<code>cluster</code>	the clusters used.

Author(s)

Dustin Tingley, Harvard University, <dtingley@gov.harvard.edu>; Teppei Yamamoto, Massachusetts Institute of Technology, <tepei@mit.edu>; Jaquilyn Waddell-Boie, Princeton University, <jwaddell@princeton.edu>; Kentaro Hirose, Princeton University, <hirose@princeton.edu>; Luke Keele, Penn State University, <ljk20@psu.edu>; Kosuke Imai, Princeton University, <kimai@princeton.edu>.

References

- Tingley, D., Yamamoto, T., Hirose, K., Imai, K. and Keele, L. (2014). "mediation: R package for Causal Mediation Analysis", *Journal of Statistical Software*, Vol. 59, No. 5, pp. 1-38.
- Imai, K., Keele, L., Tingley, D. and Yamamoto, T. (2011). Unpacking the Black Box of Causality: Learning about Causal Mechanisms from Experimental and Observational Studies, *American Political Science Review*, Vol. 105, No. 4 (November), pp. 765-789.
- Imai, K., Keele, L. and Tingley, D. (2010) A General Approach to Causal Mediation Analysis, *Psychological Methods*, Vol. 15, No. 4 (December), pp. 309-334.
- Imai, K., Keele, L. and Yamamoto, T. (2010) Identification, Inference, and Sensitivity Analysis for Causal Mediation Effects, *Statistical Science*, Vol. 25, No. 1 (February), pp. 51-71.
- Imai, K., Keele, L., Tingley, D. and Yamamoto, T. (2009) "Causal Mediation Analysis Using R" in *Advances in Social Science Research Using R*, ed. H. D. Vinod New York: Springer.

See Also

[mediate](#), [summary.medsens](#), [plot.medsens](#).

Examples

```
# Examples with JOBS II Field Experiment

# **For illustration purposes a small number of simulations are used**

data(jobs)

#####
# Example 1: Binary treatment
#####
# Fit parametric models
b <- lm(job_seek ~ treat + econ_hard + sex + age, data=jobs)
c <- lm(depress2 ~ treat + job_seek + econ_hard + sex + age, data=jobs)

# Pass model objects through mediate function
med.cont <- mediate(b, c, treat="treat", mediator="job_seek", sims=50)
# med.cont <- mediate(b, c, treat="treat", mediator="job_seek", sims=50, robustSE = T)
# jobs$cluster <- rep(1:30, each = 30)[-1]
# med.cont <- mediate(b, c, treat="treat", mediator="job_seek", sims=50, cluster = jobs$cluster)

# Pass mediate output through medsens function
sens.cont <- medsens(med.cont, rho.by=.1, eps=.01, effect.type="both")

# Use summary function to display results
```

```

summary(sens.cont)

# Plot true ACMEs and ADEs as functions of rho
par.orig <- par(mfrow = c(2,2))
plot(sens.cont, main="JOBS", ylim=c(-.2,.2))

# Plot true ACMEs and ADEs as functions of "R square tildes"
plot(sens.cont, sens.par="R2", r.type="total", sign.prod="positive")
par(par.orig)

#####
# Example 2: Categorical treatment
#####
## Not run:

# Purely for illustration, think of educ as a ``treatment``

b <- lm(job_seek ~ educ + sex, data=jobs)
c <- lm(depress2 ~ educ + job_seek + sex, data=jobs)

# compare two categories of educ --- gradwk and somcol
med.cont <- mediate(b, c, treat="educ", mediator="job_seek", sims=50,
                   control.value = "gradwk", treat.value = "somcol")
sens.cont <- medsens(med.cont, rho.by=.1, eps=.01, effect.type="both")
summary(sens.cont)

## End(Not run)

```

Description

'multimed' is used for causal mediation analysis when post-treatment mediator-outcome confounders, or alternative mediators causally preceding the mediator of interest, exist in the hypothesized causal mechanisms. It estimates the average causal mediation effects (indirect effects) and the average direct effects under the homogeneous interaction assumption based on a varying-coefficient linear structural equation model. The function also performs sensitivity analysis with respect to the violation of the homogenous interaction assumption. The function can be used for both the single experiment design and the parallel design.

Usage

```

multimed(outcome, med.main, med.alt = NULL, treat, covariates = NULL,
         experiment = NULL, data, design = c("single", "parallel"),
         sims = 1000, R2.by = 0.01, conf.level = 0.95, weight = NULL)

```

Arguments

outcome	name of the outcome variable in 'data'.
med.main	name of the mediator of interest. Under the parallel design this is the only mediator variable used in the estimation.
med.alt	vector of character strings indicating the names of the post-treatment confounders, i.e., the alternative mediators affecting both the main mediator and outcome. Not needed for the parallel design.
treat	name of the treatment variable in 'data'.
covariates	vector of character strings representing the names of the pre-treatment covariates. Cannot be used for the parallel design.
experiment	name of the binary indicator whether 'med.main' is randomly manipulated or not. Only used under the parallel design.
data	a data frame containing all the above variables.
design	experimental design used. Defaults to 'single'.
sims	number of bootstrap samples used for the calculation of confidence intervals.
R2.by	increment for the "R square tilde" parameter, i.e. the sensitivity parameter representing the proportion of residual outcome variance explained by heterogeneity in treatment-mediator interactions. Must be a numeric value between 0 and 1.
conf.level	level to be used for confidence intervals.
weight	name of the weights in 'data'.

Details

This function implements the framework proposed by Imai and Yamamoto (2012) for the estimation and sensitivity analysis for multiple causal mechanisms. It estimates the average causal mediation effects (indirect effects) with respect to the mediator of interest ('med.main'), i.e., the portion of the treatment effect on the outcome that is transmitted through that mediator, as well as the average direct effects, i.e., the portion of the treatment effect on the outcome that is not transmitted through the main mediator. Unlike the "standard" causal mediation analysis implemented by [mediate](#) and [medsens](#), this framework allows the existence of post-treatment covariates that confound the relationship between the main mediator and the outcome, or equivalently, alternative mediators ('med.alt') that causally precede the main mediator.

When the parallel design was used for the experiment (i.e. when the experiment contained an additional randomly assigned group for which both the treatment and the mediator were randomized), there is no need to specify a particular post-treatment confounder, for any such confounder (observed or unobserved) is allowed to exist by virtue of the design. Similarly, no observed covariates need to be included. The function instead requires an additional variable ('experiment') indicating whether the mediator was randomly manipulated for the unit.

The estimation and sensitivity analysis are both based on a varying-coefficient linear structural equations model, which assumes additivity but allows for an arbitrary degree of heterogeneity in model coefficients across units and thus is substantially more flexible than a traditional SEM framework. For details see Imai and Yamamoto (2012).

The function produces two sets of results. First, point estimates of the average causal mediation effects and the average direct effects are calculated, along with their (percentile) bootstrap confidence

intervals. These estimates are based on the "homogeneous interaction" assumption, or the assumption that the degree of treatment-mediator interaction is constant across all units. The estimated total treatment effect is also reported.

Second, the bounds on the average causal mediation effects and the average direct effects are also estimated and computed for various degrees of interaction heterogeneity (i.e., violation of the identification assumption), which are represented by the values of three alternative sensitivity parameters. These parameters are: (1) σ , the standard deviation of the (varying) regression coefficient on the interaction term, (2) R^2_{star} , the proportion of the residual variance that would be explained by an additional term for interaction heterogeneity, and (3) R^2_{tilde} , the proportion of the total variance explained by such a term. The confidence region is also calculated, using the Imbens and Manski (2004) formula with bootstrap standard errors. Further details are given in the above reference.

Note that rows with missing values will be omitted from the calculation of the results. Also note that the treatment variable must be a numeric vector of 1 and 0 and that both mediators and outcome variable must be numeric. The pre-treatment covariates can be of any type that `lm` can handle as predictors.

Value

`multimed` returns an object of class "multimed", a list contains the following components. The object can be passed to the `summary` and `plot` method functions for a summary table and a graphical summary.

<code>sigma</code>	values of the σ sensitivity parameter at which the bounds and confidence intervals are evaluated.
<code>R2tilde</code>	values of the R^2_{tilde} parameter.
<code>R2star</code>	values of the R^2_{star} parameter.
<code>d1.lb, d0.lb, d.ave.lb</code>	lower bounds on the average causal mediation effects under treatment, control, and the simple average of the two, respectively, corresponding to the values of the sensitivity parameters listed above. Note that the first elements of these vectors equal the point estimates under the homogeneous interaction assumption.
<code>d1.ub, d0.ub, d.ave.ub</code>	upper bounds on the average causal mediation effects.
<code>d1.ci, d0.ci, d.ave.ci</code>	confidence intervals for the average causal mediation effects at different values of the sensitivity parameters.
<code>z1.lb, z0.lb, z.ave.lb</code>	lower bounds on the average direct effects under treatment, control, and the simple average of the two, respectively, corresponding to the values of the sensitivity parameters listed above. Note that the first elements of these vectors equal the point estimates under the homogeneous interaction assumption.
<code>z1.ub, z0.ub, z.ave.ub</code>	upper bounds on the average direct effects.
<code>z1.ci, z0.ci, z.ave.ci</code>	confidence intervals for the average direct effects at different values of the sensitivity parameters.

tau point estimate of the total treatment effect.
 tau.ci confidence interval for the total treatment effect.
 conf.level confidence level used for the calculation of the confidence intervals.

Author(s)

Teppei Yamamoto, Massachusetts Institute of Technology, <teppe@mit.edu>

References

Tingley, D., Yamamoto, T., Hirose, K., Imai, K. and Keele, L. (2014). "mediation: R package for Causal Mediation Analysis", *Journal of Statistical Software*, Vol. 59, No. 5, pp. 1-38.

Imai, K. and Yamamoto, T. (2012) Identification and Sensitivity Analysis for Multiple Causal Mechanisms: Revisiting Evidence from Framing Experiments, Unpublished manuscript.

See Also

[plot.multimed](#)

Examples

```
## Not run:
# Replicates Figure 3 (right column) of Imai and Yamamoto (2012)
# Note: # of bootstrap samples set low for quick illustration

data(framing)
Xnames <- c("age", "educ", "gender", "income")
res <- multimed("immigr", "emo", "p_harm", "treat", Xnames,
               data = framing, design = "single", sims = 10)
summary(res)
plot(res, type = "point")
plot(res, type = c("sigma", "R2-total"), tgroup = "average")

# Parallel design example using the simulated data of Imai, Tingley and Yamamoto (2012)

data(boundsdata)
res.para <- multimed(outcome = "out", med.main = "med", treat = "ttt", experiment = "manip",
                    data = boundsdata, design = "parallel", sims = 10)
summary(res.para)
plot(res.para, tg = "av")

## End(Not run)
```

plot.ivmediate	<i>Plotting Local Indirect, Direct, and Total Effects from Mediation Analysis with Treatment Noncompliance</i>
----------------	--

Description

Function to plot results from `ivmediate`. The vertical axis lists the local average causal mediation effects, local average natural direct effects and local average treatment effects and the horizontal axis indicates the respective magnitudes. Most standard options for plot function available.

Usage

```
## S3 method for class 'ivmediate'
plot(x, treatment = NULL, labels = NULL,
     effect.type = c("indirect", "direct", "total"),
     conf.level = x$conf.level[1], xlim = NULL, ylim = NULL,
     xlab = "", ylab = "", main = NULL, lwd = 1.5, cex = 0.85,
     col = "black", ...)
```

Arguments

<code>x</code>	object of class <code>ivmediate</code> as produced by <code>ivmediate</code> .
<code>treatment</code>	a character string indicating the baseline treatment value of the estimated local average causal mediation effect and direct effect to plot. Can be either "control", "treated" or "both". If 'NULL' (default), both sets of estimates are plotted.
<code>labels</code>	a vector of character strings indicating the labels for the estimated effects. The default labels will be used if NULL.
<code>effect.type</code>	a vector indicating which quantities of interest to plot. Default is to plot all three quantities (indirect, direct and total effects).
<code>conf.level</code>	a numeric value for the level of the confidence intervals to plot. Must equal one of the confidence levels used to produce the <code>ivmediate</code> object.
<code>xlim</code>	range of the horizontal axis.
<code>ylim</code>	range of the vertical axis.
<code>xlab</code>	label of the horizontal axis.
<code>ylab</code>	label of the vertical axis.
<code>main</code>	main title.
<code>lwd</code>	width of the horizontal bars for confidence intervals.
<code>cex</code>	size of the dots for point estimates.
<code>col</code>	color of the dots and horizontal bars for the estimates.
<code>...</code>	additional parameters passed to 'plot'.

Author(s)

Teppei Yamamoto, Massachusetts Institute of Technology, <teppe@mit.edu>.

References

Tingley, D., Yamamoto, T., Hirose, K., Imai, K. and Keele, L. (2014). "mediation: R package for Causal Mediation Analysis", *Journal of Statistical Software*, Vol. 59, No. 5, pp. 1-38.

Yamamoto, T. (2013). Identification and Estimation of Causal Mediation Effects with Treatment Noncompliance. Unpublished manuscript.

See Also

[ivmediate](#), [summary.ivmediate](#)

Examples

```
# Examples with JOBS II Field Experiment

# ** For illustration purposes a small number of simulations are used **

require(parallel)
require(MASS)

data(jobs)

a <- lm(comply ~ treat + sex + age + marital + nonwhite + educ + income,
        data = jobs)
b <- glm(job_dich ~ comply + treat + sex + age + marital + nonwhite + educ + income,
        data = jobs, family = binomial)
c <- lm(depress2 ~ job_dich * (comply + treat) + sex + age + marital + nonwhite + educ + income,
        data = jobs)

out <- ivmediate(a, b, c, sims = 50, boot = FALSE,
                enc = "treat", treat = "comply", mediator = "job_dich")

summary(out)
plot(out)
```

plot.mediate

Plotting Indirect, Direct, and Total Effects from Mediation Analysis

Description

Function to plot results from `mediate`. The vertical axis lists indirect, direct, and total effects and the horizontal axis indicates the respective magnitudes. Most standard options for plot function available.

Usage

```
## S3 method for class 'mediate'
plot(x, treatment = NULL, labels = NULL,
     effect.type = c("indirect", "direct", "total"), xlim = NULL,
     ylim = NULL, xlab = "", ylab = "", main = NULL, lwd = 1.5,
     cex = 0.85, col = "black", ...)
```

Arguments

x	object of class <code>mediate</code> or <code>mediate.order</code> as produced by <code>mediate</code> .
treatment	a character string indicating the baseline treatment value of the estimated causal mediation effect and direct effect to plot. Can be either "control", "treated" or "both". If 'NULL' (default), both sets of estimates are plotted if and only if they differ.
labels	a vector of character strings indicating the labels for the estimated effects. The default labels will be used if NULL.
effect.type	a vector indicating which quantities of interest to plot. Default is to plot all three quantities (indirect, direct and total effects).
xlim	range of the horizontal axis.
ylim	range of the vertical axis.
xlab	label of the horizontal axis.
ylab	label of the vertical axis.
main	main title.
lwd	width of the horizontal bars for confidence intervals.
cex	size of the dots for point estimates.
col	color of the dots and horizontal bars for the estimates.
...	additional parameters passed to 'plot'.

Value

`mediate` returns an object of class "mediate". The function `summary` is used to obtain a table of the results. The `plot` function plots these quantities.

Author(s)

Dustin Tingley, Harvard University, <dtingley@gov.harvard.edu>; Teppei Yamamoto, Massachusetts Institute of Technology, <tepei@mit.edu>.

References

Tingley, D., Yamamoto, T., Hirose, K., Imai, K. and Keele, L. (2014). "mediation: R package for Causal Mediation Analysis", *Journal of Statistical Software*, Vol. 59, No. 5, pp. 1-38.

Imai, K., Keele, L. and Tingley, D. (2010) A General Approach to Causal Mediation Analysis, *Psychological Methods*, Vol. 15, No. 4 (December), pp. 309-334.

Imai, K., Keele, L. and Yamamoto, T. (2010) Identification, Inference, and Sensitivity Analysis for Causal Mediation Effects, *Statistical Science*, Vol. 25, No. 1 (February), pp. 51-71.

Imai, K., Keele, L., Tingley, D. and Yamamoto, T. (2009) "Causal Mediation Analysis Using R" in *Advances in Social Science Research Using R*, ed. H. D. Vinod New York: Springer.

See Also

[mediate](#), [plot](#)

plot.mediate.mer	<i>Plotting Indirect, Direct, and Total Effects from Mediation Analysis of Multilevel Models</i>
------------------	--

Description

Function to plot group-specific effects derived from causal mediation analysis of multilevel models.

Usage

```
## S3 method for class 'mediate.mer'
plot(x, treatment = NULL, group.plots = FALSE,
     ask = prod(par("mfcol")) < nplots, xlim = NULL, ylim = NULL,
     xlab = "", ylab = "", main = NULL, lwd = 1.5, cex = 0.85,
     col = "black", ...)
```

Arguments

x	object of class 'mediate.mer' produced by 'mediate'.
treatment	a character string indicating the baseline treatment value of the estimated causal mediation effect and direct effect to plot. Can be either "control", "treated", or "both". If 'NULL' (default), both sets of estimates are plotted if and only if they differ.
group.plots	a logical value indicating whether group-specific effects should be plotted in addition to the population-averaged effects.
ask	a logical value. If 'TRUE', the user is asked for input before a new figure is plotted. Default is to ask only if the number of plots on current screen is fewer than necessary.
xlim	range of the horizontal axis.
ylim	range of the vertical axis.
xlab	label of the horizontal axis.
ylab	label of the vertical axis.
main	main title.
lwd	width of the horizontal bars for confidence intervals .
cex	size of the dots for point estimates.
col	color of the dots and horizontal bars for the estimates..
...	additional parameters passed to 'plot'.

Author(s)

Kentaro Hirose, Princeton University, <hirose@princeton.edu>.

References

Tingley, D., Yamamoto, T., Hirose, K., Imai, K. and Keele, L. (2014). "mediation: R package for Causal Mediation Analysis", Journal of Statistical Software, Vol. 59, No. 5, pp. 1-38.

See Also

[mediate](#), [summary.mediate.mer](#).

Examples

```
# Examples with JOBS II Field Experiment

# **For illustration purposes a small number of simulations are used**
## Not run:
data(jobs)
require(lme4)

# educ: mediator group
# occp: outcome group

# Varying intercept for mediator
model.m <- glmer(job_dich ~ treat + econ_hard + (1 | educ),
                 family = binomial(link = "probit"), data = jobs)

# Varying intercept and slope for outcome
model.y <- glmer(work1 ~ treat + job_dich + econ_hard + (1 + treat | occp),
                 family = binomial(link = "probit"), data = jobs)

# Output based on mediator group
multilevel <- mediate(model.m, model.y, treat = "treat",
                    mediator = "job_dich", sims=50, group.out="educ")

#plot(multilevel, group.plots=TRUE)

## End(Not run)
```

plot.mediations

Plotting Indirect, Direct, and Total Effects from Multiple Mediation Analyses

Description

Function to plot results from multiple causal mediation analyses conducted via the [mediations](#) function. Output is a series of plots generated via [plot.mediate](#) for each treatment/mediator/outcome combination specified in the input 'mediations' object.

Usage

```
## S3 method for class 'mediations'
plot(x, which = names(x), ask = prod(par("mfcol")))
  < length(which) && dev.interactive(), ...)
```

Arguments

x output from the `mediations` function.

which subset of `names(x)`, indicating which model combinations to be plotted. Default is to plot all.

ask logical. If 'TRUE', the user is asked for input before a new figure is plotted. Default is to ask only if the number of plots on current screen is fewer the number implied by 'which'.

... arguments passed to the `plot.mediate` function for individual plots.

Value

`mediations` returns an object of class `mediations`. The function `summary` is used to obtain a table of the results. The `plot` function instead plots these quantities. All additional parameters desired for the plotting of an output from `mediate` can be passed through.

Author(s)

Dustin Tingley, Harvard University, <dtingley@gov.harvard.edu>; Teppei Yamamoto, Massachusetts Institute of Technology, <teppe@mit.edu>.

See Also

[mediations](#), [plot.mediate](#), [plot](#).

plot.medsens

Plotting Results from Sensitivity Analysis for Causal Mediation Effects

Description

This function is used to plot results from the `'medsens'` function. Causal average mediation effects (as well as average direct effects and proportions mediated for selected models) can be plotted against two alternative sensitivity parameters.

Usage

```
## S3 method for class 'medsens'
plot(x, sens.par = c("rho", "R2"),
     r.type = c("residual", "total"), sign.prod = c("positive",
     "negative"), pr.plot = FALSE, smooth.effect = FALSE,
     smooth.ci = FALSE, ask = prod(par("mfcol"))) < nplots,
     levels = NULL, xlab = NULL, ylab = NULL, xlim = NULL,
     ylim = NULL, main = NULL, lwd = par("lwd"), ...)
```

Arguments

x	'medsens' object, typically output from medsens.
sens.par	a character string indicating the sensitivity parameter to be used. Default plots effects as functions of "rho". See Details.
r.type	type of the R square parameter to be used in "R2" plots. If "residual", effects are plotted against the proportions of the residual variances that are explained by the unobserved confounder. If "total", the proportions of the total variances are used as sensitivity parameters. Only relevant if 'sens.par' is "R2".
sign.prod	a value indicating the direction of hypothesized confounding in the sensitivity analysis. If "positive", the confounder is assumed to affect the mediator and outcome variable in the same direction; if "negative" the effects are assumed to be in opposite directions. Only relevant if sens.par is set to "R2".
pr.plot	a logical value. If 'TRUE', the "proportions mediated" will be plotted instead of the average causal mediation effects or direct effects. Currently only available if the object 'medsens' is based on the linear mediator and binary probit outcome models. Default is 'FALSE'.
smooth.effect	a logical value indicating whether the estimated mediation effects are smoothed via lowess before being plotted. Default is 'FALSE'.
smooth.ci	a logical value indicating whether the confidence bands are smoothed via lowess before being plotted. Default is 'FALSE'.
ask	a logical value. If 'TRUE', the user is asked for input before a new figure is plotted. Default is to ask only if the number of plots on current screen is fewer than necessary.
levels	vector of levels at which to draw contour lines. Only relevant if 'sens.par' is set to "R2". If 'NULL', default values in contour.default are used.
xlab	label for the x axis. Default labels are used if 'NULL'.
ylab	label for the y axis. Default labels are used if 'NULL'.
xlim	limits of the x axis. If 'NULL' default values are used.
ylim	limits of the y axis. If 'NULL' default values are used.
main	main title for the plot. If 'NULL', default titles are used.
lwd	width of the lines used in graphs.
...	additional arguments to be passed to plotting functions.

Details

The sensitivity analysis for causal mediation effects can be conducted in terms of two alternative sensitivity parameters, which both quantify the degree of violation of the sequential ignorability assumption. The "rho" parameter represents the correlation between the two error terms of the (latent) linear models for the mediator and outcome variables. A large value of rho indicates the existence of important common unobserved predictors for both the mediator and outcome and therefore a high degree of sequential ignorability violation, while a value close to zero implies there is no such confounders.

The resulting "rho" figures plot the estimated true values of ACME (or ADE, proportion mediated) against rho, along with the confidence intervals. When rho is zero, sequential ignorability holds,

so the estimated value at that point will be equal to the estimate returned by the `mediate`. The confidence level is determined by the `'conf.level'` value of the original `mediate` object.

The "R2" parameters represent the proportions of the mediator and outcome variances that are explained by an unobserved pre-treatment confounder, thereby indicating the importance of such a confounder in each model. When `'r.type'` is "residual", the R2 parameters represent the proportions of the residual variances of the mediator and outcome models that become explained by the inclusion of the hypothetical pre-treatment confounder. These are denoted as "R square stars" in Imai, Keele and Yamamoto (2010) and can also be specified as "star" or using a numeric value 1 in `medsens.plot`. When `'r.type'` is "total", the R2s represent the total mediator and outcome variances the unobserved confounder would explain. This option can also be specified using "tilde" or a numeric value 2.

For both types of the "R2" parameters, `'sign.prod'` indicates the hypothesized direction in which the unobserved confounder affects the mediator and outcome. (The name derives from the fact that this direction is mathematically represented by the sign of the product of two regression coefficients.) If "positive" (or a numeric value 1) is given, the confounder is assumed to affect the mediator and outcome in the same direction. If "negative" (or a numeric value -1), the effect is assumed to be in opposite directions.

The resulting contours in the "R2" plots represent the values of the ACME (or ADE) for different combinations of the mediator R2 and outcome R2 values. When both values are zero (the lower-left corner of the plot), the unobserved pre-treatment confounder has no effect on either mediator or outcome and therefore sequential ignorability is satisfied.

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)

Dustin Tingley, Harvard University, <dtingley@gov.harvard.edu>; Teppei Yamamoto, Massachusetts Institute of Technology, <teppey@mit.edu>; Jaquilyn Waddell-Boie, Princeton University, <jwaddell@princeton.edu>; Luke Keele, Penn State University, <ljk20@psu.edu>; Kosuke Imai, Princeton University, <kimai@princeton.edu>.

References

- Tingley, D., Yamamoto, T., Hirose, K., Imai, K. and Keele, L. (2014). "mediation: R package for Causal Mediation Analysis", *Journal of Statistical Software*, Vol. 59, No. 5, pp. 1-38.
- Imai, K., Keele, L. and Tingley, D. (2010) A General Approach to Causal Mediation Analysis, *Psychological Methods*, Vol. 15, No. 4 (December), pp. 309-334.
- Imai, K., Keele, L. and Yamamoto, T. (2010) Identification, Inference, and Sensitivity Analysis for Causal Mediation Effects, *Statistical Science*, Vol. 25, No. 1 (February), pp. 51-71.
- Imai, K., Keele, L., Tingley, D. and Yamamoto, T. (2009) "Causal Mediation Analysis Using R" in *Advances in Social Science Research Using R*, ed. H. D. Vinod New York: Springer.

See Also

[medsens](#), [plot](#), [contour](#).

plot.multimed	<i>Plotting the Results of Causal Mediation Analysis for Multiple Mechanisms</i>
---------------	--

Description

Function to plot results from multimed. Most standard plotting options are available.

Usage

```
## S3 method for class 'multimed'
plot(x, type = c("point", "sigma", "R2-residual",
  "R2-total"), tgroup = c("average", "treated", "control"),
  effect.type = c("indirect", "direct", "total"),
  ask = prod(par("mfcol")) < nplots, xlab = NULL, ylab = NULL,
  xlim = NULL, ylim = NULL, main = NULL, lwd = par("lwd"),
  pch = par("pch"), cex = par("cex"), las = par("las"),
  col.eff = "black", col.cbar = "black", col.creg = "gray", ...)
```

Arguments

x	object of class multimed, typically output from the multimed function.
type	type of plot(s) required. The default is to produce all, i.e., the point estimates of the effects under the homogenous interaction assumption ("point") and bounds as function of the sigma ("sigma"), R square star ("R2-residual") and R square tilde ("R2-total") parameters.
tgroup	treatment group(s) for which the estimates are produced. The default is to plot all, i.e., the average causal mediation effect when treated ("treated"), control ("control") and the simple average of these two effects ("average").
effect.type	a vector indicating which quantities of interest to plot for the point estimates. Only plotting total effects is not available.
ask	a logical value. If 'TRUE', the user is asked for input before a new figure is plotted. Default is to ask only if the number of plots on current screen is fewer than necessary.
xlab	label for the x axis. Default labels are used if 'NULL'.
ylab	label for the y axis. Default labels are used if 'NULL'.
xlim	limits of the x axis. If 'NULL' default values are used.
ylim	limits of the y axis. If 'NULL' default values are used.
main	main title for the plot. If 'NULL', default titles are used.
lwd	width of the lines used in graphs. For the "point" plot this is the width of confidence bars. For sensitivity plots this is the width of the lines for the bounds.
pch	plotting points used for the "point" plots.
cex	magnification factor for the plotting points in the "point" plots.

las	style of the y axis labels in the "point" plots.
col.eff	color of the points in the "point" plots and/or the bounds in sensitivity plots.
col.cbar	color of the confidence bars in the "point" plots.
col.creg	color of the confidence regions in sensitivity plots.
...	additional arguments to be passed to plotting functions.

Details

'type' and 'tgroup' can contain multiple character strings, in which case multiple plots are produced. For the use of graphical parameters see [plot](#) and the links it contains.

Author(s)

Teppey Yamamoto, Massachusetts Institute of Technology, <teppey@mit.edu>.

References

Tingley, D., Yamamoto, T., Hirose, K., Imai, K. and Keele, L. (2014). "mediation: R package for Causal Mediation Analysis", Journal of Statistical Software, Vol. 59, No. 5, pp. 1-38.

Imai, K. and Yamamoto, T. (2012) Identification and Sensitivity Analysis for Multiple Causal Mechanisms: Revisiting Evidence from Framing Experiments, typescript.

See Also

[multimed](#), [plot](#)

Examples

```
## Not run:

# Replicates Figure 3 (right column) of Imai and Yamamoto (2012)
# Note: # of bootstrap samples set low for quick illustration

data(framing)
Xnames <- c("age", "educ", "gender", "income")
res <- multimed("immigr", "emo", "p_harm", "treat", Xnames,
               data = framing, sims = 10)
summary(res)
plot(res, type = "point")
plot(res, type = c("sigma", "R2-total"), tgroup = "average")

## End(Not run)
```

school

School-level data

Description

The original data source is the Education Longitudinal Study of 2002. To deal with the issue on individually identifiable information, we generated hypothetical student-level data using a multiple imputation method. The Education Longitudinal Study of 2002 used a two-stage sample selection process. First, a national sample of schools was selected using stratified probability proportional to size (PPS), and school contacting resulted in 1,221 eligible public, Catholic, and other private schools from a population of approximately 27,000 schools containing 10th grade students. Of the eligible schools, 752 participated in the study. In the second stage of sample selection, a sample of approximately 26 sophomores, from within each of the participating public and private schools was selected. Each school was asked to provide a list of 10th grade students, and quality assurance (QA) checks were performed on each list that was received.

Usage

school

Format

A data matrix with 568 rows and 5 columns, containing no missing values. The data are provided only for illustrative purposes and not for inference about education effectiveness, for which the original data source should be consulted.

SCH_ID: School indicator.

coed: Indicator variable for coeducation. 1 = coeducation.

smorale: Measure of student morale in the school. 4 levels.

free: Percent of 10th grade students receiving free lunch. 1 to 7 levels.

catholic: Indicator variable for catholic school. 1 = catholic school.

Source

The complete student-level data is available from the data archives at www.icpsr.umich.edu/

References

United States Department of Education. National Center for Education Statistics

 student

Hypothetical student-level data

Description

The original data source is the Education Longitudinal Study of 2002. To deal with the issue on individually identifiable information, we generated hypothetical student-level data using a multiple imputation method. The Education Longitudinal Study of 2002 used a two-stage sample selection process. First, a national sample of schools was selected using stratified probability proportional to size (PPS), and school contacting resulted in 1,221 eligible public, Catholic, and other private schools from a population of approximately 27,000 schools containing 10th grade students. Of the eligible schools, 752 participated in the study. In the second stage of sample selection, a sample of approximately 26 sophomores, from within each of the participating public and private schools was selected. Each school was asked to provide a list of 10th grade students, and quality assurance (QA) checks were performed on each list that was received.

Usage

student

Format

A data matrix with 9,679 rows and 17 columns, containing no missing values. The data are provided only for illustrative purposes and not for inference about education effectiveness, for which the original data source should be consulted.

SCH_ID: School indicator.

fight: Indicator variable for fight at school. 1 = fight.

attachment: Indicator variable for attachment to school. 1 = like.

work: Indicator variable for part-time job. 1 = work.

score: Measure of math score.

late: Frequency in which the student was late for school. 5 levels.

coed: Indicator variable for coeducation. 1 = coeducation.

smorale: Measure of student morale in the school. 4 levels.

gender: Indicator variable for gender. 1 = female.

income: Total family income. 13 levels.

free: Percent of 10th grade students receiving free lunch. 1 to 7 levels.

pared: Parents highest level of education. 8 levels

catholic: Indicator variable for catholic school. 1 = catholic school.

Source

The complete student-level data is available from the data archives at www.icpsr.umich.edu/

References

United States Department of Education. National Center for Education Statistics

summary.ivmediate	<i>Summarizing Output from Mediation Analysis with Treatment Non-compliance</i>
-------------------	---

Description

Function to report results from mediation analysis with treatment noncompliance. Reported categories are local average causal mediation effects, local average natural direct effects and local average treatment (total) effect.

Usage

```
## S3 method for class 'ivmediate'  
summary(object, conf.level = object$conf.level[1],  
  ...)
```

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

Arguments

object	output from mediate function.
conf.level	confidence level for the intervals reported in the summary table.
...	additional arguments affecting the summary produced.
x	output from summary.mediate function.

Author(s)

Teppei Yamamoto, Massachusetts Institute of Technology, <teppe@mit.edu>.

References

Tingley, D., Yamamoto, T., Hirose, K., Imai, K. and Keele, L. (2014). "mediation: R package for Causal Mediation Analysis", Journal of Statistical Software, Vol. 59, No. 5, pp. 1-38.

Yamamoto, T. (2013). Identification and Estimation of Causal Mediation Effects with Treatment Noncompliance. Unpublished manuscript.

See Also

[ivmediate](#), [plot.ivmediate](#)

Examples

```

# Examples with JOBS II Field Experiment

# ** For illustration purposes a small number of simulations are used **

require(parallel)
require(MASS)

data(jobs)

a <- lm(comply ~ treat + sex + age + marital + nonwhite + educ + income,
        data = jobs)
b <- glm(job_dich ~ comply + treat + sex + age + marital + nonwhite + educ + income,
        data = jobs, family = binomial)
c <- lm(depress2 ~ job_dich * (comply + treat) + sex + age + marital + nonwhite + educ + income,
        data = jobs)

out <- ivmediate(a, b, c, sims = 50, boot = FALSE,
                enc = "treat", treat = "comply", mediator = "job_dich")

summary(out)
plot(out)

```

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 confidence intervals. If the treatment-mediator interaction is allowed in the mediation analysis, effects are reported separately for the treatment and control conditions as well as the simple averages of these effects are displayed at the bottom of the summary table.

Usage

```

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

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

```

Arguments

object	output from mediate or mediate_tsls function.
...	additional arguments affecting the summary produced.
x	output from summary.mediate function.

Author(s)

Dustin Tingley, Harvard University, <dtingley@gov.harvard.edu>; Teppei Yamamoto, Massachusetts Institute of Technology, <tepei@mit.edu>; Luke Keele, Penn State University, <ljk20@psu.edu>; Kosuke Imai, Princeton University, <kimai@princeton.edu>.

References

- Tingley, D., Yamamoto, T., Hirose, K., Imai, K. and Keele, L. (2014). "mediation: R package for Causal Mediation Analysis", *Journal of Statistical Software*, Vol. 59, No. 5, pp. 1-38.
- Imai, K., Keele, L., Tingley, D. and Yamamoto, T. (2011). Unpacking the Black Box of Causality: Learning about Causal Mechanisms from Experimental and Observational Studies, *American Political Science Review*, Vol. 105, No. 4 (November), pp. 765-789.
- Imai, K., Keele, L. and Tingley, D. (2010) A General Approach to Causal Mediation Analysis, *Psychological Methods*, Vol. 15, No. 4 (December), pp. 309-334.
- Imai, K., Keele, L. and Yamamoto, T. (2010) Identification, Inference, and Sensitivity Analysis for Causal Mediation Effects, *Statistical Science*, Vol. 25, No. 1 (February), pp. 51-71.
- Imai, K., Keele, L., Tingley, D. and Yamamoto, T. (2009) "Causal Mediation Analysis Using R" in *Advances in Social Science Research Using R*, ed. H. D. Vinod New York: Springer.

See Also

[mediate](#), [mediate_tsls](#), [plot.mediate](#), [summary](#).

summary.mediate.design

Summarizing Output from Design Based Mediation Analysis

Description

Function to report results from design based mediation analysis. Reported categories differ depending on the design and assumptions used.

Usage

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

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

Arguments

object	object of class <code>mediate.design</code> , typically output from a function for design-based mediation analysis (such as mediate.sed).
...	additional arguments affecting the summary produced.
x	output from the summary function.

Author(s)

Dustin Tingley, Harvard University, <dtingley@gov.harvard.edu>; Teppei Yamamoto, Massachusetts Institute of Technology, <tepei@mit.edu>.

References

Tingley, D., Yamamoto, T., Hirose, K., Imai, K. and Keele, L. (2014). "mediation: R package for Causal Mediation Analysis", Journal of Statistical Software, Vol. 59, No. 5, pp. 1-38.

Imai, K., Tingley, D. and Yamamoto, T. (2012) Experimental Designs for Identifying Causal Mechanisms. Journal of the Royal Statistical Society, Series A (Statistics in Society)

Imai, K., Keele, L., Tingley, D. and Yamamoto, T. (2011). Unpacking the Black Box of Causality: Learning about Causal Mechanisms from Experimental and Observational Studies, American Political Science Review, Vol. 105, No. 4 (November), pp. 765-789.

Imai, K., Keele, L. and Yamamoto, T. (2010) Identification, Inference, and Sensitivity Analysis for Causal Mediation Effects, Statistical Science, Vol. 25, No. 1 (February), pp. 51-71.

Imai, K., Keele, L., Tingley, D. and Yamamoto, T. (2009) "Causal Mediation Analysis Using R" in Advances in Social Science Research Using R, ed. H. D. Vinod New York: Springer.

See Also

[mediate](#), [plot.mediate](#), [summary](#).

summary.mediate.mer *Summarizing Output from Mediation Analysis of Multilevel Models*

Description

Function to report results from mediation analysis of multilevel models. Reported categories are mediation effect, direct effect, total effect, and proportion of total effect mediated. All quantities reported with confidence intervals. Group-specific effects and confidence intervals reported based on the mediator or the outcome group. Group-average quantities reported as default.

Usage

```
## S3 method for class 'mediate.mer'
summary(object, output = c("default", "byeffect",
  "bygroup"), ...)

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

## S3 method for class 'summary.mediate.mer.2'
print(x, ...)

## S3 method for class 'summary.mediate.mer.3'
print(x, ...)
```

Arguments

object	output from mediate function.
output	group-specific effects organized by effect if output = "byeffect"; group-specific effects organized by group if output = "bygroup"; group-average effects reported as default.
...	additional arguments affecting the summary produced.
x	output from summary.mediate.mer function.

Author(s)

Kentaro Hirose, Princeton University, <hirose@princeton.edu>.

References

- Tingley, D., Yamamoto, T., Hirose, K., Imai, K. and Keele, L. (2014). "mediation: R package for Causal Mediation Analysis", *Journal of Statistical Software*, Vol. 59, No. 5, pp. 1-38.
- Imai, K., Keele, L., Tingley, D. and Yamamoto, T. (2011). Unpacking the Black Box of Causality: Learning about Causal Mechanisms from Experimental and Observational Studies, *American Political Science Review*, Vol. 105, No. 4 (November), pp. 765-789.
- Imai, K., Keele, L. and Tingley, D. (2010) A General Approach to Causal Mediation Analysis, *Psychological Methods*, Vol. 15, No. 4 (December), pp. 309-334.
- Imai, K., Keele, L. and Yamamoto, T. (2010) Identification, Inference, and Sensitivity Analysis for Causal Mediation Effects, *Statistical Science*, Vol. 25, No. 1 (February), pp. 51-71.
- Imai, K., Keele, L., Tingley, D. and Yamamoto, T. (2009) "Causal Mediation Analysis Using R" in *Advances in Social Science Research Using R*, ed. H. D. Vinod New York: Springer.

See Also

[mediate](#), [plot.mediate.mer](#).

Examples

```
# Examples with JOBS II Field Experiment

# **For illustration purposes a small number of simulations are used**
## Not run:
data(jobs)
require(lme4)

# educ: mediator group
# occp: outcome group

# Varying intercept for mediator
model.m <- glmer(job_dich ~ treat + econ_hard + (1 | educ),
                 family = binomial(link = "probit"), data = jobs)

# Varying intercept and slope for outcome
model.y <- glmer(work1 ~ treat + job_dich + econ_hard + (1 + treat | occp),
```

```
family = binomial(link = "probit"), data = jobs)

# Output based on mediator group
multilevel <- mediate(model.m, model.y, treat = "treat",
                    mediator = "job_dich", sims=50, group.out="educ")

# Group-average effects
summary(multilevel)

# Group-specific effects organized by effect
summary(multilevel, output="byeffect")

# Group-specific effects organized by group
summary(multilevel, output="bygroup")

## End(Not run)
```

summary.mediations *Summarizing Output from Multiple Mediation Analyses*

Description

The 'summary.mediations' function produces a summary of results from multiple causal analyses conducted via [mediations](#). Output is a series of [summary.mediate](#) outputs for all the treatment/mediator/outcome combinations used in the input 'mediations' object.

Usage

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

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

Arguments

object	output from mediations function.
...	additional arguments affecting the summary produced.
x	output from summary.mediations function.

Author(s)

Dustin Tingley, Harvard University, <dtingley@gov.harvard.edu>; Teppei Yamamoto, Massachusetts Institute of Technology, <tepei@mit.edu>.

See Also

[mediations](#), [summary.mediate](#), [summary](#).

summary.medsens	<i>Summarizing Results from Sensitivity Analysis for Causal Mediation Effects</i>
-----------------	---

Description

Functions to report results from the sensitivity analysis for causal mediation effects via [medsens](#) in a tabular form.

Usage

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

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

Arguments

object	output from medsens function.
...	additional arguments affecting the summary produced.
x	output from summary.medsens function.

Author(s)

Dustin Tingley, Harvard University, <dtingley@gov.harvard.edu>; Teppei Yamamoto, Massachusetts Institute of Technology, <teppey@mit.edu>; Jaquilyn Waddell-Boie, Princeton University, <jwaddell@princeton.edu>; Luke Keele, Penn State University, <ljk20@psu.edu>; Kosuke Imai, Princeton University, <kimai@princeton.edu>.

References

- Tingley, D., Yamamoto, T., Hirose, K., Imai, K. and Keele, L. (2014). "mediation: R package for Causal Mediation Analysis", *Journal of Statistical Software*, Vol. 59, No. 5, pp. 1-38.
- Imai, K., Keele, L., Tingley, D. and Yamamoto, T. (2011). Unpacking the Black Box of Causality: Learning about Causal Mechanisms from Experimental and Observational Studies, *American Political Science Review*, Vol. 105, No. 4 (November), pp. 765-789.
- Imai, K., Keele, L. and Tingley, D. (2010) A General Approach to Causal Mediation Analysis, *Psychological Methods*, Vol. 15, No. 4 (December), pp. 309-334.
- Imai, K., Keele, L. and Yamamoto, T. (2010) Identification, Inference, and Sensitivity Analysis for Causal Mediation Effects, *Statistical Science*, Vol. 25, No. 1 (February), pp. 51-71.
- Imai, K., Keele, L., Tingley, D. and Yamamoto, T. (2009) "Causal Mediation Analysis Using R" in *Advances in Social Science Research Using R*, ed. H. D. Vinod New York: Springer.

See Also

[medsens](#), [summary](#).

summary.multimed	<i>Summarizing Output from Mediation Analysis with Multiple Mechanisms</i>
------------------	--

Description

Function to report results from the `multimed` function.

Usage

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

Arguments

<code>object</code>	object of class <code>multimed</code> , typically output from the <code>multimed</code> function.
<code>...</code>	additional arguments affecting the summary produced.
<code>x</code>	output from the summary function.

Author(s)

Teppei Yamamoto, Massachusetts Institute of Technology, <teppe@mit.edu>.

References

Tingley, D., Yamamoto, T., Hirose, K., Imai, K. and Keele, L. (2014). "mediation: R package for Causal Mediation Analysis", *Journal of Statistical Software*, Vol. 59, No. 5, pp. 1-38.

Imai, K. and Yamamoto, T. (2012) Identification and Sensitivity Analysis for Multiple Causal Mechanisms: Revisiting Evidence from Framing Experiments, typescript.

See Also

[multimed](#), [plot.multimed](#)

Examples

```
## Not run:  
# Replicates Figure 3 (right column) of Imai and Yamamoto (2012)  
# Note: # of bootstrap samples set low for quick illustration  
  
data(framing)  
Xnames <- c("age", "educ", "gender", "income")  
res <- multimed("immigr", "emo", "p_harm", "treat", Xnames,  
               data = framing, sims = 10)  
summary(res)
```

```

plot(res, type = "point")
plot(res, type = c("sigma", "R2-total"), tgroup = "average")

## End(Not run)

```

test.modmed

Significance Test for Moderated Mediation

Description

Function to test whether the average causal mediation effects and direct effects are significantly different between two moderator strata.

Usage

```

test.modmed(object, ...)

## S3 method for class 'mediate'
test.modmed(object, covariates.1, covariates.2,
  sims = object$sims, conf.level = object$conf.level, ...)

## S3 method for class 'test.modmed.mediate'
print(x, ...)

```

Arguments

object	output from mediate function.
...	additional arguments.
covariates.1	first set of value(s) of covariates (moderators) included in the mediator and outcome models. See documentation for the <code>covariates</code> argument for the mediate function.
covariates.2	second set of value(s) of covariates (moderators) included in the mediator and outcome models.
sims	number of simulation draws the test will be based on. Defaults to the number used in the original mediate fit.
conf.level	level of the returned two-sided confidence intervals for the effect differences. By default it is set to the value used in the original mediate call.
x	output from <code>test.modmed</code> function.

Details

The function takes the original call to `mediate` and reruns the algorithm twice with the two sets of `covariates` values. It assumes that the objects in the environment in which the original `mediate` call was made also exist in the current environment under the same variable names, i.e., it evaluates the updated call in the current environment.

Value

When applied to a mediate object, `test.modmed` returns an object of class `"test.modmed.mediate"`, a list composed of `"htest"` objects. See [t.test](#) for more explanations of `htest` objects. When applied to a `mediate.order` object, the function returns an object of class `"test.modmed.mediate.order"` which is a list composed of `"htest.order"` objects.

Author(s)

Tepei Yamamoto, Massachusetts Institute of Technology, <tepei@mit.edu>.

References

Tingley, D., Yamamoto, T., Hirose, K., Imai, K. and Keele, L. (2014). "mediation: R package for Causal Mediation Analysis", *Journal of Statistical Software*, Vol. 59, No. 5, pp. 1-38.

Imai, K., Keele, L. and Tingley, D. (2010) A General Approach to Causal Mediation Analysis, *Psychological Methods*, Vol. 15, No. 4 (December), pp. 309-334.

Imai, K., Keele, L. and Yamamoto, T. (2010) Identification, Inference, and Sensitivity Analysis for Causal Mediation Effects, *Statistical Science*, Vol. 25, No. 1 (February), pp. 51-71.

Imai, K., Keele, L., Tingley, D. and Yamamoto, T. (2009) "Causal Mediation Analysis Using R" in *Advances in Social Science Research Using R*, ed. H. D. Vinod New York: Springer.

See Also

[mediate](#), [test.TMint](#)

Examples

```
# Examples with JOBS II Field Experiment

# **For illustration purposes a small number of simulations are used**

data(jobs)

# Fit mediator and outcome models allowing for interaction with moderator
b.int <- lm(job_seek ~ treat*age + econ_hard + sex, data=jobs)
d.int <- lm(depress2 ~ treat*job_seek*age + econ_hard + sex, data=jobs)

# Initial mediate fit
fit <- mediate(b.int, d.int, sims=50, treat="treat", mediator="job_seek")

# Test for significance of moderated mediation
test.modmed(fit, list(age = 20), list(age = 70), sims = 100)
```

test.TMint	<i>Significance Test for Treatment-Mediator Interaction in Causal Mediation Analysis</i>
------------	--

Description

Function to test whether the average causal mediation effects and direct effects are significantly different between the treatment and control conditions.

Usage

```
test.TMint(x, ...)  
  
## S3 method for class 'mediate'  
test.TMint(x, conf.level = x$conf.level, ...)
```

Arguments

x	output from mediate function.
...	additional arguments.
conf.level	level of the returned two-sided confidence intervals for the effect differences. By default it is set to the value used in the original mediate call.

Value

test.TMint returns an object of class "hctest" when applied to a mediate object. See [t.test](#) for more explanations of the contents. The function returns an object of class "hctest.order" which has its own print method included in this package.

Author(s)

Teppei Yamamoto, Massachusetts Institute of Technology, <teppe@mit.edu>.

References

- Tingley, D., Yamamoto, T., Hirose, K., Imai, K. and Keele, L. (2014). "mediation: R package for Causal Mediation Analysis", Journal of Statistical Software, Vol. 59, No. 5, pp. 1-38.
- Imai, K., Keele, L. and Tingley, D. (2010) A General Approach to Causal Mediation Analysis, Psychological Methods, Vol. 15, No. 4 (December), pp. 309-334.
- Imai, K., Keele, L. and Yamamoto, T. (2010) Identification, Inference, and Sensitivity Analysis for Causal Mediation Effects, Statistical Science, Vol. 25, No. 1 (February), pp. 51-71.
- Imai, K., Keele, L., Tingley, D. and Yamamoto, T. (2009) "Causal Mediation Analysis Using R" in Advances in Social Science Research Using R, ed. H. D. Vinod New York: Springer.

See Also

[mediate](#)

Examples

```
# Examples with JOBS II Field Experiment

# **For illustration purposes a small number of simulations are used**

data(jobs)

# Fit mediator and outcome models allowing for treatment-mediator interaction
b <- lm(job_seek ~ treat + econ_hard + sex + age, data=jobs)
d <- lm(depress2 ~ treat*job_seek + econ_hard + sex + age, data=jobs)

# Test for significance of interaction
fit <- mediate(b, d, sims=50, treat="treat", mediator="job_seek")
test.TMint(fit)
```

Index

- * **datasets**
 - boundsdata, 4
 - CEDdata, 5
 - framing, 6
 - jobs, 10
 - school, 52
 - student, 53
- amelidiate, 2
- boundsdata, 4
- CEDdata, 5
- contour, 49
- contour.default, 48
- framing, 6
- glm, 14
- integrate, 8
- ivmediate, 7, 43, 54
- jobs, 10
- lm, 40
- lowess, 48
- mclapply, 8
- mediate, 2, 3, 11, 22, 24, 26, 28, 31–33, 37, 39, 45, 46, 49, 56–58, 62–64
- mediate.ced, 21
- mediate.pd, 23
- mediate.ped, 25
- mediate.sed, 27, 56
- mediate_tsls, 29, 56
- mediation, 31
- mediation-package (mediation), 31
- mediations, 2, 3, 18, 26, 31, 46, 47, 59
- medsens, 18, 26, 34, 39, 49, 60
- mgcv, 14
- multimed, 38, 51, 61
- plot, 33, 35, 45, 47, 49, 51
- plot.ivmediate, 9, 42, 54
- plot.mediate, 18, 26, 43, 46, 47, 56, 57
- plot.mediate.mer, 18, 45, 58
- plot.mediations, 33, 46
- plot.medsens, 35–37, 47
- plot.multimed, 41, 50, 61
- print.htest.order (test.TMint), 64
- print.summary.ivmediate (summary.ivmediate), 54
- print.summary.mediate (summary.mediate), 55
- print.summary.mediate.design (summary.mediate.design), 56
- print.summary.mediate.mer (summary.mediate.mer), 57
- print.summary.mediations (summary.mediations), 59
- print.summary.medsens (summary.medsens), 60
- print.summary.multimed (summary.multimed), 61
- print.test.modmed.mediate (test.modmed), 62
- s, 14
- school, 52
- student, 53
- summary, 33, 35, 56, 57, 59, 60
- summary.ivmediate, 9, 43, 54
- summary.mediate, 18, 26, 55, 59
- summary.mediate.design, 22, 24, 28, 56
- summary.mediate.mer, 18, 46, 57
- summary.mediations, 33, 59
- summary.medsens, 35, 37, 60
- summary.multimed, 61
- Surv, 14
- survreg, 14

t.test, [63](#), [64](#)
test.modmed, [62](#)
test.TMint, [63](#), [64](#)