Package 'causal.decomp'

September 13, 2021

Title Causal Decomposition Analysis

Version 0.0.1

Date 2021-9-7

Depends R (>= 2.10)

Imports stats, parallel, MASS, nnet, SuppDists, CBPS, PSweight,
spelling, utils

Description We implement causal decomposition analysis using the methods pro-
posed by Park, Lee, and Qin (2020) <arXiv:2008.12812> and Park, Kang, and Lee (2021+). This pack-
age allows researchers to use the multiple-mediator-imputation, single-mediator-
imputation, and product-of-coefficients regression methods to estimate the initial disparity, dis-
parity reduction, and disparity remaining. It also allows to make the inference condi-
tional on baseline covariates.

License GPL-2

Encoding UTF-8

RoxygenNote 7.1.1

LazyData true

NeedsCompilation no

Author Suyeon Kang [aut, cre],
Soojin Park [aut]

Maintainer Suyeon Kang <skang062@ucr.edu>

Repository CRAN

Date/Publication 2021-09-13 11:40:02 UTC

R topics documented:

---

causal.decomp   *causal.decomp: Causal Decomposition Analysis.*

Description

The causal.decomp package provides three important functions: mmi, smi, and pocr.

---

mmi   *Multiple-Mediator-Imputation Estimation Method*

Description

'mmi' is used to estimate the initial disparity, disparity reduction, and disparity remaining for causal decomposition analysis, using the multiple-mediator-imputation estimation method proposed by Park et. al. (2020).

Usage

```
mmi(
  fit.r = NULL,
  fit.x,
  fit.y,
  treat,
  covariates,
  sims = 100,
  conf.level = 0.95,
  conditional = TRUE,
  cluster = NULL,
  long = FALSE,
  mc.cores = 1L
)
```

Arguments

fit.r   a fitted model object for treatment. Can be of class 'CBPS' or 'SumStat'. De-
                fault is 'NULL'. Only necessary if 'conditional' is 'FALSE'.

fit.x   a fitted model object for intermediate confounder(s). Each intermediate model
                can be of class 'lm', 'glm', 'multinom', or 'polr'. When multiple confounders
                are considered, can be of class 'list' containing multiple models.

fit.y   a fitted model object for outcome. Can be of class 'lm' or 'glm'.

treat   a character string indicating the name of the treatment variable used in the mod-
                els. The treatment can be categorical with two or more categories (two- or multi-
                valued factor).

covariates	a vector containing the name of the covariate variable(s) used in the models. Each covariate can be categorical with two or more categories (two- or multi-valued factor) or continuous (numeric).
sims	number of Monte Carlo draws for nonparametric bootstrap.
conf.level	level of the returned two-sided confidence intervals, which are estimated by the nonparametric percentile bootstrap method. Default is .95, which returns the 2.5 and 97.5 percentiles of the simulated quantities.
conditional	a logical value. If 'TRUE', the function will return the estimates conditional on those covariate values; and all covariates in mediator and outcome models need to be centered prior to fitting. Default is 'TRUE'. If 'FALSE', 'fit.r' needs to be specified.
cluster	a vector of cluster indicators for the bootstrap. If provided, the cluster bootdtrap is used. Default is 'NULL'.
long	a logical value. If 'TRUE', the output will contain the entire sets of estimates for all bootsrap samples. Default is 'FALSE'.
mc.cores	The number of cores to use. Must be exactly 1 on Windows.

Details

This function returns the point estimates of the initial disparity, disparity reduction, and disparity remaining for a categorical treatment and a variety of types of outcome and mediator(s) in causal decomposition analysis. It also returns nonparametric percentile bootstrap confidence intervals for each estimate.

The initial disparity represents the expected difference in an outcome between a comparison group $R = j$ and a reference group $R = i$ where $i \neq j$. That is,

$$\tau(i, j) = E\{Y|R = j\} - E\{Y|R = i\},$$

where R and Y are the group indicator and the outcome variable, respectively. The disparity reduction represents the expected change in an outcome for the group $R = j$ after adjusting the level of mediator(s) to the level of the reference group. That is,

$$\delta(j) = E\{Y|R = j\} - E\{Y(G_M(i))|R = j\},$$

where $G_M(i)$ is a random draw from the mediator distribution of the reference group. The disparity remaining represents the remaining disparity for the group $R = j$ even after adjusting the level of mediators to the reference group. Formally,

$$\zeta(i) = E\{Y(G_M(i))|R = j\} - E\{Y|R = i\}.$$

The disparity reduction and remaining can be estimated using the multiple-mediator-imputation method suggested by Park et al. (2020). See the references for more details.

If one wants to make the inference conditional on baseline covariates, set 'conditional = T' and center the data before fitting the models.

As of version 0.0.1, the intetmediate confounder model ('fit.x') can be of class 'lm', 'glm', 'multi-nom', or 'polr', corresponding respectively to the linear regression models and generalized linear models, multinomial log-linear models, and ordered response models. The outcome model ('fit.y') can be of class 'lm' or 'glm'. Also, the treatment model ('fit.r') can be of class 'CBPS' or 'Sum-Stat', both of which use the propensity score weighting. It is only necessary when 'conditional = F'.

Value

result   a matrix containing the point estimates of the initial disparity, disparity remain-
                ing, and disparity reduction, and the percentile bootsrap confidence intervals for
                each estimate.

all.result   a matrix containing the point estimates of the initial disparity, disparity remain-
                ing, and disparity reduction for all bootsrap samples. Returned if 'long' is
                'TRUE'.

Author(s)

Suyeon Kang, University of California, Riverside, <skang062@ucr.edu>; Soojin Park, University
of California, Riverside, <soojinp@ucr.edu>.

References

Park, S., Lee, C., and Qin, X. (2020). "Estimation and sensitivity analysis for causal decomposition
in heath disparity research", arXiv preprint arXiv:2008.12812.

Park, S., Kang, S., and Lee, C. (2021+). "Choosing an Optimal Method for Causal Decomposition
Analysis: A Better Practice for Identifying Contributing Factors to Health Disparities".

See Also

smi

Examples

```
data(sdata)

#-----------------------------------------------------------#
# Example 1-a: Continuous Outcome
#-----------------------------------------------------------#
fit.m1 <- lm(M.num ~ R + C.num + C.bin, data = sdata)
fit.m2 <- glm(M.bin ~ R + C.num + C.bin, data = sdata,
          family = binomial(link = "logit"))
require(MASS)
fit.m3 <- polr(M.cat ~ R + C.num + C.bin, data = sdata)
fit.s1 <- lm(S ~ R + C.num + C.bin, data = sdata)
require(nnet)
fit.m4 <- multinom(M.cat ~ R + C.num + C.bin, data = sdata)
fit.y1 <- lm(Y.num ~ R + M.num + M.bin + M.cat + S + C.num + C.bin,
          data = sdata)

require(PSweight)
fit.r1 <- SumStat(R ~ C.num + C.bin, data = sdata, weight = "IPW")
require(CBPS)
fit.r2 <- CBPS(R ~ C.num + C.bin, data = sdata, method = "exact",
          standardize = "TRUE")

res.1a <- mmi(fit.r = fit.r1, fit.x = fit.s1,
          fit.y = fit.y1, sims = 40, conditional = FALSE,
```

```
                covariates = c("C.num", "C.bin"), treat = "R")
        res.1a


        #-----------------------------------------------------------#
        # Example 1-b: Binary Outcome
        #-----------------------------------------------------------#
        fit.y2 <- glm(Y.bin ~ R + M.num + M.bin + M.cat + S + C.num + C.bin,
                data = sdata, family = binomial(link = "logit"))


        res.1b <- mmi(fit.r = fit.r1, fit.x = fit.s1,
                fit.y = fit.y2, sims = 40, conditional = FALSE,
                covariates = c("C.num", "C.bin"), treat = "R")
        res.1b


        #-----------------------------------------------------------#
        # Example 2-a: Continuous Outcome, Conditional on Covariates
        #-----------------------------------------------------------#
        # For conditional = TRUE, need to create data with centered covariates
        # copy data
        sdata.c <- sdata
        # center quantitative covariate(s)
        sdata.c$C.num <- scale(sdata.c$C.num, center = TRUE, scale = FALSE)
        # center binary (or categorical) covariates(s)
        # only neccessary if the desired baseline level is NOT the default baseline level.
        sdata.c$C.bin <- relevel(sdata.c$C.bin, ref = "1")


        # fit mediator and outcome models
        fit.m1 <- lm(M.num ~ R + C.num + C.bin, data = sdata.c)
        fit.m2 <- glm(M.bin ~ R + C.num + C.bin, data = sdata.c,
                family = binomial(link = "logit"))
        fit.m3 <- polr(M.cat ~ R + C.num + C.bin, data = sdata.c)
        fit.s2 <- lm(S ~ R + C.num + C.bin, data = sdata.c)
        fit.y1 <- lm(Y.num ~ R + M.num + M.bin + M.cat + S + C.num + C.bin,
                data = sdata.c)


        res.2a <- mmi(fit.x = fit.s2,
                fit.y = fit.y1, sims = 40, conditional = TRUE,
                covariates = c("C.num", "C.bin"), treat = "R")
        res.2a


        #-----------------------------------------------------------#
        # Example 2-b: Binary Outcome, Conditional on Covariates
        #-----------------------------------------------------------#
        fit.y2 <- glm(Y.bin ~ R + M.num + M.bin + M.cat + S + C.num + C.bin,
                data = sdata.c, family = binomial(link = "logit"))


        res.2b <- mmi(fit.x = fit.s2,
                fit.y = fit.y2, sims = 40, conditional = TRUE,
                covariates = c("C.num", "C.bin"), treat = "R")
        res.2b
```

---

pocr   *Product-of-Coefficients-Regression Estimation Method*

Description

'pocr' is used to estimate the initial disparity, disparity reduction, and disparity remaining for causal decomposition analysis, using the product-of-coefficients-regression estimation method proposed by Park et. al. (2021+).

Usage

```
pocr(
  fit.x = NULL,
  fit.m,
  fit.y,
  treat,
  covariates,
  sims = 100,
  conf.level = 0.95,
  cluster = NULL,
  long = FALSE,
  mc.cores = 1L
)
```

Arguments

fit.x	a fitted model object for intermediate confounder. Can be of class 'lm'. Only necessary if the mediator is categorical. Default is 'NULL'.
fit.m	a fitted model object for mediator. Can be of class 'lm', 'glm', 'multinom', or 'polr'.
fit.y	a fitted model object for outcome. Can be of class 'lm'.
treat	a character string indicating the name of the treatment variable used in the models. The treatment can be categorical with two or more categories (two- or multi-valued factor).
covariates	a vector containing the name of the covariate variable(s) used in the models. Each covariate can be categorical with two or more categories (two- or multi-valued factor) or continuous (numeric).
sims	number of Monte Carlo draws for nonparametric bootstrap.
conf.level	level of the returned two-sided confidence intervals, which are estimated by the nonparametric percentile bootstrap method. Default is to return the 2.5 and 97.5 percentiles of the simulated quantities.
cluster	a vector of cluster indicators for the bootstrap. If provided, the cluster bootstrap is used. Default is 'NULL'.
long	a logical value. If 'TRUE', the output will contain the entire sets of estimates for all bootsrap samples. Default is 'FALSE'.
mc.cores	The number of cores to use. Must be exactly 1 on Windows.

Details

This function returns the point estimates of the initial disparity, disparity reduction, and disparity remaining for a categorical treatment and a variety of types of outcome and mediator(s) in causal decomposition analysis. It also returns nonparametric percentile bootstrap confidence intervals for each estimate.

The definition of the initial disparity, disparity reduction, and disparity remaining can be found in help('mmi'). As opposed to the 'mmi' and 'smi' function, this function uses the product-of-coefficients-regression method suggested by Park et al. (2021+). It always make the inference conditional on baseline covariates. Therefore, users need to center the data before fitting the models. See the reference for more details.

As of version 0.0.1, the mediator model ('fit.m') can be of class 'lm', 'glm', 'multinom', or 'polr', corresponding respectively to the linear regression models and generalized linear models, multinomial log-linear models, and ordered response models. The outcome model ('fit.y') can be of class 'lm'. The intermediate confounder model ('fit.x') can also be of class 'lm' and only necessary when the mediator is categorical.

Value

result   a matrix containing the point estimates of the initial disparity, disparity remaining, and disparity reduction, and the percentile bootsrap confidence intervals for each estimate.

all.result   a matrix containing the point estimates of the initial disparity, disparity remaining, and disparity reduction for all bootsrap samples. Returned if 'long' is 'TRUE'.

Author(s)

Suyeon Kang, University of California, Riverside, <skang062@ucr.edu>; Soojin Park, University of California, Riverside, <soojinp@ucr.edu>.

References

Park, S., Kang, S., and Lee, C. (2021+). "Choosing an Optimal Method for Causal Decomposition Analysis: A Better Practice for Identifying Contributing Factors to Health Disparities".

See Also

[mmi](#), [smi](#)

Examples

```
data(sdata)

# To be conditional on covariates, first create data with centered covariates
# copy data
sdata.c <- sdata
# center quantitative covariate(s)
sdata.c$C.num <- scale(sdata.c$C.num, center = TRUE, scale = FALSE)
# center binary (or categorical) covariates(s)
```

```
# only neccessary if the desired baseline level is NOT the default baseline level.
sdata.c$C.bin <- relevel(sdata.c$C.bin, ref = "1")

#----------------------------------------------------------------#
# Example 1: Continuous Mediator
#----------------------------------------------------------------#
fit.m1 <- lm(M.num ~ R + C.num + C.bin, data = sdata.c)
fit.y1 <- lm(Y.num ~ R + M.num + M.num:R + S +
            C.num + C.bin, data = sdata.c)
res1 <- pocr(fit.m = fit.m1, fit.y = fit.y1, sims = 40,
        covariates = c("C.num", "C.bin"), treat = "R")
res1

#----------------------------------------------------------------#
# Example 2: Binary Mediator
#----------------------------------------------------------------#
fit.s1 <- lm(S ~ R + C.num + C.bin, data = sdata.c)
fit.m2 <- glm(M.bin ~ R + C.num + C.bin, data = sdata.c,
            family = binomial(link = "logit"))
fit.y2 <- lm(Y.num ~ R + M.bin + M.bin:R + S +
            C.num + C.bin, data = sdata.c)
res2 <- pocr(fit.x = fit.s1, fit.m = fit.m2, fit.y = fit.y2,
        sims = 40, covariates = c("C.num", "C.bin"), treat = "R")
res2

#----------------------------------------------------------------#
# Example 3: Ordinal Mediator
#----------------------------------------------------------------#
require(MASS)
fit.m3 <- polr(M.cat ~ R + C.num + C.bin, data = sdata.c)
fit.y3 <- lm(Y.num ~ R + M.cat + M.cat:R + S +
C.num + C.bin, data = sdata.c)
res3 <- pocr(fit.x = fit.s1, fit.m = fit.m3, fit.y = fit.y3,
        sims = 40, covariates = c("C.num", "C.bin"), treat = "R")
res3

#----------------------------------------------------------------#
# Example 4: Nominal Mediator
#----------------------------------------------------------------#
require(nnet)
fit.m4 <- multinom(M.cat ~ R + C.num + C.bin, data = sdata.c)
res4 <- pocr(fit.x = fit.s1, fit.m = fit.m4, fit.y = fit.y3,
        sims = 40, covariates = c("C.num", "C.bin"), treat = "R")
res4
```

---

sdata   *Synthetic Data for Illustration*

Description

A ranomly generated dataset containing 1000 rows and 9 columns with no missing values.

Usage

```
sdata
```

Format

A data frame containing the following variables. The data are provided only for explanatory purposes. The mediators are assumed to be independent of each other.

C.num: A quantitative covariate.

C.bin: A binary covariates with a value of 0 or 1.

R: A group indicator with four levels.

S: A quantitative intermediate confounder between a mediator and the outcome.

M.num: A quantitative mediator.

M.bin: A binary mediator with a value of 0 or 1.

M.cat: A categorical mediator with three levels.

Y.num: A quantitative outcome.

Y.bin: A binary outcome with a value of 0 or 1.

Details

Note that all the variables are randomly generated using the dataset used in Park et al. (2021+).

References

Park, S., Kang, S., and Lee, C. (2021+). "Choosing an Optimal Method for Causal Decomposition Analysis: A Better Practice for Identifying Contributing Factors to Health Disparities".

---

smi   *Single-Mediator-Imputation Estimation Method*

Description

'smi' is used to estimate the initial disparity, disparity reduction, and disparity remaining for causal decomposition analysis, using the single-mediator-imputation estimation method proposed by Park et. al. (2021+).

Usage

```
smi(
  fit.r = NULL,
  fit.m,
  fit.y,
  treat,
  covariates,
  sims = 100,
```

```
    conf.level = 0.95,
    conditional = TRUE,
    cluster = NULL,
    long = FALSE,
    mc.cores = 1L
)
```

Arguments

fit.r   a fitted model object for treatment. Can be of class 'CBPS' or 'SumStat'. De-
               fault is 'NULL'. Only necessary if 'conditional' is 'FALSE'.

fit.m   a fitted model object for mediator. Can be of class 'lm', 'glm', 'multinom', or
               'polr'.

fit.y   a fitted model object for outcome. Can be of class 'lm' or 'glm'.

treat   a character string indicating the name of the treatment variable used in the mod-
               els. The treatment can be categorical with two or more categories (two- or multi-
               valued factor).

covariates   a vector containing the name of the covariate variable(s) used in the models.
               Each covariate can be categorical with two or more categories (two- or multi-
               valued factor) or continuous (numeric).

sims   number of Monte Carlo draws for nonparametric bootstrap.

conf.level   level of the returned two-sided confidence intervals, which are estimated by the
               nonparametric percentile bootstrap method. Default is .95, which returns the 2.5
               and 97.5 percentiles of the simulated quantities.

conditional   a logical value. If 'TRUE', the function will return the estimates conditional on
               those covariate values; and all covariates in mediator and outcome models need
               to be centered prior to fitting. Default is 'TRUE'. If 'FALSE', 'fit.r' needs to be
               specified.

cluster   a vector of cluster indicators for the bootstrap. If provided, the cluster bootstrap
               is used. Default is 'NULL'.

long   a logical value. If 'TRUE', the output will contain the entire sets of estimates
               for all bootsrap samples. Default is 'FALSE'.

mc.cores   The number of cores to use. Must be exactly 1 on Windows.

Details

This function returns the point estimates of the initial disparity, disparity reduction, and disparity
remaining for a categorical treatment and a variety of types of outcome and mediator(s) in causal
decomposition analysis. It also returns nonparametric percentile bootstrap confidence intervals for
each estimate.

The definition of the initial disparity, disparity reduction, and disparity remaining can be found in
help('mmi'). As opposed to the 'mmi' function, this function uses the single-mediator-imputation
method suggested by Park et al. (2021+). See the reference for more details.

If one wants to make the inference conditional on baseline covariates, set 'conditional = T' and
center the data before fitting the models.

As of version 0.0.1, the mediator model ('fit.m') can be of class 'lm', 'glm', 'multinom', or 'polr', corresponding respectively to the linear regression models and generalized linear models, multinomial log-linear models, and ordered response models. The outcome model ('fit.y') can be of class 'lm' or 'glm'. Also, the treatment model ('fit.r') can be of class 'CBPS' or 'SumStat', both of which use the propensity score weighting. It is only necessary when 'conditional = F'.

Value

result   a matrix containing the point estimates of the initial disparity, disparity remaining, and disparity reduction, and the percentile bootsrap confidence intervals for each estimate.

all.result   a matrix containing the point estimates of the initial disparity, disparity remaining, and disparity reduction for all bootsrap samples. Returned if 'long' is 'TRUE'.

Author(s)

Suyeon Kang, University of California, Riverside, <skang062@ucr.edu>; Soojin Park, University of California, Riverside, <soojinp@ucr.edu>.

References

Park, S., Kang, S., and Lee, C. (2021+). "Choosing an Optimal Method for Causal Decomposition Analysis: A Better Practice for Identifying Contributing Factors to Health Disparities".

See Also

[mmi](mmi)

Examples

```
data(sdata)

#------------------------------------------------------------#
# Example 1-a: Continuous Outcome
#------------------------------------------------------------#
require(PSweight)
fit.r1 <- SumStat(R ~ C.num + C.bin, data = sdata, weight = "IPW")
require(CBPS)
fit.r2 <- CBPS(R ~ C.num + C.bin, data = sdata, method = "exact",
          standardize = "TRUE")

# Continuous mediator
fit.m1 <- lm(M.num ~ R + C.num + C.bin, data = sdata)
fit.y1 <- lm(Y.num ~ R + M.num + S + C.num + C.bin, data = sdata)
res.1a1 <- smi(fit.r = fit.r1, fit.m = fit.m1,
          fit.y = fit.y1, sims = 40, conditional = FALSE,
          covariates = c("C.num", "C.bin"), treat = "R")
res.1a1

# Binary mediator
```

```
fit.m2 <- glm(M.bin ~ R + C.num + C.bin, data = sdata,
         family = binomial(link = "logit"))
fit.y2 <- lm(Y.num ~ R + M.bin + S + C.num + C.bin, data = sdata)
res.1a2 <- smi(fit.r = fit.r1, fit.m = fit.m2,
         fit.y = fit.y2, sims = 40, conditional = FALSE,
         covariates = c("C.num", "C.bin"), treat = "R")
res.1a2

# Categorical mediator
require(MASS)
fit.m3 <- polr(M.cat ~ R + C.num + C.bin, data = sdata)
fit.y3 <- lm(Y.num ~ R + M.cat + S + C.num + C.bin, data = sdata)
res.1a3 <- smi(fit.r = fit.r1, fit.m = fit.m3,
         fit.y = fit.y3, sims = 40, conditional = FALSE,
         covariates = c("C.num", "C.bin"), treat = "R")
res.1a3

require(nnet)
fit.m4 <- multinom(M.cat ~ R + C.num + C.bin, data = sdata)
res.1a4 <- smi(fit.r = fit.r1, fit.m = fit.m4,
         fit.y = fit.y3, sims = 40, conditional = FALSE,
         covariates = c("C.num", "C.bin"), treat = "R")
res.1a4

#----------------------------------------------------------#
# Example 1-b: Binary Outcome
#----------------------------------------------------------#
# Continuous mediator
fit.y1 <- glm(Y.bin ~ R + M.num + S + C.num + C.bin,
         data = sdata, family = binomial(link = "logit"))
res.1b1 <- smi(fit.r = fit.r1, fit.m = fit.m1,
         fit.y = fit.y1, sims = 40, conditional = FALSE,
         covariates = c("C.num", "C.bin"), treat = "R")
res.1b1

# Binary mediator
fit.y2 <- glm(Y.bin ~ R + M.bin + S + C.num + C.bin,
         data = sdata, family = binomial(link = "logit"))
res.1b2 <- smi(fit.r = fit.r1, fit.m = fit.m2,
         fit.y = fit.y2, sims = 40, conditional = FALSE,
         covariates = c("C.num", "C.bin"), treat = "R")
res.1b2

# Categorical mediator
fit.y3 <- glm(Y.bin ~ R + M.cat + S + C.num + C.bin,
         data = sdata, family = binomial(link = "logit"))
res.1b3 <- smi(fit.r = fit.r1, fit.m = fit.m3,
         fit.y = fit.y3, sims = 40, conditional = FALSE,
         covariates = c("C.num", "C.bin"), treat = "R")
res.1b3

res.1b4 <- smi(fit.r = fit.r1, fit.m = fit.m4,
         fit.y = fit.y3, sims = 40, conditional = FALSE,
```

```
                covariates = c("C.num", "C.bin"), treat = "R")
res.1b4


#-------------------------------------------------------------#
# Example 2-a: Continuous Outcome, Conditional on Covariates
#-------------------------------------------------------------#
# For conditional = T, need to create data with centered covariates
# copy data
sdata.c <- sdata
# center quantitative covariate(s)
sdata.c$C.num <- scale(sdata.c$C.num, center = TRUE, scale = FALSE)
# center binary (or categorical) covariates(s)
# only neccessary if the desired baseline level is NOT the default baseline level.
sdata.c$C.bin <- relevel(sdata.c$C.bin, ref = "1")

# Continuous mediator
fit.y1 <- lm(Y.num ~ R + M.num + S + C.num + C.bin, data = sdata.c)
fit.m1 <- lm(M.num ~ R + C.num + C.bin, data = sdata.c)
res.2a1 <- smi(fit.m = fit.m1,
         fit.y = fit.y1, sims = 40, conditional = TRUE,
         covariates = c("C.num", "C.bin"), treat = "R")
res.2a1

# Binary mediator
fit.y2 <- lm(Y.num ~ R + M.bin + S + C.num + C.bin, data = sdata.c)
fit.m2 <- glm(M.bin ~ R + C.num + C.bin, data = sdata.c,
         family = binomial(link = "logit"))
res.2a2 <- smi(fit.m = fit.m2,
         fit.y = fit.y2, sims = 40, conditional = TRUE,
         covariates = c("C.num", "C.bin"), treat = "R")
res.2a2

# Categorical mediator
fit.y3 <- lm(Y.num ~ R + M.cat + S + C.num + C.bin, data = sdata.c)
fit.m3 <- polr(M.cat ~ R + C.num + C.bin, data = sdata.c)
res.2a3 <- smi(fit.m = fit.m3,
         fit.y = fit.y3, sims = 40, conditional = TRUE,
         covariates = c("C.num", "C.bin"), treat = "R")
res.2a3

fit.m4 <- multinom(M.cat ~ R + C.num + C.bin, data = sdata.c)
res.2a4 <- smi(fit.m = fit.m4,
         fit.y = fit.y3, sims = 40, conditional = TRUE,
         covariates = c("C.num", "C.bin"), treat = "R")
res.2a4


#-------------------------------------------------------------#
# Example 2-b: Binary Outcome, Conditional on Covariates
#-------------------------------------------------------------#
# Continuous mediator
fit.y1 <- glm(Y.bin ~ R + M.num + S + C.num + C.bin,
         data = sdata.c, family = binomial(link = "logit"))
res.2b1 <- smi(fit.m = fit.m1,
```

```
              fit.y = fit.y1, sims = 40, conditional = TRUE,
              covariates = c("C.num", "C.bin"), treat = "R")
res.2b1

# Binary mediator
fit.y2 <- glm(Y.bin ~ R + M.bin + S + C.num + C.bin,
              data = sdata.c, family = binomial(link = "logit"))
res.2b2 <- smi(fit.m = fit.m2,
              fit.y = fit.y2, sims = 40, conditional = TRUE,
              covariates = c("C.num", "C.bin"), treat = "R")
res.2b2

# Categorical mediator
fit.y3 <- glm(Y.bin ~ R + M.cat + S + C.num + C.bin,
              data = sdata.c, family = binomial(link = "logit"))
res.2b3 <- smi(fit.m = fit.m3,
              fit.y = fit.y3, sims = 40, conditional = TRUE,
              covariates = c("C.num", "C.bin"), treat = "R")
res.2b3

res.2b4 <- smi(fit.m = fit.m4,
              fit.y = fit.y3, sims = 40, conditional = TRUE,
              covariates = c("C.num", "C.bin"), treat = "R")
res.2b4
```

Index