Package `mma`

October 25, 2021

Type Package

Title Multiple Mediation Analysis

Version 10.5-1

Date 2021-10-25

Author Qingzhao Yu and Bin Li

Maintainer Qingzhao Yu <qyu@lsuhsc.edu>

Depends R (>= 2.14.1), gbm, splines, survival, car, gplots

Imports lattice

Suggests knitr, rmarkdown

VignetteBuilder knitr

Encoding UTF-8

Description Used for general multiple mediation analysis.


License GPL (>= 2)

URL https://www.r-project.org,

https://publichealth.lsuhsc.edu/Faculty_pages/qyu/index.html

RoxygenNote 7.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2021-10-25 04:40:02 UTC
R topics documented:

mma-package ......................................................... 2
boot.med .............................................................. 4
boot.mod ............................................................. 8
cgd1 ................................................................. 11
data.org ............................................................. 12
form.interaction ..................................................... 17
med ................................................................. 18
mma ................................................................. 23
moderate ............................................................ 28
plot.med ............................................................. 29
plot.mma ........................................................... 31
plot2.mma .......................................................... 32
print.med ........................................................... 34
print.mma .......................................................... 35
summary.med_iden .................................................. 36
summary.mma ....................................................... 38
test.moderation ..................................................... 40
weight_behavior ..................................................... 42

Index 43

mma-package  Mediation Analysis Package

Description

This package is used to identify mediators and for general mediation analysis. Mediation effect refers to the effect conveyed by intervening variables to an observed relationship between an exposure and a response variable (outcome). In this package, the exposure is called the predictor, the intervening variables are called mediators. The mediation effects include the total effect, direct effect, and indirect effect, which are defined and the statistical inferences described in Yu et al.(2014).

Details

<table>
<thead>
<tr>
<th>Package</th>
<th>mma</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Package</td>
</tr>
<tr>
<td>Version</td>
<td>10</td>
</tr>
<tr>
<td>Date</td>
<td>2020-03-18</td>
</tr>
<tr>
<td>License</td>
<td>GPL (&gt;= 2)</td>
</tr>
</tbody>
</table>

"data.org" is used to identify potential mediators. "med" and "boot.med" are used for statistical inferences on the mediation effects when the predictor is binary or continuous. "mma" is a combined function that automatically identify potential mediators, based on which to make statistical
inference on the mediation effects.

Author(s)
Qingzhao Yu <qyu@lsuhsc.edu> and Bin Li <bli@lsu.edu>
Maintainer: Qingzhao Yu <qyu@lsuhsc.edu>

References

Examples

data("weight_behavior")
# binary predictor
# binary y
x=weight_behavior[,c(2,4:14)]
pred=weight_behavior[,3]
y=weight_behavior[,15]
temp.b.b.glm<-mma(x,y,pred=pred,contmed=c(7:9,11:12),binmed=c(6,10),binref=c(1,1),
catmed=5,catref=1,predref="M",alpha=0.4,alphad=0.4,n=2,n2=2)
temp.b.b.mart<-mma(x,y,pred=pred,contmed=c(7:9,11:12),binmed=c(6,10),binref=c(1,1),
catmed=5,catref=1,predref="M",alpha=0.4,alphad=0.4,nonlinear=TRUE,n=2,n2=5)
#continuous y
x=weight_behavior[,c(2,4:14)]
pred=weight_behavior[,3]
y=weight_behavior[,1]
temp.b.c.glm<-mma(x,y,pred=pred,mediator=5:12,jointm=list(n=1,j1=7:9),
                  predref="M",alpha=0.4, alpha2=0.4, n2=20)
temp.b.c.mart<-mma(x,y,pred=pred,mediator=5:12,jointm=list(n=1,j1=7:9),
                  predref="M",alpha=0.4, alpha2=0.4,
                  n=2, nonlinear=TRUE, n2=20)

---

**boot.med**

*Statistical Inference on Mediation Analysis with Continuous or Binary Predictor*

**Description**

To make inferences on the mediation effects when the predictor is continuous or binary.

**Usage**

```r
boot.med(data,x=data$x, y=data$y, dirx=data$dirx, binm=data$binm,
          contm=data$contm, catm=data$catm, jointm=data$jointm,
          cova=data$cova, margin=1, n=20,
          nonlinear=FALSE, df1=1, nu=0.001, D=3, distn=NULL, family1=data$family1,
          n2=50, w=rep(1,nrow(x)), refy=NULL, x.new=x, pred.new=dirx,
          cova.new=cova, binpred=data$binpred, type=NULL,
          w.new=NULL, all.model=FALSE, xmod=NULL, custom.function=NULL, para=FALSE)
```

**Arguments**

- **data**
  the list of result from data.org that organize the covariates, mediators, predictor and outcome. If data is FALSE, then need to set x, y, dirx, contm, catm, and jointm.

- **x**
  a data frame contains all mediators and covariates. Need to set up only when data is FALSE. All variables in x that are not identified as potential mediators (by mediator, contmed, binmed, catmed, or jointm) are forced in mediation analysis as covariates.

- **y**
  the vector of outcome variable. Need to set up only when data is FALSE.

- **dirx**
  the vector/matrix of predictor(s). Need to set up only when data is FALSE.

- **binm**
  the variable names or the column number of x that locates the binary mediators. Need to set up only when data is FALSE.

- **contm**
  the variable names or the column numbers of x that locate the potential continuous mediators. Need to set up only when data is FALSE.

- **catm**
  categorical mediators should be binarized and be presented as a list, where the first item is the number of categorical variables and the following items are the names or the column numbers of each binarized categorical variable in x. data.org organizes the categorical mediators in this format after they pass the mediator tests. Need to set up only when data is FALSE.
The document contains various parameters and variables related to statistical analysis, particularly in the context of mediation analysis. Here is a breakdown of the key terms and definitions:

- **jointm**: A list where the first item is the number of groups of joint mediators to be considered, and each of the following items identifies the names or the column numbers of the mediators in x for each group of joint mediators. Need to set up only when data is FALSE.

- **cova**: The data frame of covariates to be used to predict the mediator(s). The covariates in cova cannot be potential mediators. If the covariates are for some specific potential mediators, cova$for.m is the vector of names of potential mediators that use the covariates. Works for continuous predictors (pred) only, also the specified covariates should have no missing if only some mediators uses the covariates. Works mainly for continuous predictors (pred), also the specified covariates should have no missing if only some mediators uses the covariates.

- **margin**: The change in predictor when calculating the mediation effects, see Yu et al. (2014).

- **n**: The time of resampling in calculating the indirect effects, default is n=20, see Yu et al. (2014).

- **nonlinear**: If TRUE, Multiple Additive Regression Trees (MART) will be used to fit the final full model in estimating the outcome. The default value of nonlinear is FALSE, in which case, a generalized linear model will be used to fit the final full model.

- **df1**: If nonlinear is TRUE, natural cubic spline will be used to fit the relationship between the predictor and each mediator. The df is the degree of freedom in the ns() function, the default is 1.

- **nu**: Set the parameter "shrinkage" in gbm function if MART is to be used, by default, nu=0.001. See also help(gbm.fit).

- **D**: Set the parameter "interaction.depth" in gbm function if MART is to be used, by default, D=3. See also help(gbm.fit).

- **distn**: The assumed distribution of the outcome if MART is used for final full model. The default value of distn is "gaussian" for continuous y, and "bernoulli" for binary y.

- **family1**: Define the conditional distribution of y given x, and the linkage function that links the mean of y with the system component if generalized linear model is used as the final full model. The default value of family1 is gaussian(link="identity") for continuous y, and binomial(link = "logit") for binary y.

- **n2**: The number of times of bootstrap resampling. The default value is n2=50.

- **w**: The weight for observations.

- **refy**: If y is binary, the reference group of y.

- **x.new**: Of the same format as x, with a new set of covariates and mediators on which to calculate the mediation effects.

- **pred.new**: A new set of predictor(s).

- **cova.new**: A new set of covariate(s).

- **binpred**: If TRUE, the predictor is binary.

- **type**: The type of prediction when y is class Surv. Is "risk" if not specified.

- **w.new**: The weights for new.x.
all.model  save all the fitted model from bootstrap samples if TRUE. This needs to be true to make inferences on moderation effects.

xmod  If there is a moderator, xmod gives the moderator’s name in cova and/or x.

custom.function  a string of customer defined final model for predicting the outcome(s). The response variable should be noted as “responseY”, the dataset should be noted as “dataset123”. The weights for observations should be noted as “weights123”. The covariates should be in x or pred. Use "~." for all variables in x and pred is allowed. The customer defined model should be able to make prediction by using "predict(object, newdata=...)", where the object is the results of the fitted model. If a specific package needs to be called to fit the model, the user should call the package first. For example, if the gamlss package is used to fit a piece-wise polynomial with "age" to be the mediator and "race" be the predictor, we can set custom.function=“gamlss(responseY~b(age,df=3)+race,data=dataset123,tace=FALSE)”. The length of custom.function should be the same as the dimension of y. If custom.function[j] is NA, the usual method will be used to fit y[j].

para  It is for binary predictors. If it is true, we would like the x-m relationship be fitted parametrically.

Details

The mediators are not tested in this function. data.org should be used for the tests and data organizing, and then the resulting list from data.org can be used directly to define the arguments in this function. boot.med considers all variables in x as mediators or covariates in the final model and all variables identified by contm, binm, catm, or jointm as mediators.

Value

Returns an mma object.

estimation  list the estimation of ie (indirect effect), te (total effect), and de (direct effect from the predictor) separately.

bootsresults  a list where the first item, ie, is a matrix of n2 rows where each column gives the estimated indirect effect from the corresponding mediator (identified by the column name) from the n2 bootstrap samples; the second item, te, is a vector of estimated total effects from the bootstrap sample; and the 3rd item, de, is a vector of estimated direct effect of the predictor from the bootstrap sample.

model  a list where the first item, MART, is TRUE if mart is fitted for the final model; the second item, Survival, is TRUE if a survival model is fitted; the third item, type, is the type of prediction when a survival model is fitted; the fourth item, model, is the fitted final full model where y is the outcome and all predictor, covariates, and mediators are the explanatory variables; and the fourth item, best.iter is the number of best iterations if MART is used to fit the final model.

data  a list that contains all the used data: x=x, y=y, dirx=dirx, binm=binm, contm=contm, catm=catm, jointm=jointm, binpred=FALSE.

boot.detail  a list that contains the mediation effects on each row of new.x: new.x=new.x, te1, de1, ie1.
boot.med

- `all_model` a list with all fitted models from bootstrap samples if all.model is TRUE.
- `all_iter` if all.model is TRUE, a matrix with all fitted best iterations if MART is used. Each row is from one bootstrap sample.
- `all_boot` if all.model is TRUE, a matrix with all bootstrap samples. Each row is one bootstrap sample.

**Author(s)**
Qingzhao Yu <qyu@lsuhsc.edu>

**References**


**See Also**

"med" just estimate the mediation effects.

**Examples**

data("weight_behavior")
#binary y
#binary y
x=weight_behavior[,c(2,4:14)]
pred=weight_behavior[,3]
y=weight_behavior[,15]
data.bin<-data.org(x,y,pred=pred,contmed=c(7:9,11:12),binmed=c(6,10),binref=c(1,1),
catmed=5,catref=1,predref="M",alpha=0.4,alpha2=0.4)
temp1<-boot.med(data=data.bin,n=2,n2=4)
temp2<-boot.med(data=data.bin,n=2,n2=4,nu=0.05,nonlinear=TRUE)

#continuous y
x=weight_behavior[,c(2,4:14)]
pred=weight_behavior[,3]
y=weight_behavior[,1]
data.cont<-data.org(x,y,pred=pred,mediator=5:12,jointm=list(n=1,j1=7:9),
   predref="M",alpha=0.4,alpha2=0.4)
temp3<-boot.med(data=data.cont,n=2,n2=4)
temp4<-boot.med(data=data.cont,n=2,n2=4,nu=0.05, nonlinear=TRUE)

##Surv class outcome (survival analysis)
data(cgd1) #a dataset in the survival package
x=cgd1[,c(4:5,7:12)]
pred=cgd1[,6]
status<-ifelse(is.na(cgd1$etime1),0,1)
y=Surv(cgd1$futime,status)
#for continuous predictor
data.surv.contx<-data.org(x,y,pred=pred,mediator=1:ncol(x),
   alpha=0.5,alpha2=0.5)
temp9.contx<-boot.med(data=data.surv.contx,n=1,n2=2, type="lp")
temp10.contx<-boot.med(data=data.surv.contx,nonlinear=TRUE,n=1,n2=2)

#for binary predictor
x=cgd1[,c(5:12)]
pred=cgd1[,4]
data.surv.binx<-data.org(x,y,pred=pred,mediator=1:ncol(x),
   alpha=0.4,alpha2=0.4)
temp9.binx<-boot.med(data=data.surv.binx,n=1,n2=2, type="lp")
summary(temp9.binx)
temp10.binx<-boot.med(data=data.surv.binx,nonlinear=TRUE,n=1,n2=2)

---

**.boot.mod**

*Statistical Inference on Mediation Analysis with Continuous or Binary Predictor at different level of the moderator*

**Description**

To make inferences on the mediation effects when the predictor is continuous or binary at different level of the moderator.

**Usage**

```r
boot.mod(mma1, vari, continuous.resolution=10,
   w=NULL, n=20,
   x.new=NULL, w.new=NULL, pred.new=NULL, cova.new=NULL, xj=1, margin=1,
   xmod=vari, df1=1, para=FALSE)
```
Arguments

mma
the mma project generated from the function mma or boot.med.

vari
The name of the moderator. The moderator should be included in the original data as a covariate for the response (in x) and/or for mediators (in cova).

continuous.resolution
The number of equally space points (for continuous moderator) or levels (for categorical moderator) at which to evaluate mediation effects separately. If the moderator is categorical, continuous.resolution=10 implies using all individual levels.

w
the weight for observations.

n
the time of resampling in calculating the indirect effects, default is n=20, see Yu et al. (2014).

x.new
of the same format as x, with a new set of covariates and mediators on which to calculate the mediation effects.

w.new
the weights for new.x.

pred.new
a new set of predictor(s).

cova.new
a new set of covariate for mediator(s).

xj
the inference of mediation effects at different levels of moderator is made on the xjth predictor.

margin
the change in predictor when calculating the mediation effects, see Yu et al. (2014).

xmod
If there is a moderator, xmod gives the moderator’s name in cova and/or x.

df1
if nonlinear is TURE, natural cubic spline will be used to fit the relationship between the predictor and each mediator. The df is the degree of freedom in the ns() function, the default is 1.

para
It is for binary predictors. If it is true, we would like the x-m relationship be fitted parametrically.

Details

calculate the mediation effects from the xjth predictor to the response variable(s) at each level of the mediator.

Value

Returns an mma.mod object, which is similar to the mma object. Instead of calculating mediation effects for each predictor, the function returns the mediation effects of the xjth predictor at each level of the moderator.

Author(s)

Qingzhao Yu <qyu@lsuhsc.edu>
References


See Also

"med" just estimate the mediation effects.

Examples

#binary x and categorical moderator
#linear model
data("weight_behavior")
pred=weight_behavior[,3]
x=weight_behavior[,c(2,4:14)]
inter=form.interaction(x,pred,inter.cov=c("race"),predref="M")
x=cbind(x,inter)
y=weight_behavior[,15]
head(x)
data.bin<-data.org(x,y,pred=pred,contmed=c(7:9,11:12),binmed=c(6,10),
                   binref=c(1,1),catmed=5,catref=1,predref="M",alpha=0.4,alpha2=0.4)
temp1<-boot.med(data=data.bin,n=2,n2=4,all.model=TRUE)
temp1.mode=boot.mod(temp1,vari="race",continuous.resolution=10)
plot2.mma(temp1.mode,vari="exercises",moderator="race")
plot2.mma(temp1.mode,vari="sports",moderator="race")

#nonlinear model
x=weight_behavior[,c(2,4:14)]
pred=weight_behavior[,3]
y=weight_behavior[,15]
data.bin<-data.org(x,y,pred=pred,contmed=c(7:9,11:12),binmed=c(6,10),binref=c(1,1),
                  catmed=5,catref=1,predref="M",alpha=0.4,alpha2=0.4)
temp2<-boot.med(data=data.bin,n=2,n2=4,all.model=TRUE)
temp2.mode=boot.mod(temp2,vari="race",continuous.resolution=10)
plot2.mma(temp2.mode,vari="exercises",moderator="race")
plot2.mma(temp2.mode,vari="sports",moderator="race")
#binary x and continuous moderator
x=weight_behavior[,c(2,4:14)]
pred=weight_behavior[,3]
y=weight_behavior[,1]
inter=form.interaction(x,pred,inter.cov=c("age"),predref="M")
x=cbind(x,inter)
head(x)
data.cont<-data.org(x,y,pred=pred,contmed=c(7:9,11:12),binmed=c(6,10),
                   binref=c(1,1),catmed=5,catref=1,predref="M",alpha=0.4, alpha2=0.4)
temp3<-boot.med(data=data.cont,n=2,n2=4,all.model=TRUE)
temp3.mode=boot.mod(temp3,vari="age",continuous.resolution=2)
summary(temp3.mode)

plot2.mma(temp3.mode,vari="exercises",moderator="age")
plot2.mma(temp3.mode,vari="sports",moderator="age")

x=weight_behavior[,c(2,4:14)]
pred=weight_behavior[,3]
y=weight_behavior[,1]
data.cont<-data.org(x,y,pred=pred,mediator=5:12, jointm=list(n=1,j1=7:9),
                   predref="M",alpha=0.4, alpha2=0.4)
temp4<-boot.med(data=data.cont,n=2,n2=4,nu=0.05, nonlinear=TRUE, all.model=TRUE)
temp4.mode=boot.mod(temp4,vari="age",continuous.resolution=2,n=20)
summary(temp4.mode)
summary(temp4.mode,bymed=TRUE)

cgd1

cgd1 Data Set

Description
This database was obtained from the survival package containing a time-to-event data.

Usage
data(weight_behavior)

Format
The data set contains many variables.

Examples
data(cgd1)
names(cgd1)
data.org

Data Organization and Identify Potential Mediators

Description

Do a preliminary data analysis to identify potential mediators and covariates. Each variable listed in jointm is forced in the final estimation model as a mediator. Also organize the data into a format that can be directly used for the mediation analysis functions.

Usage

```r
data.org(x,y,pred,mediator=NULL,contmed=NULL,binmed=NULL,binref=NULL,catmed=NULL,
catref=NULL,jointm=NULL,refy=rep(NA,ncol(data.frame(y))),
family1=as.list(rep(NA,ncol(data.frame(y)))),
predref=rep(NA,ncol(data.frame(pred))),alpha=0.1,alpha2=0.1,
testtype=1, w=NULL,cova=NULL)
```

Arguments

- **x**: a data frame contains the predictor, all potential mediators and covariates. Note that x has to be a data frame. All variables in x that are not identified as potential mediators (by mediator, contmed, binmed, catmed, or jointm) are forced in mediation analysis as covariates.
- **y**: the vector of outcome variable. The outcome can be binary, continuous, or of "Surv" class (see survival package for help).
- **pred**: the column or matrix of predictor(s): the predictor is the exposure variable, it can be a binary or multi-categorical factor or one/a few continuous variable(s).
- **mediator**: the list of mediators (column numbers in x or by variable names). The mediators to be checked can be identified by "contmed", "binmed" and "catmed", or by this argument, "mediator", where binary and categorical mediators in x are identified as factors or characters, the reference group is the first level of the factor or factorized character. If a mediator has only two unique values, the mediator is identified as binary. If the reference groups need to be changed, the binary or categorical mediators can be listed in binref or catref, and the corresponding reference group in binref or catref.
- **contmed**: a vector of variable names or column numbers that locate the potential continuous mediators in x.
- **binmed**: a vector of column numbers that locate the potential binary mediators in x.
- **binref**: the defined reference groups of the binary potential mediators in binmed. The first levels of the mediators if is null.
- **catmed**: a vector of variable names or column numbers that locate the potential categorical mediators in x. The first levels of the mediators if is null.
- **catref**: the defined reference groups of the categorical potential mediators in catmed.
jointm a list that identifies the mediators that need to be considered jointly, where the first item indicates the number of groups of mediators to be considered jointly, and each of the following items identifies the variable names or column numbers of the mediators in x for each group of joint mediators.

refy if y is binary, the reference group of y. The default is the first level of as.factor(y). y for the reference group is assigned as 0.

family1 define the conditional distribution of y given x, and the linkage function that links the mean of y with the system component in generalized linear model. The default value of family1 is binomial(link = "logit") for binary y, and gaussian(link="identity") for continuous y.

predref if the predictor is categorical, identify the reference group of the predictor. The default is the first level of the predictor. The value of the predictor is 0 for the reference group.

alpha the significance level at which to test if the potential mediators (identified by contmed, binnmed, and catmed) can be used as a covariate or mediator in estimating y when all variables in x are included in the model. The default value is alpha=0.1

alpha2 the significant level at which to test if a potential mediator is related with the predictor. The default value is alpha2=0.1.

testtype if the testtype is 1 (by default), covariates/mediators are identified using full model; if the testtype is 2, covariates/mediators are tested one by one in models with the predictor only.

w the weight for data analysis, by default is rep(1,length(y)).

cova The data frame of covariates to be used to predict the mediator(s). The covariates in cova cannot be potential mediators. If the covariates are for some specific potential mediators, cova$for.m is the vector of names of potential mediators that use the covariates. Works for continuous predictors (pred) only, also the specified covariates should have no missing if only some mediators uses the covariates.

Value

data.org returns a list with the organized data and identifiers of the potential mediators in the organized data set. The list has two components, bin.results if there is binary or categorical exposure(s), cont.results if there is continuous exposures.

x the organized data frame that include all potential mediators and covariates that should be used to estimate the outcome.

dirx the vector/matrix of predictor(s)/exposure variable(s).

contm the column numbers of x that locate the potential continuous mediators.

binm when the predictor is continuous, binm gives the column numbers of x that locate the potential binary mediators.

catm when the predictor is binary, catm gives the column numbers of x that locate the potential binary or categorical mediators; when the predictor is continuous, catm gives a list where the first item is the number of potential categorical mediators, and the following items give the column numbers of each binarized categorical mediator in x.
**jointm** a list where the first item is the number of groups of joint mediators, and each of the following items identifies the column numbers of the mediators in the newly organized x for each group of joint mediators.

**y** the vector/matrix of outcome(s).

**y_type** the variable type of outcome(s): 1 is continuous, 2 is binary, 3 is reserved for multi-categorical (no 3 would show in y_type, since all categorical responses are binarized), and 4 is survival.

**fullmodel** a list with each item the full linear model fitted with all potential mediators and covariates for each response.

**rela** p-values of tests on the realtionship between the predictor(s) and each potential mediator.

**P1** If testtype=2, P1 gives the p-value of the corresponding variables in predicting the outcome(s) when only the variable and predictor are covariates in the model.

**binpred** The column numbers of binary predictors in pred.

**contpred** The column numbers of continuous predictors in pred.

**catpred** A list with each item includes the column numbers of a categorical predictor in pred.

**Note**

All other variables in x but not identified by mediator, contmed, binmed, or catmed are forced in the final model as covariates. Compared with data.org, joint mediators are considered in this function. Every variable in the jointm should be listed in contmed, binmed, or catmed, and these variables are forced to be included as mediators for further mediation analysis. A variables can be included in more than one groups of joint mediators in jointm.

**Author(s)**

Qingzhao Yu <qyu@lsuhsc.edu>

**References**


See Also

"data.org" that does not consider joint mediators, which can be added freely in the mediation analysis functions later.

Examples

```r
data("weight_behavior")
#binary predictor
#binary y
x=weight_behavior[,c(2,4:14)]
pred=weight_behavior[,3]
y=weight_behavior[,15]
data.b.b.2.1<-data.org(x,y,mediator=5:12,jointm=list(n=1,j1=c(5,7,9)),
pred=pred,predref="M", alpha=0.4, alpha2=0.4)
summary(data.b.b.2.1)

#Or you can specify the potential mediators and change the reference
#group for binary or categorical mediators. In the following code,
#potential continuous mediators are columns 8,9,10,12, and 13 of x,
#binary mediators are columns 7 and 11, and categorical mediator is
#column 6 of x with 1 to be the reference group for all categorical
#and binary mediators.

data.b.b.2<-data.org(x,y,pred=pred,contmed=c(7:9,11:12),binmed=c(6,10),
                    binref=c(1,1), catmed=5, catref=1,
                    jointm=list(n=1,j1=c(5,7,9)),
                    predref="M", alpha=0.4, alpha2=0.4)
summary(data.b.b.2)

#use the mediator argument instead of contmet, binmed and catmed

#multivariate predictor

x=weight_behavior[,c(2:3,5:14)]
pred=weight_behavior[,4]
y=weight_behavior[,15]
data.b.b.2.3<-data.org(x,y,mediator=5:12,jointm=list(n=1,j1=c(5,7,9)),
pred=pred,predref="OTHER", alpha=0.4, alpha2=0.4)
summary(data.b.b.2.3)

#multivariate responses

x=weight_behavior[,c(2:3,5:14)]
pred=weight_behavior[,4]
y=weight_behavior[,1:15]
data.b.b.2.4<-data.org(x,y,mediator=5:12,jointm=list(n=1,j1=c(5,7,9)),
pred=pred,predref="OTHER", alpha=0.4, alpha2=0.4)
summary(data.b.b.2.4)

#continuous y

x=weight_behavior[,c(2,4:14)]
pred=weight_behavior[,3]
```
y = weight_behavior[, 1]
data.b.c.2 <- data.org(x, y, pred = pred, mediator = 5:12, jointm = list(n = 1, j1 = 7:9),
  predref = "M", alpha = 0.4, alpha2 = 0.4)
summary(data.b.c.2)

# continuous predictor
# binary y
x = weight_behavior[, 3:14]
pred = weight_behavior[, 2]
y = weight_behavior[, 15]
data.c.b.2 <- data.org(x, y, pred = pred, mediator = 5:12, catref = 1,
  jointm = list(n = 2, j1 = 7:9, j2 = c(5, 7)), alpha = 0.4, alpha2 = 0.4)
summary(data.c.b.2)

# multivariate predictors
x = weight_behavior[, c(3:12, 14)]
pred = weight_behavior[, c(2, 13)]
y = weight_behavior[, 15]
data.c.b.2.2 <- data.org(x, y, pred = pred, mediator = 5:11, catref = 1,
  jointm = list(n = 2, j1 = 7:9, j2 = c(5, 7)), alpha = 0.4, alpha2 = 0.4)
summary(data.c.b.2.2)

# continuous y
x = weight_behavior[, 3:14]
pred = weight_behavior[, 2]
y = weight_behavior[, 1]
data.c.c.2 <- data.org(x, y, pred = pred, contmed = c(7:9, 11:12),
  binmed = c(6, 10),
  binref = c(1, 1), catmed = 5, catref = 1, jointm = list(n = 2, j1 = 7:9, j2 = c(5, 7)),
  alpha = 0.4, alpha2 = 0.4)
summary(data.c.c.2)

# multivariate responses
x = weight_behavior[, c(2:3, 5:14)]
pred = weight_behavior[, 4]
y = weight_behavior[, c(1, 15)]
data.b.c.2.4 <- data.org(x, y, mediator = 5:12, jointm = list(n = 1, j1 = c(5, 7, 9)),
  pred = pred, predref = "OTHER", alpha = 0.4, alpha2 = 0.4)
summary(data.b.c.2.4)

# Surv class outcome (survival analysis)
data(cgd1)  # a dataset in the survival package
x = cgd1[, c(4:5, 7:12)]
pred = cgd1[, 6]
status <- ifelse(is.na(cgd1$etime1), 0, 1)
y = Surv(cgd1$futime, status)
# for continuous predictor
# all other variables are considered as potential mediator
data.surv.contx <- data.org(x, y, pred=pred, mediator=(1:ncol(x)),
                           alpha=0.5, alpha2=0.5)
summary(data.surv.contx)

# for binary predictor
x <- cgd[, c(5:12)]
pred <- cgd[, 4]
data.surv.binx <- data.org(x, y, pred=pred, mediator=(1:ncol(x)),
                          alpha=0.4, alpha2=0.4)
summary(data.surv.binx)

form.interaction

Create interaction terms of predictor(s) and potential moderator(s).

Description
Create interaction terms of predictor(S) and potential moderator(s). Mainly for linear models.

Usage
form.interaction(x, pred, inter.cov, predref=NULL, kx=NULL)

Arguments
- **x**: As the x in data.org.
- **pred**: The predictor(s) that might have interaction effect with the potential moderators.
- **inter.cov**: The vector of names of potential moderators, which are included in x as covariates.
- **predref**: The reference group of the predictor if the predictor is categorical.
- **kx**: The interaction is with the kx-th predictor(s). kx can be a vector. If kx is null, the interaction is with each predictor.

Details
form.interaction is used to create interaction terms between predictor(s) and potential moderator(s). The created interaction terms should be used as covariates in linear mediation analysis. The function is not needed for nonlinear mediation method.

Value
Return a matrix with the interaction terms pred*inter.cov.

Author(s)
Qingzhao Yu <qyu@lsuhsc.edu>
See Also

"test.moderation","moderate"

Examples

data("weight_behavior")
pred=data.frame(weight_behavior[,3])
names(pred)="pred"
x=weight_behavior[,c(2,4:14)]
inter=forminteraction(x,pred,inter.cov=c("sports","sweat"),predref=NULL)
x=cbind(x,inter)
head(x)

med Mediation Analysis with Binary or Continuous Predictor

Description

To estimate the mediation effects when the predictor is binary or continuous.

Usage

med(data, x=data$bin.results$x, y=data$bin.results$y, dirx=data$bin.results$dirx, 
    binm=data$bin.results$binm, contm = data$bin.results$contm, 
    catm = data$bin.results$catm, jointm = data$bin.results$jointm, 
    cova=data$bin.results$cova, allm = c(contm, catm), 
    margin=1, n=20, nonlinear=FALSE, df1=1, nu=0.001,D=3,distn=NULL, 
    family1=data$bin.results$family1,refy=rep(0,ncol(y)),
    binpred=data$bin.results$binpred,x.new=x,pred.new=dirx, 
    cova.new=cova,type=NULL, w=NULL, w.new=NULL,xmod=NULL, 
    custom.function=NULL,para=FALSE)

Arguments

data the list of result from data.org that organize the covariates, mediators, predictor and outcome. If data is FALSE, then need to set x1, y1, dirx, contm, catm, and jointm.

x a data frame contains all mediators and covariates. Need to set up only when data is FALSE. All variables in x that are not identified as potential mediators (by mediator, contmed, binmed, catmed, or jointm) are forced in mediation analysis as covariates.

y the vector of outcome variable. Need to set up only when data is FALSE.

dirx the vector or matrix of predictor(s). The reference group is set to be 0. Need to set up only when data is FALSE.

binm the variable names or the column number of x that locates the binary mediators. Need to set up only when data is FALSE.
contm  the variable names or the column numbers of x that locate the potential continuous mediators. Need to set up only when data is FALSE.

catm  categorical mediators should be binarized and be presented as a list, where the first item is the number of categorical variables and the following items are the names or the column numbers of each binarized categorical variable in x. data.org organizes the categorical mediators in this format after they pass the mediator tests. Need to set up only when data is FALSE.

jointm a list where the first item is the number of groups of joint mediators to be considered, and each of the following items identifies the names or the column numbers of the mediators in x for each group of joint mediators. Need to set up only when data is FALSE.

cova  The data frame of covariates to be used to predict the mediator(s). The covariates in cova cannot be potential mediators. If the covariates are for some specific potential mediators, cova$for.m is the vector of names of potential mediators that use the covariates. Works for continuous predictors (pred) only, also the specified covariates should have no missing if only some mediators uses the covariates. Need to set up only when data is FALSE.

allm  the column numbers of all mediators. Need to set up only when data is FALSE. The default value of allm is c(contm,catm).

margin  the change in predictor when calculating the mediation effects, see Yu et al. (2014).

n  the time of resampling in calculating the indirect effects, default is n=20, see Yu et al. (2014).

nonlinear  if TRUE, Multiple Additive Regression Trees (MART) will be used to fit the final full model in estimating the outcome. The default value of nonlinear is FALSE, in which case, a generalized linear model will be used to fit the final full model.

df1  if nonlinear is TRUE, natural cubic spline will be used to fit the relationship between the predictor and each mediator. The df is the degree of freedom in the ns() function, the default is 1.

nu  set the parameter "shrinkage" in gbm function if MART is to be used, by default, nu=0.001. See also help(gbm.fit).

D  set the parameter "interaction.depth" in gbm function if MART is to be used, by default, D=3. See also help(gbm.fit).

distn  the assumed distribution(s) of the outcome(s) if MART is used for final full model. The default value of distn is "gaussian" for continuous y, "bernoulli" for binary y and coxph for "Surv" class outcome.

family1  a list with the ith item define the conditional distribution of y[,i] given x, and the linkage function that links the mean of y with the system component if generalized linear model is used as the final full model. The default value of family1 is gaussian(link="identity") for continuous y[,i], and binomial(link = "logit") for binary y[,i].

refy  if y is binary, the reference group of y.

binpred  if TRUE, the predict variable is binary.
x.new
A new set of covariates, of the same format as x (after data.org), on which to calculate the mediation effects. For continuous predictor only. If is NULL, the mediation effects will be calculated based on the original data set.
	pred.new
A new set of predictor(s), of the same format as pred (after data.org), on which to calculate the mediation effects. For continuous predictor only.

cova.new
A new set of covariate(s) to estimate mediators, of the same format as cova.

type
the type of prediction when y is class Surv. By default, type is "risk".

w
the weight for each case in x.

w.new
the weight for each case in x.new.

xmod
If there is a moderator, xmod gives the moderator's name in cova and/or x.

custom.function
a string of customer defined final model for predicting the outcome(s). The response variable should be noted as "responseY", the dataset should be noted as "dataset123". The weights for observations should be noted as "weights123". The covariates should be in x or pred. Use "~." for all variables in x and pred is allowed. The customer defined model should be able to make prediction by using "predict(object, newdata=...)", where the object is the results of the fitted model. If a specific package needs to be called to fit the model, the user should call the package first. For example, if the gamlss package is used to fit a piece-wise polynomial with "age" to be the mediator and "race" be the predictor, we can set custom.function="gamlss(responseV~b(age,df=3)+race,data=dataset123,tace=FALSE)". The length of custom.function should be the same as the dimension of y. If custom.function[j] is NA, the usual method will be used to fit y[j].

para
It is for binary predictors. If it is true, we would like the x-m relationship be fitted parametrically.

Details
The mediators are not tested in this function. data.org should be used first for the tests and data organizing, and then the resulting list from data.org can be used directly to define the arguments in this function. med considers all variables in x as mediators or covariates in the final model and all variables identified by contm, binm, catm, or jointm as mediators.

Value
The result is an med object with the item a.binx for results of binary or categorical exposure(s) and the item a.contx for continuous exposure(s):

demm
a matrix where each column gives the estimated direct effect not from the corresponding mediator (identified by the column name), see Yu et al. (2014) for the definition, and each row corresponding to the results from one resampling for binary predictor or the results on a row of x.new for continuous predictor.

ie
a matrix where each column gives the estimated indirect effect from the corresponding mediator (identified by the column name) and each row corresponding to the results from one resampling for binary predictor or the results on a row of x.new for continuous predictor.
med

t a vector of the estimated total effect on x.new.
model a list, where the first item, MART, is TRUE if a mart is fitted as the final model; the second item, Survial is T if a survival model is fit; the third item, type, is the type of prediction when a survival model is fitted; the fourth item, full.model, is the fitted final full model where y is the outcome and all predictor, covariates, and mediators are the explanatory variables; and the fifth item, best.iter, is the number of best iterations if MART is used to fit the final model, is NULL if the final model is a generalized linear model.

Author(s)
Qingzhao Yu <qyu@lsuhsc.edu>

References

See Also
"boot.med" to make inferences on the estimated mediation effects using bootstrap method.

Examples
data("weight_behavior")
# binary x
# binary y
x=weight_behavior[,c(2,4:14)]
pred=weight_behavior[,3]
y=weight_behavior[,15]
data.bin<-data.org(x,y,pred=pred,contmed=c(7:9,11:12),binmed=c(6,10),binref=c(1,1),catmed=5,catref=1,predref="M",alpha=0.4,alpha2=0.4)
temp1<-med(data=data.bin,n=2)
# or use self-defined final function
#temp1<-med(data=data.bin,n=2,custom.function =
#  'glm(responseY~.,data=dataset123,family="quasibinomial",
#   weights=weights123)')
temp2<-med(data=data.bin,n=2,nonlinear=TRUE)

# multiple predictors (categorical and continuous predictors)
# predictors (categorical and continuous predictors)
x=weight_behavior[,c(3,5:14)]
pred=weight_behavior[,c(2,4)]
y=weight_behavior[,15]
data.b.b.2.3<-data.org(x,y,mediator=4:11,jointm=list(n=1,j1=c(5,7,9)),
  pred=pred,predref="OTHER", alpha=0.4,alpha2=0.4)
temp1.2<-med(data.b.b.2.3,n=2)
temp2.2<-med(data.b.b.2.3,n=2,nonlinear=TRUE)

# multivariate responses
# predictors (categorical and continuous predictors)
x=weight_behavior[,c(2:3,5:14)]
pred=weight_behavior[,c(1,15)]
y=weight_behavior[,15]
data.b.b.2.4<-data.org(x,y,mediator=5:12,jointm=list(n=1,j1=c(5,7,9)),
  pred=pred,predref="OTHER", alpha=0.4,alpha2=0.4)
temp1.3<-med(data.b.b.2.4,n=2)
# or use the self defined function
#temp1.3<-med(data.b.b.2.4,n=2,custom.function =c(
  #glm(responseY~.,data=dataset123,family="quasibinomial",
  #weights=weights123),
  #glm(responseY~.,data=dataset123,family="gaussian",
  #weights=weights123)))
temp2.3<-med(data.b.b.2.4,n=2,nonlinear=TRUE)

# continuous y
# predictors (categorical and continuous predictors)
x=weight_behavior[,c(2,4:14)]
pred=weight_behavior[,3]
y=weight_behavior[,1]
data.cont<-data.org(x,y,pred=pred,mediator=5:12,jointm=list(n=1,j1=7:9),
  predref="M",alpha=0.4,alpha2=0.4)
temp3<-med(data=data.cont,n=2)
temp4<-med(data=data.cont,n=2,nonlinear=TRUE)

## continuous x
# binary y
x=weight_behavior[,3:14]
pred=weight_behavior[,2]
y=weight_behavior[,15]
data.contx<-data.org(x,y,pred=pred,mediator=4:10, alpha=0.4,alpha2=0.4)
temp5<-med(data=data.contx,n=2)
# or use the self defined function
#temp5<-med(data=data.contx,n=2,custom.function ='glm(responseY~.,
#  data=dataset123,family="quasibinomial",weights=weights123)')
temp6<-med(data=data.contx,n=2,nonlinear=TRUE,nu=0.05)

# continuous y
**mma**

### Multiple Mediation Analysis

**Description**

Test for mediators and do statistical inferences on the identified mediation effects.

**Usage**

```r
mma(x, y, pred, mediator=NULL, contmed=NULL, binmed=NULL, binref=NULL, catmed=NULL, catref=NULL, jointm=NULL, cova=NULL, refy=rep(NA, ncol(data.frame(y))), predref=rep(NA, ncol(data.frame(pred))), alpha=0.1, alpha2=0.1, margin=1, n=20, nonlinear=FALSE, df1=1, nu=0.001, D=3, distn=NULL,
```
family1=as.list(rep(NA,ncol(data.frame(y)))),n2=50,w=rep(1,nrow(x)),
testtype=1, x.new=NULL, pred.new=NULL, cova.new=NULL, type=NULL,w.new=NULL,
all.model=FALSE,xmod=NULL,custom.function = NULL,para=FALSE)

Arguments

x  a data frame contains the predictor, all potential mediators and covariates. All variables in x that are not identified as potential mediators (by mediator, contmed, binmed, catmed, or jointm) are forced in mediation analysis as covariates.
y  the vector of outcome variable.
pred  the vector/matrix of the predictor(s).
mediator  the list of mediators (column numbers in x or by variable names). The mediators to be checked can be identified by "contmed", "binmed" and "catmed", or by this argument, "mediator", where binary and categorical mediators in x are identified by factors, the reference group is the first level of the factor.
contmed  a vector of column numbers that locate the potential continuous mediators in x.
binmed  a vector of column numbers that locate the potential binary mediators in x.
binref  the defined reference groups of the binary potential mediators in binmed.
catmed  a vector of column numbers that locate the potential categorical mediators in x.
catref  the defined reference groups of the categorical potential mediators in catmed.
jointm  a list that identifies the mediators that need to be considered jointly, where the first item indicates the number of groups of mediators to be considered jointly, and each of the following items identifies the column numbers of the mediators in x for each group of joint mediators.
cova  The data frame of covariates to be used to predict the mediator(s). The covariates in cova cannot be potential mediators. If the covariates are for some specific potential mediators, cova$for.m is the vector of names of potential mediators that use the covariates. Works for continuous predictors (pred) only, also the specified covariates should have no missing if only some mediators uses the covariates. Works mainly for continuous predictors (pred), also the specified covariates should have no missing if only some mediators uses the covariates.
refy  if y is binary, the reference group of y. By default, the reference group will be the first level of as.factor(y).
predref  if predictor is binary, identify the reference group of the binary predictor. By default, the reference group will be the first level of the predictor.
alpha  the significance level at which to test if the potential mediators (identified by contmed, binmed, and catmed) can be used as a covariate or mediator in estimating y when all variables in x are included in the model. The default value is alpha=0.1
alpha2  the significant level at which to test if a potential mediator is related with the predictor. The default value is alpha2=0.1.
margin  if binpred is FALSE, define the change in predictor when calculating the mediation effects, see Yu et al. (2014).
n the time of resampling in calculating the indirect effects, default is n=20, see Yu et al. (2014).

nonlinear if TURE, Multiple Additive Regression Trees (MART) will be used to fit the final full model in estimating the outcome. The default value of nonlinear is FALSE, in which case, a generalized linear model will be used to fit the final full model.

df1 if nonlinear is TURE, natural cubic spline will be used to fit the relationship between the countinuous predictor and each mediator. The df is the degree of freedom in the ns() function, the default is 1.

nu set the parameter "interaction.depth" in gbm function if MART is to be used, by default, nu=0.001. See also help(gbm.fit).

D set the parameter "shrinkage" in gbm function if MART is to be used, by default, D=3. See also help(gbm.fit).

distn the assumed distribution of the outcome if MART is used for final full model. The default value of distn is "bernoulli" for binary y, and "gaussian" for continuous y.

family1 define the conditional distribution of y given x, and the linkage function that links the mean of y with the system component if generalized linear model is used as the final full model. The default value of family1 is binomial(link = "logit") for binary y, gaussian(link="identity") for continuous y.

n2 the number of times of bootstrap resampling. The default value is n2=50.

w the weight for each observation.

testtype if the testtype is 1 (by default), covariates/mediators are identified using full model; if the testtype is 2, covariates/mediators are tested one by one in models with the predictor only.

x.new A new set of predictor and corresponding covariates, of the same format as x (after data.org), on which to calculate the mediation effects. For continuous predictor only. If is NULL, the mediation effects will be calculated based on the original data set.

pred.new A new set of predictor(s), of the same format as x (after data.org), on which to calculate the mediation effects. For continuous predictor only.

cova.new a new set of covaraite(s).

type the type of prediction when y is class Surv. Is "risk" if not specified.

w.new the weights for new.x.

all.model save all the fitted model from bootstrap samples if TRUE. This needs to be true to make inferences on moderation effects.

xmod If there is a moderator, xmod gives the moderator’s name in cova and/or x.

custom.function a string of customer defined final model for predicting the outcome(s). The response variable should be noted as "responseY", the dataset should be noted as "dataset123". The weights for observations should be noted as "weights123". The covariates should be in x or pred. Use ":-:" for all varaibles in x and pred is allowed. The customer defined model should be able to make prediction by
using "predict(object, newdata=...)", where the object is the results of the fitted model. If a specific package needs to be called to fit the model, the user should call the package first. For example, if the gamlss package is used to fit a piecewise polynomial with "age" to be the mediator and "race" be the predictor, we can set custom.function="gamlss(responseV~b(age,df=3)+race,data=dataset123,tace=FALSE)". The length of custom.function should be the same as the dimension of y. If custom.function[j] is NA, the usual method will be used to fit y[j].

para It is for binary predictors. If it is true, we would like the x-m relationship be fitted parametrically.

Details
mma first tests if the potential mediators defined by binm, contm, and catm should be treated as mediators or covariates (if none, the variable will be deleted from further analysis). All variables identified by jointm are treated as mediators. All other variables in x that are not tested are treated as covariates. Then mma does the mediation effects estimation and inference on the selected variables.

Value
Returns an mma object.

estimation list the estimation of ie (indirect effect), te (total effect), and de (direct effect from the predictor) separately.

bootsresults a list where the first item, ie, is a matrix of n2 rows where each column gives the estimated indirect effect from the corresponding mediator (identified by the column name) from the n2 bootstrap samples; the second item, te, is a vector of estimated total effects from the bootstrap sample; and the 3rd item, de, is a vector of estimated direct effect of the predictor from the bootstrap sample.

model a list where the first item, MART, is T if mart is fitted for the final model; the second item, Survival, is T if a survival model is fitted; the third item, type, is the type of prediction when a survival model is fitted; the fourth item, model, is the fitted final full model where y is the outcome and all predictor, covariates, and mediators are the explanatory variables; and the fourth item, best.iter is the number of best iterations if MART is used to fit the final model.

data a list that contains all the used data: x=x, y=y, dirx=dirx, binm=binm, contm=contm, catm=catm, jointm=jointm, binpred=FALSE.

boots.detail for continuous predictor only: a list that contains the mediation effects on each row of new.pred: new.pred=new.pred, te1, de1, ie1.

all_model a list with all fitted models from bootstrap samples if all.model is TRUE.

all_iter if all.model is TRUE, a matrix with all fitted best iterations if MART is used. Each row is from one bootstrap sample.

all_boot if all.model is TRUE, a matrix with all bootstrap samples. Each row is one bootstrap sample.

Author(s)
Qingzhao Yu <qyu@lsuhsc.edu>
References


See Also

"data.org" is for mediator tests, and "med" , and "boot.med" for mediation analysis and inferences.

Examples

data("weight_behavior")
#binary predictor
#binary y
x=weight_behavior[,c(2,4:14)]
pred=weight_behavior[,3]
y=weight_behavior[,15]

temp.b.b.glm<-mma(x,y,pred=pred,contmed=c(7:9,11:12),binmed=c(6,10),binref=c(1,1),
catmed=5,catref=1,predref="M",alpha=0.4,alp hesitant=0.4,n=2,n2=2)

temp.b.b.mart<-mma(x,y,pred=pred,contmed=c(7:9,11:12),binmed=c(6,10),binref=c(1,1),
catmed=5,catref=1,predref="M",alpha=0.4,alp hesitant=0.4,nonlinear=TRUE,n=2,n2=5)

#continuous y
x=weight_behavior[,c(2,4:14)]
pred=weight_behavior[,3]
y=data.frame(weight_behavior[,1])
colnames(y)="bmi"

temp.b.c.glm<-mma(x,y,pred=pred,mediator=5:12,jointm=list(n=1,j1=7:9),
predref="M",alpha=0.4,alp hesitant=0.4,n2=20)
moderate <- mma(x, y, pred=pred, mediator=5:12, jointm=list(n=1, j1=7:9),
              predref="M", alpha=0.4, alpha2=0.4,
              n=2, nonlinear=TRUE, n2=20)

# Surv class outcome (survival analysis)
data(cgd1)  # a data set in the survival package
x=cgd1[,c(4:5,7:12)]
pred=cgd1[,6]
status<-ifelse(is.na(cgd1$etime1),0,1)
y=Surv(cgd1$futime, status)

# for continuous predictor
temp.cox.contx<-mma(x, y, pred=pred, mediator=1:ncol(x),
                      alpha=0.5, alpha2=0.5, n=1, n2=2, type="lp")
summary(temp.cox.contx)
temp.surv.mart.contx<-mma(x, y, pred=pred, mediator=1:ncol(x),
                          alpha=0.5, alpha2=0.5, nonlinear=TRUE, n2=2)

# for binary predictor
x=cgd1[,c(5:12)]
pred=cgd1[,4]
temp.cox.binx<-mma(x, y, pred=pred, mediator=1:ncol(x),
                  alpha=0.4, alpha2=0.4, n=1, n2=2, type="lp")
summary(temp.cox.binx)

---

**moderate**

*Calculate and plot the direct effect of the selected exposure variable at each level of the moderator.*

### Description

Calculate and plot the direct effect of the selected exposure variable at each level of the moderator.

### Usage

```r
moderate(med1, vari, j=1, kx=1, continuous.resolution=100, plot=TRUE)
```

### Arguments

- **med1**: The med object from the `med` function.
- **vari**: The name of the moderator.
- **j**: The jth response if the response is multiple.
- **kx**: The moderate effect is with the kx-th predictor(s).
- **continuous.resolution**: The number of equally spaced points at which to evaluate continuous predictors.
- **plot**: Plot the direct effect at each level of the moderator if true.
Details

Calculate and plot the direct effect of the selected exposure variable at each level of the moderator base on the result from the `med` function.

Value

The `moderate` returns a list where the item result is a data frame with two or three elements:

- `moderator` the moderator levels.
- `x` the level of the exposure variable – available only for continuous exposure and moderate with nonlinear method.
- `de` the direct effect at the corresponding moderator (and exposure) level(s).

Author(s)

Qingzhao Yu <qyu@lsuhsc.edu>

See Also

"form.interaction","test.moderation"

Examples

```r
# nonlinear model
data("weight_behavior")
x=weight_behavior[,c(2,4:14)]
pred=weight_behavior[,3]
y=weight_behavior[,15]
data.bin<-data.org(x,y,pred=pred,contmed=c(7:9,11:12),binmed=c(6,10),
    binref=c(1,1),catmed=5,catref=1,predref="M",alpha=0.4,alpha2=0.4)
temp2<-med(data=data.bin,n=2,nonlinear=TRUE)
result1=moderate(temp2,vari="race")
result2=moderate(temp2,vari="age")
```

---

`plot.med`  
*Plot the mediation effect on the fitted med object*

Description

Plot the marginal effect of the selected variable on the outcome, and the marginal effect of the predictor on the selected variable.

Usage

```r
## S3 method for class 'med'
plot(x,...,vari,xlim=NULL)
```
Arguments

- **x**: a med object created initially call to med, med.binx, or med.contx.
- **vari**: an indices or the name of the variable to plot.
- **xlim**: the range of the variable to be plotted.
- **...**: other arguments passed to the plot function.

Details

plot.med plots the marginal effect of the selected variable on the outcome, and the marginal effect of the predictor on the selected variable. If the predictor is binary, draw a histogram or boxplot of the marginal density of the variable at each different value of the predictor.

Value

No return value, called for graphs

Author(s)

Qingzhao Yu <qyu@lsuhsc.edu>

References


See Also

"med"

Examples

data("weight_behavior")
x=weight_behavior[,c(2,4:14)]
pred=weight_behavior[,3]
y=weight_behavior[,15]
data.bin<-data.org(x,y,pred=pred,contmed=c(7:9,11:12),binmed=c(6,10),
binref=c(1,1),catmed=5,catref=1,predref="M",alpha=0.4,alpha2=0.4)
temp1<-med(data=data.bin,n=2)
temp2<-med(data=data.bin,n=2,nonlinear=TRUE)
plot(temp1,data.bin,vari="exercises",xlim=c(0,50))
plot(temp2,data.bin,vari="sports")
**plot.mma**

Relative effects plot of the fitted mma object

---

**Description**

Plot the marginal effect of the selected variable on the outcome, and the marginal effect of the predictor on the selected variable.

**Usage**

```r
## S3 method for class 'mma'
plot(x,...,vari,xlim=NULL,alpha=0.95, quantile=FALSE)
```

**Arguments**

- `x`: a mma object created initially call to mma, boot.met.binx, or boot.met.contx.
- `vari`: an indices or the name of the variable to plot.
- `xlim`: the range of the variable to be plotted.
- `alpha`: for continuous predictor only, to draw the alpha confidence interval of the indirect effect.
- `quantile`: for continuous predictor only, if true to draw the alpha confidence interval of the indirect effect based on quantile, otherwise, based on the normal approximation.
- `...`: other arguments passed to the plot function.

**Details**

plot.mma plots the marginal effect of the selected variable on the outcome, and the marginal effect of the predictor on the selected variable. If the predictor is binary, draw a histogram or boxplot of the marginal density of the variable at each different value of the predictor.

**Value**

No return value, called for drawing graphs

**Author(s)**

Qingzhao Yu <qyu@lsuhsc.edu>

**References**


See Also
"mma","boot.med"

Examples

data("weight_behavior")
x=weight_behavior[,c(2,4:14)]
pred=weight_behavior[,3]
y=weight_behavior[,15]
temp.b.b glm<-mma(x,y,pred=pred,contmed=c(7:9,11:12),binmed=c(6,10),binref=c(1,1),
catmed=5,catref=1,predref="M",alpha=0.4, alpha2=0.4, n=2,n2=2) plot(temp.b.b glm,vari="exercises",xlim=c(0,50)) plot(temp.b.b glm,vari="sports")

plot2.mma

Relative effects plot of the fitted mma object with moderator

Description
Plot the marginal effect of the selected variable on the outcome, and the marginal effect of the predictor on the selected variable, at each level of the moderator.

Usage
plot2.mma(x,...,vari,xlim=NULL, alpha=0.95, quantile=FALSE, moderator,xj=1)

Arguments
x a mma object created initially call to boot.mod.
vari an indices or the name of the variable to plot.
xlim the range of the variable to be plotted.
alpha for continuous predictor only, to draw the alpha confidence interval of the indirect effect.
quantile for continuous predictor only, if true to draw the alpha confidence interval of the indirect effect based on quantile, otherwise, based on the normal approximation.

... other arguments passed to the plot function.

moderator the name of the moderator.

xj the moderation effect on the xjth predictor.

**Details**

plot2.mma plots the marginal effect of the selected variable on the outcome, and the marginal effect of the predictor on the selected variable, at each level of the moderator. If the predictor is binary, draw a histogram or boxplot of the marginal density of the variable at each different value of the predictor.

**Value**

No return value, called for graphs

**Author(s)**

Qingzhao Yu <qyu@lsuhsc.edu>

**References**


**See Also**

"boot.mod"

**Examples**

#see boot.mod menu.
print.med

Print an med object

Description

Print the estimation of mediation effects from an med object: from functions med.

Usage

## S3 method for class 'med'
print(x,...,digit=4)

Arguments

x
a med object created initially call to med.

... other arguments passed to the print function.

digit the number of digits to keep at printing.

Value

No return value, called for graphs

Author(s)

Qingzhao Yu <qyu@lsuhsc.edu>

References


See Also

"med"
print.mma

Examples

data("weight_behavior")
#binary x
#binary y
x=weight_behavior[,c(2,4:14)]
pred=weight_behavior[,3]
y=weight_behavior[,15]
data.bin<-data.org(x,y,pred=pred,contmed=c(7:9,11:12),binmed=c(6,10),
       binref=c(1,1),catmed=5,catref=1,predref="M",alpha=0.4,alpha2=0.4)
temp1<-med(data=data.bin,n=2)
temp2<-med(data=data.bin,n=2,nonlinear=TRUE)
temp1
print(temp2,digit=5)

print.mma

Print a mma object

Description

Print the estimation of mediation effects from an mma object.

Usage

## S3 method for class 'mma'
print(x,...,digit=3)

Arguments

x            a mma object created initially call to mma, boot.med.binx, or boot.med.contx.
...          other arguments passed to the print function.
digit        the number of decimal digits to keep.

Value

Return a LIST

ie            estimation of the indirect effects from the mma object.
te            estimation of the total effect from the mma object.
re            estimation of the direct effect from the mma object.

Author(s)

Qingzhao Yu <qyu@lsuhsc.edu>
References


See Also

"mma", "boot.med"

Examples

data("weight_behavior")
x=weight_behavior[,c(2,4:14)]
pred=weight_behavior[,3]
y=weight_behavior[,15]
temp.b.b.glm<-mma(x,y,pred=pred,contmed=c(7:9,11:12),binmed=c(6,10),binref=c(1,1),
catmed=5,catref=1,predref="M",alpha=0.4,alpha2=0.4,n=2,n2=2)
print(temp.b.b.glm,digit=8)

summary.med_iden

Summary method for class "med_iden".

Description

Compute the estimations, standard deviations and confidence intervals of the mediation effects.

Usage

## S3 method for class 'med_iden'
summary(object,...,only=FALSE)
## S3 method for class 'summary.med_iden'
print(x,...)
**Arguments**

- **object**: a `med_iden` object created initially call to `data.org`.
- **x**: a `summary.med_iden` object created initially call to `summary.med_iden`.
- **...**: other arguments passed to the print function.
- **only**: if only=T, show test results for selected covariates and mediators only.

**Details**

`summary.med_iden` gives a list of identified mediators, covariates and the test results.

**Value**

The function `summary.med_iden` return a list of covariates and mediators as identified by a series of tests.

- **mediator**: variable names of the identified mediators, either significant in both full model and in relation to the predictor, or being a member of the pre-identified joint mediators.
- **covariate**: variable names of covariates: being significant in the full model but not significantly relate with the predictor.
- **tests**: statistical test results.
- **results**: the original object.

**Author(s)**

Qingzhao Yu <qyu@lsuhsc.edu>

**References**


**See Also**

"mma", "boot.med"
Examples

```r
data("weight_behavior")
x = weight_behavior[, c(2, 4:14)]
pred = weight_behavior[, 3]
y = weight_behavior[, 15]
data.b.b.2 <- data.org(x, y, mediator = 5:12, jointm = list(n = 1, j1 = c(5, 7, 9)),
                       pred = pred, predref = "M", alpha = 0.4, alpha2 = 0.4)
summary(data.b.b.2)
```

**summary.mma**

Summary of an mma project

**Description**

Compute the estimations, standard deviations and confidence intervals of the mediation effects.

**Usage**

```r
## S3 method for class 'mma'
summary(object, ..., alpha = 0.05, plot = TRUE, RE = FALSE,
         quant = FALSE, ball.use = FALSE, bymed = FALSE)
## S3 method for class 'summary.mma'
print(x, ..., digit = 3)
```

**Arguments**

- `object`: a mma object created initially call to mma, boot.met.binx, or boot.met.contx.
- `x`: a summary.mma object created initially call to summary.mma.
- `...`: other arguments passed to the print function.
- `alpha`: the alpha level for confidence interval.
- `plot`: default is TRUE, if ture, draw a barplot of the mediation effects with confidence intervals.
- `RE`: default is FALSE, if ture, show the inferences on relative mediation effects.
- `quant`: default is TRUE, if ture and ball.use is FALSE, draw the confidence intervals of relative effects using quantile.
- `ball.use`: default is TRUE, if ture, draw the confidence intervals of relative effects using the confidence ball. If both quant and ball.use are false, draw the confidence intervals based on the standard deviations from bootstrap estimates.
- `digit`: the number of decimal digits to keep.
- `bymed`: if true, show results by each mediator.

**Details**

summary.mma gives a list of the estimations and summary statistics based on the bootstrap results. If plot = T, draw a barplot of the relative effects of the direct effect of the predictor and indirect effects of the mediators. Relative effects is defined as the (in)direct effect divided by the total effect. The plot is arranged in order from the largest to the smallest relative effect.
Value

The function `summary.mma` return a list. The first item, `results`, is the list for mediation effects, and the second item, `re`, is the list for relative effects. Under them, `results` have the items `ie`, `te` and `de`; `re` has the items `ie` and `de`. In each of the items, `est` is the estimation of the corresponding (relative) mediation effects based on the whole data, `mean` is the average estimated (relative) effects from the bootstrap samples, `sd` is the standard deviation of the estimates from the bootstrap sample. `upbd` and `lwbd` are the upper and lower bound of the confidence interval of the estimation using parametric method from the bootstrap sample, `upbd_q` and `lwbd_q` are the corresponding quantiles of the estimation from the bootstrap sample.

`ie`  
a matrix of statistics inference on the (relative) indirect effects from the `mma` object. `est` is the estimate using the full sample. `mean` is the estimate that average over the bootstrap estimates. `sd` is the standard deviation of the bootstrap estimates. `upbd` and `lwbd` are the confidence bounds based on sd. `upbd_q` and `lwbd_q` are the confidence bounds based on quantiles of the bootstrap estimates. `upbd_b` and `lwbd_b` are confidence ball bounds based on bootstrap estimates.

`te`  
statistics inference on the total effects from the `mma` object.

`de`  
statistics inference on the (relative) direct effects from the `mma` object.

If plot=T, draw a barplot of the relative mediation effects.

Author(s)

Qingzhao Yu <qyu@lsuhsc.edu>

References


See Also

"mma","boot.med"
Examples

data("weight_behavior")
x=weight_behavior[,c(2,4:14)]
pred=weight_behavior[,3]
y=weight_behavior[,15]
temp.b.b.glm<-mma(x,y,pred=pred,contmed=c(7:9,11:12),binmed=c(6,10),binref=c(1,1),
catmed=5,catref=1,predref="M",alpha=0.4,alpha2=0.4,n=2,n2=2)
summary(temp.b.b.glm, RE=TRUE, ball.use=FALSE)
summary(temp.b.b.glm, ball.use=FALSE)

Description

Used on a med object, test if there are moderation effects.

Usage

test.moderation(med1,vari,j=1,kx=NULL)

Arguments

med1 a med object created initially call to med.

vari If med1 was from generalized linear method, vari is the vector of names of potential moderation effects – the interaction terms formed by the potential moderator(s) times the predictor(s). Those interaction terms have to be included in mediation analysis as covariates.

If med1 was formed by nonlinear method, vari is the vector of names of potential moderators, which have been included in mediation analysis as covariates.

j To check the moderation effect on the jth response.

kx The moderation effect is with the kx-th predictor(s). kx can be a vector. If kx is null, the moderation effect is with each predictor.

Details

test.moderation is used to test whether direct moderation effect is significant based on the med object from med function (mediation analysis). If geraldized linear models were used for mediation analysis, the moderation effects (formed by the interactions of potential moderators and predictor) should be included in the mediation analysis as covariates. The function test.moderation will give the significance levels of these interaction terms. If nonlinear models were used for mediation analysis, the potential moderators (but not the interaction terms) should have been included in mediation analysis as covarites. The function test.moderation will give not only significance levels of their interaction with the predictor(s) in generalized linear model, but also the H-statistics by Friedman and Popescue (2008).
Value

No return value, called for test results.

Author(s)

Qingzhao Yu <qyu@lsuhsc.edu>

References


See Also

"med", "moderate"

Examples

data("weight_behavior")
x=weight_behavior[,c(2,4:14)]
pred=weight_behavior[,3]
y=weight_behavior[,15]
data.bin<-data.org(x,y,pred,pred=pred,contmed=c(7:9,11:12),binmed=c(6,10),
binref=c(1,1),catmed=5,catref=1,predref="M",alpha=0.4,alpha2=0.4)
temp2<-med(data=data.bin,n=2,nonlinear=TRUE)
a=test.moderation(temp2,c("sports","sweat"),j=1,kx=NULL)

x=cbind(x,form.interaction(x,pred,inter.cov=c("sports","sweat"),predref=NULL))
data.bin<-data.org(x,y,pred,pred=pred,contmed=c(7:9,11),binmed=c(6,10),
binref=c(1,1),catmed=5,catref=1,predref="M",alpha=0.4,alpha2=0.4)
temp1<-med(data=data.bin,n=2)
a=test.moderation(temp1,c("sports","sweat"),j=1,kx=NULL)
**Description**

This database was obtained from the Louisiana State University Health Sciences Center, New Orleans, by Dr. Richard Scribner. He explored the relationship between BMI and kids behavior through a survey at children, teachers and parents in Grenada in 2014. This data set includes 691 observations and 15 variables.

**Usage**

```
data(weight_behavior)
```

**Format**

The data set contains the following variables:

- `bmi` - body mass index, calculated by weight(kg)/height(cm)^2, numeric
- `age` - children’s age in years at the time of survey, numeric
- `sex` - sex of the children, factor
- `race` - race of the children, factor
- `numpeople` - number of people in family, numeric
- `car` - the number of cars in family, numeric
- `gotosch` - the method used to go to school, factor
- `snack` - eat snack or not in a day, binary
- `tvhours` - number of hours watching TV per week, numeric
- `cmpthours` - number of hours using computer per week, numeric
- `cellhours` - number of hours playing with cell phones per week, numeric
- `sports` - join in a sport team or not, 1: yes; and 2: no
- `exercises` - number of hours of exercises per week, numeric
- `sweat` - number of hours of sweating activities per week, numeric
- `overweigh` - the child is overweighed or not, binary

**Examples**

```
data(weight_behavior)
names(weight_behavior)
```
Index

* Continuous Predictor
  boot.med, 4

* Datasets
  cgd1, 11
  weight_behavior, 42

* Mediation Analysis
  boot.med, 4
  med, 18
  mma, 23

* Mediator Tests
  data.org, 12
  mma, 23

* Moderation Effect with Mediation Analysis
  boot.mod, 8

* Package
  mma-package, 2

  boot.med, 2, 4, 21, 27, 32, 36, 37, 39
  boot.mod, 8, 33

  cgd1, 11

  data.org, 2, 12, 15, 27

  form.interaction, 17, 29

  med, 2, 7, 10, 18, 27–30, 34, 41
  mma, 2, 23, 32, 36, 37, 39
  mma-package, 2
  moderate, 18, 28, 29, 41

  plot.med, 29
  plot.mma, 31
  plot2.mma, 32
  print.med, 34
  print.mma, 35
  print.summary.med_iden
    (summary.med_iden), 36
  print.summary.mma (summary.mma), 38

  summary.med_iden, 36
  summary.mma, 38

  test.moderation, 18, 29, 40

  weight_behavior, 42