Package ‘regmed’

January 20, 2023

Type  Package
Title  Regularized Mediation Analysis
Version  2.1.0
Date  2023-1-20
Description  Mediation analysis for multiple mediators by penalized structural equation models with different types of penalties depending on whether there are multiple mediators and only one exposure and one outcome variable (using sparse group lasso) or multiple exposures, multiple mediators, and multiple outcome variables (using lasso, L1, penalties).
License  GPL (>= 2)
Suggests  testthat, rmarkdown
Depends  R (>= 4.1.0), methods, graphics, glasso, igraph
Imports  knitr, Rcpp, RcppArmadillo, lavaan, gtools
LinkingTo  Rcpp, RcppArmadillo
NeedsCompilation  yes
VignetteBuilder  knitr
URL  https://cran.r-project.org/package=regmed
Author  Jason Sinnwell [aut, cre] (<https://orcid.org/0000-0003-1964-5522>),
        Daniel Schaid [aut] (<https://orcid.org/0000-0003-1457-6433>)
Maintainer  Jason Sinnwell <sinnwell.jason@mayo.edu>
Repository  CRAN
Date/Publication  2023-01-20 21:30:02 UTC

R topics documented:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>regmed-package</td>
<td>2</td>
</tr>
<tr>
<td>medsim</td>
<td>4</td>
</tr>
<tr>
<td>mvregmed.dat.check</td>
<td>5</td>
</tr>
<tr>
<td>mvregmed.edges</td>
<td>5</td>
</tr>
<tr>
<td>mvregmed.fit</td>
<td>6</td>
</tr>
<tr>
<td>mvregmed.graph.attributes</td>
<td>8</td>
</tr>
</tbody>
</table>
Description

Mediation analysis for multiple mediators by penalized structural equation models with different types of penalties depending on whether there are multiple mediators and only one exposure and one outcome variable (using sparse group lasso) or multiple exposures, multiple mediators, and multiple outcome variables (using lasso, L1, penalties).

Details

The DESCRIPTION file:

Package: regmed
Type: Package
Title: Regularized Mediation Analysis
Version: 2.1.0
Date: 2023-1-20
Authors@R: c(person("Jason", "Sinnwell", email = "sinnwell.jason@mayo.edu", comment=c(ORCID="0000-0003-1964-5522"), role = c("aut")), person("Daniel", "Schaid", email = "schaid.daniel@mayo.edu", comment=c(ORCID="0000-0003-1457-6433"), role = c("aut")))
Description: Mediation analysis for multiple mediators by penalized structural equation models with different types of penalties depending on whether there are multiple mediators and only one exposure and one outcome variable (using sparse group lasso) or multiple exposures, multiple mediators, and multiple outcome variables (using lasso, L1, penalties).
License: GPL (>= 2)
Suggests: testthat, rmarkdown
Depends: R (>= 4.1.0), methods, graphics, glasso, igraph
Imports: knitr, Rcpp, RcppArmadillo, lavaan, gtools
LinkingTo: Rcpp, RcppArmadillo
NeedsCompilation: yes
VignetteBuilder: knitr
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>medsim</td>
<td>Simulated dataset for regmed package</td>
</tr>
<tr>
<td>mvregmed.dat.check</td>
<td>Helper function to check x, y, mediator for input to mvregmed functions</td>
</tr>
<tr>
<td>mvregmed.edges</td>
<td>For an object of class mvregmed, create edges for a graph object that can be used for plots, or for creating models input to lavaan function sem</td>
</tr>
<tr>
<td>mvregmed.fit</td>
<td>Multivariate regularized mediation model</td>
</tr>
<tr>
<td>mvregmed.graph.attributes</td>
<td>Setup attributes of graph object for plotting</td>
</tr>
<tr>
<td>mvregmed.grid</td>
<td>Fit a grid of mvregmed models over a vector of lambda penalty parameters</td>
</tr>
<tr>
<td>mvregmed.grid.bestfit</td>
<td>Choose best fit model from a grid search based on minimum Bayesian Information Criterion</td>
</tr>
<tr>
<td>mvregmed.grid.data</td>
<td>Helper function to summarize fits of models across a grid of lambda values</td>
</tr>
<tr>
<td>mvregmed.grid.update</td>
<td>Helper function to update parameters in a grid search</td>
</tr>
<tr>
<td>mvregmed.init</td>
<td>Helper function to setup data and parameters for input to mvregmed.fit and mvregmed.grid</td>
</tr>
<tr>
<td>mvregmed.lavaan.dat</td>
<td>Set up data to input to lavaan sem</td>
</tr>
<tr>
<td>mvregmed.lavaan.model</td>
<td>Setup a model for input to lavaan</td>
</tr>
<tr>
<td>plot.mvregmed.grid</td>
<td>Plot penalty parameter lambda versus BIC for model fits</td>
</tr>
<tr>
<td>plot.regmed.grid</td>
<td>Plots for regmed.grid object.</td>
</tr>
<tr>
<td>regmed-package</td>
<td>Regularized Mediation Analysis</td>
</tr>
<tr>
<td>regmed.edges</td>
<td>For an object of class regmed, create edges for a graph object that can be used for plots, or for creating models input to lavaan function sem</td>
</tr>
<tr>
<td>regmed.fit</td>
<td>Regularized Mediation model for a specified lambda penalty value.</td>
</tr>
<tr>
<td>regmed.grid</td>
<td>Regularized mediation models over a vector grid of lambda penalty values.</td>
</tr>
<tr>
<td>regmed.grid.bestfit</td>
<td>Find best fitting regmed model from regmed.grid object.</td>
</tr>
<tr>
<td>regmed.lavaan.dat</td>
<td>Set up data to input to lavaan sem</td>
</tr>
<tr>
<td>regmed.lavaan.model</td>
<td>Create a lavaan model</td>
</tr>
<tr>
<td>regmed.prefilter</td>
<td>Prefilter to reduce the number of mediators for subsequent analyses</td>
</tr>
<tr>
<td>summary.mvregmed</td>
<td>Summary of mvregmed object</td>
</tr>
</tbody>
</table>
Further information is available in the following vignettes:

**regmed**  Regularized_Mediation_Examples (source)

**Author(s)**

NA  
Maintainer: NA

**References**


---

**medsim**  
*Simulated dataset for regmed package*

**Description**

Example data simulated from 2 response (y) variables, 10 exposure (x) variables, and 200 mediators (med). The all variables are generated from multivariate standard normal, with varying degrees of association between exposure, mediators, and responses.

**Usage**

data("medsim")

**Format**

Three data frames with 100 observations each:

- x  matrix with 10 columns of continuous exposure variables
- y  matrix with 2 columns of continuous response variables
- med  numeric matrix of 200 columns of simulated mediators between exposures and responses
mvregmed.dat.check

Helper function to check x, y, mediator for input to mvregmed functions

Description
Assure that x, y, mediator are matrices, check column names and fill in if NULL, and reduce x, y, mediator so no missing values among all three matrices.

Usage
mvregmed.dat.check(x, y, mediator, max.cor=0.99)

Arguments
x
matrix with columns representing "exposure" variable (sometimes called instrumental variable)
y
matrix with columns representing outcome variables
mediator
matrix with columns representing mediator variables
max.cor
maximum correlation within y, x, or mediators, so fitting is more robust

Value
list with updated x, y, mediator

Author(s)
Daniel Schaid and Jason Sinnwell

See Also
mvregmed.fit mvregmed.grid

mvregmed.edges

For an object of class mvregmed, create edges for a graph object that can be used for plots, or for creating models input to lavaan function sem

Description
Using the names of the alpha, beta, and delta matrices in the fitted object, create directed edges

Usage
mvregmed.edges(fit, eps = 0.001)
mvregmed.fit

Arguments

- **fit**: The fit as an object of class mvregmed. This can be output from either mvregmed.fit or mvregmed.grid.bestfit.
- **eps**: Threshold to determine whether any of alpha, beta, or delta parameters are close to zero to be rounded to zero.

Value

an object of class mvregmed.edges, which is a list with all.edge which is a data.frame with directed edges that are the names of x, mediator, y that are in the fitted model. Also returns data frames for alpha, beta, delta (each data frame containing vertex labels, row/col indices from which parameters were selected, and coefficient values), as well as names of x, mediator, and y.

Author(s)

Daniel Schaid and Jason Sinnwell

References


See Also

mvregmed.fit plot.mvregmed.edges

mvregmed.fit

Multivariate regularized mediation model

Description

Fit regularized mediation model for a specified lambda penalty value. Structural equation models for analysis of multiple exposures (x), multiple mediators, and multiple outcome variables (y) are fit with a lasso (L1) penalty on the model parameters. The model is x-[alpha] -> mediator-[beta] -> outcome, where alpha and beta are the parameters for the indirect effect of x on y, through the mediator. The model also allows a direct effect of x on y: x-[delta]->y.

Usage

mvregmed.fit(x, mediator, y, lambda, x.std = TRUE, med.std = TRUE, y.std = TRUE, max.outer = 5000, max.inner = 2, step.multiplier = 0.5, print.iter = FALSE, max.cor=0.99)
mvregmed.fit

Arguments

x  matrix with columns representing "exposure" variable (sometimes called instrumental variable)
mediator matrix with columns representing mediator variables
y  matrix with columns representing outcome variables
lambda lambda penalty parameter
x.std logical (TRUE/FALSE) whether to standardize x by dividing by standard deviation of x. Note that each column of x will be centered on its mean.
med.std logical (TRUE/FALSE) whether to standardize mediator by dividing by standard deviation of mediator. Note that each column of mediator will be centered on its mean.
y.std logical (TRUE/FALSE) whether to standardize y by dividing by standard deviation of y. Note that each column of y will be centered on its mean.
max.outer maximum number of outer loop iterations. The outer loop cycles over several inner loops.
max.inner maximum number of iterations for each inner loop. There is an inner loop for each parameter in the matrices alpha, beta, delta, and vary.
step.multiplier In inner loop, the step size is shrunk by the step.multiplier to assure that step size is not too large. Generally, the default of 0.5 works well.
print.iter print iteration number during fitting routine
max.cor maximum correlation within y, x, or mediators, so fitting is more robust

Value

An object of class mvregmed

Author(s)

Daniel Schaid and Jason Sinnwell

References


See Also

mvregmed.grid

Examples

data(medsim)
mvfit <- mvregmed.fit(x, med[,1:10], y, lambda=.1)
summary(mvfit)
mvregmed.graph.attributes

Setup attributes of graph object for plotting

Description

Setup attributes of graph object for plotting directed acyclic graph with graph attributes

Usage

mvregmed.graph.attributes(fit.edges, x.color = "palegreen", y.color = "palevioletred", med.color = "skyblue", v.size = 30)

Arguments

fit.edges A data.frame of edges with 1st column a vertex directed to the vertex in the 2nd column. This is all.edge from the list that is output from mvregmed.edges
x.color Color of vertices for x variables
y.color Color of vertices for y variables
med.color Color of vertices for mediators
v.size Size of vertices for plotting

Details

User can use this as template for taking advantage of more igraph attributes.

Value

List with items 1) output from graph_from_edgelist (see igraph); 2) vertex names; 3)vertex size; 4) vertex colors

Author(s)

Daniel Schaid and Jason Sinnwell

References


See Also

mvregmed.edges graph_from_edgelist
mvregmed.grid

**Description**

For each lambda in an input vector of values, fit a penalized mvregmed model.

**Usage**

```r
mvregmed.grid(x, mediator, y, lambda.vec, max.outer = 5000,
max.inner = 2, x.std = TRUE, med.std = TRUE, y.std = TRUE,
step.multiplier = 0.5, print.iter = FALSE, max.cor=0.99)
```

**Arguments**

- **x**: matrix with columns representing "exposure" variable (sometimes called instrumental variable).
- **mediator**: matrix with columns representing mediator variables.
- **y**: matrix with columns representing outcome variables.
- **lambda.vec**: Vector of values of penalty parameter lambda's.
- **max.outer**: maximum number of outer loop iterations. The outer loop cycles over several inner loops.
- **max.inner**: maximum number of iterations for each inner loop. There is an inner loop for each parameter in the matrices alpha, beta, delta, and vary.
- **x.std**: logical (TRUE/FALSE) whether to standardize x by dividing by standard deviation of x. Note that each column of x will be centered on its mean.
- **med.std**: logical (TRUE/FALSE) whether to standardize mediator by dividing by standard deviation of mediator. Note that each column of mediator will be centered on its mean.
- **y.std**: logical (TRUE/FALSE) whether to standardize y by dividing by standard deviation of y. Note that each column of y will be centered on its mean.
- **step.multiplier**: In inner loop, the step size is shrunk by the step.multiplier to assure that step size is not too large. Generally, the default of 0.5 works well.
- **print.iter**: print iteration number during fitting routine.
- **max.cor**: maximum correlation within y, x, or mediators, so fitting is more robust.

**Value**

An object of class mvregmed.grid

**Author(s)**

Daniel Schaid and Jason Sinnwell.
References

See Also
mvregmed.fit

Examples
data(medsim)
mvfit.grid <- mvregmed.grid(x, med[,1:10], y, lambda.vec=seq(.3, .04, by=-.01))
summary(mvfit.grid)
## plot(mvfit.grid)

mvregmed.grid.bestfit Choose best fit model from a grid search based on minimum Bayesian Information Criterion

Description
Search over all models fit in a grid and choose model with min BIC as best model.

Usage
mvregmed.grid.bestfit(fit.grid)

Arguments
fit.grid An object of class mvregmed.grid, output from function mvregmed.grid

Value
An object of class mvregmed, a single best fitting model.

Author(s)
Daniel Schaid and Jason Sinnwell

References

See Also
mvregmed.grid
mvregmed.grid.data

Helper function to summarize fits of models across a grid of lambda values

Description

eelper function to summarize fits of models across a grid of lambda values

Usage

mvregmed.grid.data(fit.lst, lambda.vec)

Arguments

fit.lst A list of model fits over a grid of lambda values; length of list is length of vector of lambdas.
lambda.vec A vector of penalty lambda values/

Details

Create a data.frame of summary information for each model fit in a grid.

Value

data.frame of summary information

Author(s)

Daniel Schaid and Jason Sinnwell

References


See Also

mvregmed.grid
mvregmed.grid.update  Helper function to update parameters in a grid search

Description

After a model is fit with a specific lambda, use the output of the fitted parameters as initial values for the next lambda value, thus using warm starts at each successive lambda value.

Usage

mvregmed.grid.update(fit.obj, inits)

Arguments

fit.obj  A fitted model of class mvregmed.
inits  Initial values from mvregmed.init that are subsequently updated with new values from fit.obj

Value

A list with the same components as output from mvregmed.init

Author(s)

Daniel Schaid and Jason Sinnwell

References


See Also

mvregmed.init

mvregmed.init  Helper function to setup data and parameters for input to mvregmed.fit and mvregmed.grid

Description

Helper function to setup data and parameters for input to mvregmed.fit and mvregmed.grid

Usage

mvregmed.init(dat.obj, x.std = TRUE, med.std = TRUE, y.std = TRUE)
mvregmed.lavaan.dat

Arguments

dat.obj A list that is output from mvregmed.dat.check that contains x, mediator, and y.
x.std logical (TRUE/FALSE) whether to standardize x by dividing by standard deviation of x. Note that each column of x will be centered on its mean.
med.std logical (TRUE/FALSE) whether to standardize mediator by dividing by standard deviation of mediator. Note that each column of mediator will be centered on its mean.
y.std logical (TRUE/FALSE) whether to standardize y by dividing by standard deviation of y. Note that each column of y will be centered on its mean.

Details

Center and scale (if declared) x, mediator and y. Then regress each mediator on all x to create residuals that are used to create the residual variance matrix for mediators. This variance matrix is penalized by glasso to obtain a matrix of full rank. Variance matrices for x and y variables are also created. Initial values of parameter matrices alpha, beta, and delta are created (all intital values = 0).

Value

A list of items used as input to model fitting.

Author(s)

Daniel Schaid and Jason Sinnwell

References


See Also

mvregmed.fit mvregmed.grid

Description

Set up data to input to lavaan structural equation model (sem)

Usage

mvregmed.lavaan.dat(x, mediator, y, max.cor=0.99)
mvregmed.lavaan.model

Arguments

- **x**: matrix of exposure variables
- **mediator**: matrix of mediators
- **y**: matrix of outcome variables
- **max.cor**: maximum correlation within mediators, so that fitting is more robust

Details

Use the function regmed.dat.check to standardize all variables and subset to subjects without missing data

Value

A dataframe with updated x, mediator, and y

Author(s)

Daniel Schaid and Jason Sinnwell

References


mvregmed.lavaan.model  Setup a model for input to lavaan

Description

Set up a model statement (string formula) from mvregmed object for input to lavaan

Usage

mvregmed.lavaan.model(fit.edge, fit.mvregmed)

Arguments

- **fit.edge**: Output from mvregmed.edges
- **fit.mvregmed**: A mvregmed object, either from mvregmed.fit or from mvregmed.grid.bestfit

Details

Loop through all relationships determined important from mrregmed edges object, and create model statement for lavaan, while also specifying covariances pre-estimated by mvregmed. See vignette for examples. The summary method for lavaan supersedes the summary from the lavaan package by simplifying the output to only return the coefficient table, as the covariance estimates were fixd from mvregmed.
Value

Text string to define a model as input to sem

Author(s)

Daniel Schaid and Jason Sinnwell

References


---

**plot.mvregmed.grid**  
Plot penalty parameter lambda versus BIC for model fits

Description

Plot penalty parameter lambda versus BIC for model fits

Usage

```r
## S3 method for class 'mvregmed.grid'
plot(x, xlab="lambda", ylab="BIC", pch="*", ...)
```

Arguments

- `x`  
  An object created by mvregmed.grid
- `xlab`  
  x axis label, by default set to 'lambda'
- `ylab`  
  y axis label, by default set to 'BIC' for this plot
- `pch`  
  plot character, by default set to a star (*)
- `...`  
  optional plot arguments

Author(s)

Daniel Schaid and Jason Sinnwell

References

plot.regmed.grid

Plots for regmed.grid object.

Description

Creates 2 plots: (1) BIC vs. lambda, and (2) Coefficients Alpha and Beta of mediator vs. lambda.

Usage

## S3 method for class 'regmed.grid'
plot(x, as.log=FALSE, ...)

Arguments

x  
regmed.grid object, returned by the regmed.grid() function

as.log  
Logical; if TRUE, plot lambda on the log scale

...  
optional arguments for plot method

Value

nothing is returned

Author(s)

Dan Schaid, Greg Jenkins, Jason Sinnwell

See Also

regmed.grid,

Examples

data(medsim)
fit.grid <- regmed.grid(x[,1], med[,1:10], y[,1],
lambda.vec= c(seq(from=1, to=0, by = -.1)), frac.lasso=.8)
summary(fit.grid)
For an object of class regmed, create edges for a graph object that can be used for plots, or for creating models input to lavaan function sem

**Description**

Using the names of the alpha, beta, and delta matrices in the fitted object, create directed edges

**Usage**

```r
regmed.edges(fit, type="mediators", eps = 0.001)
```

**Arguments**

- `fit`: The fit as an object of class mvregmed. This can be output from either mvregmed.fit or mvregmed.grid.bestfit.
- `type`: Character string specifying whether to only keep edges for mediators that have a non-zero coefficient with exposure and response variables ("mediator") or all edges ("all").
- `eps`: Threshold to determine whether any of alpha, beta, or delta parameters are close to zero to be rounded to zero.

**Value**

A list with class "regmed.edges" containing all.edge which is a data.frame with directed edges that are the names of x, mediator, y that are in the fitted model, with the coefficient for that edge. The plot method will plot the edges using igraph plotting options.

**Author(s)**

Daniel Schaid and Jason Sinnwell

**References**


**See Also**

- `regmed.fit`
- `plot.regmed.edges`
**Description**

Fit regularized mediation model for a specified lambda penalty value. Structural equation models for analysis of multiple mediators are extended by creating a sparse group lasso penalized model such that the penalty considers the natural groupings of the pair of parameters that determine mediation, as well as encourages sparseness of the model parameters. The model is $x$-[alpha] -> mediator-[beta] -> outcome, where alpha and beta are the parameters for the indirect effect of x on y, through the mediator. The model also allows a direct effect of x on y: $x$-[delta] -> y.

**Usage**

```r
regmed.fit(x, mediator, y, lambda, frac.lasso=0.8, x.std=TRUE, med.std=TRUE, 
max.outer=5000, max.inner=100, step.multiplier = 0.5, wt.delta = .5, 
print.iter=FALSE, max.cor=0.99)
```

**Arguments**

- **x**: vector representing "exposure" variable (sometimes called instrumental variable)
- **mediator**: matrix of mediators, rows are observations, columns are different mediators
- **y**: vector representing outcome
- **lambda**: lambda penalty parameter
- **frac.lasso**: fraction of penalty (lambda) that is allocated to L1 penalty (lasso). The remaining fraction, (1-frac.lasso) is allocated to group-lasso penalty, where the group is the pair of parameters alpha and beta that determine mediation ($x$-[alpha] -> mediator -> [beta] y).
- **x.std**: logical (TRUE/FALSE) whether to standardize x by dividing by standard deviation of x. Note that x will be centered on its mean.
- **med.std**: logical (TRUE/FALSE) whether to standardize mediators by dividing each mediator by its standard deviation. Note that mediators will be centered on their means.
- **max.outer**: maximum number of outer loop iterations. The outer loop cycles over several inner loops.
- **max.inner**: maximum number of iterations for each inner loop. There is an inner loop for each pair of alpha-beta parameters for each mediator, an inner loop for direct effect (delta), and inner loops for residual variances for x and for y.
- **step.multiplier**: a value between 0 and 1 for backtracking, to shrink step size. Value of 0.5 is typical default.
- **wt.delta**: a weight >=0 for how much weight should be given to shrinking delta parameter, by penalaty lambda*wt.delta.
- **print.iter**: print iteration history during fitting routine
- **max.cor**: maximum correlation within mediators, so that fitting is more robust
Value

regmed object, with S3 methods available: plot, print, summary

Author(s)

Dan Schaid, Greg Jenkins, Jason Sinnwell

References


See Also

regmed.edges summary.regmed

Examples

data(medsim)
filter5 <- regmed.prefilter(x[,1], med, y[,1], k=5)
fit.regmed <- regmed.fit(x[,1], med[,1:5], y[,1], lambda = 0.2, frac.lasso=.8)
summary(fit.regmed)

Description

Fit regularized mediation models over a vector grid of lambda penalty values. Structural equation models for analysis of multiple mediators are extended by creating a sparse group lasso penalized model such that the penalty considers the natural groupings of the pair of parameters that determine mediation, as well as encourages sparseness of the model parameters. The model is x-[alpha] -> mediator-[beta] -> outcome, where alpha and beta are the parameters for the indirect effect of x on y, through the mediator. The model also allows a direct effect of x on y: x-[delta] -> y.

Usage

regmed.grid(x, mediator, y, lambda.vec, frac.lasso=0.8, max.outer=5000, max.inner=100, x.std=TRUE, med.std=TRUE, step.multiplier = 0.5, wt.delta = .5, print.iter=FALSE, max.cor=0.99)
Arguments

- **x**: vector representing "exposure" variable (sometimes called instrumental variable)
- **mediator**: matrix of mediators, rows are observations, columns are different mediators
- **y**: vector representing outcome
- **lambda.vec**: vector of lambda penalty parameters
- **frac.lasso**: fraction of penalty (lambda) that is allocated to L1 penalty (lasso). The remaining fraction, (1-frac.lasso) is allocated to group-lasso penalty, where the group is the pair of parameters alpha and beta that determine mediation (x [alpha] -> mediator -> [beta] y).
- **max.outer**: maximum number of outer loop iterations. The outer loop cycles over several inner loops.
- **max.inner**: maximum number of iterations for each inner loop. There is an inner loop for each pair of alpha-beta parameters for each mediator, an inner loop for direct effect (delta), and inner loops for residual variances for x and for y.
- **x.std**: logical (TRUE/FALSE) whether to standardize x by dividing by standard deviation of x. Note that x will be centered on its mean.
- **med.std**: logical (TRUE/FALSE) whether to standardize mediators by dividing each mediator by its standard deviation. Note that mediators will be centered on their means.
- **step.multiplier**: a value between 0 and 1 for backtracking, to shrink step size. Value of 0.5 is typical default.
- **wt.delta**: a weight >=0 for how much weight should be given to shrinking delta parameter, by penalaty lambda*wt.delta.
- **print.iter**: print iteration history during fitting routine
- **max.cor**: maximum correlation within mediators, so fitting is more robust

Details

Although outcome y is not required to be scaled by its standard deviation, it can be beneficial to scale y. This helps with setting range of lambda penalty parameters, because when all x, y, and mediators are scaled, it is reasonable to consider lambda values within the range of 0 to 1. See reference for details of algorithm.

Value

- regmed.grid object

Author(s)

Dan Schaid, Jason Sinnwell

References

regmed.grid.bestfit

See Also

plot.regmed.grid regmed.fit

Examples

data(medsim)
fit.grid <- regmed.grid(x[,1], med[,1:5], y[,1],
lambda.vec= c(seq(from=1, to=0, by = -.1)),
frac.lasso=.8)
print(fit.grid)

regmed.grid.bestfit  Find best fitting regmed model from regmed.grid object.

Description

Find best fitting regmed model from regmed.grid object using minimum BIC to select model.

Usage

regmed.grid.bestfit(fit.grid)

Arguments

fit.grid  a regmed.grid object

Value

fit  best fit regmed object based on minimum BIC
grid  row out of grid.data of regmed.grid object corresponding to best fit

Author(s)

Dan Schaid, Greg Jenkins, Jason Sinnwell

References


See Also

regmed.grid
**regmed.lavaan.dat**

*Set up data to input to lavaan sem*

**Description**

Set up data to input to lavaan structural equation model (sem)

**Usage**

```
regmed.lavaan.dat(x, mediator, y)
```

**Arguments**

- `x` vector of exposure variables
- `mediator` matrix of mediator variables
- `y` vector of outcome variable

**Details**

Use the function `regmed.dat.check` to standardize all variables and subset to subjects without missing data

**Value**

A dataframe with updated `x`, `mediator`, and `y`

**Author(s)**

Dan Schaid, Jason Sinnwell

**References**


---

**regmed.lavaan.model**

*Create a lavaan model*

**Description**

Create a lavaan model for input to `lavan::sem()`

**Usage**

```
regmed.lavaan.model(fit.edge, fit.regmed)
```
Arguments

- `fit.edge` object created by `regmed.edges` function
- `fit.regmed` object created by `regmed.fit` from which the `fit.edge` object was made

Details

The `fit.regmed` object is needed for the fixed covariance estimates to be put into the model statement. The summary method supersedes the default summary from the `lavaan` package to only return the table of coefficients, as the covariances are fixed from `regmed.fit`.

Value

a character string that describes the mediation model in format of `lavaan` model

Author(s)

Dan Schaid, Greg Jenkins, Jason Sinnwell

Description

Use sure independence screening (Fan & Lv, 2008) to reduce the number of potential mediators when the number of potential mediators is large. This is based on ranking marginal correlations and then selecting the highest ranked values such that the number of parameters is less than the sample size. Because mediation depends on the two correlations, cor(x, med) and cor(med, y) we rank the absolute values of their products, |cor(x, med) * cor(med, y)|, and choose the highest k ranked values to determine which potential mediators to include in penalized mediation models. If k is not specified, the default value of k is n/2, where n is the sample size, because each mediator results in two parameters alpha and beta.

Usage

```
regmed.prefilter(x, mediator, y, k = NULL, x.std = TRUE,
                 med.std = TRUE, y.std=TRUE, max.cor=0.99)
```

Arguments

- `x` vector representing "exposure" variable (sometimes called instrumental variable)
- `mediator` matrix of mediators, rows are observations, columns are different mediators
- `y` vector representing outcome
- `k` Number of potential mediators to select. Default is n/2, where n is sample size.
- `x.std` logical (TRUE/FALSE) whether to standardize x by dividing by standard deviation of x. Note that x will be centered on its mean.
### mediators

`mediators` logical (TRUE/FALSE) whether to standardize mediators by dividing each mediator by its standard deviation. Note that mediators will be centered on their means.

`y` logical (TRUE/FALSE) whether to standardize y by dividing by standard deviation of y. Note that y will be centered on its mean.

`max.cor` maximum correlation within mediators, so that fitting is more robust

### Value

A list with x, mediator, and y, after subsetting to no missing values, applying x.std and med.std, and subsetting mediators to k top choices.

### Author(s)

Dan Schaid, Jason Sinnwell

### References


### See Also

`regmed`

### Examples

```r
data(medsim)
dim(med)
filtered <- regmed.prefilter(x[,1], med, y[,1], k=10)
dim(filtered$med)
```

### summary.mvregmed

#### Summary of mvregmed object

Summary of non-zero parameter estimates. Optional epsilon (eps) parameter controls rounding to 0.

#### Usage

```r
## S3 method for class 'mvregmed'
summary(object, eps=1e-3, ...)
```
Arguments

- **object**: mvregmed object returned from mvregmed.fit or mvregmed.grid.bestfit
- **eps**: parameters smaller than epsilon (eps) are rounded to zero.
- **...**: optional arguments

Value

Nothing is returned

Author(s)

Daniel Schaid and Jason Sinnwell

References


See Also

- `mvregmed.fit`
- `mvregmed.grid.bestfit`
Index

* datasets
  medsim, 4

* models
  mvregmed.dat.check, 5
  regmed.fit, 18
  regmed.grid, 19
  regmed.grid.bestfit, 21
  regmed.lavaan.model, 22
  regmed.prefilter, 23

* model
  mvregmed.edges, 5
  mvregmed.fit, 6
  mvregmed.graph.attributes, 8
  mvregmed.grid, 9
  mvregmed.grid.bestfit, 10
  mvregmed.grid.data, 11
  mvregmed.grid.update, 12
  mvregmed.init, 12
  mvregmed.lavaan.dat, 13
  mvregmed.lavaan.model, 14
  plot.mvregmed.grid, 15
  regmed.edges, 17
  regmed.grid, 16
  regmed.grid.bestfit, 21
  summary.mvregmed, 24

* package
  regmed-package, 2
  graph_from_edgelist, 8

med (medsim), 4
medsim, 4
mvregmed.dat.check, 5
mvregmed.edges, 5, 6, 10, 13, 25
mvregmed.fit, 5, 6, 10, 13, 25
mvregmed.graph.attributes, 8
mvregmed.grid, 5, 7, 9, 10, 11, 13
mvregmed.grid.bestfit, 10, 25
mvregmed.grid.data, 11
mvregmed.grid.update, 12
mvregmed.init, 12, 12
mvregmed.lavaan.dat, 13
mvregmed.lavaan.model, 14
plot.mvregmed.edges, 6
plot.mvregmed.edges (mvregmed.edges), 5
plot.mvregmed.grid, 15
plot.mvregmed.edges (regmed.edges), 17
plot.regmed.grid, 16, 21
print.regmed (regmed.fit), 18
print.regmed.grid (regmed.grid), 19

regmed, 24
regmed (regmed-package), 2
regmed-package, 2
regmed.dat.check (mvregmed.dat.check), 5
regmed.edges, 17, 19
regmed.fit, 17, 18, 21
regmed.grid, 16, 19, 21
regmed.grid.bestfit, 21
regmed.lavaan.dat, 22
regmed.lavaan.model, 22
regmed.prefilter, 23

summary.lavaan (regmed.lavaan.model), 22
summary.mvregmed, 24
summary.mvregmed.grid (mvregmed.grid), 9
summary.regmed, 19
summary.regmed (regmed.fit), 18
summary.regmed.grid (regmed.grid), 19

x (medsim), 4

y (medsim), 4