

# Package ‘pwSEM’

May 9, 2026

**Title** Piecewise Structural Equation Modelling

**Version** 1.0.0

**Description** Conduct dsep tests (piecewise SEM) of a directed, or mixed, acyclic graph without latent variables (but possibly with implicitly marginalized or conditioned latent variables that create dependent errors) based on linear, generalized linear, or additive models with or without a nesting structure for the data. Also included are functions to do desp tests step-by-step, exploratory path analysis, and Monte Carlo X<sup>2</sup> probabilities. This package accompanies Shipley, B, (2026). Cause and Correlation in Biology: A User's Guide to Path Analysis, Structural Equations and Causal Inference (3rd edition). Cambridge University Press.

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.3.1

**Imports** copula, gamm4, ggm, igraph, mgcv, poolr, stats

**Suggests** knitr, rmarkdown, testthat (>= 3.0.0)

**Depends** R (>= 2.10)

**LazyData** true

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** Bill Shipley [aut, cre] (ORCID:  
<<https://orcid.org/0000-0002-7026-3880>>)

**Maintainer** Bill Shipley <Bill.Shipley@USherbrooke.ca>

**Repository** CRAN

**Date/Publication** 2025-07-02 15:20:05 UTC

## Contents

basiSet.MAG . . . . .	2
-----------------------	---

CI.algorithm . . . . .	3
DAG.to.MAG.in.pwSEM . . . . .	4
generalized.covariance . . . . .	5
get.AIC . . . . .	6
MAG.to.DAG.in.pwSEM . . . . .	7
MCX2 . . . . .	7
nested_data . . . . .	8
perm.generalized.covariance . . . . .	9
pwSEM . . . . .	10
sim_normal.no.nesting . . . . .	11
sim_normal.with.nesting . . . . .	12
sim_poisson.no.nesting . . . . .	13
sim_tetrads . . . . .	13
summary.pwSEM.class . . . . .	14
vanishing.tetrads . . . . .	14
view.paths . . . . .	15

<b>Index</b>	<b>17</b>
--------------	-----------

---

basiSet.MAG

*basiSet.MAG*


---

## Description

Gets the union basis set of a MAG (mixed acyclic graph) involving either directed edges ( $X \rightarrow Y$ ) if  $X$  is a direct cause of  $Y$  or bi-directed edges ( $X \leftrightarrow Y$ ) if  $X$  is not a cause of  $Y$ ,  $Y$  is not a cause of  $X$ , but both  $X$  and  $Y$  share a common latent cause. It is easiest to create the MAG using the DAG() function of the ggm library and then modifying the binary output matrix by adding a value of 100 for each pair (row & column) of variables with a bi-directed edge.

## Usage

```
basiSet.MAG(cgraph)
```

## Arguments

cgraph	The adjacency matrix of the MAG, i.e. a square Boolean matrix of order equal to the number of nodes of the graph and with (1) a one in position (i,j) if there is an arrow from i to j; (2) a 100 in positions (i,j) and (j,i) if there is a double-headed arrow between i and j; (3) otherwise a zero in position (i,j) The rownames of the adjacency matrix are the nodes of the MAG.
--------	---

## Value

A list containing the m-separation claims in the union basis set

**Examples**

```
# W->X->Y->Z and X<->Z
# Create the DAG skeleton
mag<-ggm::DAG(X~W,Y~X,Z~Y)
# Add the dependent error
mag[2,3]<-mag[3,2]<-100
basiSet.MAG(mag)
```

---

CI.algorithm

*The CI.algorithm function*


---

**Description**

This function implements the exploratory method of causal discovery called the CI (Causal Inference) algorithm of Pearl.

**Usage**

```
CI.algorithm(
  dat,
  family = NA,
  nesting = NA,
  smooth = TRUE,
  alpha.reject = 0.05,
  constrained.edges = NA,
  write.result = TRUE
)
```

**Arguments**

dat	A data frame containing the variables for which a partially- oriented dependency graph is sought. All variables in this data frame will be used except for those listed in the nesting= argument; these variables describe the nesting structure (if present) of the other variables.
family	A data frame giving the name of the distributional type of each variable that is not Gaussian. Example: data.frame(X1="binomial") if all variables except X1 are Gaussian but X1 is binomial.
nesting	A named list. Each name in the list is the name of the variable in the data set (except for the nesting= variables) followed by a character vector giving the names of the variables in dat holding the nesting structure. The default is nesting=NA, which is used if there are no nesting variables.
smooth	A logical value stating if linear (or generalized linear) links between the variables are assumed (smooth=FALSE) or not (smooth=FALSE). The default is smooth=TRUE, in which case generalized additive models are used to measure conditional independence and therefore non-linear relationships are assumed.

<code>alpha.reject</code>	A numerical value between 0 and 1 giving the "significance level" to use when judging (conditional) independence. The default value is <code>alpha.reject=0.05</code> .
<code>constrained.edges</code>	A character object giving the edges that must be constrained to not exist ("X Y"), or X is a causal parent of Y ("X->Y") or X and Y have a common latent parent ("X<->Y"). Default is <code>constrained.edges=NA</code> (no edges are constrained).
<code>write.result</code>	A logical value indicating if you want the resulting partially-oriented dependency graph to be output to the screen ( <code>write.result = T</code> ) or just the adjacency matrix returned as output. The default value is <code>write.result = T</code> .

**Value**

Just the partially-oriented graph output to the screen or just the adjacency matrix as a binary (0/1) matrix

**Examples**

```
set.seed(11)
X1<-rnorm(500)
X2<-0.5*X1+rnorm(500,0,sqrt(1-0.5^2))
X3<-0.5*X2+rnorm(500,0,sqrt(1-0.5^2))
X4<-0.5*X2+rnorm(500,0,sqrt(1-0.5^2))
X5<-0.5*X3+0.5*X4+rnorm(500,0,sqrt(1-2*0.5^2))
my.dat<-data.frame(X1,X2,X3,X4,X5)
CI.algorithm(dat=my.dat,smooth=FALSE,alpha.reject=0.05)
```

---

DAG.to.MAG.in.pwSEM     *Title DAG.to.MAG.in.pwSEM*

---

**Description**

Title DAG.to.MAG.in.pwSEM

**Usage**

```
DAG.to.MAG.in.pwSEM(full.DAG, latents = NA, conditioning.latents = NULL)
```

**Arguments**

<code>full.DAG</code>	The DAG with latent variables, usually produced with the <code>DAG()</code> function of the <code>ggm</code> package
<code>latents</code>	A character vector giving the names of the latent variables in the DAG
<code>conditioning.latents</code>	A character vector giving the names of those latents, listed in the "latents" argument, that serve as conditioning variables for sampling (i.e. selection bias).

**Value**

A binary (0/1) matrix holding the MAG

**Examples**

```
library(ggm)
full.dag<-DAG(x1~L1, x2~x1, x3~x2+L1, x4~x3, L2~x2+x4)
DAG.to.MAG.in.pwSEM(full.DAG=full.dag, latents=c("L1", "L2"),
conditioning.latents=c("L2"))
```

---

generalized.covariance

*Generalized covariance function*

---

**Description**

This function calculates the generalized covariance statistic of Shah, R.D. & Peters, J. (2020); i.e.  $Y1 | Y2 | C$ , where  $C$  is a set of common conditioning variables.  $R1$  and  $R2$  are the response residuals from pairs of regressions of two dependent variables ( $Y1$  and  $Y2$ ) on a set of conditioning variables.

Shah, R.D. & Peters, J. (2020). The hardness of conditional independence testing and the generalized covariance measure. *The Annals of Statistics* 48:1514-1538.

**Usage**

```
generalized.covariance(R1, R2)
```

**Arguments**

R1	a numerical vector of residuals
R2	a second numerical vector of residuals

**Value**

A list containing  $T.stat$  (the test statistic) and  $prob$  (asymptotic null probability of the  $T$  statistic).

**Examples**

```
#generalized.covariance function: X1_|_X3|{X2}
R1<-residuals(mgcv::gam(X3~X2,data=sim_normal.no.nesting,family=gaussian),
type="response")
R2<-residuals(mgcv::gam(X1~X2,data=sim_normal.no.nesting,family=gaussian),
type="response")
generalized.covariance(R1,R2)
```

---

 get.AIC

*Title get.AIC*


---

### Description

Title get.AIC

### Usage

```
get.AIC(sem.model, MAG, data)
```

### Arguments

sem.model	A list containing the structural equations, each created using either the gam or the gmm functions of the mgcv package
MAG	A matrix encoding the directed acyclic graph (DAG), or the mixed acyclic graph, of the structural equations model. This is created using the DAG or the makeGM function in the gmm library
data	A data frame holding the observed data used in the calls to the models in the sem.model object

### Value

A data frame containing the log-likelihood of the full SEM (LL), the number of free parameters that were estimated (K), along with the AIC and the bias-corrected AIC (AICc)

### Examples

```
library(mgcv)
library(ggm)
set.seed(10)
N<-1000
L1<-rnorm(N)
x1<-0.5*L1+rnorm(N,0,sqrt(1-0.5^2))
x2<-0.5*x1+rnorm(N,0,sqrt(1-0.5^2))
x3<-0.5*L1+0.5*x2+rnorm(N,0,sqrt(1-2*0.5^2))
x4<-0.5*x3+rnorm(N,0,sqrt(1-0.5^2))
my.dat<-data.frame(x1,x2,x3,x4)

my.list<-list(gam(x1~1,data=my.dat),
             gam(x2~x1,data=my.dat),
             gam(x3~x2,data=my.dat),
             gam(x4~x3,data=my.dat))

full.dag<-DAG(x1~L1,x2~x1,x3~x2+L1,x4~x3)
my.mag<-DAG.to.MAG.in.pwSEM(full.dag=full.dag,latents=c("L1"))
get.AIC(sem.model=my.list,MAG=my.mag,data=my.dat)
```

---

MAG.to.DAG.in.pwSEM      *Title MAG.to.DAG.in.pwSEM*

---

### Description

This function converts a MAG, usually created with the `makeMG()` function of the `ggm` library, with 0/1 for the directed paths, 100 for marginalized latents and 10 for conditioned latents.

### Usage

```
MAG.to.DAG.in.pwSEM(MAG, marginalized.latents, conditioned.latents)
```

### Arguments

`MAG`                      a MAG input as a matrix  
`marginalized.latents`                      a list containing pairs of variables that share a common latent marginalized cause; eg `list(X2~~X3, X4~~X5)`  
`conditioned.latents`                      a list containing pairs of variables that cause a common latent conditioned cause; eg `list(X2~~X3, X4~~X5)`

### Value

A binary matrix of 0/1 values representing the DAG produced by adding the latent variables (named as L1, L2 etc)

### Examples

```
library(ggm)
my.mag<-makeMG(dg=DAG(X2~X1, X3~X2, X4~X3), bg=UG(~X2*X4))
DAG.with.latent<-MAG.to.DAG.in.pwSEM(my.mag,
marginalized.latents=list(X2~~X4), conditioned.latents=NULL)
```

---

MCX2

*Title Monte Carlo chi-square (MCX2)*

---

### Description

The maximum likelihood chi-square statistic that is commonly calculated in structural equations modelling only asymptotically follows a theoretical chi-squared distribution; with small sample sizes it can deviate enough from the theoretical distribution to make it problematic. This function estimates the empirical probability distribution of the Maximum Likelihood Chi-Square statistic, given a fixed sample size and degrees of freedom, and outputs the estimated null probability given this sample size and degrees of freedom.

**Usage**

```
MCX2(model.df, n.obs, model.chi.square, n.sim = 10000, plot.result = FALSE)
```

**Arguments**

```
model.df      the model degrees of freedom
n.obs         the number of observations (lines) in the data set
model.chi.square
               the maximum likelihood chi-squared statistic
n.sim         the number of independent simulation runs
plot.result   a graphical output of the result
```

**Value**

A list containing MCprobability (Monte carlo null probability), and MLprobability (maximum likelihood probability)

**Examples**

```
MCX2(model.df=10, n.obs=15, model.chi.square=18.2, plot.result=TRUE)
```

---

nested\_data

*nested\_data:*

---

**Description**

Empirical example with nesting structure, a binomial variable, and 3-level nesting structure.

**Usage**

```
nested_data
```

**Format****'nested\_data':**

A data frame with 1309 rows and 8 columns:

**year** Year observation is collected

**nest** Nest in which the observation is collected

**ind** the individual within a given nest

**XR** A binary variable

**XM** A Gaussian variable

**XH** A Gaussian variable

**HP** A Gaussian variable

**XF** A Gaussian variable

---

perm.generalized.covariance  
*perm.generalized.covariance*

---

## Description

This performs a permutation version of the generalized covariance test (see: `generalized.covariance`), which tests for conditional independence of two random variables ( $Y1$ ,  $Y2$ ) conditional of a common set of conditioning variables  $C$ ; see Shah, R.D. & Peters, J. (2020). i.e.  $Y1 \perp Y2 \mid C$ .  $R1$  and  $R2$  are the response residuals from pairs of any type of appropriate regressions of two dependent variables ( $Y1$  and  $Y2$ ) on a set of conditioning variables.

Shah, R.D. & Peters, J. (2020). The hardness of conditional independence testing and the generalized covariance measure. *The Annals of Statistics* 48:1514-1538.

## Usage

```
perm.generalized.covariance(R1, R2, nperm = 5000)
```

## Arguments

<code>R1</code>	a numerical vector (typically residuals of the first regression)
<code>R2</code>	a numerical vector (typically residuals of the second regression)
<code>nperm</code>	the number of permutations (defaults to 5000)

## Value

A list containing the T statistic (`T.stat`), `permutation.prob`: the estimated null probability of independence of  $R1$  and  $R2$ , based on the chosen number of permutations, `lower.95.CI` and `upper.95.CI`: the 95% confidence intervals of the estimated null probability

## Examples

```
R1<-residuals(mgcv::gam(X3~X2,data=sim_normal.no.nesting,family=gaussian),
type="response")
R2<-residuals(mgcv::gam(X1~X2,data=sim_normal.no.nesting,family=gaussian),
type="response")

#perm.generalized.covariance function
perm.generalized.covariance(R1,R2,nperm=5000)
```

pwSEM

*The pwSEM function***Description**

This function performs a "piecewise" structural equation model without explicit latent variables (a "path" model), including with implicitly marginalized latents and implicitly conditioned latents ("correlated errors"), based on generalized linear or additive models, possibly in a mixed model context, and then tests the causal structure against an empirical data set using a dsep test. Therefore, it is able to model linear, generalized linear, generalized linear mixed, additive, generalized additive, and generalized additive mixed models.

**Usage**

```
pwSEM(
  sem.functions,
  marginalized.latents = NULL,
  conditioned.latents = NULL,
  data,
  use.permutations = FALSE,
  n.perms = 5000,
  do.smooth = FALSE,
  all.grouping.vars = NULL
)
```

**Arguments**

- `sem.functions` A list giving the `gamm4` (`gamm4` package) or `gam` (`mgcv` package) models associated with each variable in the sem, INCLUDING exogenous variables.
- `marginalized.latents` A list giving any dependent errors (correlated error variables), given in the form of `list(X~~Y,...,X~~Z)`. The same pair cannot also be listed in `conditioned.latents`.
- `conditioned.latents` A list giving any implicitly conditioned latents (selection bias), given in the form of `list(X~~Y,...,X~~Z)`. The same pair cannot also be listed in `marginalized.latents`.
- `data` A data frame containing the empirical data
- `use.permutations` A logical value (TRUE, FALSE) indicating if you want to use permutation probabilities for the d-separation tests. Defaults to FALSE. You should use TRUE for smaller data sets.
- `n.perms` The number of permutation runs to use for permutation probabilities. Defaults to 5000.
- `do.smooth` A logical value indicating if you want to use regression smoothers (generalized additive models) for the dsep tests. Defaults to FALSE. TRUE will fit nonlinear

(regression smoothers) when evaluating the d-separation claims, but this will slow down the function.

all.grouping.vars

A character vector giving the names of all variables involved in the sem functions that define groups for random effects.

## Value

A list containing the following elements: causal.graph, dsep.equivalent.causal.graph, basis.set, dsep.probs, sem.functions,C.statistic, prob.C.statistic, AIC, n.data.lines, use.permutations, n.perms

## Examples

```
# Simulated data with correlated errors involving endogenous
# variables, normally-distributed data and with a 2-level grouping
# structure and using smoothing splines for the d-separation tests.
# Data generated using this mixed acyclic graph:
# X1->X2->X3->X4 and X2<->X4

my.list<-list(gamm4::gamm4(X1~1,random=~(1|group),data=sim_normal.with.nesting,family=gaussian),
  gamm4::gamm4(X2~X1,random=~(1|group),data=sim_normal.with.nesting,family=gaussian),
  gamm4::gamm4(X3~X2,random=~(1|group),data=sim_normal.with.nesting,family=gaussian),
  gamm4::gamm4(X4~X3,random=~(1|group),data=sim_normal.with.nesting,family=gaussian))
# RUN THE pwSEM FUNCTION WITH PERMUTATION PROBABILITIES AND INCLUDING THE DEPENDENT ERRORS
out<-pwSEM(sem.functions=my.list,marginalized.latents=list(X4~~X2),
  data=sim_normal.with.nesting,use.permutations = TRUE,
  do.smooth=TRUE,all.grouping.vars=c("group"))
summary(out,structural.equations=TRUE)

# see vignette("pwSEM")
```

---

sim\_normal.no.nesting *sim\_normal.no.nesting Simulated data with correlated errors involving endogenous variables, normally-distributed data and without any grouping structure Data generated using this mixed acyclic graph: X1->X2->X3->X4 and X2<->X4*

---

## Description

sim\_normal.no.nesting Simulated data with correlated errors involving endogenous variables, normally-distributed data and without any grouping structure Data generated using this mixed acyclic graph: X1->X2->X3->X4 and X2<->X4

## Usage

```
sim_normal.no.nesting
```

**Format****'sim\_normal.no.nesting':**

A data set with 100 rows and 4 columns

**X1** A Gaussian variable

**X2** A Gaussian variable

**X3** A Gaussian variable

**X4** A Gaussian variable

---

sim\_normal.with.nesting

*sim\_normal.with.nesting: Simulated data with correlated errors involving endogenous variables, normally-distributed data and without any grouping structure Data generated using this mixed acyclic graph:  $X1 \rightarrow X2 \rightarrow X3 \rightarrow X4$  and  $X2 \leftarrow X4$*

---

**Description**

sim\_normal.with.nesting: Simulated data with correlated errors involving endogenous variables, normally-distributed data and without any grouping structure Data generated using this mixed acyclic graph:  $X1 \rightarrow X2 \rightarrow X3 \rightarrow X4$  and  $X2 \leftarrow X4$

**Usage**

sim\_normal.with.nesting

**Format****'sim\_normal.with.nesting':**

A data set with 100 rows and 5 columns. The last column "group" holds the grouping structure

**X1** A Gaussian variable

**X2** A Gaussian variable

**X3** A Gaussian variable

**X4** A Gaussian variable

**group** A character variable holding the group names

---

sim\_poisson.no.nesting

*sim\_poisson.no.nesting: Simulated data with correlated errors involving endogenous variables, Poisson-distributed data and without any grouping structure Data generated using this mixed acyclic graph:  $X1 \rightarrow X2 \rightarrow X3 \rightarrow X4$  and  $X2 \leftarrow X4$*

---

### Description

sim\_poisson.no.nesting: Simulated data with correlated errors involving endogenous variables, Poisson-distributed data and without any grouping structure Data generated using this mixed acyclic graph:  $X1 \rightarrow X2 \rightarrow X3 \rightarrow X4$  and  $X2 \leftarrow X4$

### Usage

sim\_poisson.no.nesting

### Format

**'sim\_poisson.no.nesting':**

A data set with 100 rows and 4 columns

**X1** A Gaussian variable

**X2** A Poisson variable

**X3** A Poisson variable

**X4** A Poisson variable

---

sim\_tetrads

*sim\_tetrads: Simulated data to be used with the vanishing.tetrads function Data generated using this directed acyclic graph, with L being latent:  $L \rightarrow X1, L \rightarrow X2, L \rightarrow X3 \rightarrow X4$*

---

### Description

sim\_tetrads: Simulated data to be used with the vanishing.tetrads function Data generated using this directed acyclic graph, with L being latent:  $L \rightarrow X1, L \rightarrow X2, L \rightarrow X3 \rightarrow X4$

### Usage

sim\_tetrads

**Format****'sim\_tetrads':**

A data set with 500 rows and 4 columns

**X1** A Gaussian variable

**X2** A Gaussian variable

**X3** A Gaussian variable

**X4** A Gaussian variable

---

summary.pwSEM.class     *Summary Method for pwSEM Class*

---

**Description**

Provides a summary for objects of class pwSEM.

**Usage**

```
## S3 method for class 'pwSEM.class'
summary(object, structural.equations = FALSE, ...)
```

**Arguments**

object	An object of class pwSEM.
structural.equations	Logical (TRUE outputs the structural equations)
...	other arguments

**Value**

A summary of the pwSEM object.

Returns a summary of the pwSEM object that is written to the screen

---

vanishing.tetrads     *The vanishing.tetrads function*

---

**Description**

This function implements the vanishing tetrads theorem of Spirtes, Glymour & Scheines (1993). If a set of four variables in dat has a saturated unoriented dependency graph in CI.algorithm, and a tetrad equation is zero, then this is evidence for a latent variable.

**Usage**

```
vanishing.tetrads(dat, sig = 0.05, bootstrap = FALSE, B = 1000)
```

**Arguments**

dat	A data frame containing the observed variables. No other variables can be in this file, such as ones describing the nesting structure.
sig	A numerical value between 0 and 1 giving the significance level to use when judging (conditional) independence. The default value is 0.05.
bootstrap	A logical value specifying if you want bootstrap probabilities or not. Defaults to FALSE
B	The number of bootstrap samples required. Defaults to 1000.

**Value**

Just output to the screen listing each tetrad equation, its value and its significance level.

**Examples**

```
#Determines which of the three tetrad equations are zero in this data set
#having 500 observations and 4 variables. Since this set of 4 variables
#has a saturated partially oriented dependency graph,
#as shown using the CI.algorithm function, the tetrad equations
#that are zero (i.e. vanish) identify where latent variables occur that
#are common causes of these variables
#Since this is a saturated partially oriented dependency graph:
vanishing.tetrads(dat=sim_tetrads,sig=0.05)
```

---

view.paths

*view.paths*

---

**Description**

This is a function, usually called after pwSEM, to allow you to visually see how two variables in the DAG relate to each other along all directed paths from one to the other and to see how the 1st derivative of this relationship changes as the "from" variable changes. For linear relationships, this is a constant (the path coefficient).

**Usage**

```
view.paths(
  from,
  to,
  sem.functions,
  data,
  minimum.x = NULL,
  maximum.x = NULL,
  scale = "response",
  return.values = FALSE,
  dag
)
```

**Arguments**

from	The name (character) of the variable at the beginning of the path
to	The name (character) of the variable at the end of the path
sem.functions	A list containing the gam or gamm4 functions that define the structural equations. This is normally the sem.functions returned from pwSEM
data	The data frame containing the empirical data used in the SEM functions
minimum.x	The minimum value of the "from" variable; defaults to NULL, in which case the minimum and maximum values of the "from" variable in the data set are used.
maximum.x	The maximum value of the "to" variable; defaults to NULL, in which case the minimum and maximum values of the "from" variable in the data set are used.
scale	The chosen scale in which to express the results; either "response" (which uses the original scale of the variable) or "link" (which uses the scale of the link function for the "to" variable)
return.values	A logical value (TRUE returns the values of "from" and an estimate of the 1st derivative of the function for from→to)
dag	The directed acyclic graph (DAG) used; usually this is the causal.graph object returned from a call to pwSEM

**Value**

Graphs are produced for each directed path from→to and (return.values=TRUE) a data frame containing the values of the "from" variable and approximate values of the 1st derivative of the function linking from→to (i.e.the path coefficient) if the relationship is linear.

**Examples**

```
# Example with correlated endogenous errors, Poisson distributed variables
#and no nesting structure in the data
# DAG: X1→X2→X3→X4 and X2←X4
my.list<-list(mgcv::gam(X1~1,data=sim_poisson.no.nesting,family=gaussian),
             mgcv::gam(X2~X1,data=sim_poisson.no.nesting,family=poisson),
             mgcv::gam(X3~X2,data=sim_poisson.no.nesting,family=poisson),
             mgcv::gam(X4~X3,data=sim_poisson.no.nesting,family=poisson))
out<-pwSEM(sem.functions=my.list,marginalized.latents=list(X4~~X2),
           data=sim_poisson.no.nesting,use.permutations = TRUE,n.perms=10000)
# To see each of the effects of X1 on X4 (only one in this example), we
# use view.paths() while imputing the list of SEM functions (out$sem.functions)
# and the DAG (out$causal.graph) that are output from the pwSEM function.

view.paths(from="X1",to="X4",sem.functions=out$sem.functions,data=
sim_poisson.no.nesting,scale="response",dag=out$causal.graph)
```

# Index

## \* datasets

- nested\_data, 8
- sim\_normal.no.nesting, 11
- sim\_normal.with.nesting, 12
- sim\_poisson.no.nesting, 13
- sim\_tetrads, 13

basiSet.MAG, 2

CI.algorithm, 3

DAG.to.MAG.in.pwSEM, 4

generalized.covariance, 5

get.AIC, 6

MAG.to.DAG.in.pwSEM, 7

MCX2, 7

nested\_data, 8

perm.generalized.covariance, 9

pwSEM, 10

sim\_normal.no.nesting, 11

sim\_normal.with.nesting, 12

sim\_poisson.no.nesting, 13

sim\_tetrads, 13

summary.pwSEM.class, 14

vanishing.tetrads, 14

view.paths, 15