Package 'FIAR'

June 5, 2018

Type Package

Title Functional Integration Analysis in R

Version 0.6

Author Bjorn Roelstraete

Maintainer Bjorn Roelstraete <Bjornroelstraete@gmail.com>

Description Contains Dynamic Causal Models (DCM), Autoregressive Structural Equation Models (ARSEM), and multivariate partial and conditional Granger causality tests for analysing fMRI connectivity data.

Depends lavaan, Matrix, np

License GPL-2

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

NeedsCompilation yes

Repository CRAN

Date/Publication 2018-06-05 11:48:21 UTC

R topics documented:

ARorder 2
ARsem 3
attentiondata 4
condGranger 5
DCMatt 6
dcmCompare 7
dcmEstimate 7
dcmEvidence 8
DCMex 9
dcmGenerate 11
dcmParam 12
diffGranger 14

grangerdata 15
hrfConvolve 16
partGranger 17
pdiffGranger 19
semdata 20
SEMextract 21
X0 21

Index **22**

`ARorder` *Estimate AR order of Multivariate timeseries*

Description

Compute AIC of a series of Multivariate AR models and returns the order of the model which minimizes this AIC.

Usage

```
ARorder(data, min=1,max = 10,type='AIC')
```

Arguments

`data`	timeseries to compute autoregressive order of.
`min`	Minimum order of AR model to check.
`max`	Maximum order of AR model to check.
`type`	Use AIC or BIC to compute model order.

Value

returns the order (>=1) of the autoregressive model which minimizes the AIC or BIC

Author(s)

Bjorn Roelstraete

Examples

```
# Compute the AR order of semdata based on AIC, with a maximum order of 10 to reduce computing time.
ARorder(semdata,max=10)
```

ARsem *Auto-regressive SEM*

Description

Fit an auto-regressive SEM of the specified order. The function automatically extends the given model and dataset to the model and dataset of the specified order. The function is actually a wrapper around the function sem() from the package lavaan.

Usage

```
ARsem(model, data, order)
```

Arguments

model	A vector specifying the model of AR order 0 (AR(0)) . The vector should be written as an n by n matrix where n is the number of regions in the network (see example). For every expected connection ij from region i (column) to region j (row) the vector contains '1' and '0' otherwise.
data	contains all observations (rows) and variables (colums) in the network m0. Only variables that are in the model m0 should be in the dataset.
order	Integer. The order of the AR model.

Details

An AR model of order q contains the t-0, t-1, t-2,...t-q timeseries and these timeseries are connected based on the model of order 0. The function will transform this model in the correct AR(q) model and the data set in the lagged data set containing all lagged variables. Let us take the very simple example of a dataset with 2 variables X and Y. If there is an arrow from X to Y, the function will create the AR(1) model with an additional arrow from X-1 to Y-1, from X-1 to X and from Y-1 to Y. Variables X-1 and Y-1 are automatically created within the function.

Value

An object of class 'lavaanModel', for which several methods are available, including a 'summary' method.

Author(s)

Bjorn Roelstraete

References

Kim, J., Zhu, W., Chang, L., Bentler, P., and Ernst, T. (2007). Unified Structural Equation Modeling Approach for the analysis of Multisubject, Multivariate Functional MRI Data. Human Brain Mapping 85 93.

Examples

```
# Example dataset with three brainregions x, y, z.
head(semdata)
# Prior model with connections from (column) x to (row) y and from y to z.
model <- c(0,0,0,
           1,0,0,
   0,1,0)
# Perform classical SEM

 fit0 <- ARsem(model,semdata)
 summary(fit0)

# Calculate AR() order of the data
ARorder(semdata,max=10)

# Compute AR(3) SEM
 fit3 <- ARsem(model,semdata,order=3)
 summary(fit3)
```

`attentiondata` *Demo dataset*

Description

Time series from 'Attention to visual motion' dataset).

Usage

```
data(semdata)
```

Format

A data frame of 360 observations of 3 variables.

- `x` Time series of V1
- `y` Time series of V5
- `z` Time series of SPC

Examples

```
head(attentiondata)
```

condGranger *Conditional granger causality*

Description

Compute conditional granger causality of multivariate timeseries.

Usage

```
condGranger(data, nx = 1, ny = 1, order=1, perm = FALSE, prob=TRUE, bs = 100)
```

Arguments

data	object containing all observations (rows) and variables (columns) that are being considered. The variables should be ordered as follows: First the variables that are supposed to granger cause a set of other variables (>=1). Then the set of variables (>=1) that are Granger caused by the first set of variables. Finally, a set of variables to condition on(>=1).
nx	The number of variables (>=1) that Granger cause a set of other variables (default = 1), conditioned on a third set of variables (>=1).
ny	The number of variables (>=1) that are Granger caused by the first nx variables (default = 1), conditioned on a third set of variables (>=1).
order	Autoregressive order (>=1) of timeseries. Can be computed using ARorder().
perm	Logical. If perm = FALSE (default), only the Granger causality measure is produced. If perm = TRUE, the Granger test is computed and a permutation test is performed to do inference.
prob	Logical. If TRUE, the F statistic is returned together with the p-value.
bs	Number of permutation samples. Only works when perm = TRUE. Default=100

Value

Conditional Granger causality F statistic with p-value.

Author(s)

Bjorn Roelstraete

References

Guo, S., Seth, A.K., Kendrick, K.M., Zhou, C., Feng, J.(2008). Partial Granger Causality-Eliminating Exogenous Inputs and Latent Variables. Journal of Neuroscience Methods. 79-93.

See Also

ARorder

Examples

```
# Example data with 5 regions x, y, z, q, w
head(grangerdata)

# Calculate AR() order of the data
ARorder(grangerdata,max=10)

# Compute conditional granger causality of region x (nx =1) to regions y and z (ny=2),
# conditional on regions q and w for an AR(3) model.
F <- condGranger(grangerdata,nx=1,ny=2,order=3)

# Compute F and permutation H0 distribution
F <- condGranger(grangerdata,nx=1,ny=2,order=3, perm=TRUE)
```

DCMatt *Demo dataset*

Description

An DCM object containing all parameters for estimating the attention to visual motion DCM from the SPM8 manual.

Usage

```
data(DCMatt)
```

Format

A list containing:

- `a` Prior anatomical connections
- `b` Prior functional connections
- `c` Prior input connections
- `h` Prior hemodynamic parameters
- `ons` List containing onsets per experimental conditions in scans
- `dur` List containing durations per experimental conditions in scans
- `T` Number of timebins
- `TR` repetition time in seconds
- `TE` Echo time in seconds
- `m` Number of inputs
- `v` Number of scans
- `n` Number of regions
- `names` names of regions
- `x` Number of states (5 per region)
- `X0` Confounds or null space
- `names` Names of regions

Examples

```
names(DCMatt)
```

`dcmCompare` *DCM comparison*

Description

Compute Bayes Factor based on fitvalues of 2 DCM's

Usage

```
dcmCompare(DCM1, DCM2)
```

Arguments

`DCM1`	First DCM.
`DCM2`	Second DCM.

Value

Returns a Bayes Factor based on both AIC's of the models and BIC's.

Author(s)

Bjorn Roelstraete

See Also

spmEstimate

`dcmEstimate` *DCM estimator*

Description

Estimate parameters of a bilinear DCM

Usage

```
dcmEstimate(DCM,ts)
```

Arguments

`DCM`	DCM object.
`ts`	Timeseries to fit the model to.

Value

Returns posterior parameter values:

DCM$A	posterior anatomical connections from column j to row i.
DCM$B	posterior functional connections from column j to row i for every input [,,k].
DCM$C	posterior input connections from input k (row) to region l (column).
DCM$H	posterior hemodynamic parameters.
DCM$Cp	posterior parameter covariance.
DCM$Ce	posterior error covariance.

Author(s)

Bjorn Roelstraete

References

Friston et al. 2003. Dynamic Causal Modeling. Neuroimage, 19, 1273-1302.

See Also

dcmGenerate

Examples

```
# Not run
# Estimate posterior parameters of model DCMex with data DCMex$sim
# ts <-  dcmGenerate(DCMex, SNR=1, ar=.2, names=c('V1','V2','V3'))
# DCMex <- dcmEstimate(DCMex, ts)

#Posterior anatomical connections
 DCMex$A

#Posterior functional connections
 DCMex$B

#Posterior input connections
 DCMex$C
```

---

dcmEvidence *modelfit*

Description

Compute AIC and BIC of a DCM

Usage

```
dcmEvidence(DCM, ts)
```

Arguments

DCM	DCM object.
ts	Timeseries to fit the model to.

Value

Creates 2 extra fields DCM$AIC and DCM$BIC

Author(s)

Bjorn Roelstraete

See Also

spm.dcm.estimate

Examples

```
# Compute how well the model DCMex fits the timeseries DCMex$sim
 ts <-  dcmGenerate(DCMex, SNR=1, ar=.2, names=c('V1','V2','V3'))
 DCMex <- dcmEvidence(DCMex, ts)

 DCMex$AIC
 DCMex$BIC
```

DCMex *Demo dataset*

Description

A toy dataset containing all parameters of an estimated DCM

Usage

```
data(DCMex)
```

Format

A list containing:

`a` Prior anatomical connections

`b` Prior functional connections

`c` Prior input connections

`h` Prior hemodynamic parameters

`ons` List containing onsets per experimental conditions in scans

`dur` List containing durations per experimental conditions in scans

`T` Number of timebins

`TR` repetition time in seconds

`TE` Echo time in seconds

`m` Number of inputs

`v` Number of scans

`n` Number of regions

`HPF` Length of High Pass filter in seconds

`x` Number of states (5 per region)

`sf` Stimulus function of the experiment. One column per experimental condition

`s` Information about stimulusfunction

`T0` Information about stimulusfunction

`dt0` Information about stimulusfunction

`X0` Confounds or null space

`sim` Simulated timeseries from model with SNR=1 and ar=0

`priors` List containing prior parameter covariances (pC), parameter expectations (pE), hemodynamic expectations (qE), and hemodynamic covariances (qC)

`Ep` Posterior model parameters

`A` Posterior anatomical connections

`B` Posterior functional connections

`C` Posterior input connections

`H` Posterior hemodynamic parameters

`Cp` Posterior parameter covariances

`F` Log evidence

`Ce` Posterior error covariances

`names` Names of regions

Examples

```
names(DCMex)
```

dcmGenerate *DCM timeseries generator*

Description

Generate simulated timeseries from a specified DCM

Usage

```
dcmGenerate(DCM, SNR = 0, ar = 0,names=DCM$names)
```

Arguments

DCM	A DCM list containing all model and experimental parameters. This list can be constructed using dcm_param or manually.
SNR	SNR of the timeseries. The number represents sd(signal)/sd(noise). If SNR=0 the pure signal is generated.
ar	Autoregressioncoefficient of the noise added. 0 (default) means white, gaussian noise.
names	The names of the variables.

Value

Function creates a field DCM$sim, which contains the simulated timeseries per timepoint (rows) and region (columns).

Author(s)

Bjorn Roelstraete

See Also

dcm_param

Examples

```
# Use example DCMex to generate three timeseries V1, V2, V3 with a SNR of 1 and AR(.2)

 ts <-  dcmGenerate(DCMex, SNR=1, ar=.2, names=c('V1','V2','V3'))
 plot(ts[,1], t='b')
```

dcmParam *DCM object builder*

Description

Automated step by step procedure to enter parameters needed for the DCM analysis. Everything is stored in an DCM list that can be used to generate timeseries or estimate the model. The function also immediately calculates the High pass filter (HPF) and stimulus (SF) function.

Usage

```
dcmParam(a, b, c, ons = list(), dur = list(), v, n, m, TR,
          h = c(0.65, 0.41, 0.98, 0.32, 0.34, 0), names=c(),
          TE = 0.04, T = 16, x = 5 * n, HPF=0, auto = FALSE)
```

Arguments

a	vector of length n*n representing the anatomical connections between brain regions. The 'a' vector should look like an n*n matrix with a non zero element (in Hz.) on location ij when there is an expected connection from region j (column) to region i (row)(See examples).
b	vector of length n*n*m representing the functional connections. Should be written as m n*n matrices whith non zero elements ijk (in Hz.) if input k influences the connection from region j (column) to region i (row) and zero otherwise.
c	vector of length n*m representing the input connections. Should be written as an m by n matrix whith non zero elements kj (in Hz.) if input k (row) influences region j (columns).
ons	list containing vector of onsets (in scans) for every input.
dur	list containing vector of durations (in scans) for every input. dur=0 represents event-related inputs. If durations are equal over the entire experiment, the number only needs to be entered once. If durations differ, the length of the durationvector should match the length of the onsetvector.
v	number of scans
n	number of regions in model
m	number of inputs (experimental conditions)
TR	repetition time of the experiment
h	the parameters of the hemodynamic model
names	names of brain regions
TE	echo time in seconds (default = 0.04 s)
T	number of timebins (default = 16)
x	number of states (= 5 times number of regions)
HPF	High pass filter in seconds (default = 0 seconds)

auto — logical. If FALSE (default) the prespecified DCM object is used and the function only serves to construct the HPF and SF. If TRUE, the DCM object need not be prespecified and is constructed in a step by step procedure, whereafter the HPF and SF are constructed.

Value

DCM list containing all above mentioned model and scanner parameters.

Author(s)

Bjorn Roelstraete

Examples

```
## Specify connectivity parameters in a 3 region network with connections
## from region 1 to region 2 with a strenght of .8 Hz and region 2 to
## region 3 with .65Hz.
a <- c( 0, 0, 0,
       .7, 0, 0,
        0,.4, 0)

## Specify 2 experimental manipulations (inputs) where the first directly
## influences region 1 with .4 Hz. and the second region 2 with .2 Hz
c <- c(.4, 0, 0,
        0,.5, 0)

## Specify the functional connectivities between region 1 and 3 of .2 Hz.
## caused by input 1. Input 2 influences the functional connectivity from
## region 3 to region 2.
b <- c(0, 0, 0,
       0, 0, 0,
       0,.2, 0,
       0, 0, 0,
       0, 0,.2,
       0, 0, 0)

v <- 240            # number of scans
n <- 3              # number of regions
m <- 2              # number of inputs = number of rows in DCM$c

## The onsets of input 1 are at scan 0, 30, 60, 120, and 200. The onsets of
## input 2 at scan 30, and 120.

ons.input1 <- c(0, 60, 120, 180)
ons.input2 <- c(0, 30, 60, 90, 120)

## The duration of input 1 is always 30 scans. The duration of input 2 is 15
## scans.

dur.input1 <- 30
dur.input2 <- 15
```

```
TR <- 1
DCM <- dcmParam(a, b, c, ons=list(ons.input1,ons.input2),
                dur=list(dur.input1,dur.input2), v, n, m, TR=1)
names(DCM)

# Or construct DCM by automated step by step:
# DCM <- dcmParam(auto=TRUE)
```

`diffGranger` *Directed conditional granger causality*

Description

Compute the difference conditional granger causality of multivariate timeseries.

Usage

```
diffGranger(data, nx = 1, ny = 1, order=1, perm = FALSE, bs = 100)
```

Arguments

`data`	object containing all observations (rows) and variables (columns) that are being considered. The variables should be ordered as follows: First the variables that are supposed to granger cause a set of other variables (>=1). Then the set of variables (>=1) that are Granger caused by the first set of variables. Finally, a set of variables to condition on(>=1).
`nx`	The number of variables (>=1) that Granger cause a set of other variables (default = 1), conditioned on a third set of variables (>=1).
`ny`	The number of variables (>=1) that are Granger caused by the first nx variables (default = 1), conditioned on a third set of variables (>=1).
`order`	Autoregressive order (>=1) of timeseries. Can be computed using ARorder().
`perm`	Logical. If perm = FALSE (default), only the Granger causality measure is produced. If perm = TRUE, the Granger test is computed and a permutation test is performed to do inference.
`bs`	Number of permutation samples. Only works when perm = TRUE. Default=100

Details

The total linear dependence between X and Y can be divided in three components: a directed influence from X to Y, a directed influence from Y to X and an undirected instantaneous influence between them. The difference granger causality from X to Y computed in the function diff.granger() subtracts the conditional granger causality from Y to X from the conditional granger causality from X to Y. This can be used as a measure of how much stronger (weaker) one directed influence is compared to the opposite directed influence.

Value

Partial Granger causality measure F1 plus p -value (Only when perm=TRUE).

Author(s)

Bjorn Roelstraete

References

Roebroeck, A., Formisano, E., Goebel, R. (2005). Mapping directed influence over the brain using Granger causality. NeuroImage 230-242.

See Also

condGranger, pdiffGranger

Examples

```
# Example data with 5 regions x, y, z, q, w
head(grangerdata)

# Calculate AR() order of the data
ARorder(grangerdata, max=10)

# Compute difference conditional granger causality of region x to regions y o
# and z, conditional on regions q and w
F <- diffGranger(grangerdata, nx=1, ny=2, order=3)

# Compute F and bootstrap H0 distribution
F <- diffGranger(grangerdata, nx=1, ny=2, order=3, perm=TRUE, bs=50)
```

---

grangerdata *Demo dataset*

Description

A toy dataset containing 5 autoregressive timeseries generated from the model by Baccala and Sameshima (Biol. Cybern. 2001).

Usage

```
data(grangerdata)
```

Format

A data frame of 2000 observations of 3 variables.

- `x` Time series at region 1
- `y` Time series at region 2
- `z` Time series at region 3
- `q` Time series at region 4
- `w` Time series at region 5

Examples

```
head(grangerdata)
```

---

hrfConvolve *Convolution function*

Description

Convolute a timeseries with any double gamma function.

Usage

```
hrfConvolve(x = NULL, scans = NA, onsets = c(), durations = c(),
             rt = NA, SNR = 0, mean = FALSE, a1 = 6, a2 = 12, b1 = 0.9,
             b2 = 0.9, cc = 0.35)
```

Arguments

`x`	Single timeseries (default = NULL)
`scans`	number of scans
`onsets`	onsets of experimental condition
`durations`	duration of experimental condition
`rt`	repetition time
`SNR`	signal to noise ratio of data
`mean`	logical if mean is TRUE the timeseries is centered around 0.
`a1`	parameter of the double gamma function
`a2`	parameter of the double gamma function
`b1`	parameter of the double gamma function
`b2`	parameter of the double gamma function
`cc`	parameter of the double gamma function

Details

The function is an extension of the fmri.stimulus function in the 'fmri' pacakge (see ref.). If x = NULL, the to be convolved stimulusfunction can be specified with the parameters 'scans', 'onsets', 'durations', 'rt', and 'SNR'. If x is entered, the timeseries x is convolved and the other parameters need not be specified. The default convolution function is the canonical HRF, but can be altered by changing the parameters of the double gamma function.

Value

returns convolved timeseries. The timeseries is convolved with a mixture of 2 gamma functions (default = canonical HRF).

Author(s)

Bjorn Roelstraete

References

Polzehl, J. and Tabelow, K. (2007) _fmri: A Package for Analyzing fmri Data_, R News, 7:13-17 .

Examples

```
# Specify a stimulusfunction without noise and convolve with canonical HRF

hrfConvolve(scans = 240, onsets = c(0,60,120,180), durations = c(30),
             rt = 3, SNR = 0)

# Convolve a (part of a) timeseries with a canonical HRF.

hrfConvolve(x=grangerdata[1:100,1])
plot(hrfConvolve(grangerdata[1:100,1]))

# Compare the convolved timeseries with the raw
par(mfrow=c(2,1))
plot(x=semdata[1:100,1])
plot(hrfConvolve(x=semdata[1:100,1]))
```

---

partGranger *Partial Granger causality*

Description

Compute partial Granger causality of multivariate timeseries.

Usage

```
partGranger(data, nx = 1, ny = 1, order=1, perm = FALSE, prob=TRUE, bs = 100)
```

Arguments

data	object containing all observations (rows) and variables (columns) that are being considered. The variables should be ordered as follows: First the variables that are supposed to granger cause a set of other variables (>=1). Then the set of variables (>=1) that are Granger caused by the first set of variables. Finally, a set of variables to condition on(>=1).
nx	The number of variables (>=1) that Granger cause a set of other variables (default = 1), conditioned on a third set of variables (>=1).
ny	The number of variables (>=1) that are Granger caused by the first nx variables (default = 1), conditioned on a third set of variables (>=1).
order	Autoregressive order (>=1) of timeseries. Can be computed using ARorder().
perm	Logical. If perm = FALSE (default), only the Granger causality measure is produced. If perm = TRUE, the Granger test is computed and a permutation test is performed to do inference.
prob	Logical. If TRUE, the F statistic is returned together with the p-value.
bs	Number of permutation samples. Only works when perm = TRUE. Default=100

Value

Partial Granger causality measure F1 plus p -value.

Author(s)

Bjorn Roelstraete

References

Guo, S., Seth, A.K., Kendrick, K.M., Zhou, C., Feng, J.(2008). Partial Granger Causality-Eliminating Exogenous Inputs and Latent Variables. Journal of Neuroscience Methods. 79-93.

Examples

```
# Example data with 5 regions x, y, z, q, w
head(grangerdata)

# Calculate AR() order of the data
ARorder(grangerdata, max=10)

# Compute partial conditional granger causality of region x to regions y
# and z, conditional on regions q and w
F <- partGranger(grangerdata, nx=1, ny=2, order=3)

# Compute F and permutation H0 distribution
F <- partGranger(grangerdata, nx=1, ny=2, order=3, perm=TRUE,bs=10)
```

pdiffGranger *partial difference Granger causality*

Description

Compute partial difference conditional Granger causality of multivariate timeseries.

Usage

```
pdiffGranger(data, nx = 1, ny = 1, order=1, perm = FALSE, bs = 100)
```

Arguments

data	object containing all observations (rows) and variables (columns) that are being considered. The variables should be ordered as follows: First the variables that are supposed to Granger cause a set of other variables (>=1). Then the set of variables (>=1) that are Granger caused by the first set of variables. Finally, a set of variables to condition on(>=1).
nx	The number of variables (>=1) that are supposed to Granger cause a set of other variables (default = 1), conditioned on a third set of variables (>=1).
ny	The number of variables (>=1) that are supposed to be Granger caused by the first nx variables (default = 1), conditioned on a third set of variables (>=1).
order	Autoregressive order (>=1) of timeseries. Can be computed using ARorder().
perm	Logical. If perm = FALSE (default), only the Granger causality measure is produced. If perm = TRUE, the Granger test is computed and a permutation is performed to generate the H0 distribution.
bs	Number of permutation samples. Default=100

Details

The total linear dependence between X and Y can be divided in three components: a directed influence from X to Y, a directed influence from Y to X and an undirected instantaneous influence between them. The difference Granger causality from X to Y computed in the function diff.Granger() subtracts the partial conditional Granger causality from Y to X from the partial conditional Granger causality from X to Y. This can be used as a measure of how much stronger (weaker) one directed influence is compared to the opposite directed influence.

Value

Partial difference Granger causality measure and p value.

Author(s)

Bjorn Roelstraete

See Also

diffGranger

Examples

```
# Example data with 5 regions x, y, z, q, w
head(grangerdata)

# Calculate AR() order of the data
ARorder(grangerdata, max=10)

# Compute partial difference conditional Granger causality of region x to
# regions y and z, conditional on regions q and w
F <- pdiffGranger(grangerdata, nx=1, ny=2, order=3)

# Compute F and permutation H0 distribution
F <- pdiffGranger(grangerdata, nx=1, ny=2, order=3, perm=TRUE, bs=50)
```

---

semdata *Demo dataset*

Description

A toy dataset containing 3 autoregressive timeseries generated from the model by Baccala and Sameshima (Biol. Cybern. 2001).

Usage

```
data(semdata)
```

Format

A data frame of 2000 observations of 3 variables.

- `x` Time series at region 1
- `y` Time series at region 2
- `z` Time series at region 3

Examples

```
head(semdata)
```

SEMextract *Preparing fMRI time series for SEM analysis*

Description

Preparing fMRI time series for SEM analysis.

Usage

```
SEMextract(ts, ons, dur, TR)
```

Arguments

ts	time series for analysis
ons	Onsets of experimental condition of interest.
dur	Duration of experimental condition of interest.
TR	Repetition time of time series.

Author(s)

Bjorn Roelstraete

X0 *Demo dataset*

Description

Filtered and whitened design matrix from 'Attention to visual motion' dataset).

Usage

```
data(X0)
```

Examples

```
head(X0)
```

Index

∗Topic **\textasciitildeDCM**
  dcmCompare, 7
  dcmEstimate, 7
  dcmEvidence, 8
  dcmGenerate, 11
  dcmParam, 12
∗Topic **\textasciitildeGranger**
  condGranger, 5
  diffGranger, 14
  partGranger, 17
  pdiffGranger, 19
∗Topic **\textasciitildeSEM**
  ARsem, 3
  SEMextract, 21
∗Topic **\textasciitildeauto-regressive**
  ARsem, 3
∗Topic **\textasciitildeconditional**
  condGranger, 5
  diffGranger, 14
∗Topic **\textasciitildehrf**
  hrfConvolve, 16
∗Topic **\textasciitildeparial**
  partGranger, 17
∗Topic **\textasciitildepartial**
  pdiffGranger, 19
∗Topic **\textasciitildetimeseries**
  ARorder, 2

ARorder, 2
ARsem, 3
attentiondata, 4

condGranger, 5

DCMatt, 6
dcmCompare, 7
dcmEstimate, 7
dcmEvidence, 8
DCMex, 9
dcmGenerate, 11
dcmParam, 12
diffGranger, 14

grangerdata, 15

hrfConvolve, 16

partGranger, 17
pdiffGranger, 19

semdata, 20
SEMextract, 21

X0, 21