Package 'GNAR'

November 2, 2018

Type Package

Title Methods for Fitting Network Time Series Models

Version 0.2.9

Date 2018-11-01

Author Kathryn Leeming [aut],
Guy Nason [aut],
Matt Nunes [aut, cre],
Marina Knight [ctb]

Maintainer Matt Nunes <nunesrpackages@gmail.com>

Description Simulation of, and fitting models for, Generalised Network Autoregressive (GNAR) time series models which take account of network structure. Such models are described in Knight et al. (2016), see <arXiv:1603.03221>.

Depends igraph, wordcloud

License GPL-2

Encoding UTF-8

LazyData true

NeedsCompilation no

Repository CRAN

Date/Publication 2018-11-02 05:20:12 UTC

R topics documented:

`coef.GNARfit` *Function to return coefficients of GNARfit objects*

Description

`coef.GNARfit` returns the vector of coefficients from a GNARfit object.

Usage

```
## S3 method for class 'GNARfit'
coef(object,...)
```

Arguments

`object`	the output of a GNARfit or GNARpredict call
`...`	additional arguments, unused here.

Details

S3 method for class "GNARfit".

Value

coef.GNARfit returns a vector of coefficient values.

Examples

```
#get the coefficients of the fiveNode data GNAR fit
data(fiveNode)
coef(GNARfit())
```

---

fitted.GNARfit *Function to return fitted values of GNARfit objects*

Description

fitted.GNARfit returns the fitted values of a GNARfit object as a matrix.

Usage

```
## S3 method for class 'GNARfit'
fitted(object,...)
```

Arguments

object	the output of a GNARfit or GNARpredict call
...	additional arguments, unused here.

Details

S3 method for class "GNARfit".

Value

fitted.GNARfit returns a ts object of fitted values, with t-alphaOrder rows and nnodes columns.

Examples

```
#get the fitted values of the fiveNode GNAR fit
data(fiveNode)
fitted(GNARfit())
```

fiveNet *Example network*

Description

This example GNARnet object is a network containing five vertices connected by five undirected edges, each of length 1.

Usage

```
fiveNet
```

Format

`fiveNet` is a GNARnet object containing `$edges` and `$dist`.
`edges` is a list of length five, with `edges[[i]]` containing the vertices that node i is connected to.
`dist` is a list of length five, with `dist[[i]]` containing the length of the vertices that node i is connected to.

Examples

```
#Plot this network
data(fiveNode)
plot(fiveNet)
```

fiveNode *Example Network Time Series*

Description

A multivariate time series fiveVTS and corresponding network fiveNet.

Usage

```
data("fiveNode")
```

Format

This dataset contains two R objects:
`fiveVTS` is a ts object with a matrix of 200 rows (t=200) and 5 columns (n=5) `fiveNet` is a GNARnet object containing `$edges` and `$dist`.
`edges` is a list of length five, with `edges[[i]]` containing the vertices that node i is connected to.
`dist` is a list of length five, with `dist[[i]]` containing the length of the vertices that node i is connected to.

Examples

```
data(fiveNode)
plot(fiveNet)
image(fiveVTS)
```

`fiveVTS` *Simulated GNAR data using fiveNet*

Description

This dataset is simulated from the GNAR(1,[1]) model, using fiveNet as the accompanying network. The parameters used were $\alpha_1 = 0.2$ for each node, and $\beta_1 = 0.5$. See GNARfit for the model description.

Usage

```
fiveVTS
```

Format

`fiveVTS` is a ts object with a matrix of 200 rows (t=200) and 5 columns (n=5)

Examples

```
data(fiveNode)
#Plot using ts S3 function
plot(fiveVTS)

#Plot as heatmap
image(fiveVTS)
```

`gdp` *GDP example data.*

Description

Data and seed values for GDP example.

Usage

```
data("gdp")
```

Format

This dataset contains two R objects:
`gdpVTS` is a ts object with a matrix of 52 rows (t=52) and 35 columns (n=35) `seed.nos` is a vector containing integer values.

Examples

```
data(gdp)
image(gdpVTS)
plot(seedToNet(seed.nos[1], nnodes=35, graph.prob=0.15))
```

---

gdpVTS *Differenced GDP values for 35 countries*

Description

This dataset is from the OECD (OECD (2018), Quarterly GDP (indicator). <doi:10.1787/b86d1fc8-en> (Accessed on 29 January 2018)) and is differenced annual growth rate for 35 countries.

Usage

```
gdpVTS
```

Format

gdpVTS is a ts object with a matrix of 52 rows (t=52) and 35 columns (n=35)

Examples

```
data(gdp)
#Plot using 'ts' S3 function, can only plot up to 10 columns at once
plot(gdpVTS[,1:5])

#Plot as heatmap
image(gdpVTS)
```

GNAR *GNAR package*

Description

A package to fit, predict, and simulate time series using the Generalised Network AutoRegressive (GNAR) model. The main functions are GNARfit, which fits the model to a time series and network(s), GNARpredict which fits the GNAR model and produces a within-sample prediction of the last time point, and GNARsim which simulates from a GNAR model with specified parameters. For details of the model, see GNARfit. The package also contains an example network time series in data file fiveNode, with network fiveNet, and simulated time series fiveVTS.

References

Knight, M.I., Nunes, M.A. and Nason, G.P. Modelling, detrending and decorrelation of network time series. arXiv preprint.

GNARdesign *Function to create the GNAR design matrix*

Description

Creates the design matrix necessary for fitting the GNAR model.

Usage

```
GNARdesign(vts = GNAR::fiveVTS, net = GNAR::fiveNet, alphaOrder = 2, betaOrder = c(1,1),
 fact.var = NULL, globalalpha=TRUE, tvnets=NULL, netsstart=NULL)
```

Arguments

vts	a matrix or ts object containing the multivariate time series to be modelled. The i,j entry of this matrix should be for time i and vertex/node j.
net	the (first) network associated with the time series, containing a list with entries $edges and $dist. This network should have the same number of nodes as the number of columns of the vts matrix.
alphaOrder	a non-negative integer specifying the maximum time-lag to model.
betaOrder	a vector of length alphaOrder specifying the maximum neighbour set to model at each of the time-lags.
fact.var	a vector of factors indicating which nodes belong to each set with different parameters to be fitted.
globalalpha	a TRUE/FALSE value indivating whether to use global alpha parameters.
tvnets	a list of additional networks. Currently only NULL (the static network case) is supported.
netsstart	a vector of times corresponding to the first time points for each network of tvnets. Currently only NULL (the static network case) is supported.

Value

GNARdesign	returns a matrix containing (t-alphaOrder)nnodes rows and a column for each parameter to be fitted. The columns are in time-lag order, eg for GNAR(2,[1,0]) the columns are alpha1, beta1.1, alpha2. When a factor variable is specified the columns are labelled with the factor.

Examples

```
#Design matrix to fit GNAR(2,[1,1]) to the fiveVTS data
data(fiveNode)
GNARdesign()
```

GNARfit *Fitting function for GNAR models*

Description

Fits the GNAR model to the given inputs using `GNARdesign` and `lm`.

Usage

```
GNARfit(vts=GNAR::fiveVTS, net=GNAR::fiveNet, alphaOrder=2, betaOrder=c(1,1),
 fact.var=NULL, globalalpha=TRUE, tvnets=NULL, netsstart=NULL, ErrorIfNoNei=TRUE)
```

Arguments

`vts`	a matrix containing the multivariate time series to be modelled. The `i,j` entry of this matrix should be for time `i` and vertex/node `j`.
`net`	the (first) network associated with the time series, containing a list with entries `$edges` and `$dist`. This network should have the same number of nodes as the number of columns of the `vts` matrix.
`alphaOrder`	a non-negative integer specifying the maximum time-lag to model.
`betaOrder`	a vector of length `alphaOrder` specifying the maximum neighbour set to model at each of the time-lags.
`fact.var`	a vector of factors indicating which nodes belong to different sets with different parameters to be fitted.
`globalalpha`	a TRUE/FALSE value indivating whether to use global alpha parameters.
`tvnets`	a list of additional networks. Currently only NULL (the static network case) is supported.
`netsstart`	a vector of times corresponding to the first time points for each network of `tvnets`. Currently only NULL (the static network case) is supported.
`ErrorIfNoNei`	a TRUE/FALSE value indicating whether to stop the function call with an error if betaOrder specifies more neighbour sets than exist in the network. If FALSE the function will continue and some parameters will be NA.

Details

The GNAR model of order (p, S) is defined as

$$X_{i,t} = \sum_{j=1}^{p} \left(\alpha_{i,j} X_{i,t-j} + \sum_{c=1}^{C} \sum_{r=1}^{S_j} \beta_{j,r,c} \sum_{q \in N_t^{(r)}(i)} \omega_{i,q,c}^{(t)} X_{q,t-j} \right) + u_{i,t}$$

where p is the maximum time lag, $S = (S_1, ..., S_p)$ and S_j is the maximum stage of neighbour dependence for time lag j, $N_t^{(r)}(i)$ is the rth stage neighbour set of node i at time t, $\omega_{i,q,c}^{(t)}$ is the connection weight between node i and node q at time t if the path corresponds to covariate c. Here,

we consider a sum from one to zero to be zero and $\{u_{i,t}\}$, are assumed to be independent and identically distributed at each node i, with mean zero and variance σ_i^2. Currently, only a single network GNAR model can be fitted. The connection weight, $\omega_{i,q,c}^{(t)}$, is the inverse of the distance between nodes `i` and `q`, normalised so that they sum to 1 for each `i,t`. See is.GNARnet for `GNARnet` object information and example construction.

Value

`mod`	the `lm` output from fitting the GNAR model.
`y`	the original response values, with NAs left in.
`dd`	the output of `GNARdesign` containing the design matrix, with NAs left in.
`frbic`	inputs to the `GNARfit` function.

References

Knight, M.I., Nunes, M.A. and Nason, G.P. Modelling, detrending and decorrelation of network time series. arXiv preprint.

Examples

```
#Fit the GNAR(2,[1,1]) model to the fiveVTS data
data(fiveNode)
GNARfit()

#Convert the residuals to matrix form
residToMat(GNARfit())$resid
```

`GNARpredict` *Fits and predicts using the GNAR model*

Description

Fits the GNAR model to up to observation `t-1` and produces a within-sample prediction for time `t`.

Usage

```
GNARpredict(vts=GNAR::fiveVTS, net=GNAR::fiveNet, alphaOrder=2, betaOrder=c(1,1),
 fact.var=NULL, globalalpha=TRUE, tvnets=NULL, netsstart=NULL, ErrorIfNoNei=TRUE)
```

Arguments

`vts`	a matrix containing the multivariate time series to be modelled. The `i,j` entry of this matrix should be for time `i` and vertex/node `j`.
`net`	the (first) network associated with the time series, containing a list with entries `$edges` and `$dist`. This network should have the same number of nodes as the number of columns of the `vts` matrix.
`alphaOrder`	a non-negative integer specifying the maximum time-lag to model.

betaOrder	a vector of length alphaOrder specifying the maximum neighbour set to model at each of the time-lags.
fact.var	a vector of factors indicating which nodes belong to differents set with different parameters to be fitted.
globalalpha	a TRUE/FALSE value indivating whether to use global alpha parameters.
tvnets	a list of additional networks. Currently only NULL (the static network case) is supported.
netsstart	a vector of times corresponding to the first time points for each network of tvnets. Currently only NULL (the static network case) is supported.
ErrorIfNoNei	a TRUE/FALSE value indicating whether to stop the function call with an error if betaOrder specifies more neighbour sets than exist in the network. If FALSE the function will continue and some parameters will be NA.

Details

See GNARfit for GNAR model information. Note that the prediction is for the final time observation of the input data, to predict out-of-sample add a row of zeros to the input data matrix. Only coefficients with p-value smaller than 0.05 are used to calculate prediction. See is.GNARnet for GNARnet object information and example construction.

Value

pred	the prediction for time t.
mod	the lm output from fitting the GNAR model up to t-1.
ys	the original response values up to t-1, with NAs left in.
ds	the output of GNARdesign containing the design matrix up to t-1, with NAs left in.
ypred	the original response values at t.
dpred	the time t entries of the design matrix.
frbic	inputs to the GNARpredict function.

Examples

```
#Fit and predict the fiveVTS data with the GNAR(2,[1,1]) model
data(fiveNode)
GNARpredict()
```

GNARsim *Simulates a GNAR process*

Description

Simulates a GNAR process with Normally distributed innovations.

Usage

```
GNARsim(n=200, net=GNAR::fiveNet, alphaParams=list(c(rep(0.2,5))),
 betaParams=list(c(0.5)), sigma=1, tvnets=NULL, netsstart=NULL)
```

Arguments

n	time length of simulation.
net	network used for the GNAR simulation.
alphaParams	a list containing vectors of auto-regression parameters for each time-lag.
betaParams	a list of equal length as alphaParams containing the network-regression parameters for each time-lag.
sigma	the standard deviation for the innovations.
tvnets	Only NULL is currently supported.
netsstart	Only NULL is currently supported.

Details

Parameter lists should not be NULL, set unused parameters to be zero. See GNARfit for model description.

Value

GNARsim returns the multivariate time series as a ts object, with n rows and a column for each of the nodes in the network.

References

Knight, M.I., Nunes, M.A. and Nason, G.P. Modelling, detrending and decorrelation of network time series. arXiv preprint.

Examples

```
#Simulate a GNAR(1,[1]) process with the fiveNet network
data(fiveNode)
GNARsim()
```

GNARtoigraph *Converts a GNAR network to a weighted igraph object*

Description

Takes an input network and neighbour stage and returns it in igraph form.

Usage

```
GNARtoigraph(net=GNAR::fiveNet, stage=1, normalise=FALSE)
```

Arguments

net	a GNARnet object containing $edges and dist.
stage	the neighbour set that the adjacency matrix is created for.
normalise	whether to normalise each to non-zero row to have sum one.

Details

With normalisation this is a non-invertible transform. See NofNeighbours for neighbour set definition. See is.GNARnet for GNARnet object information and example construction.

Value

GNARtoigraph returns an 'igraph' object with weights as the inverse distances of the input network.

Examples

```
#fiveNet as an igraph object
data(fiveNode)
GNARtoigraph()
```

GNARtoWMat *Converts a GNAR network to a weighted adjacency matrix*

Description

Takes an input GNARnet and neighbour stage and outputs the corresponding adjacency matrix.

Usage

```
GNARtoWMat(net=GNAR::fiveNet, stage=1, normalise=FALSE)
```

Arguments

net	a GNARnet object containing $edges and $dist.
stage	the neighbour set that the adjacency matrix is created for.
normalise	whether to normalise each to non-zero row to have sum one.

Details

With normalisation this is a non-invertible transform. See NofNeighbours for neighbour set definition.

Value

GNARtoWMat returns a square matrix with the number of rows and columns equal to the length of the $edges list. Entry i,j of the matrix will be non-zero if node j is in the stage neighbour set of i.

Examples

```
#fiveNet as an adjacency matrix
data(fiveNode)
GNARtoWMat()
```

---

igraphtoGNAR *Converts an igraph to GNAR network*

Description

Converts an 'igraph' to the GNARnet form for use as an input to GNAR functions.

Usage

```
igraphtoGNAR(ig)
```

Arguments

ig	an 'igraph' object.

Details

The values in the $dist list are the reciprocal of the values from the weighted adjacency matrix.

Value

igraphtoGNAR returns a GNARnet: a list with elements $edges and $dist.

Examples

```
#Convert fiveNet to igraph and back again
data(fiveNode)
igraphtoGNAR(GNARtoigraph(fiveNet))
```

`is.GNARfit` *Function to check GNARfit objects*

Description

`is.GNARfit` returns either TRUE or FALSE according to a series of GNARfit checks.

Usage

```
is.GNARfit(x)
```

Arguments

`x`	the object to be tested

Details

The `is.GNARfit` function checks whether the object passes a series of tests that correspond to it being the output of GNARfit or GNARpredict:

- Is it a list containing `$mod` and `$frbic`
- Does it contain either `$y` and `$dd` or `$ys` and `$ds`
- Is `$mod` a lm object
- Does `$frbic` have the components to calculate the BIC with lmToBIC

Value

`is.GNARfit` returns `TRUE` or `FALSE` corresponding to passing the above tests.

Examples

```
#check that the example fit meets the criteria above
data(fiveNode)
is.GNARfit(GNARfit())
#also check the predict function
is.GNARfit(GNARpredict())
```

`is.GNARnet` *Functions to check and create GNARnet objects*

Description

`is.GNARnet` returns either TRUE or FALSE according to a series of GNARnet checks. `as.GNARnet` returns a GNARnet object from an input weights matrix, 'igraph' object, or a GNARnet without assigned class.

Usage

```
is.GNARnet(x)
as.GNARnet(x)
```

Arguments

`x`	the network to be tested or object to be converted

Details

The `is.GNARnet` function checks whether the network passes a series of tests:

- Is it a list containing $edges and $dist
- Are the $edges and $dist lists of the same length
- Are each of the elements of $edges the same length as the corresponding $dist element
- Do the edges only contain valid entries, 1,...,nnodes
- Is it labelled as `GNARnet` class
- Are no duplicate edges present

The `as.GNARnet` function converts igraph objects to GNARnet form, other possible inputs are adjacency matrices, and lists with `$edges` and `$dist` entries of the correct form.

Value

`is.GNARnet` returns `TRUE` or `FALSE` corresponding to passing the above tests. `as.GNARnet` returns a `GNARnet` object.

Examples

```
#check that the example network meets the criteria above
data(fiveNode)
is.GNARnet(fiveNet)

#convert to igraph and back again
as.GNARnet(GNARtoigraph(fiveNet))

#generate a new network with three nodes
```

```
#edges 1->2, 2->1, 2->3
#dist 1, 2, 1
#note 1->2 and 2->1 are of different lengths
threeEdge <- list(c(2), c(1,3), NULL)
threeDist <- list(c(1), c(2,1), NULL)
threeNet <- list(edges=threeEdge, dist=threeDist)
#check if this is a GNARnet
is.GNARnet(threeNet)
#use as.GNARnet to change the class
threeNet <- as.GNARnet(threeNet)
#check if this is a GNARnet now
is.GNARnet(threeNet)
```

`lmToBIC` *Calculates the BIC for a GNAR model fit*

Description

Returns the value of the BIC for a GNARfit or GNARpredict object.

Usage

```
lmToBIC(GNARobj.in=GNARpredict())
```

Arguments

`GNARobj.in`	output of `GNARfit` or `GNARpredict`.

Details

For a GNAR(p,[S]) model, the BIC is calculated as

$$BIC(p, S) = \ln |\Sigma_{p,S}| + T^{-1} M \ln(T)$$

where $\Sigma_{p,S} = T^{-1}U'U$, U is the matrix of residuals, and $M = np + C\sum_{j=1}^{p} S_j$ when `globalalpha=FALSE`, $M = p + C\sum_{j=1}^{p} S_j$ when `globalalpha=TRUE`, and `C` is the number of factors.

For definiteness, we seek to minimize the BIC. Often, better models have a small BIC.

Value

`lmToBIC` returns the value of the BIC for the input model fit.

Examples

```
#BIC for the GNAR(2,[1,1]) model using GNARpredict on fiveVTS
data(fiveNode)
lmToBIC()
```

matrixtoGNAR *Converts an adjacency matrix to GNAR network*

Description

Converts an adjacency matrix to the GNARnet form for use as an input to GNAR functions.

Usage

```
matrixtoGNAR(input.mat)
```

Arguments

input.mat	an adjacency matrix whose dimension equals the number of nodes in the resulting network.

Details

The values in the $dist list are the reciprocal of the values from the weighted adjacency matrix.

Value

matrixtoGNAR returns a GNARnet list with elements $edges and $dist.

Examples

```
#Convert fiveNet to an adjacency matrix and back again
data(fiveNode)
matrixtoGNAR(GNARtoWMat())
```

na.row *Identifies which rows of a matrix have NAs*

Description

Returns a vector with elements TRUE/FALSE identifying which rows contain NA elements.

Usage

```
na.row(mat)
```

Arguments

mat	a matrix object.

Details

This function is used in the unstacking of residuals into a residual matrix and replacing NAs where they were previously present.

Value

`na.row` returns a vector of length equal to the number of rows in `mat`. Each element is either TRUE or FALSE.

Examples

```
#Check if there are and NAs in fiveVTS
data(fiveNode)
na.row(fiveVTS)
```

NofNeighbours *Calculates stage-neighbours of a network*

Description

Calculates neighbour sets of a particular node in the network and their distances.

Usage

```
NofNeighbours(node=1, stage=2, net=GNAR::fiveNet)
```

Arguments

`node`	is an integer specifying which node to calculate the neighbours of.
`stage`	is an integer specifying the maximum neighbour-stage to calculate to.
`net`	a `GNARnet` object with edge list and distance list.

Details

Note that the distances are calculated as the sum along the shortest path; do not use this with a weights (rather than distance) list. Stage-`r` neighbours of node `i` are denoted $N^(r)(i)$, and are nodes that are `r` edges (but no fewer) away from `i`. Hence stage-1 neighbours are the immediate neighbours, stage-2 neighbours are the neighbours of neighbours and so on.

Value

`edges`	is a list of length `stage`, where `edges[[i]]` is a vector containing the nodes that are stage-`i` neighbours of `node`.
`dist`	is a list of length `stage`, where `dist[[i]]` is a vector containing the distances from `node` to its stage-`i` neighbours, with ordering as in `edges[[i]]`.

Examples

```
#First and second stage neighbours of node 1 in fiveNet
data(fiveNode)
NofNeighbours()
```

`plot.GNARnet` *Plot function for GNAR networks*

Description

Plots a GNAR network using the 'igraph' package.

Usage

```
## S3 method for class 'GNARnet'
plot(x, ...)
```

Arguments

`x`	the network`GNARnet` object associated with the time series, containing a list with entries `$edges` and `$dist`.
`...`	additional arguments for the `igraph` plotting function, see plot.igraph.

Details

S3 method for class "GNARnet".

Examples

```
#Plot fiveNet
data(fiveNode)
plot.GNARnet(fiveNet)
```

`print.GNARfit` *Function to print the model and coefficients of GNARfit objects*

Description

`print.GNARfit` prints model, call, and coefficients of a GNARfit object.

Usage

```
## S3 method for class 'GNARfit'
print(x,...)
```

Arguments

x	the output of a GNARfit or GNARpredict call
...	additional arguments, unused here.

Details

S3 method for class "GNARfit".

Examples

```
#print the information of the fiveNode GNAR fit
data(fiveNode)
print(GNARfit())
```

---

print.GNARnet *Print function for GNAR networks*

Description

Prints information about a GNAR network.

Usage

```
## S3 method for class 'GNARnet'
print(x, ...)
```

Arguments

x	the networkGNARnet object associated with the time series, containing a list with entries $edges and $dist.
...	additional arguments, unused here.

Details

S3 method for class "GNARnet".

Examples

```
#print fiveNet information
data(fiveNode)
print.GNARnet(fiveNet)
```

residToMat — *Converts the output of a GNARfit or GNARpredict to fitted and residual value matrices*

Description

Unstacks the entries of the GNARfit or GNARpredict fitted and residual values to return matrices of a similar form to the multivariate time series input.

Usage

```
residToMat(GNARobj=GNARpredict(), nnodes=5)
```

Arguments

GNARobj	the output from the GNARfit or GNARpredict function
nnodes	the number of nodes in the original network time series

Details

This function also replaces the NAs that were removed in fitting.

Value

resid	is the matrix of residual values, with t-alphaOrder rows and nnodes columns.
fit	is the matrix of fitted values, with t-alphaOrder rows and nnodes columns.

Examples

```
#Get residual and fitted matrices from GNARpredict fit of fiveVTS
data(fiveNode)
residToMat()
```

residuals.GNARfit — *Function to return residuals of GNARfit objects*

Description

residuals.GNARfit returns the residuals of a GNARfit object as a matrix.

Usage

```
## S3 method for class 'GNARfit'
residuals(object,...)
```

Arguments

object	the output of a GNARfit or GNARpredict call
...	additional arguments, unused here.

Details

The function first checks if the object is of GNARfit class, then uses residToMat to return the residuals.

Value

residuals.GNARfit returns a 'ts' object of residuals, with t-alphaOrder rows and nnodes columns.

Examples

```
#get the residuals of the fiveNode GNAR fit
data(fiveNode)
residuals(GNARfit())
```

---

seed.nos *Vector of seed numbers*

Description

Seed numbers for reproducible random graphs.

Usage

```
seed.nos
```

Format

seed.nos is a vector of length 10,000 containing integers.

Examples

```
data(gdp)
g <- seedToNet(seed.nos[1], nnodes=35, graph.prob=0.15)
plot(g, vertex.label=colnames(gdpVTS), vertex.size=0)
```

seedToNet *Produces a random network from a seed value*

Description

Produces a reproducible undirected Erdos-Reyni random network using a particular seed value.

Usage

```
seedToNet(seed.no, nnodes=34, graph.prob=0.5)
```

Arguments

seed.no	a valid number to set the seed to.
nnodes	the number of nodes in the produced network.
graph.prob	the probability that each pair of nodes is connected.

Details

graph.prob effectively controls the sparsity of the network. All distances are set to 1.

Value

A GNARnet object.

Examples

```
#Generate the random graph from seed 10, with 5 nodes and connection prob 0.5
seedToNet(10,nnodes=5,graph.prob=0.5)
```

summary.GNARfit *Returns model summary for a GNAR model fit*

Description

Returns the summary of a GNARfit or GNARpredict object, including BIC.

Usage

```
## S3 method for class 'GNARfit'
summary(object, ...)
```

Arguments

object	output of GNARfit or GNARpredict.
...	additional arguments affecting the summary produced.

Details

The output is the summary of the fit using summary.lm, and BIC calculated using lmToBIC.

Value

`summary.GNARfit` prints the model summary and the value of the BIC.

Examples

```
#summary for the GNAR(2,[1,1]) model using GNARpredict on fiveVTS
data(fiveNode)
summary(GNARpredict())
```

`summary.GNARnet` *Summary function for GNAR networks*

Description

Prints brief information about a GNAR network.

Usage

```
## S3 method for class 'GNARnet'
summary(object, ...)
```

Arguments

`object`	the network`GNARnet` object associated with the time series, containing a list with entries `$edges` and `$dist`.
`...`	additional arguments, unused here.

Details

S3 method for class "GNARnet".

Examples

```
#print fiveNet summary information
data(fiveNode)
summary.GNARnet(fiveNet)
```

vswind *Wind Speed example network time series*

Description

A suite of data objects concerning wind speed analysis. The dataset contains a multivariate time series of wind speeds, two network descriptions, a vector of names for weather stations, and the coordinates of the weather stations.

Usage

```
data("vswind")
```

Format

This dataset contains six R objects:
`vswindts` is a ts object with a matrix of 721 rows (t=721) and 102 columns (n=102). This corresponds to 721 observations made through time at 102 weather stations. `vswindnetD` is a `GNARnet` object containing `$edges` and `$dist`.
`edges` is a list of length 102, with `edges[[i]]` containing the vertices that node i is connected to.
`dist` is a list of length 102, with `dist[[i]]` containing the length of the vertices that node i is connected to. `vswindnet` is the same as `vswindnetD` except that all the distances are replaced by 1. `vswindnames` is a character vector of length 102 containing the wind speed site names and `vswindcoords` is a matrix with 102 rows (one for each wind station) and two columns providing the x and y coordinates of the weather stations.

Source

The base data were obtained from the http://wow.metoffice.gov.uk UK Met Office Weather-ObservationsWebsite distributed under the UK Open Government License http://www.nationalarchives.gov.uk/doc/open-government-licence/version/1/ Contains public sector information licensed under the Open Goverment Licence v1.0.

See Also

windnetplot

Examples

```
data(vswind)
#
# The name entry for Bristol
#
vswindnames[77]
#[1] "BRIST"
#
# plot the distance network
#
## Not run: windnetplot()
```

windnetplot *Produce bespoke plot of the wind data network*

Description

Plots the wind speed data network with distance information.

Usage

```
windnetplot()
```

Arguments

None.

Details

The wind speed data is to be found in the vswind data set. This function contains commands, using functionality from the wordcloud package, to plot the network, with node names and edges. Distances between nodes are plotted next to the edges.

See Also

vswind

Examples

```
data(vswind)
## Not run: windplotnet()
```

Index