

Package 'RSiena'

August 19, 2009

Type Package

Title Siena - Simulation Investigation for Empirical Network Analysis

Version 1.0.5

Date 2009-08-10

Author Various

Depends R (>= 2.7.0)

Imports Matrix

Suggests tcltk, snow, rlecuyer, network

SystemRequirements GNU make, tcl/tk 8.5, Tktable

Maintainer <ruth@stats.ox.ac.uk>

Description Fits models to longitudinal networks

License GPL (>= 2)

LazyLoad yes

LazyData yes

URL <http://www.stats.ox.ac.uk/~snijders/siena>

Repository CRAN

Repository/R-Forge/Project rsiena

Repository/R-Forge/Revision 11

Date/Publication 2009-08-19 08:14:31

R topics documented:

RSiena-package	2
coCovar	3
coDyadCovar	4
getEffects	5
installGui	7
model.create	7
print01Report	9
s501	10
s502	11
s503	11
s50a	12
siena01Gui	12
siena07	13
sienaCompositionChange	15
sienaDataCreate	16
sienaDataCreateFromSession	18
sienaGroupCreate	20
sienaModelOptions	22
sienaNet	24
sienaNodeSet	25
simstats0c	26
tmp3	27
tmp4	28
varCovar	28
varDyadCovar	29
Index	31

RSiena-package

*Simulation Investigation for Empirical Network Analysis***Description**

Fits models to longitudinal sets of networks.

Details

Use [siena07](#) to fit a model.

Data objects can be created via the Gui displayed by [siena01Gui](#), via [sienaDataCreateFromSession](#) or directly from matrices and vectors using [sienaNet](#), [coCovar](#) etc. and finally [sienaDataCreate](#).

Effects are selected using an *effects* object, which can be created using [getEffects](#).

Control of the fitting requires a *model* object, which can be created by [model.create](#).

More detailed help is available in the manual which you can display using `RShowDoc("s_man400", package="RSiena")`

Package: RSiena
Type: Package
Version: 1.0.5
Date: 2009-08-10
License: GPL-2
LazyLoad: yes

Author(s)

Who wrote it

Maintainer: Who to complain to <ruth@stats.ox.ac.uk>

References

See <http://www.stats.ox.ac.uk/~snijders/siena/>

See Also

[siena07](#)

Examples

```
myNet1 <- sienaNet(array(c(s501, s502, s503), dim=c(50, 50, 3)))  
mydata <- sienaDataCreate(myNet1)  
myeff <- getEffects(mydata)  
myModel <- model.create(findiff=FALSE, fn=simstats0c)  
ans <- siena07(myModel, data=mydata, effects=myeff, batch=TRUE)
```

coCovar

Function to create a constant covariate object

Description

This function creates a constant covariate object from a vector.

Usage

```
coCovar(val, nodeSet='Actors')
```

Arguments

val	Vector of covariate values
nodeSet	Name of node set: character string

Details

When part of a Siena data object, the covariate is associated with the node set `nodeSet` of the Siena data object.

Value

Returns the covariate as an object of class "coCovar", in which form it can be used as an argument to SienaData.create.

Author(s)

Ruth Ripley

References

See <http://www.stats.ox.ac.uk/~snijders/siena/>

See Also

[sienaDataCreate](#)

Examples

```
myconstCovar <- coCovar(s50a[,1])
```

coDyadCovar

Function to create a constant dyadic covariate object.

Description

This function creates a constant dyadic covariate object from a matrix.

Usage

```
coDyadCovar(val, nodeSets=c("Actors", "Actors"))
```

Arguments

`val` Matrix of covariate values.
`nodeSets` The name of the node sets with which this covariate is associated.

Details

When part of a Siena data object, the covariate is assumed to be associated with the node sets named in `nodeSets` of the Siena data object. The name of the associated node sets will only be checked when the Siena data object is created.

Value

Returns the covariate as an object of class "coDyadCovar", in which form it can be used as an argument to SienaData.create.

Author(s)

Ruth Ripley

ReferencesSee <http://www.stats.ox.ac.uk/~sniijders/siena/>**See Also**[sienaDataCreate](#)**Examples**

```
mydyadvar <- coDyadCovar(s503)
```

`getEffects`*Function to create a Siena effects object*

Description

Creates a basic list of effects for each of the dependent variables in the input `siena` object.

Usage

```
getEffects(x, nintn = 10)
```

Arguments

`x` an object of class 'siena' or 'sienaGroup'
`nintn` Number of lines for user defined interactions. (maybe superfluous now!)

Details

Considers all the elements of the input `siena` data object and creates lists of effects for use in `siena` model fits. Note that the class of the return object may be lost if the data.frame is edited using `fix`.

Value

An object of class "effects" or "groupEffects": this is a data frame, each part of which relates to one dependent variable in the input object, with columns

<code>name</code>	name of the dependent variable
<code>effectName</code>	name of the effect
<code>functionName</code>	name of the function
<code>shortName</code>	short name for the effect

interaction1 second variable in interaction, if any
 interaction2 third variable in interaction, if any
 type "eval", "endow", or "rate"
 basicRate boolean: whether a basic rate parameter
 include boolean: include or not
 randomEffects
 boolean: random or fixed effect
 fix boolean: fix value or not
 test boolean: test required or not
 initialValue starting value for estimation
 parm parameter values
 functionType "objective", "rate"
 period period for basic rate parameters
 rateType "Structural", "covariate"
 effectFn here NULL, but could be replaced by a function later
 statisticFn here NULL, but could be replaced by a function later
 untrimmedValue
 Used to store initial values which could be trimmed
 netType "oneMode", "Behavior", "Bipartite"
 groupName name of relevant data object
 group sequential number of relevant data object in total

Author(s)

Ruth Ripley

References

See <http://www.stats.ox.ac.uk/~snijders/siena/>

See Also

[sienaDataCreate](#)

Examples

```

mynet1 <- sienaNet(array(c(s501, s502, s503), dim=c(50, 50, 3)))
mynet2 <- sienaNet(s50a, type='behavior')
mycovar <- coCovar(rnorm(50))
mydyadcovar <- coDyadCovar(matrix(as.numeric(rnorm(2500) > 2), nrow=50))
mydata <- sienaDataCreate(mynet1, mynet2, mycovar, mydyadcovar)
myeff <- getEffects(mydata)

```

installGui	<i>Function to start up the installer for the standalone Gui.</i>
------------	---

Description

Starts the installer for the standalone version of RSiena. Only for Windows.

Usage

```
installGui()
```

Value

None.

Author(s)

Ruth Ripley

References

See <http://www.stats.ox.ac.uk/~sniijders/siena/>

Examples

```
## Not run: installGui()
```

model.create	<i>Function to create a model object for parameters to Siena07</i>
--------------	--

Description

Creates a basic object which can be used as an argument for Siena07

Usage

```
model.create(fn = simstats0c, usesimstats0c=TRUE,  
projname = "Siena", MaxDegree = 0, useStdInits = FALSE,  
n3 = 1000, nsub = 4, maxlike = FALSE, diag = TRUE,  
condvarno = 0, condname = "", firstg = 0.2, cond = FALSE,  
findiff = FALSE, seed = NULL)
```

Arguments

<code>fn</code>	function to do one simulation in the Robbins-Monro algorithm
<code>usesimstats0c</code>	Boolean. If true the standard algorithm is being used which can be used with multiple processors. Just used for validation.
<code>projname</code>	character string name of project. No embedded spaces
<code>MaxDegree</code>	Named vector of maximum degree values for corresponding networks
<code>useStdInits</code>	Boolean. If TRUE, the initial values in the effects object will be ignored and default values used instead.
<code>n3</code>	Number of iterations in phase 3.
<code>nsub</code>	Number of subphases in phase 2.
<code>maxlike</code>	whether to use maximum likelihood method or straightforward simulation
<code>diag</code>	Boolean: if FALSE, use the complete estimated derivative matrix in
<code>condvarno</code>	If conditional, the sequential number of the network or behavior variable on which to condition
<code>condname</code>	If conditional, the name of the dependent variable on which to condition. Use one or other of <code>condname</code> or <code>condvarno</code> to specify the variable.
<code>firstg</code>	initial value of scaling parameter in the Robbins-Monro procedure.
<code>cond</code>	Boolean. If TRUE, use conditional simulation
<code>findiff</code>	Boolean: If TRUE, estimate derivatives using finite differences. If FALSE use scores.
<code>seed</code>	Integer. Starting value of random seed. Not used if parallel testing.

Details

Model specification is done via this object for Siena7. This function creates an object with the correct elements. Those not available as arguments can be changed manually where desired.

Value

Returns a model object of class "sienaModel" containing:

<code>projname</code>	String value of name of project
<code>useStdInits</code>	Boolean, see above
<code>checktime</code>	Boolean, set to TRUE, report time in the phases or not
<code>n3</code>	number of iterations in Phase 3
<code>firstg</code>	Initial value of the scale in the Robbins-Munro algorithm
<code>maxrat</code>	value used to control the maximum size of the jumps
<code>maxmaxrat</code>	value used to control the maximum size of the jumps
<code>FRAN</code>	Simulation fn
<code>maxlike</code>	is FRAN using maximum likelihood?
<code>cconditional</code>	is FRAN using conditional estimation?

condvarno	number of dependent variable on which to condition
condname	name of dependent variable on which to condition
FinDiff.method	Boolean, should we calculate derivatives using finite differences?
nsub	Number of subphases in phase 2
diag	Boolean - use just the diagonal of the derivative matrix?
ModelType	always 1
MaxDegree	named vector of maximum degree values, or NULL

Author(s)

Ruth Ripley

ReferencesSee <http://www.stats.ox.ac.uk/~snijders/siena/>**Examples**

```
mymodel <- model.create(findiff=TRUE, fn = simstats0c)
```

print01Report *Function to produce the Siena01 report from R objects*

Description

Prints a report of a Siena data object and its default effects.

Usage

```
print01Report(data, myeff, modelname = "Siena", session = NULL)
```

Arguments

data	a Siena data object
myeff	a Siena Effects object
modelname	Character string used to name the output file "modelname.out"
session	Used to pass in a Siena01Gui() style session object so that data file names can be printed.

Details

First deletes any file of the name "modelname.out", then prints a new one.

Value

No value returned.

Author(s)

Ruth Ripley

References

See <http://www.stats.ox.ac.uk/~snijders/siena/>

See Also

[siena01Gui](#)

Examples

```
mymodel <- model.create(findiff=TRUE, fn=simstats0c)
mynet1 <- sienaNet(array(c(s501, s502, s503), dim=c(50, 50, 3)))
mydata <- sienaDataCreate(mynet1)
myeff <- getEffects(mydata)
## Not run: print01Report(mydata, myeff)
```

s501

Network 1 data: excerpt from 'Teenage Friends and Lifestyle Study' data.

Description

First timepoint network data from an excerpt of 50 girls from the Teenage Friends and Lifestyle Study data set. Useful as a small example of network and behaviour, for which models can be fitted quickly.

Format

The adjacency matrix for the network at time point 1.

Source

s50-network1.dat from Stocnet

References

See <http://www.stats.ox.ac.uk/~snijders/siena/>

s502 *Network 2 data: excerpt from 'Teenage Friends and Lifestyle Study' data.*

Description

Second timepoint network data from an excerpt of 50 girls from the Teenage Friends and Lifestyle Study data set. Useful as a small example of network and behaviour, for which models can be fitted quickly.

Format

Adjacency matrix for the network at time point 2.

Source

s50-network2.dat from Stocnet

References

See <http://www.stats.ox.ac.uk/~snijders/siena/>

s503 *Network 3 data: excerpt from 'Teenage Friends and Lifestyle Study' data.*

Description

Second timepoint network data from an excerpt of 50 girls from the Teenage Friends and Lifestyle Study data set. Useful as a small example of network and behaviour, for which models can be fitted quickly.

Format

Adjacency matrix for the network at time point 3.

Source

s50-network3.dat from Stocnet

References

See <http://www.stats.ox.ac.uk/~snijders/siena/>

s50a

Alcohol use data: excerpt from 'Teenage Friends and Lifestyle Study' data

Description

Data from an excerpt of 50 girls from the Teenage Friends and Lifestyle Study data set. Useful as a small example of network and behaviour, for which models can be fitted quickly.

Format

A matrix of variables relating to the use of alcohol for the actors in the network. Two columns, one for each time point. Coding is 1–5, high values indicating higher consumption.

Source

part of s50-alcohol.dat from the Stocnet package.

References

See <http://www.stats.ox.ac.uk/~snijders/siena/>

siena01Gui

User interface

Description

Gui to allow entry of the data for a siena model fit.

Usage

```
siena01Gui()
```

Details

This function provides a graphical user interface for fitting Siena models. It can be run from within an R session, but is also called by `siena.exe` (Windows) and `sienascript` (Linux or Mac). It allows entry of details of the data files required, either using the gui or by loading a session file. The Apply button causes a call to `sienaDataCreateFromSession` followed by a display of the `sienaModelOptions` screen.

The required format for the column entries is described on the help page for `sienaDataCreateFromSession`, as this function can also be called directly.

The entries for the table can be loaded from a file by using the buttons `Load new session from file` or `Continue session from file`. The former will create a new report file

and produce a descriptive report. The latter will use an existing report file and omit the descriptive report.

Alternatively, use `Add` and `Remove` buttons to enter the file names, and adjust the other columns to describe your data (see help page for `sienaDataCreateFromSession`).

The `Save to file` button will save the entries in the table to a session file.

The `Clear` button will empty the table.

The `Apply` button will prompt to save the session, then create the data objects and display the `sienaModelOptions` screen.

Exit by using the menu `File/Quit` or by closing the Window.

Value

None, although various objects made will still be in the directory if you are using this within an R session.

Author(s)

Ruth Ripley

References

See <http://www.stats.ox.ac.uk/~snidjers/siena/>

See Also

`sienaDataCreateFromSession`

Examples

```
## Not run: siena01Gui()
```

siena07

Function to fit a model

Description

Fits a model using method of moments, based on straightforward simulation, conditional or otherwise, or on an MCMC simulation. Estimation is done using Robbins-Monro algorithm. Note that the particular model to be used is passed in on the model object, and data for the model must be passed in using named arguments as the . . . (See examples)

Usage

```
siena07(x, batch=FALSE, verbose=FALSE, useCluster=FALSE,  
nbrNodes=2, initC=FALSE, clusterString=rep("localhost", nbrNodes),  
tt=NULL, parallelTesting=FALSE, ...)
```

Arguments

<code>x</code>	A model object
<code>batch</code>	Desired interface: 'batch' is a small amount of printout to the console
<code>verbose</code>	Produces various output to the console if TRUE
<code>useCluster</code>	Boolean: whether to use a cluster of processes
<code>nbrNodes</code>	Number of processes to use if useCluster is TRUE
<code>initC</code>	Boolean: set to TRUE if the simulation will use C routines (currently always needed). Only for use if using multiple processors, to ensure all copies are initialised correctly.
<code>clusterString</code>	Definitions of clusters. Default set up to use the localmachine only.
<code>tt</code>	A <code>tcltk</code> toplevel window. Used if called from the model options screen.
<code>parallelTesting</code>	Boolean. If TRUE, sets up random numbers to parallel those in Siena 3.
<code>...</code>	Arguments for the simulation function

Details

Runs a three-phase model estimation. Phase 1 does a few iterations to estimate the derivative matrix of the targets with respect to the parameter vector. Phase 2 does the estimation. Phase 3 runs a simulation to estimate standard errors and check the model. The simulation function is called once for each iteration in these phases and also once to initialise the model fitting and once to complete it. Unless in batch mode, displays a `tcl/tk` screen to allow interruption and to show progress.

Value

Returns an object of class "sienaFit", some parts of which are:

<code>theta</code>	Fitted value of theta
<code>covtheta</code>	Estimated covariance matrix of theta
<code>dfra</code>	Matrix of estimated derivatives
<code>sf</code>	Matrix of deviations from target in phase 3

Author(s)

Ruth Ripley

References

See <http://www.stats.ox.ac.uk/~sniijders/siena/>

See Also

[model.create](#)

Examples

```

mymodel <- model.create(findiff=TRUE, fn=simstats0c)
mynet1 <- sienaNet(array(c(s501, s502, s503), dim=c(50, 50, 3)))
mydata <- sienaDataCreate(mynet1)
myeff <- getEffects(mydata)
ans <- siena07(mymodel, data=mydata, effects=myeff, batch=TRUE)

#or for conditional estimation
mymodel$condname <- 'mynet1'
mymodel$sconditional <- TRUE
ans <- siena07(mymodel, data=mydata, effects=myeff, batch=TRUE)

```

sienaCompositionChange

Functions to create a Siena composition change object

Description

Used to create a list of events describing the changes over time of a Siena actor set

Usage

```

sienaCompositionChange(changelist, nodeSet = "Actors", option = 1)
sienaCompositionChangeFromFile(filename, nodeSet = "Actors",
  fileobj=NULL, option = 1)

```

Arguments

changelist	A list with an entry for each actor in the node set. Each entry a vector of numbers (may be as characters) indicating intervals during which the corresponding actor was present.
filename	Name of file containing change information. One line per actor, each line a series of space delimited numbers indicating intervals.
fileobj	The result of <code>readLines</code> on filename.
nodeSet	Character string containing the name of a Siena node set.
option	Integer controlling the processing of the network entries for the actors not currently present. Values (default is 1)

- 1 0 before entry, final value carried forward after leaving
- 2 0 before entry, missing after (final value carried forward, but treated as missing)
- 3 missing whenever not in the network. Previous values will be used where available, but always treated as missing values.
- 4 Convert to structural zeros (not available at present).

Details

Intervals are treated as closed at each end.

Value

An object of class "compositionChange", a list of numeric vectors, with attributes:

NodeSet	Name of node set
Option	Option

Author(s)

Ruth Ripley

References

See <http://www.stats.ox.ac.uk/~snijders/siena/>

See Also

[sienaNodeSet](#)

Examples

```
clist <- list(c(1, 3), c(1.4, 2.5))
#or
clist <- list(c('1', '3'), c('1.4', '2.5'))

compChange <- sienaCompositionChange(clist)

## Not run:
filedata <- c("1 3", "1.4 2.5")
write.table(filedata, "cc.dat", row.names=FALSE, col.names=FALSE,
           quote=FALSE)
## file will be
## 1 3
## 1.4 2.5
compChange <- sienaCompositionChangeFromFile("cc.dat")
## End(Not run)
```

sienaDataCreate *Function to create a Siena data object*

Description

Creates a Siena data object from input networks, covariates and composition change objects.

Usage

```
sienaDataCreate(..., nodeSets=NULL)
```

Arguments

... objects of class "sienaNet", "coCovar", "varCovar", "coDyadCovar", "varDyadCovar", "compositionChange"

nodeSets list of Siena node sets. Default is the single node set named 'Actors', length equal to the number of rows in the first object of class "sienaNet"

Details

Checks that the objects fit, that there is at least one network, and adds various attributes to each dependent variable describing the data. If there is more than one nodeSet they must all be specified.

Value

An object of class "siena" which is designed to be used in a siena model fit The components of the object are.

nodeSets	List of node sets involved
observations	Integer indicating number of waves of data
depvars	List of networks and behavior variables
cCovars	List of constant covariates
vCovars	List of changing covariates
dycCovars	List of constant dyadic covariates
dycCovars	List of changing dyadic covariates
compositionChange	List of composition change objects corresponding to the node sets

Author(s)

Ruth Ripley

References

See <http://www.stats.ox.ac.uk/~snidjers/siena/>

See Also

[sienaNet](#), [coCovar](#), [varCovar](#), [coDyadCovar](#), [varDyadCovar](#), [sienaCompositionChange](#)

Examples

```
myNet1 <- sienaNet(array(c(s501, s502, s503), dim=c(50, 50, 3)))
myNet2 <- sienaNet(s50a, type='behavior')
mydata <- sienaDataCreate(myNet1, myNet2)
```

```
sienaDataCreateFromSession
```

Creates a Siena data object from a Siena session file

Description

Reads in a Siena session from file or `siena01Gui()` and creates a Siena data or group object.

Usage

```
sienaDataCreateFromSession(filename = NULL, session = NULL,
  modelName = "Siena", edited = NULL, files = NULL)
```

Arguments

<code>filename</code>	Input session file
<code>session</code>	Input session (from <code>siena01Gui</code>)
<code>modelName</code>	Character string of project name
<code>edited</code>	Boolean, indicates whether a file has been edited and therefore should not be re-read. Used internally by <code>siena01Gui</code> .
<code>files</code>	List of data files, used internally by <code>siena01Gui</code>

Details

Allows creation of data objects of class "Siena" direct from data files rather than from the various Siena network and covariate objects. Is always called by `siena01Gui` but can also be used directly.

The columns of the gui screen should have the format described below. If a session file is used for input, it should have columns with exactly the same names and in exactly the same order as those below with a row of column headings and no row numbers.

Group Used to identify the groups when using the multi-group option described in the Manual. Must not contain embedded blanks, and should be identical for all rows which relate to the same group.

Name Network files or dyadic covariates should use the same name for each file of the set. Other files should have unique names, a list of space separated ones for constant covariates.

File Name in `siena01Gui`, usually entered by using a file selection box, after clicking Add.

Format Only relevant for networks or dyadic covariates. Can be matrix, a single Pajek network (.net) or a Siena network file (and edgelist with three or four columns: from, to, value, wave (optional)). Not tested for dyadic covariates yet!

Period(s) Only relevant for networks and dyadic covariates. All other files cover all the relevant periods. Indicates the order of the network and dyadic covariate files. Should range from 1 to n within each group. Enter multiple integers with spaces between for Siena network multi-wave files. Use the value 1 or blank for other files which cover multiple periods.

ActorSet If you have more than one set of nodes, use this column to indicate which is relevant to each file. Should not contain embedded blanks.

Type Indicate here what type of data the file contains. Options are "network", "behavior", "constant covariate", "changing covariate", "constant dyadic covariate", "changing dyadic covariate", "exogenous event".

Selected Yes or No. Only files with Yes will be included in the model.

Missing Values Enter any values which indicate missingness, with spaces between different entries.

Nonzero Codes Enter any values which indicate ties, with spaces between different entries.

NbrOfActors For Siena network files, enter the number of actors here.

If using a file for input, it should be of one of the following types:

Extension	Type
.csv	Comma separated
.dat or .prn	Space delimited
.txt	Tab delimited

Network and covariate files should be text files with a row for each node. The numbers should be separated by spaces or tabs. Exogenous events should be specified by a file with a row for each node. Each row should be consist of a set of pairs of numbers which indicate the periods during which the corresponding actor was present. e.g.

```
1 3
1.5 3
1 1.4 2.3 3
2.4 3
```

would describe a network with 4 nodes, and 3 observations. Actor 1 is present all the time, actor 2 joins at time 1.5, actor 3 leaves and time 1.4 then rejoins at time 2.3, actor 4 joins at time 2.4. All intervals are treated as closed.

Value

A list with the following components:

OK	Boolean, TRUE indicating success
mydata	A Siena data or group object, of class <code>siena</code> or <code>sienaGroup</code>
myeff	Effects object associated with <code>mydata</code>

Author(s)

Ruth Ripley

References

See <http://www.stats.ox.ac.uk/~snidjers/siena/>

See Also

[sienaDataCreate](#), [siena](#)

`sienaGroupCreate` *Function to group together several Siena data objects*

Description

Creates an object of class "sienaGroup" from a list of Siena data objects.

Usage

```
sienaGroupCreate(objlist, singleOK = FALSE)
```

Arguments

<code>objlist</code>	List of objects of class "siena"
<code>singleOK</code>	Boolean: is it OK to only have one object?

Details

This function creates a Siena group object from several Siena data objects, all of which use the same networks, covariates and actor sets (although possibly different subsets of the actor set in each object). It can be used as data input to `siena07` for the multigroup option. Also used internally for convenience with a single Siena data object.

Value

<code>...</code>	List containing the input objects, with attributes:
<code>netnames</code>	names of the dependent variables in each set
<code>symmetric</code>	vector of booleans, one for each dependent variable. TRUE if all occurrences of the network are symmetric.
<code>structural</code>	vector of booleans, indicating whether structurally fixed values occur in this network
<code>allUpOnly</code>	vector of booleans, indicating whether changes are all upwards in all the occurrences of this network
<code>allDownOnly</code>	similar to previous, but for downward changes
<code>anyUpOnly</code>	vector of booleans, indicating whether changes are all upwards in any of the occurrences of this network
<code>anyDownOnly</code>	similar to previous, but for downward changes
<code>types</code>	vector of network types of the dependent variables
<code>observations</code>	Total number of periods to process
<code>periodNos</code>	Sequence of numbers of periods which are not skipped in multigroup processing

netnodeSets	list of names of the node sets corresponding to the dependent variables
cCovars	names of the constant covariates, if any
vCovars	names of the changing covariates, if any
dycCovars	names of the constant dyadic covariates, if any
dyvCovars	names of the changing dyadic covariates, if any
ccnodeSets	list of the names of the node sets corresponding to the constant covariates
cvnodeSets	list of the names of the node sets corresponding to the changing covariates
dycnodeSets	list of the names of the node sets corresponding to the constant dyadic covariates
dyvcnodeSets	list of the names of the node sets corresponding to the changing dyadic covariates
compositionChange	boolean: any composition change at all?
exoptions	named vector of composition change options for the node sets
names	Either from the input objects or "Data1", "Data2" etc
class	"sienaGroup" inheriting from "siena"
balmean	vector of means for balance calculations
bRange	vector of difference between maximum and minimum values for behavior variables, NA for other dependent variables
behRange	matrix of maximum and minimum values for behavior variables, NA for other dependent variables
bSim	vector of similarity means for behavior variables, NA for other dependent variables
bPoszvar	vector of booleans indicating positive variance for behavior variables. NA for other dependent variables
bMoreThan2	vector of booleans indicating whether the behavior variables take more than 2 distinct values
cCovarPoszvar	vector of booleans indicating positive variance for constant covariates
cCovarMoreThan2	vector of booleans indicating whether the constant covariates take more than 2 distinct values
cCovarRange	vector of difference between maximum and minimum values for constant covariates
cCovarRange2	matrix of maximum and minimum values for constant covariates
cCovarSim	vector of similarity means for constant covariates
cCovarMean	vector of means for constant covariates
vCovarRange	vector of difference between maximum and minimum values for changing covariates
vCovarSim	vector of similarity means for changing covariates
vCovarMoreThan2	vector of booleans indicating whether the changing covariates take more than 2 distinct values

vCovarPoszvar	vector of booleans indicating positive variance for changing covariates
vCovarMean	vector of means for changing covariates
dycCovarMean	vector of means for constant dyadic covariates
dycCovarRange	vector of ranges for constant dyadic covariates
dycCovarRange2	matrix of maximum and minimum values for constant dyadic covariates
dyvCovarRange	vector of ranges for changing dyadic covariates
dyvCovarMean	vector of means for changing dyadic covariates
anyMissing	vector of booleans, one for each dependent variable, indicating the presence of any missing values
netRanges	matrix of maximum and minimum values for dependent networks, NA for behavior variables

Author(s)

Ruth Ripley

References

See <http://www.stats.ox.ac.uk/~snijders/siena/>

See Also

[sienaDataCreate](#)

Examples

```
## not very useful, but will work
mynet1 <- sienaNet(array(c(s501,s502),dim=c(50,50,2)))
mydata <- sienaDataCreate(mynet1)
mygroup <- sienaGroupCreate(list(mydata, mydata))
```

sienaModelOptions *Function to allow entry of model options*

Description

Displays a Gui with model options, and allows editing of effects plus running of Siena07

Details

Called from the `Apply` function in `siena01Gui`. An internal function of `siena01Gui`.

Various parameters can be set on the upper part of the screen:

Estimation Method 0 for Unconditional fitting, 1 for Conditional. If there are multiple dependent variables, a list will be displayed from which to choose.

Standard starting value If checked, the estimation will ignore the initial values of the parameters in the effects object, and use the default ones.

Specify random seed If you wish your run to be repeatable, check this box and then choose any integer as the seed.

Number of processors If checked, a box will appear for you to select the number of processors to be used. All processes will run on the same machine.

Initial value of gain parameter A parameter to control the Robbins-Monro algorithm. Will be multiplied by the number of processors before use.

Number of phase 2 subphases Default 4. To omit phase 2, set this to 0.

Derivative method 0 for finite differences, 1 for score function. Default 1.

Number of phase 3 iterations Default 1000.

If you wish to restrict the degree of the simulations, enter the value in the table on the bottom left.

Desired effects can be selected by using the button `Edit effects`. Change the `Include` column to a 1 to select, 0 to deselect.

Initial values can be specified in the `initialValues` column.

If it is desired to fix a parameter, set the `fix` column to 1.

To request a test, set both the `test` and `fix` columns to 1 and specify the value against which to test in the `initialValue` column.

Some effects have parameter values: these can be specified in the `parm` column.

Check the included effects by using the `Show included effects` button.

The model can be fitted by using the `Estimate` button.

The data objects can be saved to an R data set using `Save to file`.

The results object can be saved to an R data set using `Save results`.

The `Display Results` button is a toggle and should display or remove the display of the results file.

`Exit Model Options` allows you to return to the previous screen.

Value

None

Author(s)

Ruth Ripley

References

See <http://www.stats.ox.ac.uk/~snijders/siena/>

See Also

[siena01Gui](#), [siena07](#)

sienaNet

Function to create a Siena network object

Description

Creates a Siena network object from a matrix or array or list of sparse matrix of triples.

Usage

```
sienaNet(netarray, type = c("oneMode", "bipartite", "behavior"),
nodeSet = "Actors", sparse = is.list(netarray))
```

Arguments

netarray	matrix (type="behavior" only) or array of values or list of sparse matrices of type "dgTMatrix"
type	type of network, default "oneMode"
nodeSet	character string naming the appropriate node set. A vector containing 2 character strings for a bipartite network: "rows" first, then "columns".
sparse	logical: set to TRUE if the data is in sparse matrix format, FALSE otherwise

Details

Adds attributes so that the array or list of matrices can be used in a Siena model fit.

Value

An object of class "sienaNet".

Author(s)

Ruth Ripley

References

See <http://www.stats.ox.ac.uk/siena/>

See Also

[sienaDataCreate](#)

Examples

```
myNet1 <- sienaNet(array(c(s501,s502,s503),dim=c(50,50,3)))
```

sienaNodeSet *Function to create a node set*

Description

Creates a Siena node set which can be used as the nodes in a siena network.

Usage

```
sienaNodeSet(n, nodeSetName='Actors', names = NULL)
```

Arguments

n integer, size of set
nodeSetName character string naming the node set
names optional character string vector of length n of the names of the nodes.

Value

Returns a Siena node set, an integer vector, possibly with names, plus the attributes, class equal to 'sienaNodeSet', and nodeSetName equal to the argument nodeSetName.

Author(s)

Ruth Ripley

References

See <http://www.stats.ox.ac.uk/~snijders/siena/>

Examples

```
students <- sienaNodeSet(50, 'student')
```

simstats0c *A version of FRAN*

Description

The function to be called as 'FRAN'. Calls compiled C++.

Usage

```
simstats0c(z, x, INIT = FALSE, TERM = FALSE, initC=FALSE,
           data=NULL, effects=NULL, fromFiniteDiff=FALSE,
           profileData=FALSE, prevAns=NULL, returnDeps=FALSE)
```

Arguments

<code>z</code>	control object, passed in automatically in Siena07
<code>x</code>	model object, passed in automatically in Siena07
<code>INIT</code>	if TRUE, do initial processing. May be required to set up <code>z</code>
<code>TERM</code>	if TRUE, do end processing.
<code>initC</code>	if TRUE, call is to setup the data and model in C++. For use with multiple processes only
<code>data</code>	A siena object
<code>effects</code>	list of data frames as returned by <code>getEffects</code>
<code>fromFiniteDiff</code>	Boolean used during calculation of derivatives by finite differences. Not for user use.
<code>profileData</code>	boolean to force dumping of the data for profiling with <code>sienaProfile.exe</code>
<code>prevAns</code>	An object of class "sienaFit", from which scaling information (derivative matrix and standard deviation of the deviations) will be extracted along with the latest version of the parameters which will be used as the initial values, unless the model requests the use of standard initial values. Phase 1 will then be omitted.
<code>returnDeps</code>	Boolean. Whether to return the simulated networks in Phase 3. Currently only returns the first network.

Details

This should be used for the element FRAN of the model object. The arguments with no defaults must be passed in on the call to `siena07`.

Value

Depends on the call. If `INIT` or `initC` or `TERM` are true, returns `z`, the control object. Otherwise, returns a list containing:

<code>fra</code>	Simulated statistics
<code>sc</code>	Scores with which to calculate the derivative (not phase 2 or if using finite differences or maximum likelihood)
<code>dff</code>	For maximum likelihood, 2nd deriv??
<code>ntim</code>	For conditional processing, time taken
<code>feasible</code>	currently set to <code>TRUE</code>
<code>OK</code>	could be set to <code>FALSE</code> if serious error has occurred
<code>nets</code>	A list of simulated networks, as a list of edgelists for each period. Only if <code>returnDeps</code> is <code>TRUE</code> .

Author(s)

Ruth Ripley

References

See <http://www.stats.ox.ac.uk/~snijders/siena/>

See Also

[siena07](#)

Examples

```
myNet1 <- sienaNet(array(c(tmp3,tmp4),dim=c(32,32,2)))
mydata <- sienaDataCreate(myNet1)
myeff<- getEffects(mydata)
myeff[myeff$effectName=='transitive triplets'&
      myeff$type=='eval','include']<- TRUE
mymodel<- model.create(findiff=TRUE, fn = simstats0c)
ans<- siena07(mymodel, data=mydata, effects=myeff, batch=TRUE)
```

tmp3

van de Bunt's Freshman dataset, time point 3

Description

Third timepoint of van de Bunt's freshman dataset.

Format

Adjacency matrix for the network at time point 3.

Source

vrnd32t3.dat from Stocnet

References

See <http://www.stats.ox.ac.uk/~snijders/siena/>

tmp4

van de Bunt's Freshman dataset, time point 4

Description

Fourth timepoint of van de Bunt's freshman dataset

Format

Adjacency matrix for the network at time point 4.

Source

see siena web page

References

See <http://www.stats.ox.ac.uk/~snijders/siena/>

varCovar

Function to create a changing covariate object.

Description

This function creates a changing covariate object from a matrix.

Usage

```
varCovar(val, nodeSet='Actors')
```

Arguments

`val` Matrix of covariate values, one row for each actor, one column for each period.
`nodeSet` Character string containing the name of the associated node set

Details

When part of a Siena data object, the covariate is assumed to be associated with node set `nodeSet` of the Siena data object.

Value

Returns the covariate as an object of class 'varCovar', in which form it can be used as an argument to SienaData.create.

Author(s)

Ruth Ripley

References

See <http://www.stats.ox.ac.uk/~snijders/siena/>

See Also

[sienaDataCreate](#)

Examples

```
myvarCovar <- varCovar(s50a)
```

varDyadCovar *Function to create a changing dyadic covariate object.*

Description

This function creates a changing dyadic covariate object from an array.

Usage

```
varDyadCovar(val, nodeSets=c('Actors', 'Actors'))
```

Arguments

val Array of covariate values, Third dimension is the time
nodeSets Names (character string) of the associated node sets

Details

When part of a Siena data object, the covariate is assumed to be associated with the node sets named NodeSets of the Siena data object. The names of the associated node sets will only be checked when the Siena data object is created.

Value

Returns the covariate as an object of class 'varDyadCovar', in which form it can be used as an argument to SienaData.create.

Author(s)

Ruth Ripley

References

See <http://www.stats.ox.ac.uk/~snijders/siena/>

See Also

[sienaDataCreate](#)

Examples

```
mydyadvar <- varDyadCovar(array(c(s501,s502),dim=c(50,50,2)))
```

Index

*Topic **classes**

- coCovar, 3
- coDyadCovar, 4
- getEffects, 5
- model.create, 7
- sienaCompositionChange, 14
- sienaDataCreate, 16
- sienaDataCreateFromSession, 17
- sienaGroupCreate, 19
- sienaNet, 23
- sienaNodeSet, 24
- varCovar, 28
- varDyadCovar, 29

*Topic **datasets**

- s501, 10
- s502, 10
- s503, 11
- s50a, 11
- tmp3, 27
- tmp4, 27

*Topic **misc**

- installGui, 6
- siena01Gui, 12
- sienaModelOptions, 22

*Topic **models**

- siena07, 13
- simstats0c, 25

*Topic **package**

- RSiena-package, 2

*Topic **print**

- print01Report, 9

- coCovar, 2, 3, 17
- coDyadCovar, 4, 17

- getEffects, 2, 5

- installGui, 6

- model.create, 2, 7, 14

- print01Report, 9

- RSiena (*RSiena-package*), 2

- RSiena-package, 2

- s501, 10

- s502, 10

- s503, 11

- s50a, 11

- siena, 19

- siena (*sienaDataCreate*), 16

- siena01Gui, 2, 9, 12, 22, 23

- siena07, 2, 13, 23, 26

- sienaCompositionChange, 14, 17

- sienaCompositionChangeFromFile (*sienaCompositionChange*), 14

- sienaDataCreate, 2–4, 6, 16, 19, 22, 24, 28, 29

- sienaDataCreateFromSession, 2, 12, 13, 17

- sienaFit (*siena07*), 13

- sienaGroup, 19

- sienaGroup (*sienaGroupCreate*), 19

- sienaGroupCreate, 19

- sienaModel (*model.create*), 7

- sienaModelOptions, 12, 22

- sienaNet, 2, 17, 23

- sienaNodeSet, 15, 24

- simstats0c, 25

- tmp3, 27

- tmp4, 27

- varCovar, 17, 28

- varDyadCovar, 17, 29