Package 'dizzysNewInfec'

November 20, 2015

Type Package

Title Simulating SEIR/SIR models

Version 0.4

Date 2015-11-23

Author Tran Thi Cam Giang

Maintainer Tran Thi Cam Giang <camgiang2010@gmail.com>

Description This package permit us to simulate infectious diseases by using the SIR/SEIR models. To do that, we implement the direct algorithm of Gillespie in 1977 as described in the book ``Modeling Infectious Diseases IN HUMANS AND ANIMALS'' by Keeling and Pejman Rohani (2008), and the adaptive tau leaping to approximate the trajectory of a continuous-time stochastic process as described by Cao et al. (2007) The Journal of Chemical Physics. This package is based upon work supported by the Institute de Recherche pour Developpement at Paris in France and by Jean-Daniel Zucker and Marc Choisy.

Imports base,stats,graphics,methods, utils

License 0.4

Repository CRAN

LazyLoad true

Collate 'zzz.R' 'modelSEIRNewInfec.r' 'equiNewInfec.r' 'plotSEIRNewInfec.r' 'globSEIRNewInfec.r' 'globModelSEIRNewInfec.r' 'adapNewInfec.r'

R topics documented:

---

dizzysNewInfec-package

Modelling of Infectious Diseases

Description

Permiting to simulate infectious diseases by using the SIR/SEIR models under deterministic and stochastic models.

Details

Package:	dizzysNewInfec
Type:	Package
Version:	1.0
Date:	2013-05-02
License:	R
Depends:	methods, deSolve, rgl, survival, KMsurv

Author(s)

TRAN Thi Cam Giang <camgiang2010@gmail.com>

References

Matt J. Keeling and Pejman Rohani (2008) Modeling Infectious Diseases IN HUMANS AND ANIMALS. Princeton University Press.

Cao Y, Gillespie DT, Petzold LR,The Journal of Chemical Physics, 2007.

Norman Matloff (2009), The Art of R Programming.

See Also

package 'dizzyz'

---

coef.seirNewInfec	*Coefficient of seirNewInfec Object*

Description

coef is a generic function which extracts coefficients of parameters from objects returned by modeling functions.

Usage

```
coef(object, ...)
```

Arguments

object	an object for which the extraction of model coefficients is meaningful.
...	other arguments.

Details

The function uses an object of class 'seirNewInfec' for extracting coefficients of parameters. There are two parameter types to show. One is the initial values of state variables. The other is the values of parameters used in simulation.

Value

'coef.seirNewInfec' showing the initial values of state variables and the values of parameters used in simulation for all cities.

Author(s)

TRAN Thi Cam Giang

References

Extract Model Coefficients in R.

See Also

The generic function 'coef' in R

Examples

```
seirobj1<-globSEIRSimulNewInfec(N=1e7)
coef(seirobj1)

seirobj2<-globSEIRSimulNewInfec(nbVilles=3, N=c(1e7,1e6))
coef(seirobj2)
```

---

detSEIRNewInfec   *Creat a seir Object*

Description

This is important function, it creates a seir object for 'seirNewInfec' class.

Usage

```
detSEIRNewInfec(nbVilles=1,duration=10*365,unitTIME=1,S=NULL,E=NULL,I=NULL,R=NULL,N=1e7,periDISE=36
mu=1/(70*365),nbCONTACT0=300,nbCONTACT1=.1,probINFECTER=0.01,sigma=1/8,gamma=1/5,phiPHASE=c(0))
```

Arguments

nbVilles	number of subpopulations in the metapopulation.
duration	time values over which to perform the numerical integration.
S	initial value of the state variable 'S', this is the number of susceptible individuals at the time t0.
E	initial value of the state variable 'E', this is the number of exposed individuals at the time t0.
I	initial value of the state variable 'I', this is the number of infected individuals at the time t0.
R	initial value of the state variable 'R', this is the number of recovered individuals in immunity at the time t0.
N	number of population.
mu	per capita birth and death rates (per day).
nbCONTACT0	mean value of the number of contact per day
nbCONTACT1	amplitude of the number of contact per day
probINFECTER	the probability that a susceptible individual native from i being in contact with another infected individual native from k gets infected
sigma	transition rate from exposed (E) to infected (I), per day (inverse of the average duration of the latency period).
gamma	recovery rate (per day).
periDISE	period of the contact rate (in days).
phiPHASE	phase of the contact rate (in radians).
unitTIME	unit of time for simulation.
...	further parameters.

Details

To implement this function, we have three step. The first step, in R, is to get the initial values of state variables and the values of parameter, then, sends those values to the C++ functions. The second step is to do simulation in the C++ implementation. The third step is to send the result from the C++ function to R. We will get the simulation resultat in R.

Moreover, in the set of arguments, there are some arguments we can give each subpopulation each value's.

S, E, I, R, N: the initial values of state variables for 'nbVilles' subpopulations. Multiple values can be specified so that each subpopulation can be given its own state variables. If there are fewer values than subpopulations they are recycled in the standard fashion. Subpopulations will all be simulated in the first value specified.

mu, nbCONTACT0, nbCONTACT1,probINFECTERsigma, gamma, phiPHASE: the parameters of simulation for 'nbVilles' subpopulations. Multiple values can be specified so that each subpopulation can be given its own parameters. If there are fewer values than subpopulations they are recycled in the standard fashion. Subpopulations will all be simulated in the first value specified.

Value

Result returned is a seir object of the 'seirNewInfec' class with all values for all slots.

Author(s)

TRAN Thi Cam Giang

References

Norman Matloff (2009), The Art of R Programming.

See Also

'globSEIRNewInfec', 'stoSEIRNewInfec' fuctions in 'dizzysNewInfec' package.

Examples

```
obj<-globSEIRNewInfec(typeSIMU="deter",nbVilles=3,N=c(1e7,1e6))
plot(obj)
```

| equiNewInfec | *Finding Endemic Equilibrium of a seasonally-forced SEIR/SIR model* |

Description

'equiNewInfec' returns the values of the state variables of an SEIR/SIR model at endemic equilibrium.

Usage

```
equiNewInfec(duration=100*365,unitTIME=1,N=10e6,mu=1/(70*365),
    nbCONTACT0=300,nbCONTACT1=.1,probINFECTER=0.1,sigma=1/7,gamma=1/7,phiPHASE=c(0),periDISE=365)
```

Arguments

duration	time values over which to perform the numerical integration.
unitTIME	time unit of simulation.
N	number of population.
mu	per capita birth and death rates (per day).
nbCONTACT0	mean value of the number of contact per day
nbCONTACT1	amplitude of the number of contact per day
probINFECTER	the probability that a susceptible individual native from i being in contact with another infected individual native from k gets infected
sigma	transition rate from exposed (E) to infected (I), per day (inverse of the average duration of the latency period.
gamma	recovery rate (per day).
phiPHASE	phase of the contact rate (in radians).
periDISE	period of the contact rate (in days).

Details

The host population is supposed to be at demographic equilibrium with the births balancing the deaths (births and death occur with the same rate 'mu') and a total population size constant and equal to 1. The contact rate is forced by a sinusoid of period 'periDISE' and phase 'phiPHASE'. The SEIR epidemiological model is defined by the following set of differential equations:

beta = -nbCONTACT0*log(1-probINFECTER);

dS/dt = mu*N - nbCONTACT0 * (1 + nbCONTACT1 * cos(2 * pi * t / periDISE + phiPHASE))*log[1-(I*probINFECTER/N)] * S * I - mu * S

dE/dt = nbCONTACT0 * (1 + nbCONTACT1 * cos(2 * pi * t / periDISE + phiPHASE))*log[1-(I*probINFECTER/N)] * S * I - mu * E - sigma * E

dI/dt = sigma * E - mu * I - gamma * I

dR/dt = gamma * I - mu * R

'nbCONTACT0' is the mean value of the contact number and 'nbCONTACT1' is the amplitude (in percentage of the mean) around the mean. The endemic equilibrium value of this sinusoidally-forced epidemiological system is found in two steps. First the endemic equilibrium point (S*,E*,I*,R*) of the unforced system ('nbCONTACT1 = 0') is found from the following analytical equations:

S* = (gamma+mu)*(sigma+mu)/(beta * sigma)

E* = mu*((1/(sigma+mu)) - ((gamma+mu)/(beta * sigma)))

I* = mu*((beta*sigma - (gamma+mu)*(sigma+mu))/(beta *(gamma+mu)*(sigma+mu)))

R* = 1 - S* - E* - I*

This endemic equilibium point is used as a starting point for numerical integration of the system of differential equations. This is done by the 'detSEIRNewInfec' function that uses the 'ode' function of the 'deSolve' package. The last values of the of the state variable are returned. These correspond to a given point on the equilibrium limit cycle (assuming that this equilibrium limit cycle is reached) and the value of this point will depend on the phase 'phiPHASE'.

Value

'equiNewInfec' returns a named vector of 3 numerical proportions containing the values of the state variables S, E and I on one point of the equilibrium limit cycle (provided that this limit cycle is reached). In case this limit cycle is reached, the value of the S, E and I state variables will depend on the input phase value 'phi'.

Author(s)

TRAN Thi Cam Giang

References

Anderson RM & May RM (1991) Infectious Diseases of Humans - Dynamics and Control. Oxford University Press.

See Also

'seir' in package 'dizzyz' and 'ode' in package 'deSolve'.

Examples

```
## The point on the limit cycle depends on the input phase value 'phi':
res<-equiNewInfec(duration=100*365,unitTIME=1,N=10e6,mu=1/(70*365),
   nbCONTACT0=300,nbCONTACT1=.1,probINFECTER=0.1,sigma=1/7,gamma=1/7,phiPHASE=c(0),periDISE=365)
print(res)
```

| globSEIRNewInfec | *Creat a seir Object* |

Description

This is important function, it creates a seir object for 'seirNewInfec' class.

Usage

```
globSEIRNewInfec(typeSIMU="stoch",duration=5*365,method="direct",unitTIME=1,mu=1/(70*365),
nbCONTACT0=100,nbCONTACT1=0.0,probINFECTER=0.01,probVISITER=0.01,
sigma=1/8,gamma=1/5,
periDISE=365,phiPHASE=c(0),nbVilles=1,seed=as.numeric(Sys.time()),typeRNG="good",
S=NULL,E=NULL,I=NULL,R=NULL,N=1e5)
```

Arguments

typeSIMU	type of simulation, 'deterministic' or 'stochastic'.
duration	time values over which to perform the numerical integration.
S	initial value of the state variable 'S', this is the number of susceptible individuals at the time t0.
E	initial value of the state variable 'E', this is the number of exposed individuals at the time t0.
I	initial value of the state variable 'I', this is the number of infected individuals at the time t0.
R	initial value of the state variable 'R', this is the number of recovered individuals in immunity at the time t0.
N	number of population.
mu	per capita birth and death rates (per day).
nbCONTACT0	mean value of the number of contact per day
nbCONTACT1	amplitude of the number of contact per day
probINFECTER	the probability that a susceptible individual native from i being in contact with another infected individual native from k gets infected
probVISITER	the probability that an individual from subpopulation i visits subpopulation j
sigma	transition rate from exposed (E) to infected (I), per day (inverse of the average duration of the latency period).
gamma	recovery rate (per day).
periDISE	period of the contact rate (in days).
phiPHASE	phase of the contact rate (in radians).
nbVilles	number of subpopulations in the metapopulation.
seed	random seed.
typeRNG	random number generator, 'good' or 'fast'. If we chose typeRNG="good", it means that we use the random number generator in C++. In the other side, we chose typeRNG="fast", it means that we use the random number generator of the professor Yann Chevaleyre at university of Paris 13. With the random generator's Yann Chevaleyre, we find that it is faster than the random generator in C++, in contrast, its accuracy is less than that of the random generator's Yann Chevaleyre.
method	simulation algorithm,'direct' or 'adaptivetau'. If we chose method="direct", it means that we implement the direct algorithm of Gillespie in 1977. In contrast, if we chose method="adaptivetau", it means that we implement the adaptive tau-leaping approximation for simulating the trajectory of a continuous-time Markov process.
unitTIME	unit of time for simulation.
...	further parameters.

Details

To implement this function, we have three step. The first step, in R, is to get the initial values of state variables and the values of parameter, then, sends those values to the C++ functions. The second step is to do simulation in the C++ implementation. The third step is to send the result from the C++ function to R. We will get the simulation resultat in R.

Moreover, in the set of arguments, there are some arguments we can give each subpopulation each value's.

S, E, I, R, N: the initial values of state variables for 'nbVilles' subpopulations. Multiple values can be specified so that each subpopulation can be given its own state variables. If there are fewer values than subpopulations they are recycled in the standard fashion. Subpopulations will all be simulated in the first value specified.

mu, nbCONTACT0, nbCONTACT1,probINFECTER,probVISITER, sigma, gamma, phiPHASE: the parameters of simulation for 'nbVilles' subpopulations. Multiple values can be specified so that each subpopulation can be given its own parameters. If there are fewer values than subpopulations they are recycled in the standard fashion. Subpopulations will all be simulated in the first value specified.

Value

Result returned is a seir object of the 'seirNewInfec' class with all values for all slots.

Author(s)

TRAN Thi Cam Giang

References

Norman Matloff (2009), The Art of R Programming.

See Also

'globSEIRNewInfec', 'detSEIRNewInfec', 'stoSEIRNewInfec' fuctions in 'dizzysNewInfec' package.

Examples

```
obj<-globSEIRNewInfec()
plot(obj)
obj<-globSEIRNewInfec(typeSIMU="sto",nbVilles=3,N=c(1e7,1e6))
plot(obj)
obj<-globSEIRNewInfec(typeSIMU="deter",nbVilles=3,N=c(1e7,1e6))
plot(obj)
```

---

globSEIRSimulNewInfec   *Redoing or Continuing a Simulation.*

Description

Redoing or continuing a simulation by using values of parameters given.

Usage

```
globSEIRSimulNewInfec(object,typeSIMU="stoch",continue=F,
duration=5*365,method="direct",unitTIME=1,mu=1/(70*365),
nbCONTACT0=200,nbCONTACT1=.1,probINFECTER=0.01,probVISITER=0.1,
sigma=1/8,gamma=1/5,
periDISE=365,phiPHASE=c(0),nbVilles=1,seed=as.numeric(Sys.time()),
typeRNG="good", append=TRUE, t0=NULL,
S=NULL,E=NULL,I=NULL,R=NULL,N=1e7)
```

Arguments

object	a seir object. If missing this object, we create a new seir object by using the values of state variables and of parameters given.
typeSIMU	type of simulation, 'deterministic' or 'stochastic'.
continue	logical (T/F). If continue=F, it means to redo silulation. Else, if continue=T, it means to continue to do silulation.
duration	time values over which to perform the numerical integration.
method	simulation algorithm,'direct' or 'adaptivetau'. If we chose method="direct", it means that we implement the direct algorithm of Gillespie in 1977. In contrast, if we chose method="adaptivetau", it means that we implement the adaptive tau-leaping approximation for simulating the trajectory of a continuous-time Markov process.
unitTIME	unit of time.
mu	per capita birth and death rates (per day).
nbCONTACT0	mean value of the number of contact per day
nbCONTACT1	amplitude of the number of contact per day
probINFECTER	the probability that a susceptible individual native from i being in contact with another infected individual native from k gets infected
probVISITER	the probability that an individual from subpopulation i visits subpopulation j
sigma	transition rate from exposed (E) to infected (I), per day (inverse of the average duration of the latency period).
gamma	recovery rate (per day).
periDISE	period of the contact rate (in days).
phiPHASE	phase of the contact rate (in radians).

nbVilles	number of subpopulations in the metapopulation.
seed	random seed for random number generator.
typeRNG	random number generator, 'good' or 'fast'. If we chose typeRNG="good", it means that we use the random number generator in C++. In the other side, we choose typeRNG="fast", it means that we use the random number generator of the professor Yann Chevaleyre at university of Paris 13. With the random generator's Yann Chevaleyre, we find that it is faster than the random generator in C++, in contrast, its accuracy is less than that of the random generator's Yann Chevaleyre.
S	initial value of the state variable 'S', this is the number of susceptible individuals at the time t0.
E	initial value of the state variable 'E', this is the number of exposed individuals at the time t0.
I	initial value of the state variable 'I', this is the number of infected individuals at the time t0.
R	initial value of the state variable 'R', this is the number of recovered individuals in immunity at the time t0.
N	number of population.
phiPHASE	phase of the contact rate (in radians).
append	logical(T/F). It is available when continue=T. If continue=T and append=T, it means that we want our new object contains the old data + the new. If continue=T and append=F, we only get the new.
t0	numeric. It means that if we want the start time after what has already been simulated ("t0 = NULL") or if we prefer that time restarts any value (for example "t0 = 1").

Details

For this fucntion, to get the values of arguments, first, for the arguments missed, we get its values from the seir object, in contrast, we get its given values. Moreover, in the set of arguments, there are some arguments we can give each subpopulation each value's. **S, E, I, R, N**: the initial values of state variables for 'nbVilles' subpopulations. Multiple values can be specified so that each subpopulation can be given its own state variables. If there are fewer values than subpopulations they are recycled in the standard fashion. Subpopulations will all be simulated in the first value specified. **mu, nbCONTACT0, nbCONTACT1, probVISITER, probINFECTER, sigma, gamma, phiPHASE**: the parameters of simulation for 'nbVilles' subpopulations. Multiple values can be specified so that each subpopulation can be given its own parameters. If there are fewer values than subpopulations they are recycled in the standard fashion. Subpopulations will all be simulated in the first value specified.

Value

Result returned is an object of the 'seir' class.

Author(s)

TRAN Thi Cam Giang

See Also

'simul' function in the 'dizzys' package.

Examples

```
#STO, STO
sto<-globSEIRNewInfec(N=1e6,typeSIMU="stoch",duration=5*365,nbVilles=2)
plot(globSEIRSimulNewInfec(sto,typeSIMU="stoch",continue=TRUE,duration=5*365,nbCONTACT1=0,phiPHASE=c(pi/2,0)),
#DET, DET
det<-globSEIRNewInfec(N=10e4,typeSIMU="deter",duration=50*365)
plot(globSEIRSimulNewInfec(det,typeSIMU="deter",continue=TRUE,duration=5*365,nbCONTACT1=0.1,phiPHASE=pi))
```

---

lines.seirNewInfec   *Add Connected Line Segments of an seir object to a Plot 2D/3D*

Description

A generic function taking coordinates given of the slot 'pop' of a seir object in various ways and joining the corresponding points with line segments.

Usage

```
lines(object, x, y, z, pop, col, type, unitTIME, proj,...)
```

Arguments

object	a seir object, this is the important parameter.
x, y, z	the names or the positions of the columns in the slot 'pop' of the seir object. This arguments are corresponding to the x, y and z coordinates of the plot. Normally, if 'z' is NULL, we only can add a line 2D to a plot 2D, on the other side, if 'z' is one of the names or the positions of the columns ("time","S","E","P","R","N") or (1,2,3,4,5,6), we can add a line 3D to a plot 3D.
pop	numeric. What subpopulations are chosen to add from the subpopulation set. The subpopulation set is a series of natural number from one to n, where n is the number of subpopulation in a metapopualtion of n subpopulation. Normally, 'pop' is NULL, it allows us to add all lines of the n subpopulations to the plot. In the other side, 'pop' is a good numeric, it means that 'pop' is in the series (1,...,n), we only can add the lines of the subpopulation in 'pop' to the plot.
col	the colors for lines. Multiple colors can be specified so that each line can be given its own color. If there are fewer colors than number of lines they are recycled in the standard fashion. Lines will all be plotted in the first colour specified.
unitTIME	the unit of time. It is 'day' corresponding to unitTIME=1, is 'week' corresponding to unitTIME=7, is 'month' corresponding to unitTIME=30 and is 'year' corresponding to unitTIME=365. Normally, unitTIME is equal to 1, however we can change it.

type	what type of plot should be drawn. Possible types are: "l", "b", "s",...
proj	list of the plane names. This argument is only used in the plot 3D. It allows us to add the projection of the lines 3D on the planes. It is necessary to give the value of 'proj' corresponding to the x, y and z axes.
...	further graphical parameters such as 'lwd', 'lty'.

Details

The function 'lines' is the partial analogue of plot.seir with the argument add=T. Moreover, for the x, y, and z coordinates, the slot 'pop' of the seir object is a list of the data frames corresponding to each subpopulation in a metapopulation of n subpopulations. The each data frame often has six columns that are 'time' being a time column, at time t, 'S' being the number of susceptibles, 'E' being the number of exposed individuals, 'P' being the number of infectives, 'R' being the number of removed individuals with immunity and 'N' being the number of population, it is the sum of S, E, P, R.

Author(s)

TRAN Thi Cam Giang

References

Murrell, P. (2005) R Graphics. Chapman & Hall/CRC Press.

Examples

```
#creating a plot
#adding a line to the plot
globSEIRSimulNewInfec(nbVilles=2)->obj
globSEIRSimulNewInfec(nbVilles=1)->obj1
#2D
plot(obj,col="red")
lines(obj1,col="blue",lwd=2)
#3D
plot(obj,z="S",col="red",proj=list(c("time","P")))
lines(obj1,z="S",col="blue",proj=list(c("time","P")))
```

---

persNewInfec	*persNewInfec in a Metapopulation*

Description

Generic function for calculating number of subpopulations in a metapopulation not extincts at time t.

Usage

```
persNewInfec(object)
```

Arguments

object   a seir object.

Details

The function 'persNewInfec' gives us the number of subpopulations, in a metapopualtion, in which the disease is not extinct at time t by using the parameter 'object' given.

Value

'persNewInfec' returns a seir object with the value of the slot 'persNewInfec'. The slot 'persNewInfec' is a data frame. This data frame has five columns, 'time' means being the time at which there are some extinct subpopulations, at time t, we have, 'nbVilles' is the number of not extinct subpopulations before the time t, 'ndie' means being the number of died subpopulations, 'dieVille' gives us which subpopulation is died, 'remain' is the number of not extinct subpopulations after the time t.

Author(s)

TRAN Thi Cam Giang

References

Matt J. Keeling and Pejman Rohani (2008) Modeling Infectious Diseases IN HUMANS AND ANIMALS. Princeton University Press.

Examples

```
obj1<-obj1<-globSEIRSimulNewInfec(nbVilles=5,N=1e5,nbCONTACT0=100,duration=365*30)
objper<-persNewInfec(obj1)
objper@persistence
```

---

  plot.persNewInfec   *Plotting Kaplan Meier Survival Curve*

Description

Plotting Kaplan–Meier survival curve based on persistence value or adding one or more straight lines through the current plot based on the slot 'pop' of seir object.

Usage

```
plot(object, x, y, type, col, xlim, ylim, curvetype, vilabline,...)
```

Arguments

`object`	a seir object with the value of the slot "persistence".
`x, y`	the names or the positions of the columns in the slot 'pop' of the seir object. This arguments are corresponding to the x and y coordinates of the plot. In this function, we only plot lines 2D, 'x, y' are one of the names or the positions of the columns ("time","S","E","P","R","N") or (1,2,3,4,5,6)
`type`	what type of plot should be drawn. Possible types are "p", "l", "b", "c", ect.
`col`	the colors for lines. Multiple colors can be specified so that each line can be given its own color. If there are fewer colors than number of lines they are recycled in the standard fashion. Lines will all be plotted in the first colour specified.
`xlim`	the x limits of the plot.
`ylim`	the y limits of the plot.
`curvetype`	type of curve for plot. There are two value for 'curvetype'. If curvetype=="KM" (Kaplan–Meier), it means that we will plot Kaplan–Meier survival curve based on the slot 'persistence' of the object. In the other side, if curvetype=="population", it means that we will plot all lines by using the slot 'pop' of the object with straight lines.
`vilabline`	numeric. What subpopulations are chosen to add straight lines to plot, from the subpopulation set. The subpopulation set is a series of natural number from one to n, where n is the number of subpopulation in a metapopualtion of n subpopulation. Normally, 'vilabline' is NULL, we don't add any straight lines to the plot. In the other side, 'vilabline' is a good numeric, it means that 'vilabline' is in the series (1,...,n), we only can add the lines of the subpopulation in 'vilabline' to the plot.
`...`	other arguments.

Details

This function allows us to plot two types of curve to the plot. Plotting Kaplan–Meier survival curve based on persistence value or adding one or more straight lines through the current plot based on the slot 'pop' of seir object.

Author(s)

TRAN Thi Cam Giang

References

David G. Kleinbaum and Mitchel Klein, Survival Analysis.

See Also

Kaplan–Meier survival curve

Examples

```
p<-persNewInfec(globSEIRSimulNewInfec(nbVilles=5,N=1e5,nbCONTACT0=100,duration=365*30))
plot.persNewInfec(p)
x11()
plot.persNewInfec(p,curvetype="pop",col=c("green","blue"),vilabline=c(1,3))
```

---

plot.seirNewInfec   *Plotting 2D/3D a seir Object*

Description

Generic function of 'seir' class for plotting 2D/3D of seir objects.

Usage

```
plot(object,x="time",y=4,z=NULL,pop=c(),col="black",type="l",unitTIME=1,
proj=list(),add=F,xlim=NULL,ylim=NULL,zlim=NULL,xlab=x,ylab="#infected",zlab=z)
```

Arguments

object	seir object.
x, y, z	the names or the positions of the columns in the slot 'pop' of the seir object. This arguments are corresponding to the x, y and z coordinates of the plot. Normally, if 'z' is NULL, we only can add a line 2D to a plot 2D, on the other side, if 'z' is one of the names or the positions of the columns ("time","S","E","P","R","N") or (1,2,3,4,5,6), we can add a line 3D to a plot 3D.
pop	numeric. What subpopulations are chosen to add from the subpopulation set. The subpopulation set is a series of natural number from one to n, where n is the number of subpopulation in a metapopualtion of n subpopulation. Normally, 'pop' is NULL, it allows us to add all lines of the n subpopulations to the plot. In the other side, 'pop' is a good numeric, it means that 'pop' is in the series (1,...,n), we only can add the lines of the subpopulation in 'pop' to the plot.
col	the colors for lines. Multiple colors can be specified so that each line can be given its own color. If there are fewer colors than number of lines they are recycled in the standard fashion. Lines will all be plotted in the first colour specified.
type	what type of plot should be drawn. Possible types are "p", "l", "b", "c", ect.
unitTIME	the unit of time. It is 'day' corresponding to unitTIME=1, is 'week' corresponding to unitTIME=7, is 'month' corresponding to unitTIME=30 and is 'year' corresponding to unitTIME=365. Normally, unitTIME is equal to 1, however we can change it.
proj	list of the plane names. This argument is only used in the plot 3D. It allows us to add the projection of the lines 3D on the planes. It is necessary to give the value of 'proj' corresponding to the x, y and z axes.

add	adding lines of the object to a current plot, if add=T. In contrast, if add=F, we plot lines of the object to a new plot. By default, add=F.
xlim, ylim, zlim	
	limits to use for the coordinates.
xlab, ylab, zlab	
	labels for the coordinates.
...	further graphical parameters such as 'lwd', 'lty'

Details

Moreover, for the x, y, and z coordinates, the slot 'pop' of the seir object is a list of the data frames corresponding to each subpopulation in a metapopulation of n subpopulations. The each data frame often has six columns that are 'time' being a time column, at time t, 'S' being the number of susceptibles, 'E' being the number of exposed individuals, 'P' being the number of infectives, 'R' being the number of removed individuals with immunity and 'N' being the number of population, it is the sum of S, E,P,R.

In the set of arguments, there are some arguments we can give each subpopulation each value's.

S, E, I, R, N: the initial values of state variables for 'nbVilles' subpopulations. Multiple values can be specified so that each subpopulation can be given its own state variables. If there are fewer values than subpopulations they are recycled in the standard fashion. Subpopulations will all be simulated in the first value specified.

mu, nbCONTACT0, nbCONTACT1, probVISITER, probINFECTER, sigma, gamma, phi: the parameters of simulation for 'nbVilles' subpopulations. Multiple values can be specified so that each subpopulation can be given its own parameters. If there are fewer values than subpopulations they are recycled in the standard fashion. Subpopulations will all be simulated in the first value specified.

Author(s)

TRAN Thi Cam Giang

References

Generic X-Y Plotting in R. plot.default in R.

See Also

'plot.seirNewInfec' in 'dizzys' package.

Examples

```
obj<-globSEIRSimulNewInfec(nbVilles=3, N=1e6, nbCONTACT0=100)
plot(obj,col=c("red","blue"),lwd=2,xlab="time (day)", ylab="number of infectives")
pause()
plot(obj,z="S",col=c("red","blue"),lwd=2,xlab="time (day)", ylab="number of infectives",zlab="number of suscept:
pause()
#plot(obj,z="S",col=c("red","blue"),lwd=2,proj=list(c("time","P"),c("time","S")),box=F,xlab="time (day)", ylab=
```

| pop.seirNewInfec | *Extract Values of State Variables of each City according to Time.* |

Description

Generic function allowing to extract number of S, I, R individuals of each population in a metapopulation of n subpopulations according to time.

Usage

```
pop.seirNewInfec(object, subset, fct,...)
```

Arguments

object	a seir object.
subset	numeric. What subpopulations are chosen to extract from the subpopulation set. The subpopulation set is a series of natural number from one to n, where n is the number of subpopulation in a metapopualtion of n subpopulation. Normally, 'subset' is NULL, it allows us to extract all n subpopulations. In the other side, 'subset' is a good numeric, it means that 'subset' is in the series (1,...,n), we only can extract the subpopulations in 'subset' argument.
fct	this is a function such as "sum", "mean",etc. With this function, we can do it with the extracted data from subpopulations.
...	other parameters.

Details

Result returned is dependent on the values of the parameter 'subset' and 'fct'.

if subset=NULL and fct=NULL, the result is the list of the 'pop' slot in the object given.

if subset=NULL and fct is a function, the result is a matrix after doing the 'fct' function on the list of the 'pop' slot in the object given.

if subset!=NULL and fct=NULL, the result is a list of the subpopulations extracted from the list of the 'pop' slot in the object given.

if subset!=NULL and fct is a function, the result is a matrix after doing the 'fct' function on the list of the subpopulations extracted in the object given.

Value

Viewing in detail in the 'detail' part.

Author(s)

TRAN Thi Cam Giang

References

Norman Matloff (2009), The Art of R Programming.

See Also

'pop.seirNewInfec' fucntion in 'dizzys' package.

Examples

```
obj<-globSEIRSimulNewInfec(nbVilles=3, N=1e6)
tpobj<-pop(obj)
class(tpobj)
tpobj<-pop(obj,fct="sum")
class(tpobj)
tpobj<-pop(obj,subset=c(1,2),fct="sum")
class(tpobj)
```

---

stoSEIRNewInfec   *Creat a seir stochastic Object*

Description

This is important function, it creates a seir stochastic object for 'seirNewInfec' class.

Usage

```
stoSEIRNewInfec(sigma=1/8,gamma=1/5,mu=1/(70*365),seed=as.numeric(Sys.time()),S=NULL,E=NULL,I=NULL,
nbCONTACT0=300, nbCONTACT1=0.1,phiPHASE=c(0),
probVISITER=0.01, probINFECTER=0.01, duration=1000, nbVilles=2, unitTIME=1,periDISE=365,
typeRNG="good",typeSIMU="stoch",method="direct",statSTATE=FALSE)
```

Arguments

sigma	transition rate from exposed (E) to infected (I), per day (inverse of the average duration of the latency period).
gamma	recovery rate (per day).
mu	per capita birth and death rates (per day).
duration	time values over which to perform the numerical integration.
S	initial value of the state variable 'S', this is the number of susceptible individuals at the time t0.
E	initial value of the state variable 'E', this is the number of exposed individuals at the time t0.
I	initial value of the state variable 'I', this is the number of infected individuals at the time t0.
R	initial value of the state variable 'R', this is the number of recovered individuals in immunity at the time t0.

N	number of population.
nbCONTACT0	mean value of the number of contact per day
nbCONTACT1	amplitude of the number of contact per day
probINFECTER	the probability that a susceptible individual native from i being in contact with another infected individual native from k gets infected
probVISITER	the probability that an individual from subpopulation i visits subpopulation j
periDISE	period of the contact rate (in days).
phiPHASE	phase of the contact rate (in radians).
nbVilles	number of subpopulations in the metapopulation.
seed	random seed.
typeSIMU	what type of model should be simulated : "deterministic" or "stochastic"
typeRNG	random number generator, 'good' or 'fast'. If we chose rng="good", it means that we use the random number generator in C++. In the other side, we chose rng="fast", it means that we use the random number generator of the professor Yann Chevaleyre at university of Paris 13. With the random generator's Yann Chevaleyre, we find that it is faster than the random generator in C++, in contrast, its accuracy is less than that of the random generator's Yann Chevaleyre.
method	simulation algorithm,'direct' or 'adaptivetau'. If we chose method="direct", it means that we implement the direct algorithm of Gillespie in 1977. In contrast, if we chose method="adaptivetau", it means that we implement the adaptive tau-leaping approximation for simulating the trajectory of a continuous-time Markov process.
unitTIME	unit of time for simulation.
statSTATE	state of metapopulation is stationary? all subpopulation in a metapopulation is in the same state? If statSTATE=FALSE, we execute simulations with the initial values of variables given, else statSTATE=FALSE, we calculate the stationary state for the metapopulation, all initial values of variables in the metapopulation will be in the stationary state.
...	further parameters.

Details

To implement this function, we have three step. The first step, in R, is to get the initial values of state variables and the values of parameter, then, sends those values to the C++ functions. The second step is to do simulation in the C++ implementation. The third step is to send the result from the C++ function to R. We will get the simulation resultat in R.

Moreover, in the set of arguments, there are some arguments we can give each subpopulation each value's.

S, E, I, R, N: the initial values of state variables for 'nbVilles' subpopulations. Multiple values can be specified so that each subpopulation can be given its own state variables. If there are fewer values than subpopulations they are recycled in the standard fashion. Subpopulations will all be simulated in the first value specified.

mu, nbCONTACT0, nbCONTACT1,probINFECTER,probVISITER, sigma, gamma, phiPHASE, statSTATE: the parameters of simulation for 'nbVilles' subpopulations. Multiple values can be

specified so that each subpopulation can be given its own parameters. If there are fewer values than subpopulations they are recycled in the standard fashion. Subpopulations will all be simulated in the first value specified.

Value

Result returned is a seir object of the 'seirNewInfec' class with all values for all slots.

Author(s)

TRAN Thi Cam Giang

References

Norman Matloff (2009), The Art of R Programming.

See Also

'globSEIRNewInfec', 'detSEIRNewInfec' fuctions in 'dizzysNewInfec' package.

Examples

```
obj<-globSEIRNewInfec(typeSIMU="sto",nbVilles=3,N=c(1e7,1e6))
plot(obj)
```

Index