

# Package ‘epimdr2’

March 31, 2026

**Title** Functions and Data for “Epidemics: Models and Data in R (2nd Edition)”

**Version** 1.1-1

**Date** 2026-03-26

**Author** Ottar N. Bjornstad [aut, cre]

**Maintainer** Ottar N. Bjornstad <onb1@psu.edu>

**Description** Functions, data sets and shiny apps for “Epidemics: Models and Data in R (2nd edition)” by Ottar N. Bjornstad (2022, ISBN: 978-3-031-12055-8) <[doi:10.1007/978-3-031-12056-5](https://doi.org/10.1007/978-3-031-12056-5)>. The package contains functions to study the Susceptible-Exposed-Infected-Removed SEIR model, spatial and age-structured Susceptible-Infected-Removed SIR models; time-series SIR and chain-binomial stochastic models; catalytic disease models; coupled map lattice models of spatial transmission and network models for social spread of infection.

**Depends** R (>= 4.0.0), shiny, DT, deSolve, plotly, polyspline

**Imports** ggplot2

**Suggests** ade4, bbmle, fields, forecast, imputeTS, lme4, ncf, nleqslv, nlme, nlts, plotrix, pomp, rootSolve, Rwave, rworldmap, statnet, scatterplot3d

**License** GPL-3

**URL** <<https://github.com/objornstad/epimdr2>>

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.3.3

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2026-03-31 10:10:02 UTC

## Contents

barabasiAlbert . . . . .	3
black . . . . .	4

burnett . . . . .	5
ces . . . . .	5
chabaudi . . . . .	6
cholera . . . . .	7
coyne . . . . .	7
coyne2 . . . . .	8
cspring . . . . .	9
dalziel . . . . .	9
ebola . . . . .	10
euthamia . . . . .	11
ferrari . . . . .	11
filipendula . . . . .	12
fiv . . . . .	12
flowField . . . . .	13
flu . . . . .	16
gillespie . . . . .	17
gonnet . . . . .	17
gravity . . . . .	18
gypsomoth . . . . .	19
icelandflu . . . . .	19
jacobian . . . . .	20
leslie . . . . .	21
litter . . . . .	22
llik.cb . . . . .	22
llik.pc . . . . .	23
m4494 . . . . .	24
magono . . . . .	24
may.app . . . . .	25
meas . . . . .	26
nbmod . . . . .	26
nb spat.app . . . . .	27
nextgenR0 . . . . .	28
niamey . . . . .	29
niamey_daily . . . . .	30
orv.app . . . . .	30
pagiard . . . . .	31
paili . . . . .	31
palymes . . . . .	32
pameasle . . . . .	32
pdv . . . . .	33
pertcop . . . . .	33
peru . . . . .	34
plot.cm . . . . .	35
plot.netSir . . . . .	35
polymod . . . . .	36
rOfun . . . . .	36
rabbit . . . . .	37
rabies . . . . .	37

retrospec . . . . .	38
ringlattice . . . . .	39
ross.app . . . . .	40
seir.app . . . . .	40
seirmod . . . . .	41
seirmod2 . . . . .	41
seirs.app . . . . .	42
silene . . . . .	42
sim.cb . . . . .	43
sir.app . . . . .	44
sirAgemod . . . . .	44
sirChainmod . . . . .	45
sirmod . . . . .	46
sirNetmod . . . . .	46
sirSpatmod . . . . .	47
sirvmod . . . . .	48
sirwmod . . . . .	48
summary.cm . . . . .	49
summary.netSir . . . . .	50
tau . . . . .	50
tsir.app . . . . .	51
tsirLyap . . . . .	52
tsirLyap . . . . .	52
tsirSim . . . . .	53
tsirSim2 . . . . .	54
tsirSpat . . . . .	54
twostrain . . . . .	55
twostrain.app . . . . .	56
tydiphtheria . . . . .	56
tymeasles . . . . .	57
tyscarlet . . . . .	57
tywhooping . . . . .	58
us . . . . .	58
usflu . . . . .	59
variants . . . . .	60
waller . . . . .	60
wattsStrogatz . . . . .	61

**Index** **62**

---

barabasiAlbert	<i>Function to generate a Barabasi-Albert network</i>
----------------	---

---

**Description**

Function to generate a Barabasi-Albert network

**Usage**

```
barabasiAlbert(N, K)
```

**Arguments**

**N** the number of nodes  
**K** the number of neighbors to which each node is connected so degree = 2\*K

**Value**

An object of class CM (contact matrix)

**Examples**

```
cm3=barabasiAlbert(200, 4)
```

---

black	<i>Black's measles seroprevalence data.</i>
-------	---

---

**Description**

Seroprevalence-by-age-bracket for measles in prevaccination New Haven as studied by Black (1959).

**Usage**

```
black
```

**Format**

A data frame with 42 rows and 3 variables:

**age** age-bracket (in years)  
**mid** mid-point of age-bracket (in years)  
**n** number of tests  
**pos** number seropositive  
**neg** number seronegative  
**f** seroprevalence

**Source**

Black (1959) Measles antibodies in the population of New Haven, Connecticut. *Journal of Immunology* 83:74-83

---

burnett

*Burnett's Parasitoid-Host data.*

---

**Description**

Data is of 22 generations of greenhouse white flies (*Trialeurodes vaporariorum*) and its parasitoid, *Encarsia formosa*. Column names are self explanatory.

**Usage**

burnett

**Format**

A data frame with 22 rows and 7 variables:

**Generation**

**NumberofHostsExposed**

**NumberofHostsParasitized**

**NumberofHostsUnparasitized**

**NumberofParasiteEggsLaid**

**NumberofParasitesSearching**

**PercentageofHostsParasitized**

**Source**

Burnett, T. A. (1958) Model of host-parasite interaction Proceedings of the 10th International Congress, Entomology, 1958, 2, 679-686

---

ccs

*UK measles CCS data.*

---

**Description**

The fraction of weeks measles was absent from each of the 954 cities and towns of England and Wales between 1944 and 1965.

**Usage**

ccs

**Format**

A data frame with 954 rows and 14 variables:

**fade3** Average duration of fadeout (of at least 3 weeks of length)

**ext** Fraction of time when measles was absent

**size** Median population size

**fade** Average duration of fadeouts (of a week or longer)

**se3** Standard error fade3

**se** Standard error of fade

**n3** The number of fadeouts (of at least 3 weeks of length)

**n** The number of fadeout of a week or longer

**names** City/town name

**Source**

Bjornstad and Grenfell (2008) Hazards, spatial transmission and timing of outbreaks in epidemic metapopulations. *Environmental and Ecological Statistics* 15: 265-277. <doi:10.1007/s10651-007-0059-3>

---

chabaudi

*Daily measures of malaria infected mice.*

---

**Description**

Daily data on laboratory mice infected with various strains of *Plasmodium chabaudi*

**Usage**

chabaudi

**Format**

A data frame with 1300 rows and 11 variables:

**Line** line number

**Day** day of infection

**Box** Cage number

**Mouse** Mouse identifier

**Treatment** Plasmodium strain

**Ind2** Unique mouse identifier

**Weight** Mouse weight

**Glucose** Blood glucose level

**RBC** Red blood cell count

**Sample** Sample number

**Para** Parasite count

**Source**

Sylvie Huijben

---

cholera*Dacca cholera death data.*

---

**Description**

Monthly deaths from cholera in Dacca, East Bengal between 1891 and 1940.

**Usage**

cholera

**Format**

A data frame with 600 rows and 4 variables:

**Year** Year**Month** Month of the year**Dacca** Monthly cholera deaths**Population** Population size of district**Source**King, A.A., Ionides, E.L., Pascual, M. and Bouma, M. J. (2008) Inapparent infections and cholera dynamics. *Nature*, 454:877-880. <doi:10.1038/nature07084>

---

coyne*Edition 1 Gradient-function for Coyne et al's rabies model*

---

**Description**

Edition 1 Gradient-function for Coyne et al's rabies model

**Usage**

coyne(t, logx, parms)

**Arguments**

t Implicit argument for time

logx A vector with values for the log-states

parms A vector with parameter values for the dynamical system

**Value**

A list of gradients for the log system

**Examples**

```
require(deSolve)
times = seq(0, 50, by=1/520)
paras = c(gamma = 0.0397, b = 0.836, a = 1.34, sigma = 7.5,
alpha = 66.36, beta = 33.25, c = 0, rho = 0.8)
start = log(c(X=12.69/2, H1=0.1, H2=0.1, Y = 0.1, I = 0.1))
out = as.data.frame(ode(start, times, coyne, paras))
```

---

coyne2

---

*Edition 2 Gradient-function for Coyne et al's rabies model*


---

**Description**

Edition 2 Gradient-function for Coyne et al's rabies model

**Usage**

```
coyne2(t, logx, parms)
```

**Arguments**

t	Implicit argument for time
logx	A vector with values for the log-states
parms	A vector with parameter values for the dynamical system

**Value**

A list of gradients for the log system

**Examples**

```
require(deSolve)
times = seq(0, 50, by=1/520)
paras = c(gamma = 0.0397, b = 0.836, a = 1.34, sigma = 7.5,
alpha = 66.36, beta = 33.25, c = 0, rho = 0.8)
start = log(c(S=12.69/2, E1=0.1, E2=0.1, I = 0.1, R = 0.1))
out = as.data.frame(ode(start, times, coyne, paras))
```

---

cspring

*Colorado Springs network*

---

**Description**

Network and individual characteristics among 749 sex workers and clients in Colorado Springs as surveyed between 1988 and 1991

**Usage**

cspring

**Format**

A list of two items the first (\$nodes) is a frame with 749 rows and 6 variables, The second (\$cm) is a 749 x 749 relational matrix of presence/absence of sexual contacts among each pair of individuals.

**\$nodes\$id** individual identifier

**\$nodes\$gender** gender; 1 = female, 2 = male

**\$nodes\$sex.worker** sex worker status; 1 = yes, 0 = no

**\$nodes\$pimp** pimp status; 1 = yes, 0 = no

**\$nodes\$sex.work.client** client status; 1 = yes, 2 = no

**\$nodes\$type** node classifier; 1 = client, 2 = worker, 3 = both

**\$cm** the relational (contact) matrix among the individuals in the network.

**Source**

Woodhouse et al. (1994) Mapping a social network of heterosexuals at high risk for HIV Infection. AIDS 8:1331-1336. doi:10.1097/00002030-199409000-00018

Klovdahl et al. (1994) Social networks and infectious disease: The Colorado Springs study. Social Science and Medicine 38:79-88. <doi:10.1016/0277-9536(94)90302-6>

---

dalziel

*Measles incidence across 40 US cities*

---

**Description**

A dataset of Measles incidence across 40 US cities with relevant demographic data

**Usage**

dalziel

**Format**

A data frame with 44,720 rows and 10 variables:

**biweek** biweek of the year

**cases** incidence

**year** year

**loc** city name

**pop** population size

**rec** susceptible recruits

**country** country

**lon** city longitude

**lat** city latitude

**decimalYear** time counter

**Source**

Dalziel et al. 2016. Persistent chaos of measles epidemics in the prevaccination United States caused by a small change in seasonal transmission patterns. PLoS Computational Biology 2016: e1004655. <doi:10.1371/journal.pcbi.1004655>

---

ebola

*Sierra-Leone Ebola 2015 data.*

---

**Description**

The daily number of cases of ebola in Sierra Leone during the 2015 epidemic.

**Usage**

ebola

**Format**

A data frame with 103 rows and 4 variables:

**date** date

**day** day

**cum\_cases** cumulative incidence

**cases** incidence calculated by differencing the cumcases and setting negatives to zero.

**Source**

<https://stacks.cdc.gov/view/cdc/31408>

---

euthamia	<i>Euthamia graminifolia rust data.</i>
----------	---

---

**Description**

Data on a fungal pathogen of the aster *Euthamia graminifolia* collected by Jennifer Keslow.

**Usage**

euthamia

**Format**

A data frame with 360 rows and 8 variables:

**block** the block

**row** row

**plot** plot within block

**xloc** x coordinates

**yloc** y coordinate

**comp** plot composition

**water** treatment: dry or wet

**score** the rust score

---

ferrari	<i>Ferrari et al. 2005 outbreak data.</i>
---------	---

---

**Description**

The incidence aggregated by serial interval of a number of outbreaks studied by Ferrari et al. 2005.

**Usage**

ferrari

**Format**

A data frame with 15 rows and 7 variables:

**Eboladeaths00** Number of deaths from ebola during the 2000 Uganda outbreak

**Ebolacases00** Number of cases of ebola during the 2000 Uganda outbreak

**Ebolacases95** Number of cases of ebola during the 1995 DRC outbreak

**FMDfarms** Number of farms infected with FMD during the 2000-01 UK outbreak

**HogCholera** Number of cases of swine fever in pigs in the 1997-98 outbreak in the Netherlands

**SarsHk** Number of cases of SARS in Hong Kong during the 2003 outbreak

**SarsSing** Number of cases of SARS in Singapore during the 2003 outbreak

**Source**

Ferrari et al. (2005) Estimation and inference of R-0 of an infectious pathogen by a removal method. *Mathematical Biosciences* 198: 14-26. <doi:10.1016/j.mbs.2005.08.002>

---

filipendula

*Filipendula rust data.*

---

**Description**

Rust infection status of 162 populations of *Filipendula ulmaria* in a Swedish Island archipelago

**Usage**

filipendula

**Format**

A data frame with 162 rows and 4 variables:

**y94** infection status in 1994

**y95** infection status in 1995

**X** X coordinate

**Y** Y coordinate

**Source**

Smith et al. 2003. Epidemiological patterns at multiple spatial scales: an 11-year study of a *Triphragmium ulmariae* – *Filipendula ulmaria* metapopulation. *Journal of Ecology*, 91(5), pp.890-903. <doi:10.1046/j.1365-2745.2003.00811.x>

---

fiv

*FIV infection in cats.*

---

**Description**

Immunological measures on cats infected with different strains of FIV

**Usage**

fiv

**Format**

A data frame with 238 rows and 18 variables:

**Id** Individual identifier  
**CD4** CD4 cell count  
**CD8B** CD8B cell count  
**CD25** CD25 cell count  
**FAS\_L** FAS ligand  
**FAS** FAS  
**IFNg** Interferon gamma  
**IL\_10** Interleukin 10  
**IL\_12** Interleukin 12  
**IL\_4** Interleukin 4  
**lymphocyte** lymphocyte count  
**neutrophils** neutrophil count  
**TNF\_a** Tumor necrosis factor  
**provirus** provirus count  
**viremia** viremia  
**Day** day  
**No** unique identifier  
**Treatment** Experimental treatment

**Source**

Roy et al. 2009. Multivariate statistical analyses demonstrate unique host immune responses to single and dual lentiviral infection. PLoS one 4, e7359. <doi:10.1371/journal.pone.0007359>

---

flowField

*flowField*

---

**Description**

Plots the flow or velocity field for a one- or two-dimensional autonomous ODE system. This function was written Michael J Grayling as part of the archived phaseR package and re-published in the this 'epimdr2' package under original GPL-3 licence

**Usage**

```

flowField(
  deriv,
  xlim,
  ylim,
  parameters = NULL,
  system = "two.dim",
  points = 21,
  col = "gray",
  arrow.type = "equal",
  arrow.head = 0.05,
  frac = 1,
  add = TRUE,
  xlab = if (system == "two.dim") state.names[1] else "t",
  ylab = if (system == "two.dim") state.names[2] else state.names[1],
  ...
)

```

**Arguments**

deriv	A function computing the derivative at a point for the ODE system to be analysed. Discussion of the required format of these functions can be found in the help file for the function <code>ode</code> .
xlim	In the case of a two-dimensional system, this sets the limits of the first dependent variable in which gradient reflecting line segments should be plotted. In the case of a one-dimensional system, this sets the limits of the independent variable in which these line segments should be plotted. Should be a <a href="#">numeric vector</a> of <a href="#">length</a> two.
ylim	In the case of a two-dimensional system this sets the limits of the second dependent variable in which gradient reflecting line segments should be plotted. In the case of a one-dimensional system, this sets the limits of the dependent variable in which these line segments should be plotted. Should be a <a href="#">numeric vector</a> of <a href="#">length</a> two.
parameters	Parameters of the ODE system, to be passed to <code>deriv</code> . Supplied as a <a href="#">numeric vector</a> ; the order of the parameters can be found from the <code>deriv</code> file. Defaults to <code>NULL</code> .
system	Set to either <code>"one.dim"</code> or <code>"two.dim"</code> to indicate the type of system being analysed. Defaults to <code>"two.dim"</code> .
points	Sets the density of the line segments to be plotted; <code>points</code> segments will be plotted in the x and y directions. Fine tuning here, by shifting <code>points</code> up and down, allows for the creation of more aesthetically pleasing plots. Defaults to 11.
col	Sets the colour of the plotted line segments. Should be a <a href="#">character vector</a> of <a href="#">length</a> one. Will be reset accordingly if it is of the wrong <a href="#">length</a> . Defaults to <code>"gray"</code> .

arrow.type	Sets the type of line segments plotted. If set to "proportional" the <code>length</code> of the line segments reflects the magnitude of the derivative. If set to "equal" the line segments take equal lengths, simply reflecting the gradient of the derivative(s). Defaults to "equal".
arrow.head	Sets the length of the arrow heads. Passed to <code>arrows</code> . Defaults to <code>0.05</code> .
frac	Sets the fraction of the theoretical maximum length line segments can take without overlapping, that they can actually attain. In practice, <code>frac</code> can be set to greater than 1 without line segments overlapping. Fine tuning here assists the creation of aesthetically pleasing plots. Defaults to 1.
add	Logical. If TRUE, the flow field is added to an existing plot. If FALSE, a new plot is created. Defaults to TRUE.
xlab	Label for the x-axis of the resulting plot.
ylab	Label for the y-axis of the resulting plot.
...	Additional arguments to be passed to either <code>plot</code> or <code>arrows</code> .

### Value

Returns a `list` with the following components (the exact make up is dependent on the value of system):

add	As per input.
arrow.head	As per input.
arrow.type	As per input.
col	As per input, but with possible editing if a <code>character vector</code> of the wrong <code>length</code> was supplied.
deriv	As per input.
dx	A <code>numeric matrix</code> . In the case of a two-dimensional system, the values of the derivative of the first dependent derivative at all evaluated points.
dy	A <code>numeric matrix</code> . In the case of a two-dimensional system, the values of the derivative of the second dependent variable at all evaluated points. In the case of a one-dimensional system, the values of the derivative of the dependent variable at all evaluated points.
frac	As per input.
parameters	As per input.
points	As per input.
system	As per input.
x	A <code>numeric vector</code> . In the case of a two-dimensional system, the values of the first dependent variable at which the derivatives were computed. In the case of a one-dimensional system, the values of the independent variable at which the derivatives were computed.
xlab	As per input.
xlim	As per input.

<code>y</code>	A <a href="#">numeric vector</a> . In the case of a two-dimensional system, the values of the second dependent variable at which the derivatives were computed. In the case of a one-dimensional system, the values of the dependent variable at which the derivatives were computed.
<code>ylab</code>	As per input.
<code>ylim</code>	As per input.

**Author(s)**

Michael J Grayling

**See Also**

[arrows](#), [plot](#)

**Examples**

#See archived phaseR package for examples

---

<code>flu</code>	<i>Boarding school influenza data.</i>
------------------	--

---

**Description**

The daily number of children confined to bed in a boarding school in North England during an outbreak in 1978 of the reemerging A/H1N1 strain. The school had 763 boys of which 512 boys were confined to bed sometime during the outbreak.

**Usage**

`flu`

**Format**

A data frame with 14 rows and 2 variables:

**day** day since beginning of outbreak

**cases** number of sick children

**Source**

Anonymous (1978) EPIDEMIOLOGY: Influenza in a boarding school. British Medical Journal, 4 March 1978 p.587.

---

gillespie                      *Gillespie exact algorithm*

---

**Description**

Function simulating a dynamical system using the Gillespie exact algorithm

**Usage**

```
gillespie(rateqs, eventmatrix, parameters, initialvals, numevents)
```

**Arguments**

rateqs	a list with rate equations
eventmatrix	a matrix of changes in state variables associated with each event
parameters	a vector of parameter values
initialvals	a vector of initial values for the states
numevents	number of events to be simulated

**Value**

A data frame with simulated time series

**Examples**

```
rlist=c(quote(mu * (S+I+R)), quote(mu * S), quote(beta * S * I / (S+I+R)),
  quote(mu * I), quote(gamma * I), quote(mu*R))
emat=matrix(c(1,0,0,-1,0,0,-1,1,0,0,-1,0,0,-1,1,0,0,-1),ncol=3, byrow=TRUE)
paras = c(mu = 1, beta = 1000, gamma = 365/20)
inits = c(S=100, I=2, R=0)
sim=gillespie(rlist, emat, paras, inits, 100)
```

---

gonnet                      *De et al. 2004 gonorrhoea contact matrix*

---

**Description**

The directed contact network from De et al. (2004) contact-tracing of the spread of gonorrhoea across asexual network in Alberta, Canada

**Usage**

```
gonnet
```

**Format**

A matrix with 89 rows and 89 columns:

**gonet** a matrix of directional contacts of disease spread

**Source**

De et al (2004). Sexual network analysis of a gonorrhea outbreak. Sexually transmitted infections 80: 280-285. <doi:10.1136/sti.2003.007187>

---

gravity	<i>A function to calculate the matrix of gravity coupling based on distance and population size</i>
---------	---

---

**Description**

A function to calculate the matrix of gravity coupling based on distance and population size

**Usage**

```
gravity(tau1, tau2, rho, pop, distance)
```

**Arguments**

tau1	recipient exponent
tau2	donor exponent
rho	distance exponent
pop	a vector of population sizes
distance	a matrix of distances

**Value**

A matrix of gravity coupling

**See Also**

[sirSpatmod](#)

**Examples**

```
require(ncf)
data(usflu)
usdist = gcdist(usflu$Longitude, usflu$Latitude)
G = gravity(0.3, 0.6, 3, usflu$Pop, usdist)
```

---

gypsymoth

*Defoliated by gypsy moth each in northeast US 1975-2002.*

---

### Description

A list containing the fraction of forest defoliated by the gypsy moth in 20km x 20km pixels across northeast US in each year between 1975 and 2002.

### Usage

gypsymoth

### Format

A list with two matrices each with 1086 rows:

**xy** A matrix with two columns representing UTM coordinates

**defoliation** A matrix with 28 columns representing pixel-wise defoliation between 1976 and 2002

### Source

Bjornstad, O. N., Robinet, C., & Liebhold, A. M. (2010). Geographic variation in North American gypsy moth cycles: subharmonics, generalist predators, and spatial coupling. *Ecology*, 91(1), 106-118. <doi:10.1890/08-1246.1>

---

icelandflu

*Monthly incidence of influenza-like illness in Iceland between 1980 and 2009.*

---

### Description

A dataset containing the monthly ILI incidence in Iceland between 1980 and 2009.

### Usage

icelandflu

### Format

A data frame with 360 rows and 3 variables:

**month** the month

**year** the year

**ili** ILI incidence

**Source**

Bjornstad ON, Viboud C. Timing and periodicity of influenza epidemics. Proceedings of the National Academy of Sciences. 2016 Nov 15;113(46):12899-901. <doi:10.1073/pnas.1616052113>

---

jacobian

*A Jacobian matrix calculator*


---

**Description**

A general-purpose function to construct and evaluate Jacobian matrices

**Usage**

```
jacobian(states, elist, parameters, pts)
```

**Arguments**

states	a vector naming all state variables
elist	a list that contains equations (as quotes) for all state variables
parameters	a labeled vector of parameters
pts	a labeled vector of the point in the phase plane in which to evaluate the Jacobian (often the endemic or disease-free equilibrium if working in mathematical epidemiology)

**Value**

The Jacobian matrix

**Examples**

```
#The SEIR model
states=c("S", "E", "I", "R")
elist=c(dS = quote(mu * (N - S) - beta * S * I / N),
  dE = quote(beta * S * I / N - (mu + sigma) * E),
  dI = quote(sigma * E - (mu + gamma+alpha) * I),
  dR = quote(gamma * I - mu * R))
paras = c(mu = 1/50, N = 1, beta = 1000,
  sigma = 365/8, gamma = 365/5, alpha=0)
deq=list(S = 1, E = 0, I = 0, R = 0)
jacobian(states=states, elist=elist, parameters=paras, pts=deq)
```

---

leslie	<i>A function to calculate asymptotic growth, sensitivity and elasticity for age-structured populations</i>
--------	---

---

**Description**

A function to calculate asymptotic growth, sensitivity and elasticity for age-structured populations

**Usage**

```
leslie(L)
```

**Arguments**

L                    the Leslie matrix

**Value**

A list consisting of the following components:

lambda	the dominant eigen value of the Leslie matrix.
right.eigenvector	the dominant right eigen vector of the Leslie matrix, proportional to the stable age-distribution.
left.eigenvector	the dominant left eigen vector of the Leslie matrix representing the age-specific reproductive values.
elasticity	the elasticities.
sensitivity	the sensitivities.

**References**

Caswell, H. 2001. Matrix Population Models: Construction, Analysis, and Interpretation. 2nd edn Sinauer Associates Inc., Sunderland, MA,

**Examples**

```
fa<-c(0, 0.5, 1.2)
sa<-c(0.8, 0.8, 0)
L<-matrix(0, nrow=3, ncol=3)
#inserting fa vector in first row
L[1,]<-fa
#inserting sa in the subdiagonal:
L[row(L)==col(L)+1] <-sa[1:2]
leslie(L)
```

---

litter	<i>Bordetella bronchiseptica</i> in rabbit kittens.
--------	---

---

**Description**

Data on *Bordetella bronchiseptica* in rabbit kittens in a breeding facility.

**Usage**

litter

**Format**

A data frame with 494 rows and 8 variables:

**Facility** breeding facility

**sick** infection status

**Date** date sampled

**Animal.code** animal identifier

**msick** dams infection status

**Litter** litter identifier

**CFU** bacterial count

**Description** unique litter identifier

**Source**

Long et al (2010) Identifying the Age Cohort Responsible for Transmission in a Natural Outbreak of *Bordetella bronchiseptica*. PLoS Pathogens 6(12): e1001224. <doi:10.1371/journal.ppat.1001224>

---

llik.cb	<i>Negative log-likelihood function for the chain-binomial model</i>
---------	--

---

**Description**

Negative log-likelihood function for the chain-binomial model

**Usage**

llik.cb(S0, beta, I)

**Arguments**

S0 a scalar with value for S0

beta a scalar with value for beta

I a vector incidence aggregated at serial interval

**Value**

the negative log-likelihood for the model

**Examples**

```
twoweek=rep(1:15, each=2)
niamey_cases1=sapply(split(niamey$cases_1[1:30], twoweek), sum)
llik.cb(S0=6500, beta=23, I=niamey_cases1)
```

---

llik.pc	<i>Function to estimate parameters for the piecewise-constant catalytic model</i>
---------	---

---

**Description**

This function uses binomial likelihoods to estimate the piecewise-constant FoI model from age-incidence data

**Usage**

```
llik.pc(par, age, num, denom, up)
```

**Arguments**

par	a vector with initial guesses
age	a vector with the ages
num	a vector with number infected by age
denom	a vector with number tested by age
up	a vector with upper age-bracket cut-offs

**Value**

The negative log-likelihood for a candidate piecewise constant catalytic model

**Examples**

```
x=c(1,4,8,12,18,24)
para=rep(0.1,length(x)+1)
optim(par=log(para),fn=llik.pc, age=rabbit$a, num=rabbit$inf, denom=rabbit$n, up=x)
```

m4494

*Measles in England and Wales 1944-1994.***Description**

The weekly reported cases of measles in each of the 354 cities and villages between 1944 and 1994.

**Usage**

m4494

**Format**

A list with four items:

**measles** A matrix with 354 rows and 2661 columns. Rows represents community and columns represent week.

**ps** A matrix with 354 rows and 41 columns. Rows represents community and columns represent population size for each of the years in the data set.

**longlat** A matrix with 354 rows and two columns. Rows represents community and columns longitude and latitude.

**year** A vector of length 354 that represents time for each week as yearly decimals.

**coverage** A vector of length 2661 that represents reported annual vaccine coverage for each week.

**Source**

Grenfell, B.T., Bjornstad, O.N., & Kappey, J. 2001. Travelling waves and spatial hierarchies in measles epidemics. *Nature* 414: 716-723. <doi:10.1038/414716a>

Lau, M.S.Y., Becker, A.D, Korevaar, H.M., Caudron, Q., Shaw, D.J., Metcalf, C.J.E., Bjornstad, O.N. and Grenfell, B.T. 2020. A competing-risks model explains hierarchical spatial coupling of measles epidemics en route to national elimination. *Nature Ecology & Evolution*. <doi:10.1038/s41559-020-1186-6>

magono

*Massachusetts gonorrhoea data.***Description**

Weekly cases of gonorrhoea in Massachusetts between 2006 and 2015.

**Usage**

magono

### Format

A data frame with 422 rows and 4 variables:

**number** Weekly case reports

**year** Year

**week** Week of the year

**time** Time in fractions of year

### Source

<https://www.tycho.pitt.edu>

---

may.app

*Launch a shiny-app simulating May's Parasitoid-host Model model*

---

### Description

Launch a shiny-app simulating May's Parasitoid-host Model model

### Usage

```
may.app
```

### Format

An object of class shiny.appobj of length 5.

### Details

Launch app for details

### Examples

```
if(interactive()){may.app}
```

meas

*Bi-weekly measles incidence in London from 1944-65.*

---

**Description**

A dataset containing the biweekly incidence of measles in London from 1944 to 1965

**Usage**

meas

**Format**

A data frame with 546 rows and 5 variables:

**year** year

**week** week of the year

**time** time

**London** incidence

**B** Biweekly births

**Details**

Birth numbers are annual, so in the data set, this number is evenly distributed across the 26 bi-weeks of each year.

**Source**

Bjornstad et al. (2002) Endemic and epidemic dynamics of measles: Estimating transmission rates and their scaling using a time series SIR model. *Ecological Monographs* 72: 169-184. <doi:10.2307/3100023>

---

nbmod*The Nicholson-Bailey model*

---

**Description**

Function to simulate the Nicholson-Bailey Parasitoid-host model

**Usage**

nbmod(R, a, T = 100, H0 = 10, P0 = 1)

**Arguments**

R	the host reproductive rate
a	the parasitoid search efficiency
T	the length of simulation (number of time-steps)
H0	initial host numbers
P0	initial parasitoid numbers

**Value**

A list of simulated Host and Parasitoid numbers

**Examples**

```
sim= nbmod(R=1.1,a=0.1)
```

---

nbspat.app	<i>Launch a shiny-app simulating the spatially-extended Nicholson-Bailey parasitoid model</i>
------------	---

---

**Description**

Launch a shiny-app simulating the spatially-extended Nicholson-Bailey parasitoid model

**Usage**

```
nbspat.app
```

**Format**

An object of class shiny.appobj of length 5.

**Details**

Launch app for details

**Examples**

```
if(interactive()){nbspat.app}
```

---

 nextgenR0

*Next generation matrix R0 calculator*


---

**Description**

Calculates R0 for arbitrarily complex compartmental flows using the method of Diekmann et al. (1990).

**Usage**

```
nextgenR0(Istates, Flist, Vlist, parameters, dfe)
```

**Arguments**

Istates	a vector naming all Infected classes
Flist	a list that contains equations (as quotes) for completely new infections entering each infected compartment for each class
Vlist	a list that contains the equations (as quotes) for losses out of each infected compartment minus the equations (as quotes) for all gains into each infected compartment that does not represent new infections but transfers among infectious classes
parameters	a labeled vector of parameters
dfe	a labeled vector of all states at the disease-free equilibrium

**Value**

The next generation matrix estimate of R0

**Source**

Diekmann, O., Heesterbeek, J. A. P. and Metz, J. A. J. 1990. On the Definition and the Computation of the Basic Reproduction Ratio R0 in Models for Infectious-Diseases in Heterogeneous Populations. *Journal of Mathematical Biology* 28: 365-382.

**Examples**

```
#The SEIR model
#Infected classes are $E$ and $I$
istates=c("E", "I")
flist=c(dEdt=quote(beta * S * I / N), dIdt=quote(0))
Vm1=quote(mu * E + sigma * E)
Vm2=quote(mu * I + alpha * I + gamma * I)
Vp1=0
Vp2=quote(sigma * E)
V1=substitute(a-b, list(a=Vm1, b=Vp1))
V2=substitute(a-b, list(a=Vm2, b=Vp2))
vlist = c(V1,V2)
```

```
para = list(mu = 0, alpha = 0, beta = 5, gamma = .8, sigma = 1.2, N = 1)
df = list(S = 1, E = 0, I = 0, R = 0)
nextgenR0(Istates=istates, Flist=flist, Vlist=vlist, parameters=para, dfe=df)
```

---

niamey

*Weekly measles incidence from 2003-04 in Niamey, Niger.*


---

### Description

A dataset containing the weekly incidence of measles in Niamey, Niger during the 2003-04 outbreak

### Usage

```
niamey
```

### Format

A data frame with 31 rows and 13 variables:

**absweek** week since beginning of outbreak

**week** week of the year

**tot\_cases** weekly incidence for the whole city

**tot\_mort** weekly deaths for the whole city

**lethality** weekly case fatality rate

**tot\_attack** weekly attack rates for the whole city

**cases\_1** weekly incidence for district 1

**attack\_1** weekly attack rates for district 1

**cases\_2** weekly incidence for district 2

**attack\_2** weekly attack rates for district 2

**cases\_3** weekly incidence for district 3

**attack\_3** weekly attack rates for district 3

**cum\_cases** weekly cumulative incidence for the whole city

### Source

Grais et al (2008) Time is of the essence: exploring a measles outbreak response vaccination in Niamey, Niger. *Journal of the Royal Society Interface* 5: 67-74. <doi:10.1098/rsif.2007.1038>

---

niamey_daily	<i>Day of appearance of each measles case from 2003-04 outbreak in Niamey, Niger.</i>
--------------	---

---

**Description**

A dataset containing the day of appearance of each measles case in Niamey, Niger during the 2003-04 outbreak.

**Usage**

```
niamey_daily
```

**Format**

A data frame with 10,937 rows and 1 variables:

**day** the day of appearance of each case since day of outbreak

**Source**

Grais et al. (2008) Time is of the essence: exploring a measles outbreak response vaccination in Niamey, Niger. *Journal of the Royal Society Interface* 5: 67-74. <doi:10.1098/rsif.2007.1038>

---

orv.app	<i>Launch a shiny-app to study outbreak-response vaccination campaigns</i>
---------	--

---

**Description**

Launch a shiny-app to study outbreak-response vaccination campaigns

**Usage**

```
orv.app
```

**Format**

An object of class shiny.appobj of length 5.

**Details**

Launch app for details

**Examples**

```
if(interactive()){orv.app}
```

---

pagiard	<i>Weekly incidence of giardia in Pennsylvania between 2006 and 2014.</i>
---------	---

---

**Description**

A dataset containing the weekly incidence of giardia in Pennsylvania between 2006 and 2014.

**Usage**

pagiard

**Format**

A data frame with 448 rows and 3 variables:

**PENNSYLVANIA** weekly incidence

**YEAR** the year

**WEEK** the week

**Source**

<https://www.tycho.pitt.edu>

---

paili	<i>Weekly deaths from Influenza-like illness in Pennsylvania between 1972 and 1998.</i>
-------	---

---

**Description**

A dataset containing the weekly ILI related deaths in Pennsylvania between 1972 and 1998.

**Usage**

paili

**Format**

A data frame with 1404 rows and 3 variables:

**PENNSYLVANIA** weekly deaths

**YEAR** the year

**WEEK** the week

**Source**

<https://www.tycho.pitt.edu>

---

palymes	<i>Weekly incidence of Lymes disease in Pennsylvania between 2006 and 2014.</i>
---------	---

---

**Description**

A dataset containing the weekly incidence of Lymes disease in Pennsylvania between 2006 and 2014.

**Usage**

palymes

**Format**

A data frame with 448 rows and 3 variables:

**PENNSYLVANIA** weekly incidence

**YEAR** the year

**WEEK** the week

**Source**

<https://www.tycho.pitt.edu>

---

pameasle	<i>Weekly incidence of measles in Pennsylvania between 1928 and 1969.</i>
----------	---

---

**Description**

A dataset containing the weekly incidence of measles in Pennsylvania between 2006 and 2014.

**Usage**

pameasle

**Format**

A data frame with 448 rows and 3 variables:

**PENNSYLVANIA** weekly incidence

**YEAR** the year

**WEEK** the week

**Source**

<https://www.tycho.pitt.edu>

---

 pdv

---

*Cumulative death count of harbor seals from CDV.*


---

**Description**

The cumulative count of dead seals washed ashore across 25 Northern European areas during the 2002 epidemic starting in May and running through the end of the year.

**Usage**

pdv

**Format**

A list with three items:

**coord** A data frame with 3 columns and 25 rows. Column location represents location name, latitude is latitude and longitude is longitude

**ts** A data frame with 26 columns and 269 rows. The first column is the day since beginning of outbreak and the next 25 columns are cumulative count of stranded seal carcasses

**fs** A 25-by-25 matrix representing the seaway friction distance among the haulouts.

**Source**

Harding, K. C., Härkönen, T. and Caswell, H. (2002), The 2002 European seal plague: epidemiology and population consequences. *Ecology Letters*, 5: 727-732. <doi:10.1046/j.1461-0248.2002.00390.x>

---

 pertcop

---

*Weekly whooping cough incidence from 1900-1937 in Copenhagen, Denmark.*


---

**Description**

A dataset containing the weekly incidence of whooping cough from Copenhagen, Denmark between January 1900 and December 1937

**Usage**

pertcop

**Format**

A data frame with 1982 rows and 9 variables:

**date** date

**births** births

**day** day of month

**month** month of year

**year** year

**cases** weekly incidence

**deaths** weekly deaths

**popsize** weekly population size interpolated from census data

**Source**

Lavine et al. 2013. Immune boosting explains regime- shifts in prevaccine-era pertussis dynamics. PLoS ONE, 8(8):e72086. <doi:10.1371/journal.pone.0072086>

---

peru

*Rubella in Peru data.*

---

**Description**

Rubella incidence by age as studied by Metcalf et al (2011).

**Usage**

peru

**Format**

A data frame with 95 rows and 2 variables:

**age** end of age-bracket (in years)

**cumulative** cumulative number of rubella cases

**incidence** number of rubella cases

**n** total cases

**Source**

Metcalf et al (2011) Rubella metapopulation dynamics and importance of spatial coupling to the risk of congenital rubella syndrome in Peru. Journal of the Royal Society Interface 8: 369-376. <doi:10.1371/journal.pone.0072086>

---

plot.cm                      *Function to plot an object of class CM*

---

**Description**

Function to plot an object of class CM

**Usage**

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

**Arguments**

x	an object of class cm
...	other arguments

**Value**

A plot of the contact matrix

**Examples**

```
cm=ringlattice(N=20,K=4)  
plot(cm)
```

---

plot.netSir                      *Function to plot a netSir object*

---

**Description**

Function to plot a netSir object

**Usage**

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

**Arguments**

x	an object of class netSir
...	other arguments

**Value**

A network plot of a netSir object

**See Also**

[sirNetmod](#)

---

polymod

*POLYMOD contact-rate data by Age.*

---

**Description**

Age-specific contact rates from the diary study by Mossong et al. 2008.

**Usage**

polymod

**Format**

A data frame with 900 rows and 3 variables:

**contactor** end of age-bracket (in years) of contactor group

**contactee** end of age-bracket (in years) of contactee group

**contact.rate** average contact rate

**Source**

Mossong et al. 2008 Social contacts and mixing patterns relevant to the spread of infectious diseases PLoS Med, Public Library of Science 5:e74. <doi:10.1371/journal.pmed.0050074>.

---

r0fun

*Function to calculate R0 from a contact matrix*

---

**Description**

Function to calculate R0 from a contact matrix

**Usage**

r0fun(CM, tau, gamma)

**Arguments**

CM an object of class CM

tau = probability of infection across an edge

gamma = probability of removal per time step

**Value**

the R0

**Examples**

```
cm1=barabasiAlbert(N=200,K=2)
r0fun(cm1, 0.3, 0.1)
```

---

rabbit	<i>Rabbit Bordetella bronchiseptica data.</i>
--------	---

---

**Description**

Rabbits infected by *B. bronchiseptica* by age as studied by Long et al (2010).

**Usage**

rabbit

**Format**

A data frame with 42 rows and 3 variables:

- a** end of age-bracket (in months)
- n** number of rabbits tested
- inf** number of rabbits infected with the bacterium

**Source**

Long et al (2010) Identifying the Age Cohort Responsible for Transmission in a Natural Outbreak of *Bordetella bronchiseptica*. PLoS Pathogens 6(12): e1001224. <doi:10.1371/journal.ppat.1001224>

---

rabies	<i>Raccoon rabies data.</i>
--------	-----------------------------

---

**Description**

Data is the average monthly number of reported cases of rabid raccoons across all counties within each of 11 east coast US states the time line is from the first reported case in each state (starting in late 1970s for West Virginia).

**Usage**

rabies

**Format**

A data frame with 208 rows and 12 variables:

**Month** Month since rabies appearance in the state

**CT** Connecticut

**DE** Delaware

**MD** Maryland

**MA** Massachusetts

**NJ** New Jersey

**NY** New York

**NC** North Carolina

**PA** Pennsylvania

**RI** Rhode Island

**VA** Virginia

**WV** West Virginia

**Source**

Childs et al. 2000. Predicting the local dynamics of epizootic rabies among raccoons in the United States Proceedings of the National Academy of Sciences 97:13666-13671. <doi:10.1073/pnas.240326697>

---

retrospec

*Function to predict efficacy of outbreak-response vaccination campaign*

---

**Description**

Function to predict efficacy of outbreak-response vaccination campaign

**Usage**

```
retrospec(  
  R,  
  day,  
  vaccine_efficiency,  
  target_vaccination,  
  intervention_length,  
  mtime,  
  LP = 7,  
  IP = 7,  
  N = 10000  
)
```

**Arguments**

R	reproductive ratio
day	first day of ORV campaign
vaccine_efficacy	Vaccine efficacy
target_vaccination	fraction of population vaccinated during ORV campaign
intervention_length	duration of ORV campaign
mtime	length of simulation
LP	length of latent period
IP	length of infectious period
N	initial susceptible population size

**Value**

A list of gradients

**Examples**

```
red1=retrospec(R=1.8, 161, vaccine_efficacy=0.85, target_vaccination=0.5,
  intervention_length=10, mtime=250, LP=8, IP=5, N=16000)
1-red1$redn
```

---

ringlattice

*Function to generate a ring lattice*

---

**Description**

Function to generate a ring lattice

**Usage**

```
ringlattice(N, K)
```

**Arguments**

N	the number of nodes
K	the number of neighbors to which each node is connected so degree = 2xK

**Value**

An object of class CM (contact matrix)

**Examples**

```
cm=ringlattice(N=20,K=4)
```

---

`ross.app`*Launch a shiny-app simulating a Ross-Macdonald model*

---

**Description**

Launch a shiny-app simulating a Ross-Macdonald model

**Usage**

```
ross.app
```

**Format**

An object of class `shiny.appobj` of length 5.

**Details**

Launch app for details.

**Examples**

```
if(interactive()){ross.app}
```

---

`seir.app`*Launch a shiny-app simulating the seasonal SEIR model*

---

**Description**

Launch a shiny-app simulating the seasonal SEIR model

**Usage**

```
seir.app
```

**Format**

An object of class `shiny.appobj` of length 5.

**Details**

Launch app for details

**Examples**

```
if(interactive()){seir.app}
```

---

seirmod                      *Gradient-function for the SEIR model*

---

**Description**

Gradient-function for the SEIR model

**Usage**

```
seirmod(t, y, parameters)
```

**Arguments**

t	Implicit argument for time
y	A vector with initial values for the states
parameters	A vector with parameter values for the SEIR system

**Value**

A list of gradients

**Examples**

```
require(deSolve)
times = seq(0, 10, by=1/120)
paras = c(mu = 1/50, N = 1, beta = 1000, sigma = 365/8, gamma = 365/5)
start = c(S=0.06, E=0, I=0.001, R = 0.939)
out=ode(start, times, seirmod, paras)
```

---

seirmod2                      *Gradient-function for the forced SEIR model*

---

**Description**

Gradient-function for the forced SEIR model

**Usage**

```
seirmod2(t, y, parameters)
```

**Arguments**

t	Implicit argument for time
y	A vector with initial values for the states
parameters	A vector with parameter values for the SIR system

**Value**

A list of gradients

**Examples**

```
require(deSolve)
times = seq(0, 10, by=1/120)
paras = c(mu = 1/50, N = 1, beta0 = 1000, beta1 = 0.2, sigma = 365/8, gamma = 365/5)
start = c(S=0.06, E=0, I=0.001, R = 0.939)
out=ode(start, times, seirmod2, paras)
```

---

seirs.app

*Launch a shiny-app simulating the SEIRS model*

---

**Description**

Launch a shiny-app simulating the SEIRS model

**Usage**

seirs.app

**Format**

An object of class shiny.appobj of length 5.

**Details**

Launch app for details

**Examples**

```
if(interactive()){seirs.app}
```

---

silene

*Antler smut on wild campion.*

---

**Description**

Data on a fungal pathogen of the wild campion collected by Janis Antonovics.

**Usage**

silene

**Format**

A data frame with 876 rows and 5 variables:

**X** road segment number

**lat** latitude

**long** longitude

**hmean** number of healthy plants

**dmean** number of diseased plants

**Source**

Antonovics, J. 2004. Long-term study of a plant-pathogen metapopulation. In: Hanski, Ilkka, and Oscar E. Gaggiotti. Ecology, genetics, and evolution of metapopulations. Academic Press.

---

sim.cb

*Function to simulate the chain-binomial model*

---

**Description**

Function to simulate the chain-binomial model

**Usage**

```
sim.cb(S0, beta)
```

**Arguments**

S0            a scalar with value for S0

beta           a scalar with value for beta

**Value**

A data-frame with time series of susceptibles and infected

**Examples**

```
sim=sim.cb(S0=6500, beta=23)
```

---

 sir.app

*Launch a shiny-app simulating the SIR model*


---

**Description**

Launch a shiny-app simulating the SIR model

**Usage**

```
sir.app
```

**Format**

An object of class shiny.appobj of length 5.

**Details**

Launch app for details

**Examples**

```
if(interactive()){sir.app}
```

---

 sirAgemod

*Gradient-function for the age-structured SIR model with possibly heterogeneous mixing*


---

**Description**

Gradient-function for the age-structured SIR model with possibly heterogeneous mixing

**Usage**

```
sirAgemod(t, logx, parameters)
```

**Arguments**

t	Implicit argument for time
logx	A vector with initial values for the log-states
parameters	A named list with parameter values for the age-structured SIR system. N is population size, gamma is recovery rate, mu is birth/death rate, beta is transmission rate, W is the normalized contact matrix, v is vector of age-class specific vaccination rates and r is class-specific aging rates (since age brackets may differ in width).

**Value**

A list of gradients

**Examples**

```
ra=rep(1,4)
n=length(ra)
W=matrix(1, ncol=4, nrow=4)
paras =list(N=1, gamma=365/14, mu=0.02, beta=500, W=W,v=rep(0,4), r=ra)
xstart=log(c(S=rep(0.099/n,n), I=rep(0.001/n,n), R=rep(0.9/n,n)))
times=seq(0,10,by=14/365)
out=as.data.frame(ode(xstart, times, sirAgedmod, paras))
```

---

 sirChainmod

*Gradient-function for the chain-SIR model*


---

**Description**

Gradient-function for the chain-SIR model

**Usage**

```
sirChainmod(t, logx, parameters)
```

**Arguments**

t	Implicit argument for time
logx	A vector with values for the log-states
parameters	A vector with parameter values for the chain-SIR system

**Value**

A list of gradients

**Examples**

```
require(deSolve)
times = seq(0, 10, by=1/52)
paras = c(mu = 1/75, N = 1, beta = 625, gamma = 365/14, u=5)
xstart2 = log(c(S=.06, I=c(0.001, rep(0.0001, paras["u"]-1)), R = 0.0001))
out = as.data.frame(ode(xstart2, times, sirChainmod, paras))
```

---

sirmod *Gradient-function for the SIR model*

---

**Description**

Gradient-function for the SIR model

**Usage**

```
sirmod(t, y, parameters)
```

**Arguments**

t	Implicit argument for time
y	A vector with initial values for the states
parameters	A vector with parameter values for the SIR system

**Value**

A list of gradients

**Examples**

```
require(deSolve)
times = seq(0, 26, by=1/10)
paras = c(mu = 0, N = 1, beta = 2, gamma = 1/2)
start = c(S=0.999, I=0.001, R = 0)
out=ode(start, times, sirmod, paras)
```

---

sirNetmod *Function to simulate an epidemic on a network*

---

**Description**

Function to simulate a stochastic (discrete time) Reed-Frost SIR model on a social network

**Usage**

```
sirNetmod(CM, tau, gamma)
```

**Arguments**

CM	a contact matrix
tau	the transmission probability
gamma	the recovery probability

**Value**

An object of class netSir with infectious status for each node through time

**Examples**

```
cm1=barabasiAlbert(N=200,K=2)
sim1=sirNetmod(cm1,.3,0.1)
summary(sim1)
plot(sim1)
```

---

 sirSpatmod

*Gradient function for a spatially-extended SIR model*


---

**Description**

Gradient function for a spatially-extended SIR model given some spatial topology

**Usage**

```
sirSpatmod(t, y, parameters)
```

**Arguments**

t	Implicit argument for time
y	A vector of length $L*3$ with initial values for the states. The first $1:L$ represents initial S's, $(L+1):2*L$ are initial I's and the last $(2*L+1):3*L$ are initial R's
parameters	A vector with parameter values for the spatial SIR system

**Value**

A list of gradients

**Examples**

```
require(deSolve)
require(ncf)
data(usflu)
usdist = gcdist(usflu$Longitude, usflu$Latitude)
G = gravity(0.3, 0.6, 3, usflu$Pop, usdist)
gamma = 1/3.5
R0 = 1.8
beta = R0 * gamma/usflu$Pop
m = 1 / 1000 / sum(usflu$Pop)
parms = list(beta = beta, m = m, gamma = gamma, G = G)
S = usflu$Pop
R = I = rep(0, length(usflu$Pop))
I[31] = 1
inits = c(S = S, I = I, R = R)
```

```
times = 0:200
out = ode(inits, times, sirSpatmod, parms)
L=length(usflu$Pop)
matplot(out[, 50+(1:L)], type = "l", ylab = "Prevalence", xlab = "Day")
```

---

sirvmod	<i>Gradient-function for the SIR model with outbreak-response vaccination</i>
---------	---

---

**Description**

Gradient-function for the SIR model with outbreak-response vaccination

**Usage**

```
sirvmod(t, x, parms)
```

**Arguments**

t	Implicit argument for time
x	A vector with values for the states
parms	A vector with parameter values for the SIR system

**Value**

A list of gradients

**See Also**

[retrospec](#)

---

sirwmod	<i>Gradient-function for the SIRWS model</i>
---------	--

---

**Description**

Gradient-function for the SIRWS model

**Usage**

```
sirwmod(t, logy, parameters)
```

**Arguments**

t	Implicit argument for time
logy	A vector with values for the log(states)
parameters	A vector with parameter values for the SIRWS system

**Value**

A list of gradients (in log-coordinates)

**Examples**

```
require(deSolve)
times = seq(0, 26, by=1/10)
paras = c(mu = 1/70, p=0.2, N = 1, beta = 200, omega = 1/10, gamma = 17, kappa=30)
start = log(c(S=0.06, I=0.01, R=0.92, W = 0.01))
out = as.data.frame(ode(start, times, sirwmod, paras))
```

---

summary.cm

*Function to calculate the degree distribution for an object of class CM*

---

**Description**

Function to calculate the degree distribution for an object of class CM

**Usage**

```
## S3 method for class 'cm'
summary(object, plot = FALSE, ...)
```

**Arguments**

object	an object of class cm
plot	if TRUE a bar plot of the degree distribution is produced
...	other arguments

**Value**

A plot of the contract matrix

**Examples**

```
cm=wattsStrogatz(N=20, K=4, Prw=.3)
summary(cm)
```

---

summary.netSir      *Function to summarize a netSir object*

---

### Description

Function to summarize a netSir object

### Usage

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

### Arguments

object      an object of class netSir  
 ...      other arguments

### Value

A data-frame with the time series of susceptible, infected and recovered individuals

### See Also

[sirNetmod](#)

---

tau      *Gillespie tau-leap algorithm*

---

### Description

Function simulating a dynamical system using the Gillespie tau-leap approximation

### Usage

```
tau(rateqs, eventmatrix, parameters, initialvals, deltaT, endT)
```

### Arguments

rateqs      a list with rate equations  
 eventmatrix      a matrix of changes in state variables associated with each event  
 parameters      a vector of parameter values  
 initialvals      a vector of initial values for the states  
 deltaT      the tau-leap time interval  
 endT      the time length of simulation

**Value**

A data frame with simulated time series

**Examples**

```
rlist2=c(quote(mu * (S+E+I+R)), quote(mu * S), quote(beta * S * I/(S+E+I+R)),
  quote(mu*E), quote(sigma * E), quote(mu * I), quote(gamma * I), quote(mu*R))
emat2=matrix(c(1,0,0,0,-1,0,0,0,-1,1,0,0,0,-1,0,0,0,-1,1,0,0,0,-1,0,0,0,-1,1,0,0,0,-1,0,0,0,-1,1,0,0,0,-1),
  ncol=4, byrow=TRUE)
paras = c(mu = 1, beta = 1000, sigma = 365/8, gamma = 365/5)
inits = c(S=999, E=0, I=1, R = 0)
sim2=tau(rlist2, emat2, paras, inits, 1/365, 1)
```

---

tsir.app

*Launch a shiny-app simulating TSIR model*

---

**Description**

Launch a shiny-app simulating TSIR model

**Usage**

```
tsir.app
```

**Format**

An object of class shiny.appobj of length 5.

**Details**

Launch app for details

**Examples**

```
if(interactive()){tsir.app}
```

---

 tsirLlyap

*Function to calculate the local Lyapunov exponents for the TSIR*


---

**Description**

Function to calculate the local Lyapunov exponents from an object of class lyap.

**Usage**

```
tsirLlyap(x, m = 1)
```

**Arguments**

x                    an object of class lyap (normally from a call to tsirLyap)  
 m                    number of forward iterations on the attractor

**Value**

An object of class llyap with the local Lyapunov exponent and S-I data

**Examples**

"See chapter 10 in book"

---

 tsirLyap

*Function to do Lyapunov exponent calculations from a TSIR simulation*


---

**Description**

Function to do Lyapunov exponent calculations from a TSIR simulation

**Usage**

```
tsirLyap(I, S, alpha, bt, N)
```

**Arguments**

I                    a vector containing the time series of Is  
 S                    vector containing the time series of Ss  
 alpha                the exponent on I  
 bt                    the seasonal transmission coefficients  
 N                    the population size

**Value**

An object of class `lyap` with the Lyapunov exponent, values for the Jacobians, parameters and data

**Examples**

"See chapter 10 in book"

---

 tsirSim

---

*Function to simulate the stochastic TSIR*


---

**Description**

Function to simulate the stochastic TSIR assuming stochasticity in transmission and a Poisson birth-death process

**Usage**

```
tsirSim(
  alpha = 0.97,
  B = 2300,
  beta = 25,
  sdbeta = 0,
  S0 = 0.06,
  I0 = 180,
  IT = 520,
  N = 3300000
)
```

**Arguments**

<code>alpha</code>	the exponent on I
<code>B</code>	the birth rate
<code>beta</code>	the transmission rate
<code>sdbeta</code>	the standard deviation on beta
<code>S0</code>	the initial susceptible fraction
<code>I0</code>	the initial number of infected
<code>IT</code>	the length of simulation
<code>N</code>	the population size

**Value**

A list with time series of simulated infected and susceptible hosts

**Examples**

```
out = tsirSim()
```

---

tsirSim2	<i>Function to simulate the seasonally-forced TSIR</i>
----------	--

---

**Description**

Function to simulate the stochastic TSIR assuming stochasticity in transmission and a Poisson birth-death process

**Usage**

```
tsirSim2(beta, alpha, B, N, inits = list(Snull = 0, Inull = 0), type = "det")
```

**Arguments**

beta	the seasonal transmission coefficients
alpha	the exponent on I
B	a vector of Births (the length of which determines the length of the simulation)
N	the population size
inits	a list containing initial S and I
type	an argument "det" or "stoc" that determines whether a deterministic or stochastic simulation is done

**Value**

A list with time series of simulated infected and susceptible hosts

**Examples**

"See chapter 8 in book"

---

tsirSpat	<i>Function to simulate the spatially-extended seasonally-forced TSIR</i>
----------	---

---

**Description**

Function to simulate the spatially-extended seasonally-forced TSIR for a patchily distributed host population. Coupling is assumed to be global and according to a commuter model so with  $p$  patches and a coupling of  $c$ , local transmission is reduced by a fraction  $(1-c*p)$ .

**Usage**

```
tsirSpat(beta, alpha, B, N, p, c, inits, type = "det")
```

**Arguments**

beta	the seasonal transmission coefficients
alpha	the exponent on I
B	a vector of Births (the length of which determines the length of the simulation)
N	the population size
p	the number of patches
c	the the spatial coupling
inits	a list containing a vector of initial S and a vector of initial I for each patch
type	an argument "det" or "stoc" that determines whether a deterministic or stochastic simulation is done

**Value**

A list with time series of simulated infected and susceptible hosts

**Examples**

"see chapter 15 in book"

---

twostrain	<i>Gradient-function for the two-strain SIR model</i>
-----------	---

---

**Description**

Gradient-function for the two-strain SIR model

**Usage**

```
twostrain(t, y, parameters)
```

**Arguments**

t	Implicit argument for time
y	A vector with initial values for the states
parameters	A vector with parameter values for the two-strain SIR system

**Value**

A list of gradients

**Examples**

```
require(deSolve)
times = seq(0, 30, by=1/200)
paras = c(mu = 0.02, N = 1, beta1=500, beta2=750, gamma = 365/5, Theta=0.15, Xi=0.15, Pi=0.8)
start = c(S = 0.999, I1 = 0.001, I2 = 0.00, R1=0, R2=0, J1=0, J2=0, R = 0)
out = as.data.frame(ode(start, times, twostrain, paras))
```

---

twostrain.app	<i>Launch a shiny-app simulating a two-strain SIR model</i>
---------------	---

---

**Description**

Launch a shiny-app simulating a two-strain SIR model

**Usage**

twostrain.app

**Format**

An object of class shiny.appobj of length 5.

**Details**

Launch app for details

**Examples**

```
if(interactive()){twostrain.app}
```

---

tydiphtheria	<i>Weekly incidence of diphtheria in Philadelphia between 1914 and 1947.</i>
--------------	--

---

**Description**

A dataset containing the weekly incidence incidence of diphtheria in Philadelphia between 1914 and 1947.

**Usage**

tydiphtheria

**Format**

A data frame with 1774 rows and 4 variables:

**YEAR** the year

**WEEK** the week

**PHILADELPHIA** weekly diphtheria incidence

**TIME** the time counter

**Source**

<https://www.tycho.pitt.edu>

---

tymeasles	<i>Weekly incidence of measles in Philadelphia between 1914 and 1947.</i>
-----------	---

---

**Description**

A dataset containing the weekly incidence of measles in Philadelphia between 1914 and 1947.

**Usage**

tymeasles

**Format**

A data frame with 1774 rows and 4 variables:

**YEAR** the year

**WEEK** the week

**PHILADELPHIA** weekly measles incidence

**TIME** the time counter

**Source**

<https://www.tycho.pitt.edu>

---

tyscarlet	<i>Weekly incidence of scarlet fever in Philadelphia between 1914 and 1947.</i>
-----------	---

---

**Description**

A dataset containing the weekly incidence of scarlet fever in Philadelphia between 1914 and 1947.

**Usage**

tyscarlet

**Format**

A data frame with 1774 rows and 4 variables:

**YEAR** the year

**WEEK** the week

**PHILADELPHIA** weekly scarlet fever incidence

**TIME** the time counter

**Source**

<https://www.tycho.pitt.edu>

---

tywhooping	<i>Weekly incidence of whooping cough in Philadelphia between 1925 and 1947.</i>
------------	--

---

**Description**

A dataset containing the weekly incidence incidence of whooping cough in Philadelphia between 1925 and 1947.

**Usage**

tywhooping

**Format**

A data frame with 1200 rows and 5 variables:

**YEAR** the year

**WEEK** the week

**PHILADELPHIA** weekly whooping cough incidence

**TIME** the time counter

**TM** observation counter

**Source**

<https://www.tycho.pitt.edu>

---

us	<i>2005 US Life table.</i>
----	----------------------------

---

**Description**

Survivorship and fecundities for the US in 2005 by 5 year age-brackets.

**Usage**

us

**Format**

A data frame with 20 rows and 4 variables:

- a** end of age-bracket (in years)
- la** fraction of birth cohort still alive
- fa** fecundity at age
- sa** survival probabilities per age-bracket

---

usflu	<i>US 1975/76 ILI data.</i>
-------	-----------------------------

---

**Description**

Influenza-like illness data for the lower 48 states and the District of Columbia during the 1975/76 season dominated by A/H3N2/Victoria strain

**Usage**

usflu

**Format**

A data frame with 49 rows and 7 variables:

- State** State number
- Acronym** State code
- Pop** Population size
- Latitude** Latitude
- Longitude** Longitude
- Start** Week of start of epidemic
- Peak** Week of peak of epidemic

**Source**

Viboud C, Bjornstad ON, Smith DL, Simonsen L, Miller MA, Grenfell BT (2006) Synchrony, waves, and spatial hierarchies in the spread of influenza. *Science* 312: 447-451.<doi:10.1126/science.1125237>

---

variants *US SARS-CoV-2 variant data.*

---

**Description**

Weekly fraction of identification of the various CoV-2 variants May 2021 through March 2022.  
From [coronavirus.health.ny.gov/clinical-variant-data/](https://coronavirus.health.ny.gov/clinical-variant-data/)

**Usage**

variants

**Format**

A data frame with 47 rows and 7 variables:

**date** End of week of sample

**other** Early variants

**B.1.617.2** Delta variant

**B.1.1.529** First omicron variant

**BA.1.1** Omicron variant BA.1

**BA.2** Omicron variant BA.2 and BA.2.12

**BA.2.12.1** Omicron variant BA 2.12.1

---

waller *Rabies month of first appearance across Connecticut.*

---

**Description**

First month of report of raccoon rabies for 168 townships in Connecticut from March 1991 through January 1995.

**Usage**

waller

**Format**

A data frame with 168 rows and 3 variables:

**x** Longitudinal distance (in km) from first township of appearance

**y** Latitudinal distance (in km) from first township of appearance

**month** month since first appearance in the state in March 1991

**Source**

Waller, L. A., & Gotway, C. A. (2004). Applied spatial statistics for public health data (Vol. 368). John Wiley & Sons.

---

wattsStrogatz                      *Function to generate a Watts-Strogatz network*

---

**Description**

Function to generate a Watts-Strogatz network

**Usage**

```
wattsStrogatz(N, K, Prw)
```

**Arguments**

N	the number of nodes
K	the number of neighbors to which each node is connected so degree = 2*K
Prw	the rewiring probability

**Value**

An object of class CM (contact matrix)

**Examples**

```
cm2=wattsStrogatz(N=20, K=4, Prw=.3)
```

# Index

## \* datasets

black, 4  
burnett, 5  
ccs, 5  
chabaudi, 6  
cholera, 7  
cspring, 9  
dalziel, 9  
ebola, 10  
euthamia, 11  
ferrari, 11  
filipendula, 12  
fiv, 12  
flu, 16  
gonnet, 17  
gypsomoth, 19  
icelandflu, 19  
litter, 22  
m4494, 24  
magono, 24  
may.app, 25  
meas, 26  
nbspat.app, 27  
niamey, 29  
niamey\_daily, 30  
orv.app, 30  
pagiard, 31  
paili, 31  
palymes, 32  
pameasle, 32  
pdv, 33  
pertcop, 33  
peru, 34  
polymod, 36  
rabbit, 37  
rabies, 37  
ross.app, 40  
seir.app, 40  
seirs.app, 42

silene, 42  
sir.app, 44  
tsir.app, 51  
twostrain.app, 56  
tydiphtheria, 56  
tymeasles, 57  
tyscarlet, 57  
tywhooping, 58  
us, 58  
usflu, 59  
variants, 60  
waller, 60

arrows, 15, 16

barabasiAlbert, 3  
black, 4  
burnett, 5

ccs, 5  
chabaudi, 6  
character, 14, 15  
cholera, 7  
coyne, 7  
coyne2, 8  
cspring, 9

dalziel, 9

ebola, 10  
euthamia, 11

ferrari, 11  
filipendula, 12  
fiv, 12  
flowField, 13  
flu, 16

gillespie, 17  
gonnet, 17  
gravity, 18

gypsomoth, 19  
icelandflu, 19  
jacobian, 20  
length, 14, 15  
leslie, 21  
list, 15  
litter, 22  
llik.cb, 22  
llik.pc, 23  
m4494, 24  
magono, 24  
matrix, 15  
may.app, 25  
meas, 26  
nbmod, 26  
nbspat.app, 27  
nextgenR0, 28  
niamey, 29  
niamey\_daily, 30  
numeric, 14–16  
ode, 14  
orv.app, 30  
pagiard, 31  
paili, 31  
palyes, 32  
pameasle, 32  
pdv, 33  
pertcop, 33  
peru, 34  
plot, 15, 16  
plot.cm, 35  
plot.netSir, 35  
polymod, 36  
r0fun, 36  
rabbit, 37  
rabies, 37  
retrospec, 38, 48  
ringlattice, 39  
ross.app, 40  
seir.app, 40  
seirmod, 41  
seirmod2, 41  
seirs.app, 42  
silene, 42  
sim.cb, 43  
sir.app, 44  
sirAgemod, 44  
sirChainmod, 45  
sirmod, 46  
sirNetmod, 36, 46, 50  
sirSpatmod, 18, 47  
sirvmod, 48  
sirwmod, 48  
summary.cm, 49  
summary.netSir, 50  
tau, 50  
tsir.app, 51  
tsirLlyap, 52  
tsirLyap, 52  
tsirSim, 53  
tsirSim2, 54  
tsirSpat, 54  
twostrain, 55  
twostrain.app, 56  
tydiphtheria, 56  
tymeasles, 57  
tyscarlet, 57  
tywhooping, 58  
us, 58  
usflu, 59  
variants, 60  
vector, 14–16  
waller, 60  
wattsStrogatz, 61