Package: EpiLPS (via r-universe)

October 29, 2026

Type Package

Title A Fast and Flexible Bayesian Tool for Estimating Epidemiological Parameters

Version 1.3.0

Depends R (>= 4.1.0)

Maintainer Oswaldo Gressani <oswaldo_gressani@hotmail.fr>

BugReports https://github.com/oswaldogressani/EpiLPS/issues

Description Estimation of epidemiological parameters with Laplacian-P-splines following the methodology of Gressani et al. (2022) <doi:10.1371/journal.pcbi.1010618>.

URL <https://epilps.com/>

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

LinkingTo RcppArmadillo, Rcpp

Imports Rcpp (>= 1.0.7), coda (>= 0.19-4), EpiEstim (>= 2.2-4), ggplot2 (>= 3.3.5), gridExtra (>= 2.3)

Suggests rmarkdown, knitr

VignetteBuilder knitr

NeedsCompilation yes

Author Oswaldo Gressani [aut, cre] (<https://orcid.org/0000-0003-4152-6159>), Bryan Sumalinab [ctb] (<https://orcid.org/0000-0001-8264-5336>)

Repository https://cran.r-universe.dev

Date/Publication 2024-03-09 02:27:38 UTC

RemoteUrl https://github.com/cran/EpiLPS

RemoteRef HEAD

RemoteSha 6899f2b6aa1eee1b3b29c4b37cb891e7bbc87fd1

Contents

cov19incidence2022 . . . 2
cov19mort2021 . . . 3
epicurve . . . 3
episim . . . 4
eruptions . . . 5
estimIncub . . . 6
estimR . . . 7
estimRmcmc . . . 9
histosmooth . . . 11
Idist . . . 12
incubsim . . . 13
influenza2009 . . . 15
nowcasting . . . 15
nowcastingR . . . 17
perfRestim . . . 18
plot.Idist . . . 21
plot.incubestim . . . 22
plot.nowcasted . . . 23
plot.Rt . . . 24
plot.Rtnow . . . 25
Rmodelpriors . . . 26
sars2003 . . . 26
summary.Rt . . . 27
zika2015 . . . 28

Index **29**

`cov19incidence2022` *Incidence data for Belgium in 2022*

Description

A data frame with six columns including the calendar date of COVID-19 infection, the date on which cases have been reported and the number of cases.

Usage

```
data(cov19incidence2022)
```

Format

`t` A numeric variable associated to a calendar date.

`d` A numeric variable indicating the delay of reporting.

`Date` The calendar date for incidence.

`Rep.date` The calendar date for the reporting of incidence.

`Cases` The number of cases.

`Reported` Indicates whether cases are already reported or not.

Source

https://epistat.sciensano.be/covid/

`cov19mort2021` *Mortality data for Belgium in 2021*

Description

A data frame with six columns including the calendar date of death, the date on which cases have been reported and the number of cases/deaths.

Usage

```
data(cov19mort2021)
```

Format

`t` A numeric variable associated to a calendar date.

`d` A numeric variable indicating the delay of reporting.

`Date` The calendar date of the death event.

`Rep.date` The calendar date for the reporting of the death event.

`Cases` The number of cases/deaths.

`Reported` Indicates whether cases are already reported or not.

Source

https://epistat.sciensano.be/covid/

`epicurve` *Plot the epidemic curve*

Description

This routine gives a graphical representation of the epidemic curve based on an incidence time series.

Usage

```
epicurve(incidence, dates = NULL, datelab = "7d", col = "deepskyblue4", barwidth = 1,
       title = "Epidemic curve", xtickangle = 0, smooth = NULL,  smoothcol = "orange")
```

Arguments

incidence	A vector containing the incidence time series
dates	A vector of dates in format "YYYY-MM-DD".
datelab	The spacing for ticks on the x-axis. Default "7d".
col	The color of the epidemic curve.
barwidth	The width of the bars. Default is 1.
title	Title of the plot.
xtickangle	The angle of the x-ticks. Default is 0 (horizontal).
smooth	An object of class Rt obtained with the estimR or estimRmcmc routine. It is used to draw a smoothed estimate of the epidemic curve based on the Laplacian-P-splines model.
smoothcol	The color of the smoothed curve and associated credible interval.

Value

A plot of the epidemic curve.

Author(s)

Oswaldo Gressani <oswaldo_gressani@hotmail.fr>

Examples

```
si <- c(0.05, 0.05, 0.1, 0.1, 0.1, 0.1, 0.1, 0.05, 0.05, 0.1, 0.1, 0.1)
epidemic <- episim(si = si, Rpattern = 4)
epicurve(epidemic$y)
```

---

episim *Simulation of an incidence time series*

Description

Based on a serial interval and a functional input for the reproduction number over T days, the routine generates an incidence time series following a Poisson or negative binomial model. The link between the reproduction number and the generated incidence data is governed by the renewal equation. The baseline (mean) number of cases at day 1 is fixed at 10. The mean number of cases for the remaining days of the epidemic are generated following equation (2) of Azmon et al. (2013).

Usage

```
episim(si, endepi = 50, Rpattern = 1, Rconst = 2.5,
      dist = c("poiss", "negbin"), overdisp = 1, verbose = FALSE, plotsim = FALSE)
```

Arguments

si	The serial interval distribution.
endepi	The total number of days of the epidemic.
Rpattern	Different scenarios for the true underlying curve of Rt. Six scenarios are possible with 1,2,3,4,5,6.
Rconst	The constant value of R (if scenario 1 is selected), default is 2.5.
dist	The distribution from which to sample the incidence counts. Either Poisson (default) or negative binomial.
overdisp	Overdispersion parameter for the negative binomial setting.
verbose	Should metadata of the simulated epidemic be printed?
plotsim	Create a plot of the incidence time series, the true reproduction number curve and the serial interval.

Value

An object of class `episim` consisting of a list with the generated incidence time series, the mean vector of the Poisson/negative binomial distribution, the true underlying R function for the data generating process and the chosen serial interval distribution.

Author(s)

Oswaldo Gressani <oswaldo_gressani@hotmail.fr>

References

Azmon, A., Faes, C., Hens, N. (2014). On the estimation of the reproduction number based on misreported epidemic data. *Statistics in medicine*, **33**(7):1176-1192.

Examples

```
si <- c(0.05, 0.05, 0.1, 0.1, 0.1, 0.1, 0.1, 0.05, 0.05, 0.1, 0.1, 0.1)
epidemic <- episim(si = si, Rpattern = 1)
```

---

eruptions *Eruption times in Yellowstone National Park*

Description

A vector of eruption times in minutes recorded from the Old Faithful geyser in Yellowstone National Park, USA.

Usage

```
data(eruptions)
```

Format

`eruptions` A vector of eruption times in minutes.

References

Härdle, W. (1991). Smoothing Techniques with Implementation in S. New York: Springer.

`estimIncub` *Estimation of the incubation density based on coarse data*

Description

This function computes an estimate of the incubation density based on coarse data constructed from symptom onset times and exposure windows. It uses the Laplacian-P-splines methodology with a Langevinized Gibbs algorithm to sample from the posterior distribution.

Usage

```
estimIncub(x, K = 10, niter = 1000, tmax = max(x), tgridlen = 500, verbose = FALSE)
```

Arguments

`x`	A data frame with the lower and upper bound of incubation interval.
`K`	Number of B-splines in the basis.
`niter`	The number of MCMC samples.
`tmax`	The upper bound for the B-spline basis. Default is the largest data point in `x`.
`tgridlen`	The number of grid points on which to evaluate the density.
`verbose`	Should a message be printed? Default is FALSE.

Value

A list of class `incubestim` containing summary values for the estimated incubation density.

Author(s)

Oswaldo Gressani <oswaldo_gressani@hotmail.fr>

Examples

```
set.seed(123)
simdat <- incubsim(n = 30, tmax = 20) # Simulate incubation data
data <- simdat$Dobsincub              # Incubation bounds
incubfit <- estimIncub(x = data, niter = 500, tmax = 20, verbose = TRUE)
```

estimR *Estimation of the reproduction number with Laplacian-P-splines*

Description

This routine estimates the instantaneous reproduction number R_t; the mean number of secondary infections generated by an infected individual at time t (White et al. 2020); by using Bayesian P-splines and Laplace approximations (Gressani et al. 2022). Estimation of R_t is based on a time series of incidence counts and (a discretized) serial interval distribution. The negative binomial distribution is used to model incidence count data and P-splines (Eilers and Marx, 1996) are used to smooth the epidemic curve. The link between the epidemic curve and the reproduction number is established via the renewal equation.

Usage

```
estimR(incidence, si, K = 30, dates = NULL, maxmethod = c("NelderMead","HillClimb"),
CoriR = FALSE, WTR = FALSE, optimstep = 0.3, priors = Rmodelpriors())
```

Arguments

incidence	A vector containing the incidence time series. If incidence contains NA values at certain time points, these are replaced by the average of the left- and right neighbor counts. If the right neighbor is NA, the left neighbor is used as a replacement value.
si	The (discrete) serial interval distribution.
K	Number of B-splines in the basis.
dates	A vector of dates in format "YYYY-MM-DD" (optional).
maxmethod	The method to maximize the hyperparameter posterior distribution.
CoriR	Should the R_t estimate of Cori (2013) be also computed?
WTR	Should the R_t estimate of Wallinga-Teunis (2004) be also computed?
optimstep	Learning rate for the "HillClimb" method to maximize the posterior distribution of the hyperparameters.
priors	A list containing the prior specification of the model hyperparameters as set in Rmodelpriors. See ?Rmodelpriors.

Details

The estimR routine estimates the reproduction number in a totally "sampling-free" fashion. The hyperparameter vector (containing the penalty parameter of the P-spline model and the overdispersion parameter of the negative binomial model for the incidence time series) is fixed at its maximum a posteriori (MAP). By default, the algorithm for maximization is the one of Nelder and Mead (1965). If maxmethod is set to "HillClimb", then a gradient ascent algorithm is used to maximize the hyperparameter posterior.

Value

A list with the following components:

- incidence: The incidence time series.
- si: The serial interval distribution.
- RLPS: A data frame containing estimates of the reproduction number obtained with the Laplacian-P-splines methodology.
- thetahat: The estimated vector of B-spline coefficients.
- Sighat: The estimated variance-covariance matrix of the Laplace approximation to the conditional posterior distribution of the B-spline coefficients.
- RCori: A data frame containing the estimates of the reproduction obtained with the method of Cori et al. (2013).
- RWT: A data frame containing the estimates of the reproduction obtained with the method of Wallinga-Teunis (2004).
- LPS_elapsed: The routine real elapsed time (in seconds) when estimation of the reproduction number is carried out with Laplacian-P-splines.
- Cori_elapsed: The routine real elapsed time (in seconds) when estimation of the reproduction number is carried out with the method of Cori et al. (2013).
- penparam: The estimated penalty parameter related to the P-spline model.
- K: The number of B-splines used in the basis.
- NegBinoverdisp: The estimated overdispersion parameter of the negative binomial distribution for the incidence time series.
- optimconverged: Indicates whether the algorithm to maximize the posterior distribution of the hyperparameters has converged.
- method: The method to estimate the reproduction number with Laplacian-P-splines.
- optim_method: The chosen method to to maximize the posterior distribution of the hyperparameters.

Author(s)

Oswaldo Gressani <oswaldo_gressani@hotmail.fr>

References

Gressani, O., Wallinga, J., Althaus, C. L., Hens, N. and Faes, C. (2022). EpiLPS: A fast and flexible Bayesian tool for estimation of the time-varying reproduction number. *Plos Computational Biology*, **18**(10): e1010618.

Cori, A., Ferguson, N.M., Fraser, C., Cauchemez, S. (2013). A new framework and software to estimate time-varying reproduction numbers during epidemics. *American Journal of Epidemiology*, **178**(9):1505–1512.

Wallinga, J., & Teunis, P. (2004). Different epidemic curves for severe acute respiratory syndrome reveal similar impacts of control measures. *American Journal of Epidemiology*, **160**(6), 509-516.

White, L.F., Moser, C.B., Thompson, R.N., Pagano, M. (2021). Statistical estimation of the reproductive number from case notification data. *American Journal of Epidemiology*, **190**(4):611-620.

Eilers, P.H.C. and Marx, B.D. (1996). Flexible smoothing with B-splines and penalties. *Statistical Science*, **11**(2):89-121.

Examples

```
# Illustration on simulated data
si <- Idist(mean = 5, sd = 3)$pvec
datasim <- episim(si = si, endepi = 60, Rpattern = 5, dist="negbin", overdisp = 50)
epifit_sim <- estimR(incidence = datasim$y, si = si, CoriR = TRUE)
plot(epifit_sim, addfit = "Cori")

# Illustration on the 2003 SARS epidemic in Hong Kong.
data(sars2003)
epifit_sars <- estimR(incidence = sars2003$incidence, si = sars2003$si, K = 40)
tail(epifit_sars$RLPS)
summary(epifit_sars)
plot(epifit_sars)
```

`estimRmcmc` *Estimation of the reproduction number with Laplacian-P-splines via MCMC*

Description

This routine estimates the instantaneous reproduction number R_t; the mean number of secondary infections generated by an infected individual at time t (White et al. 2020); by using Bayesian P-splines and Laplace approximations (Gressani et al. 2022). The inference approach is fully stochastic with a Metropolis-adjusted Langevin algorithm. The `estimRmcmc()` routine estimates R_t based on a time series of incidence counts and a (discretized) serial interval distribution. The negative binomial distribution is used to model incidence count data and P-splines (Eilers and Marx, 1996) are used to smooth the epidemic curve. The link between the epidemic curve and the reproduction number is established via the renewal equation.

Usage

```
estimRmcmc(incidence, si, K = 30, dates = NULL, niter = 5000, burnin = 2000,
 CoriR = FALSE, WTR = FALSE, priors = Rmodelpriors(), progressbar = TRUE)
```

Arguments

`incidence`	A vector containing the incidence time series. If `incidence` contains NA values at certain time points, these are replaced by the average of the left- and right neighbor counts. If the right neighbor is NA, the left neighbor is used as a replacement value.
`si`	The (discrete) serial interval distribution.
`K`	Number of B-splines in the basis.
`dates`	A vector of dates in format "YYYY-MM-DD" (optional).
`niter`	The number of MCMC samples.
`burnin`	The burn-in size.

CoriR	Should the R_t estimate of Cori (2013) be also computed?
WTR	Should the R_t estimate of Wallinga-Teunis (2004) be also computed?
priors	A list containing the prior specification of the model hyperparameters as set in Rmodelpriors. See ?Rmodelpriors.
progressbar	Should a progression bar indicating status of MCMC algorithm be shown? Default is TRUE.

Value

A list with the following components:

- incidence: The incidence time series.
- si: The serial interval distribution.
- RLPS: A data frame containing estimates of the reproduction number obtained with the Laplacian-P-splines methodology.
- thetahat: The estimated vector of B-spline coefficients.
- Sighat: The estimated variance-covariance matrix of the Laplace approximation to the conditional posterior distribution of the B-spline coefficients.
- RCori: A data frame containing the estimates of the reproduction obtained with the method of Cori (2013).
- RWT: A data frame containing the estimates of the reproduction obtained with the method of Wallinga-Teunis (2004).
- LPS_elapsed: The routine real elapsed time (in seconds) when estimation of the reproduction number is carried out with Laplacian-P-splines.
- penparam: The estimated penalty parameter related to the P-spline model.
- K: The number of B-splines used in the basis.
- NegBinoverdisp: The estimated overdispersion parameter of the negative binomial distribution for the incidence time series.
- optimconverged: Indicates whether the algorithm to maximize the posterior distribution of the hyperparameters has converged.
- method: The method to estimate the reproduction number with Laplacian-P-splines.
- optim_method: The chosen method to to maximize the posterior distribution of the hyperparameters.
- HPD90_Rt: The 90% HPD interval for Rt obtained with the LPS methodology.
- HPD95_Rt: The 95% HPD interval for Rt obtained with the LPS methodology.

Author(s)

Oswaldo Gressani <oswaldo_gressani@hotmail.fr>

References

Gressani, O., Wallinga, J., Althaus, C. L., Hens, N. and Faes, C. (2022). EpiLPS: A fast and flexible Bayesian tool for estimation of the time-varying reproduction number. *Plos Computational Biology*, **18(10): e1010618**.

Cori, A., Ferguson, N.M., Fraser, C., Cauchemez, S. (2013). A new framework and software to estimate time-varying reproduction numbers during epidemics. *American Journal of Epidemiology*, **178**(9):1505–1512.

Wallinga, J., & Teunis, P. (2004). Different epidemic curves for severe acute respiratory syndrome reveal similar impacts of control measures. *American Journal of Epidemiology*, **160**(6), 509-516.

White, L.F., Moser, C.B., Thompson, R.N., Pagano, M. (2021). Statistical estimation of the reproductive number from case notification data. *American Journal of Epidemiology*, **190**(4):611-620.

Eilers, P.H.C. and Marx, B.D. (1996). Flexible smoothing with B-splines and penalties. *Statistical Science*, **11**(2):89-121.

Examples

```
# Illustration on the 2009 influenza pandemic in Pennsylvania.
data(influenza2009)
epifit_flu <- estimRmcmc(incidence = influenza2009$incidence, dates = influenza2009$dates,
                         si = influenza2009$si[-1], niter = 2500,
                         burnin = 1500, progressbar = FALSE)
tail(epifit_flu$RLPS)
summary(epifit_flu)
plot(epifit_flu)
```

`histosmooth` *Histogram smoothing with Laplacian-P-splines*

Description

This function provides a smooth density estimate to a histogram using Laplacian-P-splines. The B-spline basis is computed on the midpoints of the histogram bins. The default number of (cubic) B-splines is 30 and a third-order penalty is specified. The negative binomial distribution is used to model the number of observations falling in each bin.

Usage

```
histosmooth(x, xl = min(x), xr = max(x), K = 30)
```

Arguments

`x`	A vector of real numbers from which the histogram will be constructed.
`xl`	The left bound for the domain of `x` which also coincides with the left bound of the domain of the B-spline basis. The default is taken to be the minimum of `x`.

xr	The right bound for the domain of x which also coincides with the right bound of the domain of the B-spline basis. The default is taken to be the maximum of x.
K	Number of B-splines in the basis.

Value

A list containing the left (xl) and right (xr) bounds of the domain of the estimated density, the binwidth and a function to be evaluated between xl and xr.

Author(s)

Oswaldo Gressani <oswaldo_gressani@hotmail.fr>

References

Gressani, O. and Lambert, P. (2018). Fast Bayesian inference using Laplace approximations in a flexible promotion time cure model based on P-splines. *Computational Statistical & Data Analysis* **124**: 151-167.

Gressani, O., Wallinga, J., Althaus, C. L., Hens, N. and Faes, C. (2022). EpiLPS: A fast and flexible Bayesian tool for estimation of the time-varying reproduction number. *Plos Computational Biology*, **18**(10): e1010618.

Examples

```
# Old Faithful geyser application
data(eruptions)
x <- eruptions
ffit <- histosmooth(x, xl = 1, xr = 6)
tt <- seq(ffit$xl, ffit$xr, length = 500)
dtt <- tt[2] - tt[1]
graphics::hist(x, breaks = seq(ffit$xl, ffit$xr, by = ffit$binwidth),
     freq = FALSE, ylim = c(0, 0.8), main = "Old Faithful Geyser",
    xlab = "Eruption time (minutes)")
densfit <- sapply(tt, ffit$fdens)
densfit <- densfit / (sum(densfit * dtt))
graphics::lines(tt, densfit, col = "red", lwd = 2)
```

Idist *Density function and discrete distribution for a disease interval*

Description

This function computes the probability density function and probability mass function for a disease interval (e.g. the serial interval defined as the time elapsed between the symptom onset in an infector and the onset of symptoms in the secondary cases generated by that infector). It takes as input the mean and the standard deviation of the disease interval (expressed in days) and gives as an output the interval distribution based on a chosen parametric family.

Usage

```
Idist(mean, sd, dist = c("gamma", "weibull", "lognorm"), probs = NULL)
```

Arguments

mean	The mean of the disease interval (must be larger than 1).
sd	A positive and finite real number corresponding to the standard deviation of the disease interval.
dist	A choice among a Gamma, Weibull or Log-normal distribution for the disease interval.
probs	A vector of probabilities for the interval distribution.

Details

The discretization is based on the formula in Held et al. (2019).

Value

A list of class `Idist` containing a vector of probabilities corresponding to the discrete distribution of the disease interval, the name of the chosen parametric distribution and its parameters.

Author(s)

Oswaldo Gressani <oswaldo_gressani@hotmail.fr>

References

Held, L., Hens, N., D O'Neill, P., and Wallinga, J. (2019). Handbook of infectious disease data analysis. *CRC Press*.

Examples

```
Idist(mean = 2.6, sd = 1.5)
```

---

incubsim *Simulation of incubation times*

Description

This routine simulates symptom onset times, generation times and left and right bounds of infecting exposure windows for source-recipient pairs based on a given incubation distribution and a generation time distribution. The generation time distribution is assumed to be a Weibull with shape 2.826 and scale 5.665 (Ferretti et al. 2020). The incubation distribution can be chosen by the user among:

- A LogNormal distribution with a mean of 5.5 days and standard deviation of 2.1 days (Ferretti et al. 2020).

- A Weibull distribution (shape-scale parameterization) with a mean of 6.4 days and standard deviation of 2.3 days (Backer et al. 2020).
- An artificial bimodal distribution constructed from a mixture of two Weibull distributions.
- A Gamma distribution (shape-rate parameterization) with a mean of 3.8 days and standard deviation of 2.9 days (Donnelly et al. 2003).

Usage

```
incubsim(incubdist = c("LogNormal","Weibull","MixWeibull", "Gamma"), coarseness = 1,
 n = 100, tmax = 20, tgridlen = 500, plotsim = FALSE)
```

Arguments

`incubdist`	The distribution of the incubation period.
`coarseness`	The average coarseness of the data. Default is 1 day.
`n`	The sample size.
`tmax`	The upper bound on which to evaluate the `incubdist` density.
`tgridlen`	The number of grid points on which to evaluate the density.
`plotsim`	Graphical visualization of the simulated data?

Value

A list including (among others) the left and right bounds of the incubation period.

Author(s)

Oswaldo Gressani <oswaldo_gressani@hotmail.fr>

References

Ferretti, L., Wymant, C., Kendall, et al. (2020). Quantifying SARS-CoV-2 transmission suggests epidemic control with digital contact tracing. *Science*, **368**(6491), eabb6936.

Backer, J. A., Klinkenberg, D., & Wallinga, J. (2020). Incubation period of 2019 novel coronavirus (2019-nCoV) infections among travellers from Wuhan, China, 20–28 January 2020. *Eurosurveillance*, **25**(5), 2000062.

Donnelly, C. A., Ghani, A. C., Leung, G. M., et al. (2003). Epidemiological determinants of spread of causal agent of severe acute respiratory syndrome in Hong Kong. *The Lancet*, **361**(9371), 1761-1766.

Examples

```
simdat <- incubsim(n = 50) # Simulation of incubation times for 50 cases.
simdat$Dobsincub           # Left and right bounds of incubation period.
```

influenza2009 *Data on the 2009 pandemic influenza in Pennsylvania*

Description

A list with the daily incidence of onset of symptoms among children in a school in Pennsylvania (2009), a vector of dates and a discrete serial interval distribution.

Usage

```
data(influenza2009)
```

Format

A list with three components:

`incidence` An incidence time series of length 32.

`dates` A vector of dates in "YYYY-MM-DD" format.

`si` A vector of probabilities corresponding to the serial interval distribution.

Source

https://cran.r-project.org/package=EpiEstim

References

Ferguson N.M. et al. (2005) Strategies for containing an emerging influenza pandemic in Southeast Asia. *Nature* **437**(7056), 209-214.

Cauchemez S. et al. (2011) Role of social networks in shaping disease transmission during a community outbreak of 2009 H1N1 pandemic influenza. *Proc Natl Acad Sci* USA **108**(7), 2825-2830.

nowcasting *Nowcasting and estimation of occurred but not yet reported events*

Description

This routine can be used to estimate cases that have not yet been reported also known as occurred-but-not-yet-reported-events. Daily cases are typically subject to reporting delays, so that the reported number of infected individuals is not always reflecting the true epidemic status. Nowcasting aims to correct this underreporting phenomenon by estimating the number of infections that have occurred but that have not yet been reported. The latter number is then combined with the already reported cases and interpreted as a nowcast or prediction for the true epidemic status regarding the number of daily cases. The routine is anchored around Laplacian-P-splines in an epidemic context (Gressani et al. 2022) and the detailed methodology can be found in Sumalinab et al. (2023).

Usage

```
nowcasting(data, day.effect = TRUE, ref.day = "Monday", verbose = TRUE)
```

Arguments

`data`	A data frame containing the data for each time and delay combination with the following 6 columns. The first column is a numeric variable associated to the calendar date. The second column is a numeric variable indicating the delay of reporting. The third column corresponds to the calendar date of the event (e.g. death) and the fourth column to the calendar date at which the event of interest was reported. The fifth column indicates the number of cases for each time and delay combination. Finally, the sixth column indicates whether the cases are already reported or not yet reported. To see an example of such a data structure type `data("cov19mort2021")` and then `head(cov19mort2021)`. This will illustrate the data structure used for nowcasting the daily number of cases based on mortality data for Belgium in 2021.
`day.effect`	If TRUE (defaut), include the day of the week effect.
`ref.day`	If `day.effect = TRUE`, then the reference category for the day of the week must be specified. The default is "Monday".
`verbose`	Show summary output of nowcast in console? Default is TRUE.

Value

A list with the following components:

- data: The data frame used as an input.
- cases.now: A data frame containing the nowcasting results with 95% prediction intervals.
- delay: A data frame containing the two-dimensional delay distribution.
- lambda_estim: Estimated penalty parameters of the P-splines model.
- phi_estim: Estimated overdispersion parameter from the negative binomial model.

Author(s)

Bryan Sumalinab (writing) and Oswaldo Gressani (editing).

References

Gressani, O., Wallinga, J., Althaus, C. L., Hens, N. and Faes, C. (2022). EpiLPS: A fast and flexible Bayesian tool for estimation of the time-varying reproduction number. *Plos Computational Biology*, **18**(10): e1010618.

Sumalinab, B., Gressani, O., Hens, N. and Faes, C. (2023). Bayesian nowcasting with Laplacian-P-splines. MedRxiv preprint. https://www.medrxiv.org/content/10.1101/2022.08.26.22279249v2

Examples

```
# data("cov19mort2021")
# ncast <- nowcasting(data = cov19mort2021, day.effect = FALSE)
# plot(ncast) # Show nowcasted cases
# plot(ncast, type = "delay") # Show contour of delay distribution
```

`nowcastingR` *Nowcasting the reproduction number*

Description

This routine can be used to nowcast the time-varying reproduction number. Daily cases are typically subject to reporting delays, so that the reported number of infected individuals is not always reflecting the true epidemic status. Nowcasting aims to correct this underreporting phenomenon by estimating the number of infections that have occurred but that have not yet been reported. The latter number is then combined with the already reported cases and interpreted as a nowcast or prediction for the true epidemic status regarding the number of daily cases. The routine is anchored around Laplacian-P-splines in an epidemic context (Gressani et al. 2022) and the detailed methodology can be found in Sumalinab et al. (2023). Two different models can be fitted, named M3 (the default) and M2. M3 uses a joint approach that simultaneously models the delay dimension and the time-varying reproduction number. M2 uses reported cases and a nowcast of the not yet reported cases. See Sumalinab et al. (2023) and the vignette https://epilps.com/NowcastingRt.html for more details.

Usage

```
nowcastingR(data, day.effect = TRUE, ref.day = "Monday", si, method = c("M3", "M2"))
```

Arguments

`data`	A data frame containing the data for each time and delay combination with the following 6 columns. The first column is a numeric variable associated to the calendar date. The second column is a numeric variable indicating the delay of reporting. The third column corresponds to the calendar date of the event (e.g. death) and the fourth column to the calendar date at which the event of interest was reported. The fifth column indicates the number of cases for each time and delay combination. Finally, the sixth column indicates whether the cases are already reported or not yet reported. To see an example of such a data structure type `data("cov19incidence2022")` and then `head(cov19incidence2022)`. This will illustrate the required data structure for nowcasting the reproduction number based on incidence data for Belgium in 2022.
`day.effect`	If TRUE (default), include the day of the week effect.
`ref.day`	If `day.effect = TRUE`, then the reference category for the day of the week must be specified. The default is "Monday".
`si`	The (discrete) serial interval distribution.
`method`	The model to be fitted, either M3 (default) or M2.

Value

A list with the following components:

- data: The data frame used as an input.
- Rnow: A data frame containing the nowcasted reproduction number.
- lambda_estim: Estimated penalty parameters of the P-splines model.
- phi_estim: Estimated overdispersion parameter from the negative binomial model.
- method: The model choice, i.e. either M3 or M2.

Author(s)

Bryan Sumalinab (writing) and Oswaldo Gressani (editing).

References

Gressani, O., Wallinga, J., Althaus, C. L., Hens, N. and Faes, C. (2022). EpiLPS: A fast and flexible Bayesian tool for estimation of the time-varying reproduction number. *Plos Computational Biology*, **18**(10): e1010618.

Sumalinab, B., Gressani, O., Hens, N. and Faes, C. (2023). An efficient approach to nowcasting the time-varying reproduction number. MedRxiv preprint.

Examples

```
# data("cov19incidence2022")
# si_covid <- c(0.344, 0.316, 0.168, 0.104, 0.068) # serial interval distribution
# Sys.setlocale("LC_TIME", "English")              # set system locale to English
# Rnowfit <- nowcastingR(data = cov19incidence2022, si = si_covid)
# tail(Rnowfit$Rnow)
# plot(Rnowfit)
```

`perfRestim` *Routine to measure the performance of estimR and estimRmcmc*

Description

This routine can be used to check the 'statistical performance' of the `estimR()` and `estimRmcmc()` routines to estimate the reproduction number R_t. It simulates epidemics using the `episim()` function and computes the Bias, MSE, coverage probability (CP) and width of 90% and 95% credible intervals for R_t averaged over days $t = 8, ..., T$, where T is the total number of days of the simulated epidemics. As such, it can be used to reproduce part of the results in Gressani et al. (2022) Table 1 and Table 2, respectively. Small differences in results are due to a restructuring of the code since version 1.0.6. If strict reproducible results are required, please refer to version 1.0.6 of the EpiLPS package or visit the GitHub repository https://github.com/oswaldogressani/EpiLPS-ArticleCode.

Usage

```
perfRestim(nsim = 100, scenario = 1, days = 40, K = 40,
 method = c("LPSMAP", "LPSMALA"), mcmciter = 3000, burnin = 1000,
 si = c("flu", "sars", "mers"), seed = 1325, overdisp = 1000)
```

Arguments

nsim	Total number of simulated epidemics.
scenario	The scenario to be used in `episim()`.
days	Number of days for the simulated epidemics.
K	Number of B-splines basis function in the P-spline model.
method	The method for LPS, either LPSMAP or LPSMALA.
mcmciter	Number of MCMC samples for method LPSMALA.
burnin	Burn-in for method LPSMALA.
si	The discrete serial interval distribution. Possible specifications are "flu", "sars" or "mers".
seed	A seed for reproducibility.
overdisp	The value of the overdispersion parameter for the negative binomial model in the `episim()` routine.

Value

A list with the following components:

- LPS: Results for the LPS approach.
- EpiEstim: Results for the EpiEstim approach with weekly sliding windows.
- inciplot: The simulated incidence time series.
- Rlpsplot: Estimated R_t trajectories with LPS.
- Repiestimplot: Estimated R_t trajectories with EpiEstim.

Author(s)

Oswaldo Gressani <oswaldo_gressani@hotmail.fr>

References

Gressani, O., Wallinga, J., Althaus, C. L., Hens, N. and Faes, C. (2022). EpiLPS: A fast and flexible Bayesian tool for estimation of the time-varying reproduction number. *Plos Computational Biology*, **18**(10): e1010618.

Examples

```
# # FLU serial interval (Scenarios 1-4)
# S1 <- perfRestim(si = "flu", scenario = 1, seed = 1325)
# S1mcmc <- perfRestim(si = "flu", scenario = 1, seed = 1325, method = "LPSMALA")
# suppressWarnings(gridExtra::grid.arrange(S1$inciplot, S1$Rlpsplot, S1$Repiestimplot, nrow = 1))
# S2 <- perfRestim(si = "flu", scenario = 2, seed = 1123)
# S2mcmc <- perfRestim(si = "flu", scenario = 2, seed = 1123, method = "LPSMALA")
# suppressWarnings(gridExtra::grid.arrange(S2$inciplot, S2$Rlpsplot, S2$Repiestimplot, nrow = 1))
# S3 <- perfRestim(si = "flu", scenario = 3, seed = 1314)
# S3mcmc <- perfRestim(si = "flu", scenario = 3, seed = 1314, method = "LPSMALA")
# suppressWarnings(gridExtra::grid.arrange(S3$inciplot, S3$Rlpsplot, S3$Repiestimplot, nrow = 1))
# S4 <- perfRestim(si = "flu", scenario = 4, seed = 1966)
# S4mcmc <- perfRestim(si = "flu", scenario = 4, seed = 1966, method = "LPSMALA")
# suppressWarnings(gridExtra::grid.arrange(S4$inciplot, S4$Rlpsplot, S4$Repiestimplot, nrow = 1))
#
# # SARS serial interval (Scenarios 5-8)
# S5 <- perfRestim(si = "sars", scenario = 1, seed = 1998, overdisp = 5)
# S5mcmc <- perfRestim(si = "sars", scenario = 1, seed = 1998, overdisp = 5, method = "LPSMALA")
# suppressWarnings(gridExtra::grid.arrange(S5$inciplot, S5$Rlpsplot, S5$Repiestimplot, nrow = 1))
# S6 <- perfRestim(si = "sars", scenario = 2, seed = 1870, overdisp = 5)
# S6mcmc <- perfRestim(si = "sars", scenario = 2, seed = 1870, overdisp = 5, method = "LPSMALA")
# suppressWarnings(gridExtra::grid.arrange(S6$inciplot, S6$Rlpsplot, S6$Repiestimplot, nrow = 1))
# S7 <- perfRestim(si = "sars", scenario = 3, seed = 115,  overdisp = 5)
# S7mcmc <- perfRestim(si = "sars", scenario = 3, seed = 115,  overdisp = 5, method = "LPSMALA")
# suppressWarnings(gridExtra::grid.arrange(S7$inciplot, S7$Rlpsplot, S7$Repiestimplot, nrow = 1))
# S8 <- perfRestim(si = "sars", scenario = 4, seed = 1464, overdisp = 5)
# S8mcmc <- perfRestim(si = "sars", scenario = 4, seed = 1464, overdisp = 5, method = "LPSMALA")
# suppressWarnings(gridExtra::grid.arrange(S8$inciplot, S8$Rlpsplot, S8$Repiestimplot, nrow = 1))
#
# # MERS serial interval (Scenario 9)
# S9 <- perfRestim(si = "mers", scenario = 5, days = 60,  seed = 1905, overdisp = 50)
# S9mcmc <- perfRestim(si = "mers", scenario = 5, days = 60,
# seed = 1905, overdisp = 50, method = "LPSMALA")
# suppressWarnings(gridExtra::grid.arrange(S9$inciplot, S9$Rlpsplot,
# S9$Repiestimplot, nrow = 1))
#
# #(Partially recovering Table 2 and Table 3 of Gressani et al. 2022)
# simsummary <- matrix(0, nrow = 36, ncol = 7)
# colnames(simsummary) <- c("Method", "Bias", "MSE", "CP90%", "CP95%",
#                           "CIwidth90%", "CIwidth95%")
# simsummary <- as.data.frame(simsummary)
#
# # Scenario 1
# simsummary[1,] <- c(rownames(S1$LPS),S1$LPS)
# simsummary[2,] <- c(rownames(S1mcmc$LPS),S1mcmc$LPS)
# simsummary[3,] <- c(rownames(S1$EpiEstim),S1$EpiEstim)
# simsummary[4,] <- rep("--",7)
# # Scenario 2
# simsummary[5,] <- c(rownames(S2$LPS),S2$LPS)
# simsummary[6,] <- c(rownames(S2mcmc$LPS),S2mcmc$LPS)
# simsummary[7,] <- c(rownames(S2$EpiEstim),S2$EpiEstim)
# simsummary[8,] <- rep("--",7)
```

```
# # Scenario 3
# simsummary[9,] <- c(rownames(S3$LPS),S3$LPS)
# simsummary[10,] <- c(rownames(S3mcmc$LPS),S3mcmc$LPS)
# simsummary[11,] <- c(rownames(S3$EpiEstim),S3$EpiEstim)
# simsummary[12,] <- rep("--",7)
# # Scenario 4
# simsummary[13,] <- c(rownames(S4$LPS),S4$LPS)
# simsummary[14,] <- c(rownames(S4mcmc$LPS),S4mcmc$LPS)
# simsummary[15,] <- c(rownames(S4$EpiEstim),S4$EpiEstim)
# simsummary[16,] <- rep("--",7)
# # Scenario 5
# simsummary[17,] <- c(rownames(S5$LPS),S5$LPS)
# simsummary[18,] <- c(rownames(S5mcmc$LPS),S5mcmc$LPS)
# simsummary[19,] <- c(rownames(S5$EpiEstim),S5$EpiEstim)
# simsummary[20,] <- rep("--",7)
# # Scenario 6
# simsummary[21,] <- c(rownames(S6$LPS),S6$LPS)
# simsummary[22,] <- c(rownames(S6mcmc$LPS),S6mcmc$LPS)
# simsummary[23,] <- c(rownames(S6$EpiEstim),S6$EpiEstim)
# simsummary[24,] <- rep("--",7)
# # Scenario 7
# simsummary[25,] <- c(rownames(S7$LPS),S7$LPS)
# simsummary[26,] <- c(rownames(S7mcmc$LPS),S7mcmc$LPS)
# simsummary[27,] <- c(rownames(S7$EpiEstim),S7$EpiEstim)
# simsummary[28,] <- rep("--",7)
# # Scenario 8
# simsummary[29,] <- c(rownames(S8$LPS),S8$LPS)
# simsummary[30,] <- c(rownames(S8mcmc$LPS),S8mcmc$LPS)
# simsummary[31,] <- c(rownames(S8$EpiEstim),S8$EpiEstim)
# simsummary[32,] <- rep("--",7)
# # Scenario 9
# simsummary[33,] <- c(rownames(S9$LPS),S9$LPS)
# simsummary[34,] <- c(rownames(S9mcmc$LPS),S9mcmc$LPS)
# simsummary[35,] <- c(rownames(S9$EpiEstim),S9$EpiEstim)
# simsummary[36,] <- rep("--",7)
# simsummary
```

`plot.Idist` *Plot the interval distribution from an Idist object*

Description

This routine plots the interval distribution based on an Idist object.

Usage

```
## S3 method for class 'Idist'
plot(x, barcol = "firebrick", denscol = "pink", denstransparent = 0.5,
 barwidth = 0.30, title = NULL, themetype = c("gray","classic","light","dark"),
 titlesize = 15, xtitlesize = 13, ytitlesize = 13, ...)
```

Arguments

x	An object of class Idist.
barcol	Color of the discretized interval distribution.
denscol	Color of the probability density function (pdf).
denstransparent	The transparency of the pdf.
barwidth	Width of the bars for the discrete distribution.
title	Title of the plot.
themetype	Theme of the plot.
titlesize	Size of the plot title. Default is 15.
xtitlesize	Size of title and text on x axis. Default is 13.
ytitlesize	Size of title and text on y axis. Default is 13.
...	Further arguments to be passed to plot.

Value

A plot of the interval distribution.

Author(s)

Oswaldo Gressani <oswaldo_gressani@hotmail.fr>

Examples

```
x <- Idist(mean = 3, sd = 1.6, dist = "weibull")
plot(x)
```

---

plot.incubestim *Plot the estimated incubation distribution*

Description

This routine can be used to plot the estimated incubation distribution based on an object of class incubestim.

Usage

```
## S3 method for class 'incubestim'
plot(x, type = c("pdf", "cdf", "incubwin"), ...)
```

Arguments

x	An object of class incubestim.
type	The type of plot. Default is "pdf" for the plot of the density function. Setting it to "cdf" returns the cumulative distribution function and "incubwin" gives a bar plot showing the width of the incubation window.
...	Further arguments to be passed to plot.

Value

- The probability density function of the estimated incubation period with 95% credible envelope.
- The cumulative distribution function of the estimated incubation period with 95% credible envelope.
- A bar plot showing the width of the incubation windows.

Author(s)

Oswaldo Gressani <oswaldo_gressani@hotmail.fr>

Examples

```
set.seed(123)
simdat <- incubsim(n = 30, tmax = 20) # Simulate incubation data
data <- simdat$Dobsincub              # Incubation bounds
incubfit <- estimIncub(x = data, niter = 500, tmax = 20, verbose = TRUE)
plot(incubfit)
```

---

plot.nowcasted *Plot of nowcasted cases and delay distribution*

Description

This routine plots the reported cases, the nowcasted cases and associated 95% credible interval for the nowcast. In addition, it can be used to show the contour plot of the estimated delay distribution.

Usage

```
## S3 method for class 'nowcasted'
plot(x, type = c("cases", "delay"), ...)
```

Arguments

x	An object of class nowcasted.
type	Either "cases" (default) to show the nowcasted cases or "delay" to show the contour plot of the estimated delay distribution.
...	Further arguments to be passed to plot.

Value

A plot of the reported and nowcasted cases.

Author(s)

Bryan Sumalinab (writing) and Oswaldo Gressani (editing).

`plot.Rt` *Plot the estimated reproduction number*

Description

This routine can be used to plot the estimated reproduction number based on an object of class `Rt`.

Usage

```
## S3 method for class 'Rt'
plot(x, datelab = "7d", cilevel = 0.95, col = "black", cicol = "gray",
 xtickangle = 0, legendpos = "right", title = "Estimated R",
 addfit = c("none","Cori","WT"), theme = "gray", timecut = 0,...)
```

Arguments

`x`	An object of class `Rt`.
`datelab`	Spacing for the ticks on the x-axis.
`cilevel`	Level of the credible interval.
`col`	Color of the fitted R_t curve for LPS.
`cicol`	Color for shading the credible envelope.
`xtickangle`	Angle of the x-ticks. Default is 0 (horizontal).
`legendpos`	Position of the legend.
`title`	Title of the plot.
`addfit`	Should an additional R_t fit be added?
`theme`	Theme, either "gray", "classic", "light", "dark"
`timecut`	Cut time points on plot.
`...`	Further arguments to be passed to plot.

Value

A plot of the fitted time-varying reproduction number.

Author(s)

Oswaldo Gressani <oswaldo_gressani@hotmail.fr>

Examples

```
si <- c(0.05, 0.05, 0.1, 0.1, 0.1, 0.1, 0.1, 0.05, 0.05, 0.1, 0.1, 0.1)
epidemic <- episim(si = si, Rpattern = 2, endepi = 30)
epifit <- estimR(incidence = epidemic$y, K = 30, si = si)
plot(epifit)
```

`plot.Rtnow` *Plot the nowcasted reproduction number*

Description

This routine can be used to plot the nowcasted reproduction number based on an object of class `Rtnow`.

Usage

```
## S3 method for class 'Rtnow'
plot(x, datelab = "7d", cilevel = 0.95, xtickangle = 0, legendpos = "right",
 nowcastcol = "cornflowerblue", ...)
```

Arguments

`x`	An object of class `Rtnow`.
`datelab`	Spacing for the ticks on the x-axis.
`cilevel`	Level of the credible interval.
`xtickangle`	Angle of the x-ticks. Default is 0 (horizontal).
`legendpos`	Position of the legend.
`nowcastcol`	Color of the nowcasted R_t curve.
`...`	Further arguments to be passed to plot.

Value

A plot of the nowcasted time-varying reproduction number.

Author(s)

Bryan Sumalinab (writing) and Oswaldo Gressani (editing).

Rmodelpriors *Prior specification for model hyperparameters*

Description

Specification of the hyperparameters for the Gamma prior on the roughness penalty parameter associated to the P-spline model and the Gamma prior on the overdispersion parameter of the negative binomial model underlying the incidence data.

Usage

```
Rmodelpriors(
 listcontrol = list(a_delta = 10, b_delta = 10, phi = 2, a_rho = 1e-04, b_rho = 1e-04)
)
```

Arguments

`listcontrol`	A list specifying the hyperparameters in the Gamma priors for the roughness penalty parameter of the P-spline model (named a_δ, b_δ and ϕ) and the overdispersion parameter of the negative binomial model for the incidence data (named a_ρ and b_ρ).

Value

A list with the specified hyperparameter components. By default, $a_\delta = b_\delta = 10$, $\phi = 2$ and $a_\rho = b_\rho = 10^{-4}$.

Author(s)

Oswaldo Gressani <oswaldo_gressani@hotmail.fr>

References

Gressani, O., Wallinga, J., Althaus, C. L., Hens, N. and Faes, C. (2022). EpiLPS: A fast and flexible Bayesian tool for estimation of the time-varying reproduction number. *Plos Computational Biology*, **18**(10): e1010618.

sars2003 *Daily incidence of the 2003 SARS epidemic in Hong Kong*

Description

A list with the daily incidence of onset of symptoms for the 2003 SARS outbreak in Hong Kong and a discretized serial interval distribution.

Usage

```
data(sars2003)
```

Format

A list with two components:

`incidence` A vector with 107 observations.

`si` A vector of probabilities corresponding to the serial interval distribution.

Source

https://cran.r-project.org/package=EpiEstim

References

Cori A. et al. (2009). Temporal variability and social heterogeneity in disease transmission: the case of SARS in Hong Kong. *Plos Computational Biology* **5**(8) : e1000471.

---

summary.Rt *Summarize the estimated reproduction number*

Description

This routine can be used to summarize estimation results for related to the reproduction number.

Usage

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

Arguments

`object`	An object of class `Rt`.
`...`	Further arguments to be passed.

Value

A summary output.

Author(s)

Oswaldo Gressani <oswaldo_gressani@hotmail.fr>

zika2015 *Data on the 2015 Zika virus disease in Colombia*

Description

A list containing incidence data for the 2015 Zika disease in Girardot (Colombia) from October 2015 to January 2016, a vector of dates and a discrete serial interval distribution.

Usage

```
data(zika2015)
```

Format

A list with three components:

`incidence` An incidence time series of length 93.

`dates` A vector of dates in "YYYY-MM-DD" format.

`si` A vector of probabilities corresponding to the serial interval distribution with a mean of 7 days and standard deviation of 1.5 days.

Source

https://cran.r-project.org/package=outbreaks

References

Rojas, D. P. et al. (2016). The epidemiology and transmissibility of Zika virus in Girardot and San Andres island, Colombia, September 2015 to January 2016. *Eurosurveillance*, **21**(28), 30283.

Index

∗ **datasets**
cov19incidence2022, 2
cov19mort2021, 3
eruptions, 5
influenza2009, 15
sars2003, 26
zika2015, 28

cov19incidence2022, 2
cov19mort2021, 3

epicurve, 3
episim, 4
eruptions, 5
estimIncub, 6
estimR, 7
estimRmcmc, 9

histosmooth, 11

Idist, 12
incubsim, 13
influenza2009, 15

nowcasting, 15
nowcastingR, 17

perfRestim, 18
plot.Idist, 21
plot.incubestim, 22
plot.nowcasted, 23
plot.Rt, 24
plot.Rtnow, 25

Rmodelpriors, 26

sars2003, 26
summary.Rt, 27

zika2015, 28