Package: doublIn (via r-universe)

October 18, 2024

Type Package

Title Estimate Incubation or Latency Time using Doubly Interval Censored Observations

Version 0.2.0

Maintainer Vera Arntzen <v.h.arntzen@math.leidenuniv.nl>

Description Visualize contact tracing data using a 'shiny' app and estimate the incubation or latency time of an infectious disease respecting the following characteristics in the analysis; (i) doubly interval censoring with (partly) overlapping or distinct windows; (ii) an infection risk corresponding to exponential growth; (iii) right truncation allowing for individual truncation times; (iv) different choices concerning the family of the distribution. For our earlier work, we refer to Arntzen et al. (2023) <doi:10.1002/sim.9726>. A paper describing our approach in detail will follow.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.2.3

Imports coda, flexsurv, ggplot2, rjags, magrittr, tidyverse, DT, epicontacts, lubridate, mStats, plotly, shiny, shinyWidgets, shinydashboard, shinythemes, visNetwork, xtable, dplyr, methods

Suggests testthat (>= 3.0.0), shinytest

Config/testthat/edition 3

NeedsCompilation no

Author Vera Arntzen [aut, cre]

Depends R (>= 3.5.0)

Repository CRAN

Date/Publication 2024-06-19 12:40:07 UTC

Contents

`Estimate_doublIn` *Estimate the incubation or latency time of an infectious disease, i.e. a doubly interval censored time-to-event*

Description

Estimate the distribution of doubly interval censored observations of time-to-event allowing for (i) constant risk of initial event within the window containing the time origin or a risk according to exponential growth (as for infection risk in the beginning of an outbreak); (ii) different shapes of the distribution (gamma, generalized gamma,Weibull); (iii) right truncation; (iv) (partial) overlap of the two windows. Provides estimates of the mean, median, 95th percentile and parameters, as well as diagnostics.

Usage

```
Estimate_doublIn(
  dat,
  infection_risk_distribution = "constant",
  exp_growth_rate = NULL,
  exp_growth_rate_SE = NULL,
  method = "GenGamma",
  percentiles = c(0.5, 0.9, 0.95, 0.99),
  right_truncation = FALSE,
  iters = 5000,
  burnin_period = 250,
  thin = 1,
  further_thin_plots = 10,
  plot_rm_burnin = TRUE
)
```

Arguments

`dat`	data.frame with one row per individual and the variables L0, L1, R0, R1 representing the left and right window containing the time origin and endpoint, respectively. When right truncation needs to be addressed, an additional variable Trunc is required.
`infection_risk_distribution`	either exponential growth ("exp_growth") or a constant risk of infection ("constant") is assumed within the exposure window.

`exp_growth_rate`
: when exponential growth is assumed, the estimated growth factor r.

`exp_growth_rate_SE`
: the Standard Error of the estimated growth factor.

`method`
: assumed distribution for the time-to-event; can be "gamma", "GenGamma" (generalized gamma) or "Weibull".

`percentiles`
: the percentiles of interest as a vector with probabilities.

`right_truncation`
: whether right truncation occurred in the data (T) or not (F); an additional variable 'Trunc' in the data represents the calendar truncation time.

`iters`
: the number of iterations for the MCMC chain.

`burnin_period`
: burnin_period, i.e. the number of initial iterationals to be removed before analyzing the chains.

`thin`
: a thinning factor, meaning that every so many iterations is saved.

`further_thin_plots`
: additional thinning factor for plots (default is 10).

`plot_rm_burnin`
: omits the burnin period from the diagnostic plots, as these iterations are removed from the actual analysis (default is T).

Details

The function estimates in the Bayesian framework, running JAGS via R and employing three parallel Markov Chain Monte Carlo chains per model. We extended the code by Charniga et al. (2022). The code for the diagnostic plots is written by Ronald Geskus.

Value

A list: the estimates including Gelman diagnostic criterion; the settings that were used to run the model; a diagnostic plot with the running quantiles per parameter; a diagnostic plot with the running parameter estimates.

Author(s)

Vera Arntzen, <v.h.arntzen@math.leidenuniv.nl>

References

Stacy, E. W., and G. A. Mihram, Parameter estimation for a generalized gamma distribution, Technometrics, 7 (3), 349–358, doi:10.1080/00401706.1965.10490268, 1965

Charniga, K., et al., Estimating the incubation period of monkeypox virus during the 2022 multinational outbreak, medRxiv, doi:10.1101/2022.06.22.22276713, 2022

LeBauer et al., Translating Probability Density Functions: From R to BUGS and Back Again, The R Journal, 2013

Plummer, M., JAGS user manual, 2017 https://people.stat.sc.edu/hansont/stat740/jags_user_manual.pdf

Rubio, J.F, The Generalised Gamma Distribution, 2020 https://rpubs.com/FJRubio/GG

Examples

```
# NB: the example takes a short while to run.

# Draw an exposure window width 1, 2, 3, 4, 5
L1 <- sample(1:5, 100, replace = TRUE)

# Draw the infection moment from a uniform distribution on (L0, L1)
L <- runif(100, 0, L1)

# Draw latency times (as estimated by Xin et al., 2022)
times <- rgamma(100, shape = 4.05, rate = 0.74)
R <- L + times

# Draw end of quarantine (last test moment)
Q <- L1 + sample( c(5, 10, 15, 20, 25), 100, replace = TRUE)

# Define the data set
mydat <- data.frame(R = R, L0 = 0, L1 = L1,
                    R0 = floor(R), R1 = floor(R + 1), Trunc = Q)

# Apply the truncation
mydat <- mydat[which( (mydat$R > mydat$Trunc) == FALSE), ]
mydat$R <- NULL

# If exposure ends after the last possible moment of the endpoint, end
# exposure earlier
mydat$L1 <- ifelse(mydat$L1 > mydat$R1, mydat$R1, mydat$L1)

# Run the model with truncation
 Estimate_doublIn(dat = mydat,
infection_risk_distribution = "constant",
method = "gamma", percentiles = c(0.5, 0.9, 0.95, 0.99),
right_truncation = TRUE, iters = 1000,
burnin_period = 10, thin = 1,
further_thin_plots = 1)
```

`Plot_doublIn` *Plot the incubation or latency time distribution*

Description

Plot the estimated probability density function using a doublIn output object.

Usage

```
Plot_doublIn(
  doublIn_obj,
```

```
  label_x = "Days since infection",
  label_y = "Probability",
  p_shaded = 2,
  p_linepieces = c(0.5, 0.9, 0.95)
)
```

Arguments

doublIn_obj	output list from function Estimate_doublIn containing parameter estimates.
label_x	Label for the x-axis.
label_y	Label for the y-axis.
p_shaded	Which area needs to be shaded, i.e. enclosed by which percentile from the vector p_linepieces (1, 2 or 3).
p_linepieces	Percentiles to indicate by vertical line pieces, given as vector of length three with probabilities between 0 and 1.

Details

The function plots the estimated probability density function using the parameterisation of the generalized gamma distribution by Stacy et al. that includes the gamma and Weibull distributions as a special case.

Value

Plot of the estimated probability density function.

Author(s)

Vera Arntzen, <v.h.arntzen@math.leidenuniv.nl>

References

Stacy, E. W., and G. A. Mihram, Parameter estimation for a generalized gamma distribution, Technometrics, 7 (3), 349–358, doi:10.1080/00401706.1965.10490268, 1965

Visualize_contact_tracing_data

Visualize contact tracing data

Description

Run an application that visualizes contact tracing data that can be used to estimate incubation and latency time using the doublIn package.

Usage

```
Visualize_contact_tracing_data()
```

Details

We have used this app to visualize contact tracing data that was collected using a tailor-made contact tracing form. For details and source code, have a look at the Github page of Manh Nguyen Duc.

Value

R Shiny application.

Author(s)

Vera Arntzen, <v.h.arntzen@math.leidenuniv.nl>

References

Manh Nguyen Duc (2024) Contact tracing form using KoboToolbox https://github.com/manhnguy/Contact-Tracing-for-Respiratory-Transmitted-Diseases

Examples

```
# To run the app:
if(interactive()){Visualize_contact_tracing_data()}
```

Index