Package 'Rsero'

June 21, 2024

Type Package

Title Estimate the annual force of infection using serological data

Version 2.0

Author Nathanael Hoze

Maintainer Nathanael Hoze <nathanael.hoze@gmail.com>

Description Fit models of the force of infection to serological data using Bayesian MCMC methods. The package provides a standardized format for handling serological data, contains various serocatalytic models of force of infection, fits the models to the data, and provides tools to analyse and plot the results of the fits.

License What license is it under?

Depends R (>= 4.0.2), Rcpp (>= 0.12.13), corrplot, ggplot2, binom, methods, bayesplot, rstan, rstantools, shinystan, shiny, httpuv, loo

LinkingTo StanHeaders (>= 2.1.0), rstan (>= 2.32.2), BH (>= 1.65.0.1), Rcpp (>= 0.12.13), RcppEigen (>= 0.3.3.3.0)

RoxygenNote 7.3.0

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

URL https://github.com/nathoze/Rsero

RemoteType local

RemoteUrl /Users/nathanaelhoze/Dropbox/Scientific_Projects/Rsero

NeedsCompilation yes

R topics documented:

Rsero-package   *Rsero: Estimate the annual force of infection using serological data*

Description

Fit models of the force of infection to serological data using Bayesian MCMC methods. The package provides a standardized format for handling serological data, contains various serocatalytic models of force of infection, fits the models to the data, and provides tools to analyse and plot the results of the fits.

See Also

Useful links:

- <https://github.com/nathoze/Rsero>

combine_surveys   *Combine serological surveys*

Description

The function combine_surveys creates a new object of the class SeroData from two SeroData objects. In particular, it allows combining serological surveys sampled at different dates.

Usage

```
combine_surveys(SeroData1, SeroData2)
```

Arguments

SeroData1	A first SeroData object.
SeroData2	A second SeroData object.

Value

A SeroData object.

Author(s)

Nathanael Hoze <nathanael.hoze@gmail.com>

See Also

[SeroData](#) Define the format of the serological data.

[subset](#) Extract a smaller subset of a SeroData object.

Examples

```
# Combine two simulated surveys, one acquired in 2015, with one-year age categories,
# and another one acquired in 1995 with 10-year age categories.
Years <- c(1976,1992)
FOI <- c(0.2,0.3)
data1 <- simulate_SeroData(sampling_year = 2015, epidemic_years = Years, foi = FOI)
data2 <- simulate_SeroData(sampling_year = 1995,age_class=10, epidemic_years = Years, foi = FOI)
data <- combine_surveys(data1,data2)
```

---

compute_information_criteria

Compute the AIC, DIC and WAIC

Description

This function returns the Aikake Information Criterion (AIC), the Deviance Information Criterion (DIC) and the Watanabe Aikake Information Criterion (WAIC) from a fit of the class [FOIfit](#).

Usage

```
compute_information_criteria(FOIfit, ...)
```

Arguments

FOIfit   A FOIfit object

Value

A list with the class information_criteria, which contains the following items:

- AIC: The Aikake Information Criterion.
- DIC: The Deviance Information Criterion.
- WAIC: The Wakanabe-Aikake Information Criterion.
- MLE: The Maximum-likelihood, estimated as the maximal value of the likelihood in the drawn samples, used in the AIC.
- k: Number of parameters, used in the AIC.
- Dbar: Mean deviance, used in the DIC.
- pD: Effective number of parameters of the model, used in the DIC.
- pwaic: Estimated effective number of parameters, used in the WAIC.
- lpd: log pointwise predictive density, used in the WAIC.

Author(s)

Nathanael Hoze <nathanael.hoze@gmail.com>

References

See Gelman et al., Understanding predictive information criteria for Bayesian models. Stat Comput (2014) 24:997-1016, DOI 10.1007/s11222-013-9416-2

Examples

```
data('one_peak_simulation')
model <- FOImodel(type='outbreak', K=1)
F1  = fit(model = model, data = data )
compute_information_criteria(FOIfit  = F1)
```

---

correlation_posterior   *Plot the correlation matrix of the output*

Description

Compute and plot the correlation matrix of the parameters obtained from a MCMC fit. It returns a n-by-n matrix, where n is the number of parameters.

Usage

```
correlation_posterior(FOIfit, show_coefficients = TRUE)
```

Arguments

FOIfit   An object of the class FOIfit.

show_coefficients

                    Boolean. If TRUE, shows the value of the correlation coefficient on the plotted
                    matrix. Default = TRUE.

Value

The correlation matrix.

Author(s)

Nathanael Hoze <nathanael.hoze@gmail.com>

Examples

```
data('two_peaks_simulation')
model <- FOImodel('outbreak',K=2, background = 1)
F <- fit(data=data, model=model)
M <- correlation_posterior(F)
```

---

fit | *Fit force of infection models to serological data*

Description

Runs a Bayesian MCMC using rstan to draw samples of a model of the class FOImodel, with data of the class SeroData. It is an adaptation of the function [sampling](#) in the rstan package.

Usage

```
fit(
  model,
  data = data,
  iter = 5000,
  chains = 4,
  warmup = floor(iter/2),
  thin = 1,
  seed = sample.int(.Machine$integer.max, 1),
  init = "random",
  check_data = TRUE,
  sample_file = NULL,
  diagnostic_file = NULL,
  verbose = FALSE,
  algorithm = c("NUTS", "HMC", "Fixed_param"),
  control = NULL,
  include = TRUE,
  cores = getOption("mc.cores", 1L),
 open_progress = interactive() && !isatty(stdout()) && !identical(Sys.getenv("RSTUDIO"),
    "1"),
  show_messages = TRUE,
  ...
)

## S3 method for class 'FOIfit'
print(x, digits_summary = 2, ...)

## S3 method for class 'FOIfit'
summary(x, ...)
```

Arguments

model	A FOImodel object, generated by the function [FOImodel](#).
data	A SeroData object, generated by the function [SeroData](#).
iter	integer. Number of iterations for each chain, including warmup. Default = 5000.
chains	integer. Number of independent Markov chains. Default = 4.
warmup	A positive integer specifying the number of warmup (aka burnin) iterations per chain. If step-size adaptation is on (which it is by default), this also controls the number of iterations for which adaptation is run (and hence these warmup samples should not be used for inference). The number of warmup iterations should not be larger than iter and the default is iter/2.

thin   A positive integer specifying the period for saving samples. The default is 1, which is usually the recommended value.

seed   The seed for random number generation. The default is generated from 1 to the maximum integer supported by R on the machine. Even if multiple chains are used, only one seed is needed, with other chains having seeds derived from that of the first chain to avoid dependent samples. When a seed is specified by a number, as.integer will be applied to it. If as.integer produces NA, the seed is generated randomly. The seed can also be specified as a character string of digits, such as "12345", which is converted to integer.

init   Initial values specification. See the detailed documentation for the init argument in stan.

check_data   Logical, defaulting to TRUE. If TRUE the data will be preprocessed; otherwise not. See the Note section in stan.

sample_file   An optional character string providing the name of a file. If specified the draws for *all* parameters and other saved quantities will be written to the file. If not provided, files are not created. When the folder specified is not writable, tempdir() is used. When there are multiple chains, an underscore and chain number are appended to the file name prior to the .csv extension.

diagnostic_file

                        An optional character string providing the name of a file. If specified the diagnostics data for *all* parameters will be written to the file. If not provided, files are not created. When the folder specified is not writable, tempdir() is used. When there are multiple chains, an underscore and chain number are appended to the file name prior to the .csv extension.

verbose   TRUE or FALSE: flag indicating whether to print intermediate output from Stan on the console, which might be helpful for model debugging.

algorithm   One of sampling algorithms that are implemented in Stan. Current options are "NUTS" (No-U-Turn sampler, Hoffman and Gelman 2011, Betancourt 2017), "HMC" (static HMC), or "Fixed_param". The default and preferred algorithm is "NUTS".

control   A named list of parameters to control the sampler's behavior. See the details in the documentation for the control argument in stan.

include   Logical scalar defaulting to TRUE indicating whether to include or exclude the parameters given by the pars argument. If FALSE, only entire multidimensional parameters can be excluded, rather than particular elements of them.

cores   Number of cores to use when executing the chains in parallel, which defaults to 1 but we recommend setting the mc.cores option to be as many processors as the hardware and RAM allow (up to the number of chains).

open_progress   Logical scalar that only takes effect if cores > 1 but is recommended to be TRUE in interactive use so that the progress of the chains will be redirected to a file that is automatically opened for inspection. For very short runs, the user might prefer FALSE.

show_messages   Either a logical scalar (defaulting to TRUE) indicating whether to print the summary of Informational Messages to the screen after a chain is finished or a character string naming a path where the summary is stored. Setting to FALSE is not recommended unless you are very sure that the model is correct up to numerical error.

Value

A list with the class `FOIfit`, which contains the following items:

- fit: The results of the fit, of class `stanfit`.
- data: The input data.
- model: The input model.

Author(s)

Nathanael Hoze <nathanael.hoze@gmail.com>

See Also

[SeroData](#) Define the format of the serological data.

[FOImodel](#) Define a model.

Examples

```
data <- simulate_SeroData(number_samples = 1000,
  age_class = 1,
  epidemic_years = c(1976,1992),
  foi = c(0.2,0.3))
model <- FOImodel('outbreak', K = 2)
F1 <- fit(model = model, data = data)
seroprevalence.fit(F1)
plot(F1)
```

FOImodel   *Model of the force of infection*

Description

This function creates an object of the class `FOImodel`. Inputs are the type of the model (required), additional parameters if required by the model, and hyperparameters for the prior distributions (optional). The models to be given as an input are predefined. More details are given in the vignette `models`.

Usage

```
FOImodel(
  type = "constant",
  K = 1,
  group_size = 1,
  seroreversion = 0,
  se = 1,
  sp = 1,
  prioralpha1 = 0,
  prioralpha2 = 5,
  priorbeta1 = 0,
  priorbeta2 = 1,
```

```
    priorT1 = 1,
    priorT2 = 100,
    priorC1 = 0,
    priorC2 = 10,
    priorY1 = 0,
    priorY2 = 10,
    priorRho = 2,
    cat_lambda = 1,
    fixed_parameters = NULL,
    ...
)

## S3 method for class 'FOImodel'
print(x, ...)
```

Arguments

type	A character with the name of model. The options are:

- 'constant': Constant force of infection
- 'outbreak': Series of outbreak modeled with gaussians
- 'independent': Annual independent values of the force of infection
- 'piecewise': Piecewise constant force of infection. The number of phases with a constant level is given by the variable K
- 'constantoutbreak': A combination of K outbreaks with a constant yearly force of infection
- 'independent_group': Similar to the independent model, but with piecewise constant values of the force annual force of infection in time periods of length `group_size` years.

K	integer. An additional parameter used in the outbreak, constantoutbreak and piecewise stan models. In the case of the outbreaks, this parameter is the number of Gaussians used. In the case of the piecewise constant model it is the number of constant phases. Default = 1.
group_size	integer. An additional parameter used in the independent_group models. The force of infection is averaged over `group_size` year period. By default `group_size` = 1, which is equivalent ot the independent model.
seroreversion	integer, equal to 0 or 1. If `seroreversion=0` the model includes a rate of seroreversion (waning immunity). See the vignette `models` for details. Default = 0.
se	numeric, between 0 and 1. If `se=1` the assay has a perfect sensitivity. Default = 1.
sp	numeric, between 0 and 1. If `sp=1` the assay has a perfect specificity. Default = 1.
prioralpha1	First parameter of the uniform prior distribution of the parameter alpha, used as the intensity of the force of infection in the outbreak models. Default = 0.
prioralpha2	Second parameter of the uniform prior distribution of the parameter alpha, used as the intensity of the force of infection in the outbreak models. Default = 5.
priorbeta1	First parameter of the uniform prior distribution of the parameter beta, used as the spread of the force of infection in the outbreak models. Default = 0.
priorbeta2	Second parameter of the uniform prior distribution of the parameter beta, used as the spread of the force of infection in the outbreak models. Default = 1.

priorT1	First parameter for the uniform distribution for the T parameter. The time of the outbreak is defined as the number of years before the survey (outbreak and constant outbreak models). It is the time interval for the change of FOI in the piecewise constant model. Default = 1.
priorT2	Second parameter for the uniform distribution for the T parameter. Default = 100.
priorC1	First parameter of the uniform prior distribution for the constant force of infection, used in the constant and piecewise constant models. Default = 0.
priorC2	Second parameter of the uniform prior distribution for the constant force of infection, used in the constant and piecewise constant models. Default = 10.
priorY1	First parameter of the uniform prior distribution for the annual hazard of infection, used in the independent models. Default = 0.
priorY2	Second parameter of the uniform prior distribution for the annual hazard of infection, used in the independent models. Default = 10.
priorRho	Parameter of the exponential prior distribution for rho, the seroreversion rate used when seroreversion=1. Default = 2. ##'
cat_lambda	integer, equal to 0 or 1. If `cat_lambda`=1 the force of infection varies accross the different categories defined in an objet `SeroData`. See the vignette `models` for details. Default = 1.
...	Additional arguments (not used).
x	A object of the class `FOImodel`.

Value

A list with the class `FOImodel`, which contains the following items:

- type: The type of the model.
- stanname: The name of the stan file used.
- estimated_parameters: The number of estimated parameters.
- priors: a list with the priors.
- K: the input parameter used in the outbreak and piecewise models, if given.

Author(s)

Nathanael Hoze <nathanael.hoze@gmail.com>

Examples

```
## A gaussian model, with two gaussians, and user-defined priors
model <- FOImodel('outbreak',K = 2, prioralpha1 = 0, prioralpha2 = 1)

## A piecewise model, consisting in two constant phases, with seroreversion and user-defined priors
model <- FOImodel('piecewise', K=2, seroreversion=1, priorRho1=0.1)
```

parameters_credible_intervals

Analyse a FOIfit

Description

Determines the mean and quantiles of the parameters estimated in the fit. By default, it returns the mean, the median and 95 % credible interval of the parameters.

Usage

```
parameters_credible_intervals(FOIfit, quants = c(0.025, 0.5, 0.975))
```

Arguments

FOIfit   An object of the class FOIfit.

quants   Numeric. Contains the list of the estimated quantiles given between 0 and 1.
                Default parameters are the 2.5%, 50 % and 97.5 % quantiles given by quants =
                c(0.025,0.5,0.975).

Value

A dataframe containing the 2.5 In the case of an outbreak model (outbreak or outbreak+constant), the total force of infection is given by alpha, and the probability of infection by "Outbreak Prob. Inf.". For models with constant phases, the annual probability of infection is given.

Author(s)

Nathanael Hoze <nathanael.hoze@gmail.com>

Examples

```
 data("1peakSimulation")
 model <- FOImodel(type='outbreak', seroreversion=TRUE, K=1)
 options(mc.cores = parallel::detectCores())
 Fit <- fit(data = data, model = model)
 parameters_credible_intervals(Fit)

# an example where other quantiles are specified
 data("1peakSimulation")
 model <- FOImodel(type='outbreak', seroreversion=TRUE, K=1)
 options(mc.cores = parallel::detectCores())
 Fit <- fit(data = data, model = model)
 parameters_credible_intervals(Fit, quants=c(0.25,0.5,0.75))
```

Description

Plot the mean and credible interval of the yearly force of infection, given an object of the class `FOIfit`. The function `plot` uses the package `ggplot2`. It returns a graphical object that can be further modified.

Usage

```
## S3 method for class 'FOIfit'
plot(
  FOIfit,
  mean_only = FALSE,
  individual_samples = 0,
  YLIM = 1,
  XLIM1 = NULL,
  XLIM2 = NULL,
  ...
)
```

Arguments

`FOIfit`	The `FOIfit` object to be plotted.
`mean_only`	Boolean. If `FALSE`, plot the mean force of infection as well as an envelope of the 95% credible interval. Default = `FALSE`.
`individual_samples`	
	Integer. Number of individual samples to be plotted additionally to the mean and the credible interval of the force of infection. The `individual_samples` samples are randomly chosen in the chains. Default = 0.
`YLIM`	Upper limit of the y-axis. Default = 1. The lower limit is set to 0.

Value

A list of `ggplot2` objects.

Author(s)

Nathanael Hoze <nathanael.hoze@gmail.com>

Examples

```
data <- simulate_SeroData( max_age = 50, epidemic_years = c(1976,1992), foi = c(0.2,0.3))
model <- FOImodel(type='outbreak', K = 2)
Fit <- fit(model = model, data = data)
p <- plot(Fit)
p+ylim(0,1)
p[[1]]$category # the name of the category plotted
```

---

plot_posterior   *Plot posterior distribution*

Description

Plot the posterior distribution of the parameters estimated in a `FOIfit`.

Usage

```
plot_posterior(FOIfit)
```

Arguments

FOIfit   An object of the class `FOIfit`.

Value

A list with the posterior plots. These plots are generated with the `ggplot2` package and can be modified.

Author(s)

Nathanael Hoze <nathanael.hoze@gmail.com>

Examples

```
# Here describe how to access the plots
P <- plot_posterior(FOIfit)
P[[1]]+ylim(0,200)
```

SeroData   *Create a dataset containing the serological data and other information*

Description

This function contains the definition of the class `SeroData`. It requires the age and seropositivity of individuals, and accepts more input parameters.

Usage

```
SeroData(
  age_at_sampling,
  age = NULL,
  Y,
  age_class = 1,
  max_age = NULL,
  sampling_year = NULL,
  location = NULL,
  sex = NULL,
```

```
    category = "Category 1",
    reference.category = NULL,
    class1 = NULL,
    class2 = NULL,
    ...
)
```

```
compute.age.groups(age, sampling_year)
```

Arguments

age_at_sampling
: A vector of integers containing the age of the sampled individuals at the time of sampling. Must be equal or greater than 1.

Y
: A vector containing the seropositivy status of the sampled individuals. It can be in a numeric form (1 or 0) or boolean (TRUE or FALSE). This vector must have the same size as age.

age_class
: Integer. The length in years of the age classes. Default = 1.

max_age
: Integer. The maximal age considered for the individuals in the sample. Individuals older than max_age are set equal to max_age.

sampling_year
: Integer. Defines the sampling year. It can be a single value or a vector of the same size as the number of sampled individuals. If it is a single value, all the sampled individuals have the same sampling year. Default = 2017.

location
: An optional character factor defining the sampling location. It can be a single chain of characters or a vector of the same size as the number of sampled individuals. Default = NULL.

sex
: An optional character factor defining the sex of the individuals. It can be a single chain of characters or a vector of the same size as the number of sampled individuals. Default = NULL.

category
: Character. An optional element containing the name of the categories and defining the category of the individuals. This feature is used when fitting the models assuming different risks of infection for the different categories. It can be a single character element or a matrix of characters with row size equal to the number of individuals and column size equal to the number of different category classes. Default = "Category 1".

reference.category
: Character. A vector containing the name of the reference categories. It must have the same length as the number of columns of category. By default, it will take as reference the most common element in each category.

...
: Additional arguments (not used).

Value

A list with the class SeroData, which contains the following items:

- age: A vector of integers containing the age of the sampled individuals at the time of the latest sampling, of length N.

- age_at_sampling: A vector of integers containing the age of the sampled individuals at the time of sampling, of length N.

- Y: A vector of seropositivity status, of length N.

- N: The number of individuals considered.
- A: The maximal age.
- NGroups: The total number of age groups.
- sampling_year: A vector of sampling years, of length N.
- location: A vector of the sampling location, of length N.
- sex: A vector of the sex of each individual, of length N.

Author(s)

Nathanael Hoze `<nathanael.hoze@gmail.com>`

Examples

```
## A very simple example of a serological survey with three individuals:
data = SeroData(age_at_sampling = c(10,32,24), Y=c(0,1,1), max_age = 50, age_class = 1, sampling_year = 2017)

seroprevalence(data = data)
data2 = subset(data,c(1,3))
seroprevalence(data = data2, age_class = 5)

## A example using categories
sex= c('male', 'male', 'female')
data = SeroData(age_at_sampling = c(10,32,24), Y=c(0,1,1),  category= sex )

# defining the reference category 'female'

data = SeroData(age_at_sampling = c(10,32,24), Y=c(0,1,1), category= sex, reference.category='female')

# Grouping individuals by age categories 1-10, 11-20, 21-30, etc.
data = SeroData(age_at_sampling = c(10,32,24), Y=c(0,1,1), category= sex, reference.category='female', age_cl
```

---

seroprevalence	*Compute the seroprevalence*

Description

Compute the seroprevalence using an object of the class `SeroData`. If multiple `category` are defined, it will also compute the seroprevalence for each category.

Usage

```
seroprevalence(serodata)
```

Arguments

serodata   An object of the class `SeroData`.

Value

The mean and 95 % confidence interval of the seroprevalence

Author(s)

Nathanael Hoze <nathanael.hoze@gmail.com>

Examples

```
## A very simple example of a serological survey with three individuals:
data = SeroData(age_at_sampling = c(10,32,24), Y=c(0,1,1), max_age = 50, age_class = 1, sampling_year = 2017)
seroprevalence(serodata = data)
```

---

seroprevalence.fit   *Plot the fit of the seroprevalence vs. age*

Description

Plot the mean and credible interval of the posterior of the seroprevalence, given an object of the class FOIfit. See function [seroprevalence](#) for plotting the seroprevalence from the data. The function returns graphical objects that can be further modified.

Usage

```
seroprevalence.fit(
  FOIfit,
  individual_samples = 0,
  age_class = 10,
  YLIM = 1,
  ...
)
```

Arguments

FOIfit   The FOIfit object to be plotted.

individual_samples

                  Integer. Number of individual samples to be plotted additionally to the mean and
                  the credible interval of the force of infection. The indivual_samples samples
                  are randomly chosen in the chains. Default = 0.

age_class   Integer. Length of the age groups (in years). Used if show_data = TRUE. Default
                  = 10.

YLIM   Upper limit of the y-axis. Default = 1. The lower limit is set to 0.

show_data   Boolean. If TRUE, shows the fraction of seropositive as well. By default, ages
                  are associated in groups of age_class years. Default = TRUE.

Value

A list of ggplot2 objects.

Author(s)

Nathanael Hoze <nathanael.hoze@gmail.com>

Examples

```
data <- simulate_SeroData( max_age = 50, epidemic_years = c(1976,1992), foi = c(0.2,0.3))
model <- FOImodel(type='outbreak', background=TRUE, K = 2)

Fit <- fit(model = model, data = data)

# plot the fraction of seropositive, with 2 individual samples
p <- seroprevalence.fit(Fit,  individual_samples = 2,YLIM=0.2)
```

---

seroprevalence.plot   *Plot the seroprevalence by age*

Description

Plot the mean and 95 % confidence interval of the seroprevalence by age-class. Accepts as an input a SeroData object. If multiple category are defined, it will also compute the seroprevalence for each category.

Usage

```
seroprevalence.plot(serodata, age_class = 10, YLIM = 1, ...)
```

Arguments

serodata	An object of the class SeroData.
age_class	Integer. The length in years of the age classes. Default = 10.
YLIM	Upper limit of the y-axis. Default = 1. The lower limit is set to 0.

Value

a list with plots of the seroprevalence for each category, or one plot if only one category is defined.

Author(s)

Nathanael Hoze <nathanael.hoze@gmail.com>

Examples

```
dat  = data("one_peak_simulation")
seroprevalence.plot(serodata = dat)
```

simulate_SeroData   *Simulate a serological survey*

Description

Simulate a serological survey, based on years of epidemics and force of infection.

Usage

```
simulate_SeroData(
  number_samples = 500,
  equal_number = FALSE,
  age_class = 1,
  max_age = 70,
  sampling_year = 2019,
  epidemic_years = c(2000, 1987, 1982),
  foi = c(0.3, 0.2, 0.4),
  location = NULL,
  sex = NULL,
  category = "Category 1",
  se = 1,
  sp = 1,
  rho = 0
)
```

Arguments

number_samples	integer. The number of individuals in the simulated serological survey. Default = 500.
equal_number	boolean. If FALSE, draw random ages with a uniform distribution between 1 and max_age. if TRUE, generates a SeroData object with an equal distribution of the ages. Default = FALSE.
age_class	integer. The length in years of the age classes. Default = 1.
max_age	integer. The maximal age. Default = 70.
sampling_year	integer. Default = 2019.
epidemic_years	numeric. A vector with the years where the force of infection is positive. The force of infection at the non specified years is equal to zero. Default = c(2000,1987,1982).
foi	numeric. The force of infection at the years defined in epidemic_years. It must be the same size as epidemic_years. Default = c(0.3,0.2,0.4).
location	An optional character factor defining the sampling location. It can be a single chain of characters or a vector of the same size as the number of sampled individuals. Default = NULL.
sex	An optional character factor defining the sex of the individuals. It can be a single chain of characters or a vector of the same size as the number of sampled individuals. Default = NULL.
se	numeric, between 0 and 1. Sensitivity of the assay. Default = 1.
sp	numeric, between 0 and 1. Specificity of the assay. Default = 1.
rho	numeric, 0 or positive. Seroreversion rate. Default = 0.

Value

An object of the class SeroData.

Author(s)

Nathanael Hoze <nathanael.hoze@gmail.com>

See Also

[SeroData](#) Define the format of the serological data.

Examples

```
## Example 1: Simulates a serological survey taken in a population that experienced
## a series of three outbreaks in 1972, 1988 and 1996.

data = simulate_SeroData(number_samples = 1000,
age_class = 1,
sampling_year = 2015,
max_age = 50,
epidemic_years = c(1996,1988,1972),
foi = c(0.2,0.1,0.3),
se = 1)

seroprevalence(data)


## Example 2: 500 individuals sampled in 2019, in a population that experienced two epidemics, in 1962
## and 2012.
## The assay has a sensitivity of 0.9.
years =  seq(1962,2012)
alpha=c(0.25, 0.1)
T=c(1974, 2000)
beta=c(1,0.5)
FOI = alpha[1]*exp(-(years-T[1])^2/beta[1]^2) + alpha[2]*exp(-(years-T[2])^2/beta[2]^2)
data1 <- simulate_SeroData( epidemic_years = years,foi=FOI, se=0.9)
seroprevalence(data1)

## Adding another survey, sampled in 2005, where ages are categorized in 5 year groups:
data2 <- simulate_SeroData(age_class = 5,
 epidemic_years = years,
 foi=FOI,
 se = 0.9,
 sampling_year = 2005)
data <- combine_surveys(data1,data2)
seroprevalence(data)
```

---

subset.SeroData   *Creates a subset of a serological survey*

Description

This function creates a serological survey of the class `SeroData`.

Usage

```
## S3 method for class 'SeroData'
subset(data, sub = seq(1, data$N), loc = NULL, category = NULL)
```

Arguments

sub	Used in the `subset` function. A list of the indices of the individuals to be considered in the new `SeroData` object.
loc	Used in the `subset` function. A list of the `location` of the individuals to be considered in the new `SeroData` object.
sex	An optional character factor defining the sex of the individuals. It can be a single chain of characters or a vector of the same size as the number of sampled individuals. Default = `NULL`.
...	Additional arguments (not used).

Value

A list with the class `SeroData`, which contains the following items:

- age: A vector of integers containing the age of the sampled individuals at the time of the latest sampling, of length N.

- age_at_sampling: A vector of integers containing the age of the sampled individuals at the time of sampling, of length N.

- Y: A vector of seropositivity status, of length N.

- N: The number of individuals considered.

- A: The maximal age.

- NGroups: The total number of age groups.

- sampling_year: A vector of sampling years, of length N.

- location: A vector of the sampling location, of length N.

- sex: A vector of the sex of each individual, of length N.

Author(s)

Nathanael Hoze <nathanael.hoze@gmail.com>

Examples

```
## A very simple example of a serological survey with three individuals:
data = SeroData(age_at_sampling = c(10,32,24), Y=c(0,1,1), max_age = 50, age_cats = 1, sampling_year = 2017)

seroprevalence(data = data)
data2 = subset(data,c(1,3))

# add example with subset(data, location)
```

---

traceplot_Rsero   *Markov chain Traceplot*

Description

Draw the traceplot corresponding to one or more Markov chains, providing a visual way to inspect sampling behavior and assess mixing across chains and convergence. It is an adaptation of the function traceplot in the rstan package.

Usage

```
traceplot_Rsero(
  FOIfit,
  pars = NULL,
  include = TRUE,
  unconstrain = FALSE,
  inc_warmup = FALSE,
  window = NULL,
  nrow = NULL,
  ncol = NULL,
  ...
)
```

Arguments

FOIfit	A FOIfit object.
pars	A character vector of parameter names. Defaults to relevant parameters of the model.
include	Should the parameters given by the pars argument be included (the default) or excluded from the plot? Only relevant if pars is not missing.
unconstrain	Should parameters be plotted on the unconstrained space? Defaults to FALSE.
inc_warmup	TRUE or FALSE, indicating whether the warmup sample are included in the trace plot; defaults to FALSE.
window	A vector of length 2. Iterations between window[1] and window[2] will be shown in the plot. The default is to show all iterations if inc_warmup is TRUE and all iterations from the sampling period only if inc_warmup is FALSE. If inc_warmup is FALSE the iterations specified in window should not include iterations from the warmup period.
nrow,	ncol Passed to facet_wrap.
...	Optional arguments to pass to geom_path (e.g. size, linetype, alpha, etc.).
object	An instance of class stanfit.

Author(s)

Nathanael Hoze <nathanael.hoze@gmail.com>

Examples

```
##'   data <- simulate_SeroData(number_samples = 1000,
  age_class = 1,
  epidemic_years = c(1976,1992),
  foi = c(0.2,0.3))
model <- FOImodel('outbreak', K = 2)
F1 <- fit(model = model, data = data)
traceplot_Rsero(F1)
```

Index