Package 'disbayes'

October 13, 2022

Title Bayesian Multi-State Modelling of Chronic Disease Burden Data

Date 2022-08-18

Version 1.0.0

Description Estimation of incidence and case fatality for a chronic disease, given partial information, using a multi-state model. Given data on age-specific mortality and either incidence or prevalence, Bayesian inference is used to estimate the posterior distributions of incidence, case fatality, and functions of these such as prevalence. The methods are described in Jackson et al. (2021) <arXiv:2111.14100>.

License GPL-3

Encoding UTF-8

LazyData true

Biarch true

Depends R (>= 3.5.0)

Imports dplyr, tidyr, magrittr, tibble, generics, methods, Rcpp (>= 0.12.0), rstan (>= 2.18.1), mgcv, SHELF, ggplot2, loo, matrixStats

LinkingTo BH (>= 1.66.0), Rcpp (>= 0.12.0), RcppParallel, RcppEigen (>= 0.3.3.3.0), rstan (>= 2.18.1), StanHeaders (>= 2.18.0)

Suggests knitr, rmarkdown, rstantools (>= 2.0.0.9000), tempdisagg, testthat

VignetteBuilder knitr

SystemRequirements GNU make

URL https://chjackson.github.io/disbayes/

BugReports https://github.com/chjackson/disbayes/issues

RoxygenNote 7.2.0

NeedsCompilation yes

Author Christopher Jackson [aut, cre, cph] (<https://orcid.org/0000-0002-6656-8913>)

Maintainer Christopher Jackson <chris.jackson@mrc-bsu.cam.ac.uk>

Repository CRAN

Date/Publication 2022-08-22 09:50:02 UTC

R topics documented:

`disbayes-package` *The 'disbayes' package.*

Description

Bayesian evidence synthesis for chronic disease epidemiology

References

Stan Development Team (2019). RStan: the R interface to Stan. R package version 2.19.2. https://mc-stan.org

---

`ci2num` *Convert a proportion and credible interval to a numerator and denominator*

Description

Estimate the number of events and denominator that contain roughly equivalent information to an estimate and uncertainty interval for a proportion, by interpreting the estimate and interval as a Beta posterior arising from a vague Beta(0.5,0.5) prior updated with the data consisting of that number and denominator.

Usage

```
ci2num(est, lower, upper, epsilon = 0.5, denom0 = 1000)
```

Arguments

est	Point estimate
lower	Lower 95% credible limit
upper	Upper 95% credible limit
epsilon	If any of lower are zero, then they are replaced by the minimum of epsilon and est/2. Similarly values of 1 for upper are replaced by the maximum of 1-epsilon and (1+est)/2.
denom0	Denominator to use as a default when the point estimate is exactly 0 or 1 (which is not compatible with the beta distribution). Should correspond to a guess of the population size used to produce the estimate, which should be no greater than the actual population of the area, and usually less. Should be either a scalar, or a vector of the same length as est (though note if it is a vector, then only the elements where est is 1 or 0 get used).

Details

Based on fitting a Beta distribution by least squares, using the method provided by the **SHELF** package.

Requires that the estimate and upper and lower limits are all distinct (except that est=0 is allowed and handled specially for convenience, see denom0). Vectors of estimates and limits may be supplied.

Value

A data frame with elements num and denom corresponding to the supplied estimate and limits.

References

Oakley (2020). SHELF: Tools to Support the Sheffield Elicitation Framework. R package version 1.7.0. https://CRAN.R-project.org/package=SHELF

Examples

```
est <- 3.00 / 100
upper <- 3.52 / 100
lower <- 2.60 / 100
ci2num(est, lower, upper)
```

`conflict_disbayes` *Conflict p-values*

Description

A test of the hypothesis that the direct data on a disease outcome give the same information about that outcome as an indirect evidence synthesis obtained from a fitted `disbayes` model. The outcome may be annual incidence, mortality, remission probabilities, or prevalence.

Usage

```
conflict_disbayes(x, varname)
```

Arguments

`x`	A fitted `disbayes` model.
`varname`	Either `inc`, `prev`, `mort` or `rem`.

Details

Hierarchical models are not currently supported in this function.

Value

A data frame with columns indicating age, gender and area.

`p1` is a "one-sided" p-value for the null hypothesis that $r_{obs} = r_{fit}$ against the alternative that $r_{obs} > r_{fit}$,

`p2` is the two-sided p-value for the null hypothesis that $r_{obs} = r_{fit}$ against the alternative that r_{obs} is not equal to r_{fit},

where r_{obs} is the rate informed only by direct data, and r_{fit} is the rate informed by evidence synthesis. Therefore if the evidence synthesis excludes the direct data, then these are interpreted as "conflict" p-values (see Presanis et al. 2013).

In each case, a small p-value favours the alternative hypothesis.

References

Presanis, A. M., Ohlssen, D., Spiegelhalter, D. J. and De Angelis, D. (2013) Conflict diagnostics in directed acyclic graphs, with applications in Bayesian evidence synthesis. Statistical Science, 28, 376-397.

`disbayes` *Bayesian estimation of chronic disease epidemiology from incomplete data*

Description

Estimates a three-state disease model from incomplete data. It is designed to estimate case fatality and incidence, given data on mortality and at least one of incidence and prevalence. Remission may also be included in the data and modelled.

Usage

```
disbayes(
  data,
  inc_num = NULL,
  inc_denom = NULL,
  inc_prob = NULL,
  inc_lower = NULL,
  inc_upper = NULL,
  prev_num = NULL,
  prev_denom = NULL,
  prev_prob = NULL,
  prev_lower = NULL,
  prev_upper = NULL,
  mort_num = NULL,
  mort_denom = NULL,
  mort_prob = NULL,
  mort_lower = NULL,
  mort_upper = NULL,
  rem_num = NULL,
  rem_denom = NULL,
  rem_prob = NULL,
  rem_lower = NULL,
  rem_upper = NULL,
  age = "age",
  cf_model = "smooth",
  inc_model = "smooth",
  rem_model = "const",
  prev_zero = FALSE,
  inc_trend = NULL,
  cf_trend = NULL,
  cf_init = 0.01,
  eqage = 30,
  eqagehi = NULL,
  sprior = c(1, 1, 1),
  hp_fixed = NULL,
  rem_prior = c(1.1, 1),
```

```
  inc_prior = c(2, 0.1),
  cf_prior = c(2, 0.1),
  method = "opt",
  draws = 1000,
  iter = 10000,
  stan_control = NULL,
  bias_model = NULL,
  ...
)
```

Arguments

data	Data frame containing some of the variables below. The variables below are provided as character strings naming columns in this data frame. For each disease measure available, one of the following three combinations of variables must be specified: (1) numerator and denominator (2) estimate and denominator (3) estimate with lower and upper credible limits. Mortality must be supplied, and at least one of incidence and prevalence. If remission is assumed to be possible, then remission data should also be supplied (see below). Estimates refer to the probability of having some event within a year, rather than rates. Rates per year r can be converted to probabilities p as $p = 1 - exp(-r)$, assuming the rate is constant within the year. For estimates based on registry data assumed to cover the whole population, then the denominator will be the population size.
inc_num	Numerator for the incidence data, assumed to represent the observed number of new cases within a year among a population of size inc_denom.
inc_denom	Denominator for the incidence data. The function ci2num can be used to convert a published estimate and interval for a proportion to an implicit numerator and denominator. Note that to include extra uncertainty beyond that implied by a published interval, the numerator and denominator could be multiplied by a constant, for example, multiplying both the numerator and denominator by 0.5 would give the data source half its original weight.
inc_prob	Estimate of the incidence probability
inc_lower	Lower credible limit for the incidence estimate
inc_upper	Upper credible limit for the incidence estimate
prev_num	Numerator for the estimate of prevalence, i.e. number of people currently with a disease.
prev_denom	Denominator for the estimate of prevalence (e.g. the size of the survey used to obtain the prevalence estimate)
prev_prob	Estimate of the prevalence probability
prev_lower	Lower credible limit for the prevalence estimate
prev_upper	Upper credible limit for the prevalence estimate

`mort_num` Numerator for the estimate of the mortality probability, i.e number of deaths attributed to the disease under study within a year

`mort_denom` Denominator for the estimate of the mortality probability (e.g. the population size, if the estimates were obtained from a comprehensive register)

`mort_prob` Estimate of the mortality probability

`mort_lower` Lower credible limit for the mortality estimate

`mort_upper` Upper credible limit for the mortality estimate

`rem_num` Numerator for the estimate of the remission probability, i.e number of people observed to recover from the disease within a year.

Remission data should be supplied if remission is permitted in the model, either as a numerator and denominator or as an estimate and lower credible interval. Conversely, if no remission data are supplied, then remission is assumed to be impossible. These "data" may represent a prior judgement rather than observation - lower denominators or wider credible limits represent weaker prior information.

`rem_denom` Denominator for the estimate of the remission probability

`rem_prob` Estimate of the remission probability

`rem_lower` Lower credible limit for the remission estimate

`rem_upper` Upper credible limit for the remission estimate

`age` Variable in the data indicating the year of age. This must start at age zero, but can end at any age.

`cf_model` Model for how case fatality rate varies with age.

`"smooth"` (the default). Case fatality rate is modelled as a smooth function of age, using a spline.

`"indep"` Case fatality rates are estimated independently for each year of age. This may be useful for determining how much information is in the data. That is, if the posterior from this model is identical to the prior for a certain age, then there is no information in the data alone about case fatality at that age, indicating that some other structural assumption (such as a smooth function of age) or external data are equired to give more precise estimates.

`"increasing"` Case fatality rate is modelled as a smooth and increasing function of age.

`"const"` Case fatality rate is modelled as constant with age.

`inc_model` Model for how incidence rates vary with age.

`"smooth"` (the default). Incidence rates are modelled as a smooth spline function of age.

`"indep"` Incidence rates for each year of age are estimated independently.

`rem_model` Model for how remission rates vary with age, which are typically less well-informed by data, compared to incidence and case fatality.

`"const"` (the default). Constant remission rate over all ages.

`"smooth"` Remission rates are modelled as a smooth spline function of age.

`"indep"` Remission rates estimated independently over all ages.

prev_zero — If TRUE, attempt to estimate prevalence at age zero from the data, as part of the Bayesian model, even if the observed prevalence is zero. Otherwise (the default) this is assumed to be zero if the count is zero, and estimated otherwise.

inc_trend — Matrix of constants representing trends in incidence through calendar time by year of age. There are nage rows and nage columns, where nage is the number of years of age represented in the data. The entry in the ith row and jth column represents the ratio between the incidence nage+j years prior to the year of the data, year, and the incidence in the year of the data, for a person i-1 years of age. For example, if nage=100 and the data refer to the year 2017, then the first column refers to the year 1918 and the last (100th) column refers to 2017. The last column should be all 1, unless the current data are supposed to be biased.

To produce this format from a long data frame, filter to the appropriate outcome and subgroup, and use pivot_wider, e.g.

```
trends <- ihdtrends filter(outcome=="Incidence", gender=="Female")
pivot_wider(names_from="year", values_from="p2017") select(-age, -gender,
-outcome) as.matrix()
```

cf_trend — Matrix of constants representing trends in case fatality through calendar time by year of age, in the same format as inc_trend.

cf_init — Initial guess at a typical case fatality value, for an average age.

eqage — Case fatalities (and incidence and remission rates) are assumed to be equal for all ages below this age, inclusive, when using the smoothed model.

eqagehi — Case fatalities (and incidence and remission rates) are assumed to be equal for all ages above this age, inclusive, when using the smoothed model.

sprior — Rates of the exponential prior distributions used to penalise the coefficients of the spline model. The default of 1 should adapt appropriately to the data, but Higher values give stronger smoothing, or lower values give weaker smoothing, if required.

This can be a named vector with names "inc","cf","rem" in any order, giving the prior smoothness rates for incidence, case fatality and remission. If any of these are not smoothed they can be excluded, e.g. sprior = c(cf=10, inc=1).

This can also be an unnamed vector of three elements, where the first refers to the spline model for incidence, the second for case fatality, the third for remission. If one of the rates (e.g. remission) is not being modelled with a spline, any number can be supplied here and it is just ignored.

hp_fixed — A list with one named element for each hyperparameter to be fixed. The value should be either

- a number (to fix the hyperparameter at this number)
- TRUE (to fix the hyperparameter at the posterior mode from a training run where it is not fixed)

If the element is either NULL, FALSE, or omitted from the list, then the hyperparameter is given a prior and estimated as part of the Bayesian model.

The hyperparameters that can be fixed are

- scf Smoothness parameter for the spline relating case fatality to age.
- sinc Smoothness parameter for the spline relating incidence to age.

For example, to fix the case fatality smoothness to 1.2 and fix the incidence smoothness to its posterior mode, specify `hp_fixed = list(scf=1.2, sinc=TRUE)`.

`rem_prior` Vector of two elements giving the Gamma shape and rate parameters of the prior for the remission rate, used in both `rem_model="const"` and `rem_model="indep"`.

`inc_prior` Vector of two elements giving the Gamma shape and rate parameters of the prior for the incidence rate. Only used if `inc_model="indep"`, for each age-specific rate.

`cf_prior` Vector of two elements giving the Gamma shape and rate parameters of the prior for the case fatality rate. Only used if `cf_model="const"`, or if `cf_model="indep"`, for each age-specific rate, and for the rate at eqage in `cf_model="increasing"`.

`method` String indicating the inference method, defaulting to `"opt"`.

If `method="mcmc"` then a sample from the posterior is drawn using Markov Chain Monte Carlo sampling, via **rstan**'s `rstan::sampling()` function. This is the most accurate but the slowest method.

If `method="opt"`, then instead of an MCMC sample from the posterior, `disbayes` returns the posterior mode calculated using optimisation, via **rstan**'s `rstan::optimizing()` function. A sample from a normal approximation to the (real-line-transformed) posterior distribution is drawn in order to obtain credible intervals.

If the optimisation fails to converge (non-zero return code), try increasing the number of iterations from the default 1000, e.g. `disbayes(..., iter=10000, ...)`, or changing the algorithm to `disbayes(..., algorithm="Newton", ...)`.

If there is an error message that mentions `chol`, then the computed Hessian matrix is not positive definite at the reported optimum, hence credible intervals cannot be computed. This can occur either because of numerical error in computation of the Hessian, or because the reported optimum is wrong. If you are willing to believe the optimum and live without credible intervals, then set `draws=0` to skip computation of the Hessian. To examine the problematic Hessian, set `hessian=TRUE,draws=0`, then look at the `$fit$hessian` component of the `disbayes` return object. If it can be inverted, do `sqrt(diag(solve()))` on the Hessian, and check for `NaN`s, indicating the problematic parameters. Otherwise, diagonal entries of the Hessian matrix that are very small may indicate parameters that are poorly identified from the data, leading to computational problems.

If `method="vb"`, then variational Bayes methods are used, via **rstan**'s `rstan::vb()` function. This is labelled as "experimental" by **rstan**. It might give a better approximation to the posterior than `method="opt"`, but has not been investigated much for `disbayes` models.

`draws` Number of draws from the normal approximation to the posterior when using `method="opt"`.

`iter` Number of iterations for MCMC sampling, or maximum number of iterations for optimization.

`stan_control` (`method="mcmc"` only). List of options supplied as the `control` argument to `rstan::sampling()` in **rstan** for the main model fit.

`bias_model` Experimental model for bias in the incidence estimates due to conflicting information between the different data sources. If `bias_model=NULL` (the default)

no bias is assumed, and all data are assumed to be generated from the same age-specific incidences.

Otherwise there are assumed to be two alternative curves of incidence by age (denoted 2 and 1) where curve 2 is related to curve 1 via a constant hazard ratio that is estimated from the data, given a standard normal prior on the log scale. Three distinct curves would not be identifiable from the data.

If `bias_model="inc"` then the incidence data is assumed to be generated from curve 2, and the prevalence and mortality data from curve 1.

`bias_model="prev"` then the prevalence data is generated from curve 2, and the incidence and mortality data from curve 1.

If `bias_model="incprev"` then both incidence and prevalence data are generated from curve 2, and the mortality data from curve 1.

`...` Further arguments passed to `rstan::sampling()` to control MCMC sampling, or `rstan::optimizing()` to control optimisation, in Stan.

Value

A list including the following components

`call`: Function call that was used.

`fit`: An object containing posterior samples from the fitted model, in the `stanfit` format returned by the `stan` function in the **rstan** package.

`method`: Optimisation method that was chosen.

`nage`: Number of years of age in the data

`dat`: A list containing the input data in the form of numerators and denominators.

`stan_data`: Full list of data supplied to Stan

`stan_inits`: Full list of parameter initial values supplied to Stan

`hp_fixed` Values of any hyperparameters that are fixed during the main model fit.

Use the `tidy.disbayes` method to return summary statistics from the fitted models, simply by calling `tidy()` on the fitted model.

References

Jackson C, Zapata-Diomedi B, Woodcock J. "Bayesian multistate modelling of incomplete chronic disease burden data" https://arxiv.org/abs/2111.14100

`disbayes_hier` *Bayesian estimation of chronic disease epidemiology from incomplete data - hierarchical model for case fatalities.*

Description

A variant of `disbayes` in which data from different areas can be related in a hierarchical model and, optionally, the effect of gender can be treated as additive with the effect of area. This is much more computationally intensive than the basic model in `disbayes`. Time trends are not supported in this function.

Usage

```
disbayes_hier(
  data,
  group,
  gender = NULL,
  inc_num = NULL,
  inc_denom = NULL,
  inc_prob = NULL,
  inc_lower = NULL,
  inc_upper = NULL,
  prev_num = NULL,
  prev_denom = NULL,
  prev_prob = NULL,
  prev_lower = NULL,
  prev_upper = NULL,
  mort_num = NULL,
  mort_denom = NULL,
  mort_prob = NULL,
  mort_lower = NULL,
  mort_upper = NULL,
  rem_num = NULL,
  rem_denom = NULL,
  rem_prob = NULL,
  rem_lower = NULL,
  rem_upper = NULL,
  age = "age",
  cf_init = 0.01,
  eqage = 30,
  eqagehi = NULL,
  cf_model = "default",
  inc_model = "smooth",
  rem_model = "const",
  prev_zero = FALSE,
  sprior = c(1, 1, 1),
  hp_fixed = NULL,
  nfold_int_guess = 5,
  nfold_int_upper = 100,
  nfold_slope_guess = 5,
  nfold_slope_upper = 100,
  mean_int_prior = c(0, 10),
  mean_slope_prior = c(5, 5),
  gender_int_priorsd = 0.82,
  gender_slope_priorsd = 0.82,
  inc_prior = c(1.1, 0.1),
  rem_prior = c(1.1, 1),
  method = "opt",
  draws = 1000,
  iter = 10000,
```

```
  stan_control = NULL,
  ...
)
```

Arguments

data	Data frame containing some of the variables below. The variables below are provided as character strings naming columns in this data frame. For each disease measure available, one of the following three combinations of variables must be specified: (1) numerator and denominator (2) estimate and denominator (3) estimate with lower and upper credible limits. Mortality must be supplied, and at least one of incidence and prevalence. If remission is assumed to be possible, then remission data should also be supplied (see below). Estimates refer to the probability of having some event within a year, rather than rates. Rates per year r can be converted to probabilities p as $p = 1 - exp(-r)$, assuming the rate is constant within the year. For estimates based on registry data assumed to cover the whole population, then the denominator will be the population size.
group	Variable in the data representing the area (or other grouping factor).
gender	If NULL (the default) then the data are one homogenous gender, and there should be one row per year of age. Otherwise, set gender to a character string naming the variable in the data representing gender (or other categorical grouping factor). Gender will then treated as a fixed additive effect, so the linear effect of gender on log case fatality is the same in each area. The data should have one row per year of age and gender.
inc_num	Numerator for the incidence data, assumed to represent the observed number of new cases within a year among a population of size inc_denom.
inc_denom	Denominator for the incidence data. The function ci2num can be used to convert a published estimate and interval for a proportion to an implicit numerator and denominator. Note that to include extra uncertainty beyond that implied by a published interval, the numerator and denominator could be multiplied by a constant, for example, multiplying both the numerator and denominator by 0.5 would give the data source half its original weight.
inc_prob	Estimate of the incidence probability
inc_lower	Lower credible limit for the incidence estimate
inc_upper	Upper credible limit for the incidence estimate
prev_num	Numerator for the estimate of prevalence, i.e. number of people currently with a disease.
prev_denom	Denominator for the estimate of prevalence (e.g. the size of the survey used to obtain the prevalence estimate)
prev_prob	Estimate of the prevalence probability
prev_lower	Lower credible limit for the prevalence estimate

`prev_upper` Upper credible limit for the prevalence estimate

`mort_num` Numerator for the estimate of the mortality probability, i.e number of deaths attributed to the disease under study within a year

`mort_denom` Denominator for the estimate of the mortality probability (e.g. the population size, if the estimates were obtained from a comprehensive register)

`mort_prob` Estimate of the mortality probability

`mort_lower` Lower credible limit for the mortality estimate

`mort_upper` Upper credible limit for the mortality estimate

`rem_num` Numerator for the estimate of the remission probability, i.e number of people observed to recover from the disease within a year.

Remission data should be supplied if remission is permitted in the model, either as a numerator and denominator or as an estimate and lower credible interval. Conversely, if no remission data are supplied, then remission is assumed to be impossible. These "data" may represent a prior judgement rather than observation - lower denominators or wider credible limits represent weaker prior information.

`rem_denom` Denominator for the estimate of the remission probability

`rem_prob` Estimate of the remission probability

`rem_lower` Lower credible limit for the remission estimate

`rem_upper` Upper credible limit for the remission estimate

`age` Variable in the data indicating the year of age. This must start at age zero, but can end at any age.

`cf_init` Initial guess at a typical case fatality value, for an average age.

`eqage` Case fatalities (and incidence and remission rates) are assumed to be equal for all ages below this age, inclusive, when using the smoothed model.

`eqagehi` Case fatalities (and incidence and remission rates) are assumed to be equal for all ages above this age, inclusive, when using the smoothed model.

`cf_model` The following alternative models for case fatality are supported:

`"default"` (the default). Random intercepts and slopes, and no further restriction.

`"interceptonly"`. Random intercepts, but common slopes.

`"increasing"`. Case fatality is assumed to be an increasing function of age (note it is constant below `"eqage"` in all models) with a common slope for all groups.

`"common"` Case fatality is an unconstrained function of age which is common to all areas, i.e. it has the same parameter values in every area. This and `"increasing_common"` are used in situations where you want to compare a model with area-specific rates with a single model for the data aggregated over areas. Modelling the area-disaggregated data using a common function for all areas is equivalent to a model for the aggregated data, and can be statistically compared (using cross-validation) with a model with area-specific rates.

`"increasing_common"` Case fatality is an increasing function of age which is common to all areas.

"const" Case fatality is assumed to be constant with age, for all ages, but different in each area.

"const_common" Case fatality is a constant over all ages and areas.

In all models, case fatality is a smooth function of age.

inc_model Model for how incidence varies with age.

"smooth" (the default). Incidence is modelled as a smooth spline function of age, independently for each area (and gender).

"indep" Incidence rates for each year of age, area (and gender) are estimated independently.

rem_model Model for how remission varies with age. Currently supported models are "const" for a constant remission rate over all ages, "const" for a smooth spline, or "indep" for a different remission rates estimated independently for each age with no smoothing.

prev_zero If TRUE, attempt to estimate prevalence at age zero from the data, as part of the Bayesian model, even if the observed prevalence is zero. Otherwise (the default) this is assumed to be zero if the count is zero, and estimated otherwise.

sprior Rates of the exponential prior distributions used to penalise the coefficients of the spline model. The default of 1 should adapt appropriately to the data, but Higher values give stronger smoothing, or lower values give weaker smoothing, if required.

This can be a named vector with names "inc","cf","rem" in any order, giving the prior smoothness rates for incidence, case fatality and remission. If any of these are not smoothed they can be excluded, e.g. sprior = c(cf=10, inc=1).

This can also be an unnamed vector of three elements, where the first refers to the spline model for incidence, the second for case fatality, the third for remission. If one of the rates (e.g. remission) is not being modelled with a spline, any number can be supplied here and it is just ignored.

hp_fixed A list with one named element for each hyperparameter to be fixed. The value should be either

- a number (to fix the hyperparameter at this number)
- TRUE (to fix the hyperparameter at the posterior mode from a training run where it is not fixed)

If the element is either NULL, FALSE, or omitted from the list, then the hyperparameter is given a prior and estimated as part of the Bayesian model.

The hyperparameters that can be fixed are

- scf Smoothness parameter for the spline relating case fatality to age.
- sinc Smoothness parameter for the spline relating incidence to age.
- scfmale Smoothness parameter for the spline defining how the gender effect relates to age. Only for models with additive gender and area effects.
- sd_int Standard deviation of random intercepts for case fatality.
- sd_slope Standard deviation of random slopes for case fatality.

For example, to fix the case fatality smoothness to 1.2, fix the incidence smoothness to its posterior mode, and estimate all the other hyperparameters, specify hp_fixed = list(scf=1.2, sinc=TRUE).

`nfold_int_guess`
: Prior guess at the ratio of case fatality between a high risk (97.5% quantile) and low risk (2.5% quantile) area.

`nfold_int_upper`
: Prior upper 95% credible limit for the ratio in average case fatality between a high risk (97.5% quantile) and low risk (2.5% quantile) area.

`nfold_slope_guess, nfold_slope_upper`
: This argument and the next argument define the prior distribution for the variance in the random linear effects of age on log case fatality. They define a prior guess and upper 95% credible limit for the ratio of case fatality slopes between a high trend (97.5% quantile) and low risk (2.5% quantile) area. (Note that the model is not exactly linear, since departures from linearity are defined through a spline model. See the Jackson et al. paper for details.).

`mean_int_prior`
: Vector of two elements giving the prior mean and standard deviation respectively for the mean random intercept for log case fatality.

`mean_slope_prior`
: Vector of two elements giving the prior mean and standard deviation respectively for the mean random slope for log case fatality.

`gender_int_priorsd`
: Prior standard deviation for the additive effect of gender on log case fatality

`gender_slope_priorsd`
: Prior standard deviation for the additive effect of gender on the linear age slope of log case fatality

`inc_prior`
: Vector of two elements giving the Gamma shape and rate parameters of the prior for the incidence rate. Only used if `inc_model="indep"`, for each age-specific rate.

`rem_prior`
: Vector of two elements giving the Gamma shape and rate parameters of the prior for the remission rate, used in both `rem_model="const"` and `rem_model="indep"`.

`method`
: String indicating the inference method, defaulting to `"opt"`.

  If `method="mcmc"` then a sample from the posterior is drawn using Markov Chain Monte Carlo sampling, via **rstan**'s `rstan::sampling()` function. This is the most accurate but the slowest method.

  If `method="opt"`, then instead of an MCMC sample from the posterior, `disbayes` returns the posterior mode calculated using optimisation, via **rstan**'s `rstan::optimizing()` function. A sample from a normal approximation to the (real-line-transformed) posterior distribution is drawn in order to obtain credible intervals.

  If the optimisation fails to converge (non-zero return code), try increasing the number of iterations from the default 1000, e.g. `disbayes(..., iter=10000, ...)`, or changing the algorithm to `disbayes(..., algorithm="Newton", ...)`.

  If there is an error message that mentions `chol`, then the computed Hessian matrix is not positive definite at the reported optimum, hence credible intervals cannot be computed. This can occur either because of numerical error in computation of the Hessian, or because the reported optimum is wrong. If you are willing to believe the optimum and live without credible intervals, then set `draws=0` to skip computation of the Hessian. To examine the problematic Hessian, set `hessian=TRUE,draws=0`, then look at the `$fit$hessian` component

of the disbayes return object. If it can be inverted, do sqrt(diag(solve())) on the Hessian, and check for NaNs, indicating the problematic parameters. Otherwise, diagonal entries of the Hessian matrix that are very small may indicate parameters that are poorly identified from the data, leading to computational problems.

If method="vb", then variational Bayes methods are used, via **rstan**'s rstan::vb() function. This is labelled as "experimental" by **rstan**. It might give a better approximation to the posterior than method="opt", but has not been investigated much for disbayes models.

draws — Number of draws from the normal approximation to the posterior when using method="opt".

iter — Number of iterations for MCMC sampling, or maximum number of iterations for optimization.

stan_control — (method="mcmc" only). List of options supplied as the control argument to rstan::sampling() in **rstan** for the main model fit.

... — Further arguments passed to rstan::sampling() to control MCMC sampling, or rstan::optimizing() to control optimisation, in Stan.

Value

A list including the following components

call: Function call that was used.

fit: An object containing posterior samples from the fitted model, in the stanfit format returned by the stan function in the **rstan** package.

method: Optimisation method that was chosen.

nage: Number of years of age in the data

narea: Number of areas (or other grouping variable that defines the hierarchical model).

ng: Number of genders (or other categorical variable whose effect is treated as additive with the area effect).

groups: Names of the areas (or other grouping variable), taken from the factor levels in the original data.

genders: Names of the genders (or other categorical variable), taken from the factor levels in the original data.

dat: A list containing the input data in the form of numerators and denominators.

stan_data: Full list of data supplied to Stan

stan_inits: Full list of parameter initial values supplied to Stan

trend: Whether a time trend was modelled

hp_fixed Values of any hyperparameters that are fixed during the main model fit.

References

Jackson C, Zapata-Diomedi B, Woodcock J. "Bayesian multistate modelling of incomplete chronic disease burden data" https://arxiv.org/abs/2111.14100

ihdengland *Ischemic heart disease in England*

Description

Ischemic heart disease in England

Usage

```
ihdengland
```

Format

A data frame with columns:

`sex`: `"male"` or `"female"`.

`ageyr`. Year of age.

`location`. Name of the location, which is either a city region or region in England.

`num_mort`. Numerator behind the estimate of mortality

`num_inc`. Numerator behind the estimate of incidence

`num_prev`. Numerator behind the estimate of prevalence

`denom_mort`. Denominator behind the estimate of mortality

`denom_inc`. Denominator behind the estimate of incidence

`denom_prev`. Denominator behind the estimate of prevalence

Details

The data were processed to

* change the geography to refer to England city regions and the remaining English regions,

* change counts by 5-year age groups to estimated 1-year counts,

* obtain estimated numerators and denominators from the published point estimates and uncertainty intervals. A point estimate of the risk is equivalent to the numerator divided by the denominator. The denominator is related to the extent of uncertainty around this estimate, and obtained using the Bayesian method implemented in `ci2num`.

The script given in https://github.com/chjackson/disbayes/blob/master/data-raw/gbd_process.Rmd shows these steps.

Source

Global Burden of Disease, 2017

References

Jackson C, Zapata-Diomedi B, Woodcock J. "Bayesian multistate modelling of incomplete chronic disease burden data" https://arxiv.org/abs/2111.14100.

`ihdtrends` *Trends in ischemic heart disease in England*

Description

Trends in ischemic heart disease in England

Usage

```
ihdtrends
```

Format

A data frame with columns:

`gender`: `"male"` or `"female"`.

`age`: Year of age.

`year`: Calendar year.

`p2017`: Estimated ratio between the outcome in the calendar year and the outcome in 2017.

`outcome`: Outcome referred to (incidence or case fatality).

Details

The data were interpolated and smoothed to produce a matrix by year of age and calendar year, using the script at https://github.com/chjackson/disbayes/blob/master/data-raw/trends.r.

Source

Scarborough, P., Wickramasinghe, K., Bhatnagar, P. and Rayner, M. (2011) Trends in coronary heart disease, 1961-2001. British Heart Foundation.

Smolina, K., Wright, F. L., Rayner, M. and Goldacre, M. J. (2012) Determinants of the decline in mortality from acute myocardial infarction in England between 2002 and 2010: linked national database study. BMJ, 344.

British Heart Foundation (2020) Heart and Circulatory Disease Statistics 2020. British Heart Foundation.

loo.disbayes — *Leave-one-out cross validation for disbayes models*

Description

Leave-one-out cross validation for disbayes models

Usage

```
## S3 method for class 'disbayes'
loo(x, outcome = "overall", ...)
```

Arguments

x	A model fitted by disbayes. Any of the computation methods are supported.
outcome	Either "overall", to assess the fit to all data, or one of "inc", "prev", "mort" or "rem", to assess the fit to the incidence data, prevalence data, mortalidy data or remission data, respectively.
...	Other arguments (currently unused).

Value

An object of class "loo" as defined by the **loo** package.

See Also

loo_indiv to return tidied observation-specific contributions to the overall model fit computed here.

loo_indiv — *Extract observation-specific contributions from a disbayes leave-one-out cross validation*

Description

Extract observation-specific contributions from a disbayes leave-one-out cross validation

Usage

```
loo_indiv(x, agg = FALSE)

looi_disbayes(x, agg = FALSE)
```

Arguments

x	For loo_indiv, an object returned by loo.disbayes. For looi_disbayes, an object returned by disbayes.
agg	If TRUE then the observation-specific contributions are aggregated over outcome type, returning a data frame with one row for each of incidence, prevalence, mortality and remission (if remission is included in the model), and one column for each of "elpd_loo", "p_loo" and "looic".

Value

A data frame with one row per observed age-specific mortality, incidence, prevalence and/or remission age-specific data-point, containing leave-one-out cross validation statistics representing how well the model would predict that observation if it were left out of the fit.

These are computed with the **loo** package.

loo_indiv acts on the objects that are returned by running loo on disbayes objects. looi_disbayes acts directly on disbayes objects. Both of those functions return a data frame with LOO contributions for each data point.

Functions

- looi_disbayes: Observation-level leave-one-out cross validation statistics for a disbayes model

---

plot.disbayes *Quick and dirty plot of estimates from disbayes models against age*

Description

Posterior medians and 95

Usage

```
## S3 method for class 'disbayes'
plot(x, variable = "cf", ...)
```

Arguments

x	Object returned by disbayes
variable	Name of the variable of interest to plot against age, by default case fatality rates.
...	Other arguments. Currently unused

Value

A ggplot2 object that can be printed to show the plot, or customised by adding geoms.

Better plots can be drawn by tidying the object returned by disbayes, and using ggplot2 directly on the tidy data frame that this produces. See the vignette for examples.

plot.disbayes_hier *Quick plot of estimates from hierarchical disbayes models against age*

Description

Posterior medians and 95% credible intervals for a quantity of interest are plotted against year of age.

Usage

```
## S3 method for class 'disbayes_hier'
plot(x, variable = "cf", ci = FALSE, ...)
```

Arguments

x	Object returned by disbayes_hier
variable	Name of the variable of interest to plot against age, by default case fatality rates.
ci	Show 95% credible intervals with ribbons.
...	Other arguments. Currently unused

Value

A ggplot2 object that can be printed to show the plot, or customised by adding geoms.

Better plots can be drawn by tidying the object returned by disbayes, and using ggplot2 directly on the tidy data frame that this produces. See the vignette for examples.

plotfit_data_disbayes *Create tidy data for a check of observed against fitted outcome probability estimates from disbayes*

Description

Create tidy data for a check of observed against fitted outcome probability estimates from disbayes

Usage

```
plotfit_data_disbayes(x)
```

Arguments

x	Fitted model from disbayes

Value

A data frame containing observed data in the form of outcome probabilities, as extracted by tidy_obsdat, and estimates of the corresponding probability parameters from the fitted model.

plotfit_disbayes *Graphical check of observed against fitted outcome probabilities from disbayes*

Description

The data behind the plot can be produced using plotfit_data_disbayes, to enable customised plots to be produced by hand with ggplot2.

Usage

```
plotfit_disbayes(x, agemin = 50)
```

Arguments

x	Fitted model from disbayes
agemin	Minimum age to show on the horizontal axis.

Value

A ggplot2 object containing the plot.

tidy.disbayes *Form a tidy data frame from the estimates from a disbayes fit*

Description

Simply call this after fitting disbayes, as, e.g.

```
res <- disbayes(...)
tidy(res)
```

Usage

```
## S3 method for class 'disbayes'
tidy(x, startyear = 1, ...)

## S3 method for class 'disbayes_hier'
tidy(x, ...)
```

Arguments

x	Object returned by disbayes
startyear	Only used for models with time trends. Numeric year represented by year 1 in the data. For example, set this to 1918 to convert years 1-100 to years 1918-2017.
...	Other arguments (currently unused)

Value

A data frame with one row per model parameter, giving summary statistics for the posterior distribution for that parameter. For array parameters, e.g. those that depend on age or area, then the age and area are returned in separate columns, to make it easier to summarise and plot the results, e.g. using **ggplot2**.

Model parameters might include, depending on the model specification,

- `cf`, `inc`, `rem`: Case fatality, incidence, remission rates
- `inc_prob`, `rem_prob`, `mort_prob`, `cf_prob`: Annual incidence, remission, mortality and case fatality risks (probabilities).
- `prev_prob` Prevalence (probability).
- `state_probs` State occupancy probabilities.
- `beta`, `beta_inc` Coefficients of the spline basis for case fatality and incidence respectively.
- `lambda_cf`, `lambda_inc` Smoothness parameters of the spline functions.
- `prevzero` Prevalence at age zero
- `cfbase` Case fatality at the baseline age (only in models where case fatality is increasing).
- `dcf` Annual increments in case fatality (only in models where case fatality is increasing).
- `bias_loghr` Log hazard ratio describing bias in case fatality between datasets (only in models where `bias_model` has been set).

For models with time trends:

- `cf_yr`, `inc_yr`, `state_probs_yr` Case fatality rates, incidence rates and state occupancy probabilities in years prior to the current year. `cf` and `inc` refer to the rates for the current year, the one represented in the data.

Only for hierarchical models:

- `mean_inter`, `mean_slope`,`sd_inter`,`sd_slope`. Mean and standard deviation for random effects distribution for the intercept and slope of log case fatality.
- `lambda_cf_male`, `lambda_inc_male`. Smoothness of the additive gender effect on case fatality and incidence.
- `bareat` Area-level contribution to spline basis coefficients.
- `barea` Normalised spline basis coefficients.

Functions

- `tidy.disbayes_hier`: Tidy method for hierarchical disbayes models

`tidy_obsdat` *Extract observed data from a disbayes model fit*

Description

Extract observed data from a disbayes model fit

Usage

```
tidy_obsdat(x)
```

Arguments

`x`	Fitted `disbayes` model

Value

A data frame with columns `num` and `denom` giving the incidence, prevalence and mortality (and remission if used) numerators and denominators used in the model fit. The column `var` indicates which of incidence, prevalence etc. the numbers refer to. The column `prob` is derived as `num` divided by `denom`. Columns `lower` and `upper` define credible intervals for the "data-based" point estimate `prob`, obtained from the Beta posterior assuming a Beta(0.5, 0.5) prior.

This "data-based" point estimate can be compared with estimates from the model using the functions `plotfit_data_disbayes` and `plotfit_disbayes`.

Index