Package ‘BayesMoFo’

July 21, 2025

Title Bayesian Mortality Forecasting

Version 0.1.0

Description Carry out Bayesian estimation and forecasting for a variety of stochastic mortality models using vague prior distributions. Models supported include numerous well-established approaches introduced in the actuarial and demographic literature, such as the Lee-Carter (1992) <doi:10.1080/01621459.1992.10475265>, the Cairns-Blake-Dowd (2009) <doi:10.1080/10920277.2009.10597538>, the Li-Lee (2005) <doi:10.1353/dem.2005.0021>, and the Plat (2009) <doi:10.1016/j.insmatheco.2009.08.006> models. The package is designed to analyse stratified mortality data structured as a 3-dimensional array of dimensions p × A × T (strata × age × year). Stratification can represent factors such as cause of death, country, deprivation level, sex, geographic region, insurance product, marital status, socioeconomic group, or smoking behavior. While the primary focus is on analysing stratified data ($p > 1$), the package can also handle mortality data that are not stratified ($p = 1$). Model selection via the Deviance Information Criterion (DIC) is supported.

License GPL (>= 2)

Encoding UTF-8

RoxygenNote 7.3.2

Imports rjags, insight, coda, tidyverse, dplyr, magrittr, rlang

Depends R (>= 4.0.0)

LazyData true

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Jackie Siaw Tze Wong [aut, cre] (ORCID:
<`https://orcid.org/0000-0002-0314-6684`>),
Alex Diana [ctb],
Aniketh Pittea [ctb]

Maintainer Jackie Siaw Tze Wong <`jw19203@essex.ac.uk`>

Repository CRAN

Date/Publication 2025-07-14 17:30:06 UTC

Contents

BayesMoFo 3
converge_diag_fn 3
converge_diag_param_fn 5
converge_diag_rates_fn 6
data_summarised 8
DIC_fn 9
Ext_array_country 10
Ext_array_product 11
Ext_array_sex 12
fit_APCI 13
fit_CBD_M3 16
fit_CBD_M5 19
fit_CBD_M6 21
fit_CBD_M7 24
fit_CBD_M8 27
fit_LC 30
fit_M1A 33
fit_M1M 35
fit_M1U 38
fit_M2A1 40
fit_M2A2 42
fit_M2Y1 45
fit_M2Y2 47
fit_MLiLee 49
fit_PLAT 52
fit_RH 55
plot_deaths_fn 58
plot_param_fn 60
plot_rates_fn 61
predict_deaths_fn 62
preparedata_fn 63
quantile_fn 65
runBayesMoFo 66
summary_fn 69
uk_deathscausedata 70
uk_mortalitydata 71

Index **72**

`BayesMoFo` *BayesMoFo: Bayesian Mortality Forecasting*

Description

The BayesMoFo package performs Bayesian inference in a wide variety of mortality models (see vignette for a full list of supported models). Inference is performed using JAGS via the **rjags** interface.

Details

Additional author and maintainer information is available in the DESCRIPTION file.

Author(s)

Maintainer: Jackie Siaw Tze Wong `<jw19203@essex.ac.uk>` (ORCID)

Other contributors:

- Alex Diana `<ad23269@essex.ac.uk>` [contributor]
- Aniketh Pittea `<aniketh.pittea@uk.gt.com>` [contributor]

`converge_diag_fn` *A function to check for overall convergence of fitted stochastic mortality models for monitoring purposes*

Description

Compute several common MCMC convergence diagnostics of parameters fitted under stochastic mortality models using posterior samples stored in "fit_result" object.

Usage

```
converge_diag_fn(result, tol = 0.2, plot_gelman = FALSE, plot_geweke = FALSE)
```

Arguments

`result`	object of type either "fit_result" or "BayesMoFo".
`tol`	A numeric value between 0 and 1 specifying the tolerance percentage of the convergence diagnostics, i.e. if the proportion of convergence diagnostic checks/tests (either Gelman's or Heidel's) exceeds `tol`, warning messages will be triggered. Default is `tol=0.20`.
`plot_gelman`	A logical value indicating whether to produce a plot of Gelman's R statistic, PSRF ("potential scale reduction factor") for visualisation.
`plot_geweke`	A logical value indicating whether to produce a plot of Geweke's statistic for visualisation.

Value

A message of recommendations and (possibly) convergence diagnostics plots. Additionally, a list with components:

`gelman_diag`	A `gelman.diag` object which is a list containing the estimated R statistic (PSRF) along with the upper confidence intervals, and also the multivariate PSRF (Brooks and Gelman, 1998). See Gelman and Rubin (1992) for more details.
`geweke_diag`	A `geweke.diag` object which is a list containing the estimated Geweke's Z scores and the corresponding fractions used for equality of means test. See Geweke (1992) for more details.
`heidel_diag`	A `heidel.diag` object which is a matrix containing results from both Stationarity and Half-width tests. See Heidelberger and Welch (1981), Heidelberger and Welch (1983) for more details.
`n`	The total number of posterior samples (if more than one chain were generated, they are merged into one long chain).

References

Andrew Gelman, Donald B. Rubin. (1992). "Inference from Iterative Simulation Using Multiple Sequences," Statistical Science, Statist. Sci. 7(4), 457-472. doi: 10.1214/ss/1177011136

Brooks, SP. and Gelman, A. (1998). General methods for monitoring convergence of iterative simulations. Journal of Computational and Graphical Statistics, 7, 434-455. doi:10.1080/10618600.1998.10474787

Geweke, J. (1992) Evaluating the accuracy of sampling-based approaches to calculating posterior moments. In Bayesian Statistics 4 (ed JM Bernado, JO Berger, AP Dawid and AFM Smith). Clarendon Press, Oxford, UK. doi:10.21034/sr.148

Heidelberger P and Welch PD. (1981). A spectral method for confidence interval generation and run length control in simulations. Comm. ACM. 24, 233-245. doi:10.1145/358598.358630

Heidelberger P and Welch PD. (1983) Simulation run length control in the presence of an initial transient. Opns Res., 31, 1109-44. doi:10.1287/opre.31.6.1109

Examples

```
#load and prepare data
data("dxt_array_product");data("Ext_array_product")
death<-preparedata_fn(dxt_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)
expo<-preparedata_fn(Ext_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)

#fit any mortality model
runBayesMoFo_result<-runBayesMoFo(death=death,expo=expo,models="M1A",n_iter=1000,
  family="poisson",n.chain=5,thin=1)

#compute convergence diagnostics (with plots)
converge_runBayesMoFo_result<-converge_diag_fn(runBayesMoFo_result,
  plot_gelman=TRUE,plot_geweke=TRUE)

#Gelman's R (PSRF)
converge_runBayesMoFo_result$gelman_diag
```

```
#Geweke's Z statistics
converge_runBayesMoFo_result$geweke_diag$z

#Heidel's Stationary and Halfwidth Tests
converge_runBayesMoFo_result$heidel_diag
```

`converge_diag_param_fn` *A function to assess convergence of the posterior sampling of fitted parameters for monitoring purposes*

Description

Produce several convergence diagnostic tools (e.g. trace/density/acf plots and effective sample sizes) from the posterior samples of fitted parameters.

Usage

```
converge_diag_param_fn(
  result,
  plot_params = NULL,
  trace = TRUE,
  density = TRUE,
  acf_plot = FALSE,
  ESS_all = FALSE
)
```

Arguments

`result`	object of type either "fit_result" or "BayesMoFo".
`plot_params`	A vector of character strings specifying which set of parameters to plot for visualisation. If not specified, a random selection of the parameters will be included in the plots (see `fit_result$param` or `runBayesMoFo_result$result$best$param` for a full list) will be chosen. Note that a specific combination of alpha, beta, kappa, and gamma, is to be plotted, then users need to specify the exact indices of them, e.g. `plot_params="gamma[1,2]"`. Otherwise, only three randomly selected of them will be plotted. To see a complete list of parameters, i.e. `colnames(fit_result$post_sample[[1]])[!startsWith(colnames(fit_result$post_sample[[1` or `colnames(runBayesMoFo_result$result$best$post_sample[[1]])[!startsWith(colnames(r`
`trace`	A logical value to indicate if trace plots of posterior samples of death rates should be shown (default) or suppressed (e.g. to aid visibility).
`density`	A logical value to indicate if density plots of posterior samples of death rates should be shown (default) or suppressed (e.g. to aid visibility).
`acf_plot`	A logical value to indicate if auto-correlation plots should be shown or suppressed (default).

`ESS_all`	A logical value indicating if effective sample sizes are to be computed for all parameters. The default is FALSE where only chosen parameters will be evaluated, if TRUE all parameters will be assessed.

Value

Some convergence-related plots of posterior samples of fitted parameters.

`ESS` The effective sample sizes of the chosen parameters.

Examples

```
#load and prepare data
data("dxt_array_product");data("Ext_array_product")
death<-preparedata_fn(dxt_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)
expo<-preparedata_fn(Ext_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)

#fit any mortality model
runBayesMoFo_result<-runBayesMoFo(death=death,expo=expo,models="APCI")

#default plot
converge_runBayesMoFo_result<-converge_diag_param_fn(runBayesMoFo_result)

#ESS
converge_runBayesMoFo_result$ESS

#plot specific parameters

runBayesMoFo_result$result$best$param  #run this line to check parameters of the model
converge_diag_param_fn(runBayesMoFo_result,plot_params=c("rho","sigma2_kappa","beta"))
#note only three betas were plotted

colnames(runBayesMoFo_result$result$best$post_sample[[1]])[!startsWith(
  colnames(runBayesMoFo_result$result$best$post_sample[[1]]),"q[")]
#run the above line to check full list of parameters of the model
converge_diag_param_fn(runBayesMoFo_result,plot_params=c("beta[1,2]","gamma[3,2]"))

#ACF plot
converge_diag_param_fn(runBayesMoFo_result,plot_params=c("beta[1,2]","gamma[3,2]"),
trace=FALSE,density=FALSE,acf_plot=TRUE)
```

`converge_diag_rates_fn` *A function to assess convergence of the posterior sampling of death rates for monitoring purposes*

Description

Produce several convergence diagnostic tools (e.g. trace/density/acf plots and effective sample sizes) from the posterior samples of death rates.

Usage

```
converge_diag_rates_fn(
  result,
  plot_ages = NULL,
  plot_years = NULL,
  plot_strata = NULL,
  trace = TRUE,
  density = TRUE,
  acf_plot = FALSE,
  ESS_all = FALSE
)
```

Arguments

`result`	object of type either "fit_result" or "BayesMoFo".
`plot_ages`	A numeric vector specifying which range of ages to plot for visualisation. If not specified, a random selection of ages that were used to fit the model (i.e. `fit_result$death$ages`) will be chosen.
`plot_years`	A numeric vector specifying which range of years to plot for visualisation. If not specified, a random selection of years that were used to fit the model (i.e. `fit_result$death$years`) will be chosen.
`plot_strata`	A vector of character strings specifying which strata to plot for visualisation. If not specified, a random selection of the strata used to fit the model (i.e. `fit_result$death$strat_name`) will be chosen.
`trace`	A logical value to indicate if trace plots of posterior samples of death rates should be shown (default) or suppressed (e.g. to aid visibility).
`density`	A logical value to indicate if density plots of posterior samples of death rates should be shown (default) or suppressed (e.g. to aid visibility).
`acf_plot`	A logical value to indicate if auto-correlation plots should be shown or suppressed (default).
`ESS_all`	A logical value indicating if effective sample sizes are to be computed for all rates. The default is FALSE where only chosen rates will be evaluated, if TRUE all rates will be assessed.

Value

Some convergence-related plots of posterior samples of mortality rates.

`ESS` The effective sample sizes of the chosen parameters.

Examples

```
#load and prepare data
data("dxt_array_product");data("Ext_array_product")
death<-preparedata_fn(dxt_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)
expo<-preparedata_fn(Ext_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)
```

```
#fit any mortality model
runBayesMoFo_result<-runBayesMoFo(death=death,expo=expo,models="APCI")

#default plot
converge_runBayesMoFo_result<-converge_diag_rates_fn(runBayesMoFo_result)

#ESS
converge_runBayesMoFo_result$ESS

#plot by age and changing pre-specified arguments
converge_diag_rates_fn(runBayesMoFo_result,plot_ages=c(40,50,60),plot_years=c(2017,2020))

#ACF plot
converge_diag_rates_fn(runBayesMoFo_result,plot_ages=c(40,50,60),plot_years=c(2017,2020),
trace=FALSE,density=FALSE,acf_plot=TRUE)
```

`data_summarised` *Sample mortality data stratified by insurance products*

Description

This is a sample data set used for demonstration purposes.

Usage

```
data("data_summarised")
```

Format

A data frame with 1278 rows of observations and 9 variables:

Product Character. The name of the insurance product associated with the observation. There are in total 4 types of products considered in the dataset:
`"ACI"`: ;
`"DB"`: ;
`"SCI"`: ;
`"Annuities"`: Note that this product contains a lot of missing values.

Age Numeric. The claim age x associated with the observation, ranging between 18-100.

Year Numeric. The claim year t associated with the observation, spanning years 2016-2020.

Exposure Numeric. The central exposure to risk, E_x^c, associated with the observation.

Claim Numeric. The number of claims ("deaths") associated with the observation.

ExpClaim Numeric. The expected number of claims associated with the observation.

Qx Numeric. The crude mortality rate associated with the observation. It can be computed as $\frac{\text{Claim}}{\text{Exposure}}$.

ExpQx Numeric. The expected crude mortality rate associated with the observation. It can be computed as $\frac{\text{ExpClaim}}{\text{Exposure}}$.

StdQx Numeric. The standard deviation of the crude mortality rate associated with the observation. It can be computed as $\sqrt{\frac{\text{Qx}(1-\text{Qx})}{\text{Exposure}}}$.

Examples

```
data("data_summarised")
str(data_summarised)
head(data_summarised)

#extracting a subset of the data (3 products)
data_summarised[data_summarised$Product==c("ACI","DB","SCI"),]

#extracting a subset of the data (ages 35-65)
data_summarised[(data_summarised$Age>=35 & data_summarised$Age<=65),]
```

`DIC_fn` *A function to calculate the DIC of stochastic mortality models using posterior samples*

Description

Compute the Deviance Information Criterion (DIC) of stochastic mortality models using posterior samples stored in "fit_result" object.

Usage

```
DIC_fn(result)
```

Arguments

`result`	object of type either "fit_result" or "BayesMoFo".

Value

A numeric value representing the DIC of the mortality model.

Examples

```
#load and prepare data
data("dxt_array_product");data("Ext_array_product")
death<-preparedata_fn(dxt_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)
expo<-preparedata_fn(Ext_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)

#a toy example
runBayesMoFo_result<-runBayesMoFo(death=death,expo=expo,models="M1A",n_iter=50,n.adapt=50)

#compute the DIC
DIC_fn(runBayesMoFo_result)
```

`Ext_array_country` *Sample mortality data stratified by country*

Description

This is a sample data set used for demonstration purposes. They consist of two 3-dimensional arrays, one for number of deaths (`dxt_array_country`) and another for the the corresponding central exposures to risk (`Ext_array_country`).

Usage

```
data("Ext_array_country")

data("dxt_array_country")
```

Format

An object of class `array` of dimension 5 x 96 x 50.

A 3-dimensional data array (dim 1: strata, dim 2: ages, dim 3: years) containing 5 countries, 96 ages, and 50 years:

Strata Character. Indicate the country (in total 5) origin of the data, labelled as:
- `"AUS"`: Australia;
- `"ITALY"`: Italy;
- `"JAPAN"`: Japan;
- `"UK"`: United Kingdom ;
- `"US"`: United States.

Ages Numeric. Ages at deaths, ranging between 0-95.

Years Numeric. Years at deaths, spanning years 1951-2000.

Examples

```
##Load exposure data
data("Ext_array_country")
str(Ext_array_country)
head(Ext_array_country)

#extracting a subset of the data (2 countries, ages 0-90, years 1961-2000)
Ext_array_country[c("AUS","JAPAN"),as.character(0:90),as.character(1961:2000)]

##Load death data
data("dxt_array_country")
str(dxt_array_country)
head(dxt_array_country)

#extracting a subset of the data (2 countries, ages 0-90, years 1961-2000)
dxt_array_country[c("AUS","JAPAN"),as.character(0:90),as.character(1961:2000)]
```

`Ext_array_product` *Sample mortality data stratified by insurance products*

Description

This is a sample data set used for demonstration purposes. They consist of two 3-dimensional arrays, one for number of deaths (`dxt_array_product`) and another for the the corresponding central exposures to risk (`Ext_array_product`).

Usage

```
data("Ext_array_product")

data("dxt_array_product")
```

Format

An object of class `array` of dimension 4 x 83 x 5.

A 3-dimensional data array (dim 1: strata, dim 2: ages, dim 3: years) with 4 strata (products), 83 ages, and 5 years:

Strata/Products Character. Names of the insurance products considered in the dataset. There are in total 4 products:
`"ACI"`: ;
`"DB"`: ;
`"SCI"`: ;
`"Annuities"`: Note that this product contains a lot of missing values.

Ages Numeric. Ages at claims, ranging between 18-100.

Years Numeric. Years at claims, spanning years 2016-2020.

Examples

```
##Load exposure data
data("Ext_array_product")
str(Ext_array_product)
head(Ext_array_product)

#extracting a subset of the data (3 products, ages 35-65, years 2016-2020)
Ext_array_product[c("ACI","DB","SCI"),as.character(35:65),as.character(2016:2020)]

##Load death data
data("dxt_array_product")
str(dxt_array_product)
head(dxt_array_product)

#extracting a subset of the data (3 products, ages 35-65, years 2016-2020)
dxt_array_product[c("ACI","DB","SCI"),as.character(35:65),as.character(2016:2020)]
```

`Ext_array_sex` *Sample mortality data stratified by gender/sex in the UK*

Description

This is a sample data set used for demonstration purposes. They consist of two 3-dimensional arrays, one for number of deaths (`dxt_array_sex`) and another for the the corresponding central exposures to risk (`Ext_array_sex`).

Usage

```
data("Ext_array_sex")

data("dxt_array_sex")
```

Format

An object of class `array` of dimension 2 x 111 x 181.

A 3-dimensional data array (dim 1: strata, dim 2: ages, dim 3: years) with 2 strata (males and females), 111 ages, and 181 years:

Strata Character. Indicate the gender/sex, labelled as:
`"Male"`: ;
`"Female"`: ..

Ages Numeric. Ages at deaths, ranging between 0-109, and 110+.

Years Numeric. Years at deaths, spanning years 1841-2021.

Examples

```
##Load exposure data
data("Ext_array_sex")
str(Ext_array_sex)
head(Ext_array_sex)

#extracting a subset of the data (2 genders, ages 0-100, years 1961-2000)
Ext_array_sex[c("Male","Female"),as.character(0:100),as.character(1961:2000)]

##Load death data
data("dxt_array_sex")
str(dxt_array_sex)
head(dxt_array_sex)

#extracting a subset of the data (2 genders, ages 0-100, years 1961-2000)
dxt_array_sex[c("Male","Female"),as.character(0:100),as.character(1961:2000)]
```

`fit_APCI` *A function to fit the APCI stochastic mortality model.*

Description

Carry out Bayesian estimation of the stochastic mortality model, the **Age-Period-Cohort-Improvement (APCI) model**. Note that this model has been studied extensively by Richards et al. (2019) and Wong et al. (2023).

Usage

```
fit_APCI(
  death,
  expo,
  n_iter = 10000,
  family = "nb",
  share_alpha = FALSE,
  share_beta = FALSE,
  share_gamma = FALSE,
  n.chain = 1,
  thin = 1,
  n.adapt = 1000,
  forecast = FALSE,
  h = 5,
  quiet = FALSE
)
```

Arguments

`death`	death data that has been formatted through the function `preparedata_fn`.
`expo`	expo data that has been formatted through the function `preparedata_fn`.
`n_iter`	number of iterations to run. Default is `n_iter=10000`.
`family`	a string of characters that defines the family function associated with the mortality model. "poisson" would assume that deaths follow a Poisson distribution and use a log link; "binomial" would assume that deaths follow a Binomial distribution and a logit link; "nb" (default) would assume that deaths follow a Negative-Binomial distribution and a log link.
`share_alpha`	a logical value indicating if $a_{x,p}$ should be shared across all strata (see details below). Default is `FALSE`.
`share_beta`	a logical value indicating if $b_{x,p}$ should be shared across all strata (see details below). Default is `FALSE`.
`share_gamma`	a logical value indicating if the cohort parameter $\gamma_{x,p}$ should be shared across all strata (see details below). Default is `FALSE`.
`n.chain`	number of parallel chains for the model.
`thin`	thinning interval for monitoring purposes.

`n.adapt`	the number of iterations for adaptation. See `?rjags::adapt` for details.
`forecast`	a logical value indicating if forecast is to be performed (default is `FALSE`). See below for details.
`h`	a numeric value giving the number of years to forecast. Default is `h=5`.
`quiet`	if TRUE then messages generated during compilation will be suppressed, as well as the progress bar during adaptation.

Details

The model can be described mathematically as follows: If `family="poisson"`, then

$$d_{x,t,p} \sim \text{Poisson}(E^c_{x,t,p} m_{x,t,p}),$$

$$\log(m_{x,t,p}) = a_{x,p} + b_{x,p}(t - \bar{t}) + k_{t,p} + \gamma_{c,p},$$

where $c = t - x$ is the cohort index, $\bar{t}$ is the mean of t, $d_{x,t,p}$ represents the number of deaths at age x in year t of stratum p, while $E^c_{x,t,p}$ and $m_{x,t,p}$ represents respectively the corresponding central exposed to risk and central mortality rate at age x in year t of stratum p. Similarly, if `family="nb"`, then a negative binomial distribution is fitted, i.e.

$$d_{x,t,p} \sim \text{Negative-Binomial}(\phi, \frac{\phi}{\phi + E^c_{x,t,p} m_{x,t,p}}),$$

$$\log(m_{x,t,p}) = a_{x,p} + b_{x,p}(t - \bar{t}) + k_{t,p} + \gamma_{c,p},$$

where ϕ is the overdispersion parameter. See Wong et al. (2018). But if `family="binomial"`, then

$$d_{x,t,p} \sim \text{Binomial}(E^0_{x,t,p}, q_{x,t,p}),$$

$$\text{logit}(q_{x,t,p}) = a_{x,p} + b_{x,p}(t - \bar{t}) + k_{t,p} + \gamma_{c,p},$$

where $q_{x,t,p}$ represents the initial mortality rate at age x in year t of stratum p, while $E^0_{x,t,p} \approx E^c_{x,t,p} + \frac{1}{2} d_{x,t,p}$ is the corresponding initial exposed to risk. Constraints used are:

$$\sum_t k_{t,p} = \sum_t t k_{t,p} = 0, \sum_c \gamma_{c,p} = \sum_c c\gamma_{c,p} = \sum_c c^2 \gamma_{c,p} = 0 \quad \text{for each stratum } p.$$

If `share_alpha=TRUE`, then the additive age-specific parameter is the same across all strata p, i. e.

$$a_x + b_{x,p}(t - \bar{t}) + k_{t,p} + \gamma_{c,p}.$$

Similarly if `share_beta=TRUE`, then the multiplicative age-specific parameter is the same across all strata p, i. e.

$$a_{x,p} + b_x(t - \bar{t}) + k_{t,p} + \gamma_{c,p}.$$

Similarly if `share_gamma=TRUE`, then the cohort parameter is the same across all strata p, i. e.

$$a_{x,p} + b_{x,p}(t - \bar{t}) + k_{t,p} + \gamma_c.$$

If `forecast=TRUE`, then the following time series models are fitted on $k_{t,p}$ and $\gamma_{c,p}$ as follows:

$$k_{t,p} = \rho k_{t-1,p} + \epsilon_{t,p} \text{ for } p = 1, \ldots, P \text{ and } t = 2, \ldots, T,$$

$$k_{1,p} = \epsilon_{1,p}$$

and

$$\gamma_{c,p} = \gamma_{c-1,p} + \rho_\gamma(\gamma_{c-1,p} - \gamma_{c-2,p}) + \epsilon^\gamma_{c,p} \text{ for } p = 1, \ldots, P \text{ and } c = 3, \ldots, C,$$

$$\gamma_2 = \gamma_1 + \frac{1}{\sqrt{1-\rho_\gamma^2}}\epsilon^\gamma_{2,p},$$

$$\gamma_1 = 100\epsilon^\gamma_{1,p}$$

where $\epsilon_{t,p} \sim N(0, \sigma_k^2)$ for $t = 1, \ldots, T$, $\epsilon^\gamma_{c,p} \sim N(0, \sigma_\gamma^2)$ for $c = 1, \ldots, C$, while $\eta_1, \eta_2, \rho, \sigma_k^2, \rho_\gamma, \sigma_\gamma^2$ are additional parameters to be estimated. Note that the forecasting models are inspired by Wong et al. (2023).

Value

A list with components:

`post_sample` An `mcmc.list` object containing the posterior samples generated.

`param` A vector of character strings describing the names of model parameters.

`death` The death data that was used.

`expo` The expo data that was used.

`family` The family function used.

`forecast` A logical value indicating if forecast has been performed.

`h` The forecast horizon used.

References

Richards S. J., Currie I. D., Kleinow T., and Ritchie G. P. (2019). A stochastic implementation of the APCI model for mortality projections. British Actuarial Journal, 24, E13. doi:10.1017/S1357321718000260

Jackie S. T. Wong, Jonathan J. Forster, and Peter W. F. Smith. (2018). Bayesian mortality forecasting with overdispersion, Insurance: Mathematics and Economics, Volume 2018, Issue 3, 206-221. doi:10.1016/j.insmatheco.2017.09.023

Jackie S. T. Wong, Jonathan J. Forster, and Peter W. F. Smith. (2023). Bayesian model comparison for mortality forecasting, Journal of the Royal Statistical Society Series C: Applied Statistics, Volume 72, Issue 3, 566–586. doi:10.1093/jrsssc/qlad021

Examples

```
#load and prepare mortality data
data("dxt_array_product");data("Ext_array_product")
death<-preparedata_fn(dxt_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)
expo<-preparedata_fn(Ext_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)

#fit the model (negative-binomial family)
#NOTE: This is a toy example, please run it longer in practice.
fit_APCI_result<-fit_APCI(death=death,expo=expo,n_iter=50,n.adapt=50)
head(fit_APCI_result)
```

```
#if sharing the cohorts only (poisson family)
fit_APCI_result2<-fit_APCI(death=death,expo=expo,n_iter=1000,family="poisson",share_gamma=TRUE)
head(fit_APCI_result2)

#if sharing all alphas, betas, and cohorts (poisson family)
fit_APCI_result3<-fit_APCI(death=death,expo=expo,n_iter=1000,family="poisson",share_alpha=TRUE,
share_beta=TRUE,share_gamma=TRUE)
head(fit_APCI_result3)

#if forecast
fit_APCI_result4<-fit_APCI(death=death,expo=expo,n_iter=1000,family="poisson",forecast=TRUE)
plot_rates_fn(fit_APCI_result4)
plot_param_fn(fit_APCI_result4)
```

`fit_CBD_M3` *A function to fit the Age-Period-Cohort (APC) stochastic mortality model.*

Description

Carry out Bayesian estimation of the stochastic mortality model, the **Age-Period-Cohort (APC) model** (Jacobsen et al., 2002). Note that this model is equivalent to "M3" under the formulation of Cairns et al. (2009).

Usage

```
fit_CBD_M3(
  death,
  expo,
  n_iter = 10000,
  family = "nb",
  share_alpha = FALSE,
  share_gamma = FALSE,
  n.chain = 1,
  thin = 1,
  n.adapt = 1000,
  forecast = FALSE,
  h = 5,
  quiet = FALSE
)
```

Arguments

`death`	death data that has been formatted through the function `preparedata_fn`.
`expo`	expo data that has been formatted through the function `preparedata_fn`.

`n_iter`	number of iterations to run. Default is `n_iter=10000`.
`family`	a string of characters that defines the family function associated with the mortality model. "poisson" would assume that deaths follow a Poisson distribution and use a log link; "binomial" would assume that deaths follow a Binomial distribution and a logit link; "nb" (default) would assume that deaths follow a Negative-Binomial distribution and a log link.
`share_alpha`	a logical value indicating if $a_{x,p}$ should be shared across all strata (see details below). Default is `FALSE`.
`share_gamma`	a logical value indicating if the cohort parameter $\gamma_{x,p}$ should be shared across all strata (see details below). Default is `FALSE`.
`n.chain`	number of parallel chains for the model.
`thin`	thinning interval for monitoring purposes.
`n.adapt`	the number of iterations for adaptation. See `?rjags::adapt` for details.
`forecast`	a logical value indicating if forecast is to be performed (default is `FALSE`). See below for details.
`h`	a numeric value giving the number of years to forecast. Default is `h=5`.
`quiet`	if TRUE then messages generated during compilation will be suppressed, as well as the progress bar during adaptation.

Details

The model can be described mathematically as follows: If `family="poisson"`, then

$$d_{x,t,p} \sim \text{Poisson}(E^c_{x,t,p} m_{x,t,p}),$$

$$\log(m_{x,t,p}) = a_{x,p} + k_{t,p} + \gamma_{c,p},$$

where $c = t - x$ is the cohort index, $d_{x,t,p}$ represents the number of deaths at age x in year t of stratum p, while $E^c_{x,t,p}$ and $m_{x,t,p}$ represents respectively the corresponding central exposed to risk and central mortality rate at age x in year t of stratum p. Similarly, if `family="nb"`, then a negative binomial distribution is fitted, i.e.

$$d_{x,t,p} \sim \text{Negative-Binomial}(\phi, \frac{\phi}{\phi + E^c_{x,t,p} m_{x,t,p}}),$$

$$\log(m_{x,t,p}) = a_{x,p} + k_{t,p} + \gamma_{c,p},$$

where ϕ is the overdispersion parameter. See Wong et al. (2018). But if `family="binomial"`, then

$$d_{x,t,p} \sim \text{Binomial}(E^0_{x,t,p}, q_{x,t,p}),$$

$$\text{logit}(q_{x,t,p}) = a_{x,p} + k_{t,p} + \gamma_{c,p},$$

where $q_{x,t,p}$ represents the initial mortality rate at age x in year t of stratum p, while $E^0_{x,t,p} \approx E^c_{x,t,p} + \frac{1}{2} d_{x,t,p}$ is the corresponding initial exposed to risk. Constraints used are:

$$\sum_t k_{t,p} = 0, \gamma_{1,p} = 0, \sum_{x,t} \gamma_{c,p} = 0 \quad \text{for each stratum } p.$$

If `share_alpha=TRUE`, then the additive age-specific parameter is the same across all strata p, i. e.

$$a_x + k_{t,p} + \gamma_{c,p}.$$

Similarly if `share_gamma=TRUE`, then the cohort parameter is the same across all strata p, i. e.

$$a_{x,p} + k_{t,p} + \gamma_c.$$

If `forecast=TRUE`, then the following time series models are fitted on $k_{t,p}$ and $\gamma_{c,p}$ as follows:

$$k_{t,p} = \eta_1 + \eta_2 t + \rho(k_{t-1,p} - (\eta_1 + \eta_2(t-1))) + \epsilon_{t,p} \text{ for } p = 1, \ldots, P \text{ and } t = 1, \ldots, T,$$

and

$$\gamma_{c,p} = \gamma_{c-1,p} + \rho_\gamma(\gamma_{c-1,p} - \gamma_{c-2,p}) + \epsilon^\gamma_{c,p} \text{ for } p = 1, \ldots, P \text{ and } c = 3, \ldots, C,$$

$$\gamma_2 = \gamma_1 + \frac{1}{\sqrt{1-\rho_\gamma^2}}\epsilon^\gamma_{2,p},$$

$$\gamma_1 = 100\epsilon^\gamma_{1,p}$$

where $\epsilon_{t,p} \sim N(0, \sigma_k^2)$ for $t = 1, \ldots, T$, $\epsilon^\gamma_{c,p} \sim N(0, \sigma_\gamma^2)$ for $c = 1, \ldots, C$, while $\eta_1, \eta_2, \rho, \sigma_k^2, \rho_\gamma, \sigma_\gamma^2$ are additional parameters to be estimated. Note that the forecasting models are inspired by Wong et al. (2023).

Value

A list with components:

`post_sample`	An `mcmc.list` object containing the posterior samples generated.
`param`	A vector of character strings describing the names of model parameters.
`death`	The death data that was used.
`expo`	The expo data that was used.
`family`	The family function used.
`forecast`	A logical value indicating if forecast has been performed.
`h`	The forecast horizon used.

References

Cairns, A. J. G., Blake, D., Dowd, K., Coughlan, G. D., Epstein, D., Ong, A., and Balevich, I. (2009). A Quantitative Comparison of Stochastic Mortality Models Using Data From England and Wales and the United States. North American Actuarial Journal, 13(1), 1–35. doi:10.1080/10920277.2009.10597538

Jacobsen R, Keiding N, and Lynge E. (2002). Long term mortality trends behind low life expectancy of Danish women. J Epidemiol Community Health. 56(3): 205-8. doi: 10.1136/jech.56.3.205

Jackie S. T. Wong, Jonathan J. Forster, and Peter W. F. Smith. (2018). Bayesian mortality forecasting with overdispersion, Insurance: Mathematics and Economics, Volume 2018, Issue 3, 206-221. doi:10.1016/j.insmatheco.2017.09.023

Jackie S. T. Wong, Jonathan J. Forster, and Peter W. F. Smith. (2023). Bayesian model comparison for mortality forecasting, Journal of the Royal Statistical Society Series C: Applied Statistics, Volume 72, Issue 3, 566–586. doi:10.1093/jrsssc/qlad021

Examples

```
#load and prepare mortality data
data("dxt_array_product");data("Ext_array_product")
death<-preparedata_fn(dxt_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)
expo<-preparedata_fn(Ext_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)

#fit the model (negative-binomial family)
#NOTE: This is a toy example, please run it longer in practice.
fit_CBD_M3_result<-fit_CBD_M3(death=death,expo=expo,n_iter=50,n.adapt=50)
head(fit_CBD_M3_result)


#if sharing the cohorts only (poisson family)
fit_CBD_M3_result2<-fit_CBD_M3(death=death,expo=expo,n_iter=1000,family="poisson",share_gamma=TRUE)
head(fit_CBD_M3_result2)

#if sharing both alphas and cohorts (poisson family)
fit_CBD_M3_result3<-fit_CBD_M3(death=death,expo=expo,n_iter=1000,family="poisson",
share_alpha=TRUE,share_gamma=TRUE)
head(fit_CBD_M3_result3)
```

`fit_CBD_M5` *A function to fit the stochastic mortality model "M5".*

Description

Carry out Bayesian estimation of the stochastic mortality model referred to as **"M5"** under the formulation of Cairns et al. (2009).

Usage

```
fit_CBD_M5(
  death,
  expo,
  n_iter = 10000,
  family = "nb",
  n.chain = 1,
  thin = 1,
  n.adapt = 1000,
  forecast = FALSE,
  h = 5,
  quiet = FALSE
)
```

Arguments

`death`	death data that has been formatted through the function `preparedata_fn`.

`expo`	expo data that has been formatted through the function `preparedata_fn`.
`n_iter`	number of iterations to run. Default is `n_iter=10000`.
`family`	a string of characters that defines the family function associated with the mortality model. "poisson" would assume that deaths follow a Poisson distribution and use a log link; "binomial" would assume that deaths follow a Binomial distribution and a logit link; "nb" (default) would assume that deaths follow a Negative-Binomial distribution and a log link.
`n.chain`	number of parallel chains for the model.
`thin`	thinning interval for monitoring purposes.
`n.adapt`	the number of iterations for adaptation. See `?rjags::adapt` for details.
`forecast`	a logical value indicating if forecast is to be performed (default is `FALSE`). See below for details.
`h`	a numeric value giving the number of years to forecast. Default is `h=5`.
`quiet`	if TRUE then messages generated during compilation will be suppressed, as well as the progress bar during adaptation.

Details

The model can be described mathematically as follows: If `family="poisson"`, then

$$d_{x,t,p} \sim \text{Poisson}(E^c_{x,t,p} m_{x,t,p}),$$

$$\log(m_{x,t,p}) = k^1_{t,p} + k^2_{t,p}(x - \bar{x}),$$

where $\bar{x}$ is the mean of x, $d_{x,t,p}$ represents the number of deaths at age x in year t of stratum p, while $E^c_{x,t,p}$ and $m_{x,t,p}$ represents respectively the corresponding central exposed to risk and central mortality rate at age x in year t of stratum p. Similarly, if `family="nb"`, then a negative binomial distribution is fitted, i.e.

$$d_{x,t,p} \sim \text{Negative-Binomial}(\phi, \frac{\phi}{\phi + E^c_{x,t,p} m_{x,t,p}}),$$

$$\log(m_{x,t,p}) = k^1_{t,p} + k^2_{t,p}(x - \bar{x}),$$

where ϕ is the overdispersion parameter. See Wong et al. (2018). But if `family="binomial"`, then

$$d_{x,t,p} \sim \text{Binomial}(E^0_{x,t,p}, q_{x,t,p}),$$

$$\text{logit}(q_{x,t,p}) = k^1_{t,p} + k^2_{t,p}(x - \bar{x}),$$

where $q_{x,t,p}$ represents the initial mortality rate at age x in year t of stratum p, while $E^0_{x,t,p} \approx E^c_{x,t,p} + \frac{1}{2} d_{x,t,p}$ is the corresponding initial exposed to risk. No constraints are needed. If `forecast=TRUE`, then the following time series models are fitted on $k^1_{t,p}$ as follows:

$$k^1_{t,p} = \eta^1_1 + \eta^1_2 t + \rho(k^1_{t-1,p} - (\eta^1_1 + \eta^1_2(t-1))) + \epsilon^1_{t,p} \text{ for } p = 1, \ldots, P \text{ and } t = 1, \ldots, T,$$

where $\epsilon^1_{t,p} \sim N(0, (\sigma^1)^2_k)$ for $t = 1, \ldots, T$, while $\eta^1_1, \eta^1_2, \rho^1, (\sigma^1)^2_k$ are additional parameters to be estimated. Similarly for $k^2_{t,p}$. Note that the forecasting models are inspired by Wong et al. (2023).

Value

A list with components:

`post_sample` An `mcmc.list` object containing the posterior samples generated.

`param` A vector of character strings describing the names of model parameters.

`death` The death data that was used.

`expo` The expo data that was used.

`family` The family function used.

`forecast` A logical value indicating if forecast has been performed.

`h` The forecast horizon used.

References

Cairns, A. J. G., Blake, D., Dowd, K., Coughlan, G. D., Epstein, D., Ong, A., and Balevich, I. (2009). A Quantitative Comparison of Stochastic Mortality Models Using Data From England and Wales and the United States. North American Actuarial Journal, 13(1), 1–35. doi:10.1080/10920277.2009.10597538

Jackie S. T. Wong, Jonathan J. Forster, and Peter W. F. Smith. (2018). Bayesian mortality forecasting with overdispersion, Insurance: Mathematics and Economics, Volume 2018, Issue 3, 206-221. doi:10.1016/j.insmatheco.2017.09.023

Jackie S. T. Wong, Jonathan J. Forster, and Peter W. F. Smith. (2023). Bayesian model comparison for mortality forecasting, Journal of the Royal Statistical Society Series C: Applied Statistics, Volume 72, Issue 3, 566–586. doi:10.1093/jrsssc/qlad021

Examples

```
#load and prepare mortality data
data("dxt_array_product");data("Ext_array_product")
death<-preparedata_fn(dxt_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)
expo<-preparedata_fn(Ext_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)

#fit the model (poisson family)
#NOTE: This is a toy example, please run it longer in practice.
fit_CBD_M5_result<-fit_CBD_M5(death=death,expo=expo,n_iter=50,n.adapt=50,family="poisson")
head(fit_CBD_M5_result)
```

`fit_CBD_M6` *A function to fit the stochastic mortality model "M6".*

Description

Carry out Bayesian estimation of the stochastic mortality model referred to as **"M6"** under the formulation of Cairns et al. (2009).

Usage

```
fit_CBD_M6(
  death,
  expo,
  share_gamma = FALSE,
  n_iter = 10000,
  family = "nb",
  n.chain = 1,
  thin = 1,
  n.adapt = 1000,
  forecast = FALSE,
  h = 5,
  quiet = FALSE
)
```

Arguments

`death`	death data that has been formatted through the function `preparedata_fn`.
`expo`	expo data that has been formatted through the function `preparedata_fn`.
`share_gamma`	a logical value indicating if the cohort parameter $\gamma_{c,p}$ should be shared across all strata (see details below). Default is `FALSE`.
`n_iter`	number of iterations to run. Default is `n_iter=10000`.
`family`	a string of characters that defines the family function associated with the mortality model. "poisson" would assume that deaths follow a Poisson distribution and use a log link; "binomial" would assume that deaths follow a Binomial distribution and a logit link; "nb" (default) would assume that deaths follow a Negative-Binomial distribution and a log link.
`n.chain`	number of parallel chains for the model.
`thin`	thinning interval for monitoring purposes.
`n.adapt`	the number of iterations for adaptation. See `?rjags::adapt` for details.
`forecast`	a logical value indicating if forecast is to be performed (default is `FALSE`). See below for details.
`h`	a numeric value giving the number of years to forecast. Default is `h=5`.
`quiet`	if TRUE then messages generated during compilation will be suppressed, as well as the progress bar during adaptation.

Details

The model can be described mathematically as follows: If `family="poisson"`, then

$$d_{x,t,p} \sim \text{Poisson}(E^c_{x,t,p} m_{x,t,p}),$$

$$\log(m_{x,t,p}) = k^1_{t,p} + k^2_{t,p}(x - \bar{x}) + \gamma_{c,p},$$

where $c = t - x$ is the cohort index, $\bar{x}$ is the mean of x, $d_{x,t,p}$ represents the number of deaths at age x in year t of stratum p, while $E^c_{x,t,p}$ and $m_{x,t,p}$ represents respectively the corresponding central exposed to risk and central mortality rate at age x in year t of stratum p. Similarly, if `family="nb"`, then a negative binomial distribution is fitted, i.e.

$$d_{x,t,p} \sim \text{Negative-Binomial}(\phi, \frac{\phi}{\phi + E^c_{x,t,p} m_{x,t,p}}),$$

$$\log(m_{x,t,p}) = k^1_{t,p} + k^2_{t,p}(x - \bar{x}) + \gamma_{c,p},$$

where ϕ is the overdispersion parameter. See Wong et al. (2018). But if `family="binomial"`, then

$$d_{x,t,p} \sim \text{Binomial}(E^0_{x,t,p}, q_{x,t,p}),$$

$$\text{logit}(q_{x,t,p}) = k^1_{t,p} + k^2_{t,p}(x - \bar{x}) + \gamma_{c,p},$$

where $q_{x,t,p}$ represents the initial mortality rate at age x in year t of stratum p, while $E^0_{x,t,p} \approx E^c_{x,t,p} + \frac{1}{2} d_{x,t,p}$ is the corresponding initial exposed to risk. Constraints used are:

$$\sum_c \gamma_{c,p} = \sum_c c\gamma_{c,p} = 0 \quad \text{for each stratum } p.$$

If `share_gamma=TRUE`, then the cohort parameter is the same across all strata p, i. e.

$$k^1_{t,p} + k^2_{t,p}(x - \bar{x}) + \gamma_c.$$

If `forecast=TRUE`, then the following time series models are fitted on $k^1_{t,p}$ as follows:

$$k^1_{t,p} = \eta^1_1 + \eta^1_2 t + \rho(k^1_{t-1,p} - (\eta^1_1 + \eta^1_2(t-1))) + \epsilon^1_{t,p} \text{ for } p = 1, \ldots, P \text{ and } t = 1, \ldots, T,$$

where $\epsilon^1_{t,p} \sim N(0, (\sigma^1)^2_k)$ for $t = 1, \ldots, T$, while $\eta^1_1, \eta^1_2, \rho^1, (\sigma^1)^2_k$ are additional parameters to be estimated. Similarly for $k^2_{t,p}$. Also,

$$\gamma_{c,p} = \gamma_{c-1,p} + \rho_\gamma(\gamma_{c-1,p} - \gamma_{c-2,p}) + \epsilon^\gamma_{c,p} \text{ for } p = 1, \ldots, P \text{ and } c = 3, \ldots, C,$$

$$\gamma_2 = \gamma_1 + \frac{1}{\sqrt{1 - \rho^2_\gamma}} \epsilon^\gamma_{2,p},$$

$$\gamma_1 = 100\epsilon^\gamma_{1,p}$$

where $\epsilon^\gamma_{c,p} \sim N(0, \sigma^2_\gamma)$ for $c = 1, \ldots, C$, while $\rho_\gamma, \sigma^2_\gamma$ are additional parameters to be estimated. Note that the forecasting models are inspired by Wong et al. (2023).

Value

A list with components:

`post_sample` An `mcmc.list` object containing the posterior samples generated.

`param` A vector of character strings describing the names of model parameters.

`death` The death data that was used.

`expo` The expo data that was used.

`family` The family function used.

`forecast` A logical value indicating if forecast has been performed.

`h` The forecast horizon used.

References

Cairns, A. J. G., Blake, D., Dowd, K., Coughlan, G. D., Epstein, D., Ong, A., and Balevich, I. (2009). A Quantitative Comparison of Stochastic Mortality Models Using Data From England and Wales and the United States. North American Actuarial Journal, 13(1), 1–35. doi:10.1080/10920277.2009.10597538

Jackie S. T. Wong, Jonathan J. Forster, and Peter W. F. Smith. (2018). Bayesian mortality forecasting with overdispersion, Insurance: Mathematics and Economics, Volume 2018, Issue 3, 206-221. doi:10.1016/j.insmatheco.2017.09.023

Jackie S. T. Wong, Jonathan J. Forster, and Peter W. F. Smith. (2023). Bayesian model comparison for mortality forecasting, Journal of the Royal Statistical Society Series C: Applied Statistics, Volume 72, Issue 3, 566–586. doi:10.1093/jrsssc/qlad021

Examples

```
#load and prepare mortality data
data("dxt_array_product");data("Ext_array_product")
death<-preparedata_fn(dxt_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)
expo<-preparedata_fn(Ext_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)

#fit the model (negative-binomial family)
#NOTE: This is a toy example, please run it longer in practice.
fit_CBD_M6_result<-fit_CBD_M6(death=death,expo=expo,n_iter=50,n.adapt=50)
head(fit_CBD_M6_result)


#if sharing the cohorts only (poisson family)
fit_CBD_M6_result2<-fit_CBD_M6(death=death,expo=expo,n_iter=1000,family="poisson",share_gamma=TRUE)
head(fit_CBD_M6_result2)
```

`fit_CBD_M7` *A function to fit the stochastic mortality model "M7".*

Description

Carry out Bayesian estimation of the stochastic mortality model referred to as **"M7"** under the formulation of Cairns et al. (2009).

Usage

```
fit_CBD_M7(
  death,
  expo,
  share_gamma = FALSE,
  n_iter = 10000,
  family = "nb",
  n.chain = 1,
```

```
  thin = 1,
  n.adapt = 1000,
  forecast = FALSE,
  h = 5,
  quiet = FALSE
)
```

Arguments

`death`	death data that has been formatted through the function `preparedata_fn`.
`expo`	expo data that has been formatted through the function `preparedata_fn`.
`share_gamma`	a logical value indicating if the cohort parameter $\gamma_{c,p}$ should be shared across all strata (see details below). Default is `FALSE`.
`n_iter`	number of iterations to run. Default is `n_iter=10000`.
`family`	a string of characters that defines the family function associated with the mortality model. "poisson" would assume that deaths follow a Poisson distribution and use a log link; "binomial" would assume that deaths follow a Binomial distribution and a logit link; "nb" (default) would assume that deaths follow a Negative-Binomial distribution and a log link.
`n.chain`	number of parallel chains for the model.
`thin`	thinning interval for monitoring purposes.
`n.adapt`	the number of iterations for adaptation. See `?rjags::adapt` for details.
`forecast`	a logical value indicating if forecast is to be performed (default is `FALSE`). See below for details.
`h`	a numeric value giving the number of years to forecast. Default is `h=5`.
`quiet`	if TRUE then messages generated during compilation will be suppressed, as well as the progress bar during adaptation.

Details

The model can be described mathematically as follows: If `family="poisson"`, then

$$d_{x,t,p} \sim \text{Poisson}(E^c_{x,t,p} m_{x,t,p}),$$

$$\log(m_{x,t,p}) = k^1_{t,p} + k^2_{t,p}(x-\bar{x}) + k^3_{t,p}((x-\bar{x})^2 - \hat{\sigma}^2_x) + \gamma_{c,p},$$

where $c = t - x$ is the cohort index, $\bar{x}$ is the mean of x, $\hat{\sigma}^2_x$ is the mean of $(x-\bar{x})^2$, $d_{x,t,p}$ represents the number of deaths at age x in year t of stratum p, while $E^c_{x,t,p}$ and $m_{x,t,p}$ represents respectively the corresponding central exposed to risk and central mortality rate at age x in year t of stratum p. Similarly, if `family="nb"`, then a negative binomial distribution is fitted, i.e.

$$d_{x,t,p} \sim \text{Negative-Binomial}(\phi, \frac{\phi}{\phi + E^c_{x,t,p} m_{x,t,p}}),$$

$$\log(m_{x,t,p}) = k^1_{t,p} + k^2_{t,p}(x-\bar{x}) + k^3_{t,p}((x-\bar{x})^2 - \hat{\sigma}^2_x) + \gamma_{c,p},$$

where ϕ is the overdispersion parameter. See Wong et al. (2018). But if `family="binomial"`, then

$$d_{x,t,p} \sim \text{Binomial}(E^0_{x,t,p}, q_{x,t,p}),$$

$$\text{logit}(q_{x,t,p}) = k^1_{t,p} + k^2_{t,p}(x-\bar{x}) + k^3_{t,p}((x-\bar{x})^2 - \hat{\sigma}^2_x) + \gamma_{c,p},$$

where $q_{x,t,p}$ represents the initial mortality rate at age x in year t of stratum p, while $E^0_{x,t,p} \approx E^c_{x,t,p} + \frac{1}{2}d_{x,t,p}$ is the corresponding initial exposed to risk. Constraints used are:

$$\sum_c \gamma_{c,p} = \sum_c c\gamma_{c,p} = \sum_c c^2\gamma_{c,p} = 0 \quad \text{for each stratum } p.$$

If `share_gamma=TRUE`, then the cohort parameter is the same across all strata p, i. e.

$$k^1_{t,p} + k^2_{t,p}(x-\bar{x}) + k^3_{t,p}((x-\bar{x})^2 - \hat{\sigma}^2_x) + \gamma_c.$$

If `forecast=TRUE`, then the following time series models are fitted on $k^1_{t,p}$ as follows:

$$k^1_{t,p} = \eta^1_1 + \eta^1_2 t + \rho(k^1_{t-1,p} - (\eta^1_1 + \eta^1_2(t-1))) + \epsilon^1_{t,p} \text{ for } p = 1,\ldots,P \text{ and } t = 1,\ldots,T,$$

where $\epsilon^1_{t,p} \sim N(0,(\sigma^1)^2_k)$ for $t = 1,\ldots,T$, while $\eta^1_1, \eta^1_2, \rho^1, (\sigma^1)^2_k$ are additional parameters to be estimated. Similarly for $k^2_{t,p}$ and $k^3_{t,p}$. Also,

$$\gamma_{c,p} = \gamma_{c-1,p} + \rho_\gamma(\gamma_{c-1,p} - \gamma_{c-2,p}) + \epsilon^\gamma_{c,p} \text{ for } p = 1,\ldots,P \text{ and } c = 3,\ldots,C,$$

$$\gamma_2 = \gamma_1 + \frac{1}{\sqrt{1-\rho^2_\gamma}}\epsilon^\gamma_{2,p},$$

$$\gamma_1 = 100\epsilon^\gamma_{1,p}$$

where $\epsilon^\gamma_{c,p} \sim N(0,\sigma^2_\gamma)$ for $c = 1,\ldots,C$, while $\rho_\gamma, \sigma^2_\gamma$ are additional parameters to be estimated. Note that the forecasting models are inspired by Wong et al. (2023).

Value

A list with components:

`post_sample` An `mcmc.list` object containing the posterior samples generated.

`param` A vector of character strings describing the names of model parameters.

`death` The death data that was used.

`expo` The expo data that was used.

`family` The family function used.

`forecast` A logical value indicating if forecast has been performed.

`h` The forecast horizon used.

References

Cairns, A. J. G., Blake, D., Dowd, K., Coughlan, G. D., Epstein, D., Ong, A., and Balevich, I. (2009). A Quantitative Comparison of Stochastic Mortality Models Using Data From England and Wales and the United States. North American Actuarial Journal, 13(1), 1–35. doi:10.1080/10920277.2009.10597538

Jackie S. T. Wong, Jonathan J. Forster, and Peter W. F. Smith. (2018). Bayesian mortality forecasting with overdispersion, Insurance: Mathematics and Economics, Volume 2018, Issue 3, 206-221. doi:10.1016/j.insmatheco.2017.09.023

Jackie S. T. Wong, Jonathan J. Forster, and Peter W. F. Smith. (2023). Bayesian model comparison for mortality forecasting, Journal of the Royal Statistical Society Series C: Applied Statistics, Volume 72, Issue 3, 566–586. doi:10.1093/jrsssc/qlad021

Examples

```
#load and prepare mortality data
data("dxt_array_product");data("Ext_array_product")
death<-preparedata_fn(dxt_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)
expo<-preparedata_fn(Ext_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)

#fit the model (negative-binomial family)
#NOTE: This is a toy example, please run it longer in practice.
fit_CBD_M7_result<-fit_CBD_M7(death=death,expo=expo,n_iter=50,n.adapt=50)
head(fit_CBD_M7_result)


#if sharing the cohorts only (poisson family)
fit_CBD_M7_result2<-fit_CBD_M7(death=death,expo=expo,n_iter=1000,family="poisson",share_gamma=TRUE)
head(fit_CBD_M7_result2)
```

`fit_CBD_M8` *A function to fit the stochastic mortality model "M8".*

Description

Carry out Bayesian estimation of the stochastic mortality model referred to as **"M8"** under the formulation of Cairns et al. (2009).

Usage

```
fit_CBD_M8(
  death,
  expo,
  share_gamma = FALSE,
  n_iter = 10000,
  family = "nb",
  n.chain = 1,
  thin = 1,
  n.adapt = 1000,
  forecast = FALSE,
  h = 5,
  quiet = FALSE
)
```

Arguments

`death`	death data that has been formatted through the function `preparedata_fn`.
`expo`	expo data that has been formatted through the function `preparedata_fn`.
`share_gamma`	a logical value indicating if the cohort parameter $\gamma_{c,p}$ should be shared across all strata (see details below). Default is `FALSE`.

`n_iter`	number of iterations to run. Default is `n_iter=10000`.
`family`	a string of characters that defines the family function associated with the mortality model. "poisson" would assume that deaths follow a Poisson distribution and use a log link; "binomial" would assume that deaths follow a Binomial distribution and a logit link; "nb" (default) would assume that deaths follow a Negative-Binomial distribution and a log link.
`n.chain`	number of parallel chains for the model.
`thin`	thinning interval for monitoring purposes.
`n.adapt`	the number of iterations for adaptation. See `?rjags::adapt` for details.
`forecast`	a logical value indicating if forecast is to be performed (default is `FALSE`). See below for details.
`h`	a numeric value giving the number of years to forecast. Default is `h=5`.
`quiet`	if TRUE then messages generated during compilation will be suppressed, as well as the progress bar during adaptation.

Details

The model can be described mathematically as follows: If `family="poisson"`, then

$$d_{x,t,p} \sim \text{Poisson}(E^c_{x,t,p} m_{x,t,p}),$$

$$\log(m_{x,t,p}) = k^1_{t,p} + k^2_{t,p}(x - \bar{x}) + \gamma_{c,p}(c_p - x),$$

where $c = t - x$ is the cohort index, $\bar{x}$ is the mean of x, c_p are stratum-specific parameters, $d_{x,t,p}$ represents the number of deaths at age x in year t of stratum p, while $E^c_{x,t,p}$ and $m_{x,t,p}$ represents respectively the corresponding central exposed to risk and central mortality rate at age x in year t of stratum p. Similarly, if `family="nb"`, then a negative binomial distribution is fitted, i.e.

$$d_{x,t,p} \sim \text{Negative-Binomial}(\phi, \frac{\phi}{\phi + E^c_{x,t,p} m_{x,t,p}}),$$

$$\log(m_{x,t,p}) = k^1_{t,p} + k^2_{t,p}(x - \bar{x}) + \gamma_{c,p}(c_p - x),$$

where ϕ is the overdispersion parameter. See Wong et al. (2018). But if `family="binomial"`, then

$$d_{x,t,p} \sim \text{Binomial}(E^0_{x,t,p}, q_{x,t,p}),$$

$$\text{logit}(q_{x,t,p}) = k^1_{t,p} + k^2_{t,p}(x - \bar{x}) + \gamma_{c,p}(c_p - x),$$

where $q_{x,t,p}$ represents the initial mortality rate at age x in year t of stratum p, while $E^0_{x,t,p} \approx E^c_{x,t,p} + \frac{1}{2} d_{x,t,p}$ is the corresponding initial exposed to risk. Constraints used are:

$$\sum_{x,t} \gamma_{c,p} = 0 \quad \text{for each stratum } p.$$

If `share_gamma=TRUE`, then the cohort parameter is the same across all strata p, i. e.

$$k^1_{t,p} + k^2_{t,p}(x - \bar{x}) + \gamma_c(c_p - x).$$

If `forecast=TRUE`, then the following time series models are fitted on $k^1_{t,p}$ as follows:

$$k^1_{t,p} = \eta^1_1 + \eta^1_2 t + \rho(k^1_{t-1,p} - (\eta^1_1 + \eta^1_2(t-1))) + \epsilon^1_{t,p} \text{ for } p = 1, \ldots, P \text{ and } t = 1, \ldots, T,$$

where $\epsilon^1_{t,p} \sim N(0, (\sigma^1)^2_k)$ for $t = 1, \ldots, T$, while $\eta^1_1, \eta^1_2, \rho^1, (\sigma^1)^2_k$ are additional parameters to be estimated. Similarly for $k^2_{t,p}$. Also,

$$\gamma_{c,p} = \gamma_{c-1,p} + \rho_\gamma(\gamma_{c-1,p} - \gamma_{c-2,p}) + \epsilon^\gamma_{c,p} \text{ for } p = 1, \ldots, P \text{ and } c = 3, \ldots, C,$$

$$\gamma_2 = \gamma_1 + \frac{1}{\sqrt{1-\rho^2_\gamma}}\epsilon^\gamma_{2,p},$$

$$\gamma_1 = 100\epsilon^\gamma_{1,p}$$

where $\epsilon^\gamma_{c,p} \sim N(0, \sigma^2_\gamma)$ for $c = 1, \ldots, C$, while $\rho_\gamma, \sigma^2_\gamma$ are additional parameters to be estimated. Note that the forecasting models are inspired by Wong et al. (2023).

Value

A list with components:

`post_sample` An `mcmc.list` object containing the posterior samples generated.

`param` A vector of character strings describing the names of model parameters.

`death` The death data that was used.

`expo` The expo data that was used.

`family` The family function used.

`forecast` A logical value indicating if forecast has been performed.

`h` The forecast horizon used.

References

Cairns, A. J. G., Blake, D., Dowd, K., Coughlan, G. D., Epstein, D., Ong, A., and Balevich, I. (2009). A Quantitative Comparison of Stochastic Mortality Models Using Data From England and Wales and the United States. North American Actuarial Journal, 13(1), 1–35. doi:10.1080/10920277.2009.10597538

Jackie S. T. Wong, Jonathan J. Forster, and Peter W. F. Smith. (2018). Bayesian mortality forecasting with overdispersion, Insurance: Mathematics and Economics, Volume 2018, Issue 3, 206-221. doi:10.1016/j.insmatheco.2017.09.023

Jackie S. T. Wong, Jonathan J. Forster, and Peter W. F. Smith. (2023). Bayesian model comparison for mortality forecasting, Journal of the Royal Statistical Society Series C: Applied Statistics, Volume 72, Issue 3, 566–586. doi:10.1093/jrsssc/qlad021

Examples

```
#load and prepare mortality data
data("dxt_array_product");data("Ext_array_product")
death<-preparedata_fn(dxt_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)
expo<-preparedata_fn(Ext_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)
```

```
#fit the model (poisson family)
#Note that this model sometimes fails numerically.
try(fit_CBD_M8_result<-fit_CBD_M8(death=death,expo=expo,n_iter=1000,family="poisson"))

#if sharing the cohorts only (negative-binomial family)
try(fit_CBD_M8_result2<-fit_CBD_M8(death=death,expo=expo,n_iter=1000,share_gamma=TRUE))
```

`fit_LC` *A function to fit the stochastic mortality model by Lee and Carter (1992).*

Description

Carry out Bayesian estimation of the stochastic mortality model, the **Lee-Carter model** (Lee and Carter, 1992). Note that this model is equivalent to "M1" under the formulation of Cairns et al. (2009).

Usage

```
fit_LC(
  death,
  expo,
  n_iter = 10000,
  family = "nb",
  share_alpha = FALSE,
  share_beta = FALSE,
  n.chain = 1,
  thin = 1,
  n.adapt = 1000,
  forecast = FALSE,
  h = 5,
  quiet = FALSE
)
```

Arguments

`death`	death data that has been formatted through the function `preparedata_fn`.
`expo`	expo data that has been formatted through the function `preparedata_fn`.
`n_iter`	number of iterations to run. Default is `n_iter=10000`.
`family`	a string of characters that defines the family function associated with the mortality model. "poisson" would assume that deaths follow a Poisson distribution and use a log link; "binomial" would assume that deaths follow a Binomial distribution and a logit link; "nb" (default) would assume that deaths follow a Negative-Binomial distribution and a log link.

`share_alpha`	a logical value indicating if $a_{x,p}$ should be shared across all strata (see details below). Default is `FALSE`.
`share_beta`	a logical value indicating if $b_{x,p}$ should be shared across all strata (see details below). Default is `FALSE`.
`n.chain`	number of parallel chains for the model.
`thin`	thinning interval for monitoring purposes.
`n.adapt`	the number of iterations for adaptation. See `?rjags::adapt` for details.
`forecast`	a logical value indicating if forecast is to be performed (default is `FALSE`). See below for details.
`h`	a numeric value giving the number of years to forecast. Default is `h=5`.
`quiet`	if TRUE then messages generated during compilation will be suppressed, as well as the progress bar during adaptation.

Details

The model can be described mathematically as follows: If `family="poisson"`, then

$$d_{x,t,p} \sim \text{Poisson}(E^c_{x,t,p} m_{x,t,p}),$$

$$\log(m_{x,t,p}) = a_{x,p} + b_{x,p} k_{t,p},$$

where $d_{x,t,p}$ represents the number of deaths at age x in year t of stratum p, while $E^c_{x,t,p}$ and $m_{x,t,p}$ represents respectively the corresponding central exposed to risk and central mortality rate at age x in year t of stratum p. Similarly, if `family="nb"`, then a negative binomial distribution is fitted, i.e.

$$d_{x,t,p} \sim \text{Negative-Binomial}(\phi, \frac{\phi}{\phi + E^c_{x,t,p} m_{x,t,p}}),$$

$$\log(m_{x,t,p}) = a_{x,p} + b_{x,p} k_{t,p},$$

where ϕ is the overdispersion parameter. See Wong et al. (2018). But if `family="binomial"`, then

$$d_{x,t,p} \sim \text{Binomial}(E^0_{x,t,p}, q_{x,t,p}),$$

$$\text{logit}(q_{x,t,p}) = a_{x,p} + b_{x,p} k_{t,p},$$

where $q_{x,t,p}$ represents the initial mortality rate at age x in year t of stratum p, while $E^0_{x,t,p} \approx E^c_{x,t,p} + \frac{1}{2} d_{x,t,p}$ is the corresponding initial exposed to risk. Constraints used are:

$$\sum_x b_{x,p} = 1, \sum_t k_{t,p} = 0 \quad \text{for each stratum } p.$$

If `share_alpha=TRUE`, then the additive age-specific parameter is the same across all strata p, i. e.

$$a_x + b_{x,p} k_{t,p}.$$

Similarly if `share_beta=TRUE`, then the multiplicative age-specific parameter is the same across all strata p, i. e.

$$a_{x,p} + b_x k_{t,p}.$$

If `forecast=TRUE`, then a time series model (an AR(1) with linear drift) will be fitted on $k_{t,p}$ as follows:

$$k_{t,p} = \eta_1 + \eta_2 t + \rho(k_{t-1,p} - (\eta_1 + \eta_2(t-1))) + \epsilon_{t,p} \text{ for } p = 1, \ldots, P \text{ and } t = 1, \ldots, T,$$

where $\epsilon_{t,p} \sim N(0, \sigma_k^2)$, $\eta_1, \eta_2, \rho, \sigma_k^2$ are additional parameters to be estimated. In principle, there are many other options for forecasting the mortality time trend. But currently, we assume that this serves as a general purpose forecasting model for simplicity.

Value

A list with components:

`post_sample`	An `mcmc.list` object containing the posterior samples generated.
`param`	A vector of character strings describing the names of model parameters.
`death`	The death data that was used.
`expo`	The expo data that was used.
`family`	The family function used.
`forecast`	A logical value indicating if forecast has been performed.
`h`	The forecast horizon used.

References

Cairns, A. J. G., Blake, D., Dowd, K., Coughlan, G. D., Epstein, D., Ong, A., and Balevich, I. (2009). A Quantitative Comparison of Stochastic Mortality Models Using Data From England and Wales and the United States. North American Actuarial Journal, 13(1), 1–35. doi:10.1080/10920277.2009.10597538

Lee, R. D., and Carter, L. R. (1992). Modeling and Forecasting U.S. Mortality. Journal of the American Statistical Association, 87(419), 659–671. doi:10.1080/01621459.1992.10475265

Jackie S. T. Wong, Jonathan J. Forster, and Peter W. F. Smith. (2018). Bayesian mortality forecasting with overdispersion, Insurance: Mathematics and Economics, Volume 2018, Issue 3, 206-221. doi:10.1016/j.insmatheco.2017.09.023

Jackie S. T. Wong, Jonathan J. Forster, and Peter W. F. Smith. (2023). Bayesian model comparison for mortality forecasting, Journal of the Royal Statistical Society Series C: Applied Statistics, Volume 72, Issue 3, 566–586. doi:10.1093/jrsssc/qlad021

Examples

```
#load and prepare mortality data
data("dxt_array_product");data("Ext_array_product")
death<-preparedata_fn(dxt_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)
expo<-preparedata_fn(Ext_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)

#fit the model (negative-binomial family)
#NOTE: This is a toy example, please run it longer in practice.
fit_LC_result<-fit_LC(death=death,expo=expo,n_iter=50,n.adapt=50)
head(fit_LC_result)
```

```
#fit the model (poisson family)
fit_LC_result<-fit_LC(death=death,expo=expo,n_iter=1000,family="poisson")
head(fit_LC_result)

#if sharing the betas
fit_LC_result2<-fit_LC(death=death,expo=expo,n_iter=1000,family="poisson",share_beta=TRUE)
head(fit_LC_result2)

#if sharing both alphas and betas
fit_LC_result3<-fit_LC(death=death,expo=expo,n_iter=1000,family="poisson",
share_alpha=TRUE,share_beta=TRUE)
head(fit_LC_result3)

#if forecast
fit_LC_result4<-fit_LC(death=death,expo=expo,n_iter=1000,family="poisson",forecast=TRUE)
plot_rates_fn(fit_LC_result4)
plot_param_fn(fit_LC_result4)
```

`fit_M1A` *A function to fit the stochastic mortality model M1A.*

Description

Carry out Bayesian estimation of the stochastic mortality **model M1A**.

Usage

```
fit_M1A(
  death,
  expo,
  n_iter = 10000,
  family = "nb",
  n.chain = 1,
  thin = 1,
  n.adapt = 1000,
  forecast = FALSE,
  h = 5,
  quiet = FALSE
)
```

Arguments

`death`	death data that has been formatted through the function `preparedata_fn`.
`expo`	expo data that has been formatted through the function `preparedata_fn`.
`n_iter`	number of iterations to run. Default is `n_iter=10000`.

`family`	a string of characters that defines the family function associated with the mortality model. "poisson" would assume that deaths follow a Poisson distribution and use a log link; "binomial" would assume that deaths follow a Binomial distribution and a logit link; "nb" (default) would assume that deaths follow a Negative-Binomial distribution and a log link.
`n.chain`	number of parallel chains for the model.
`thin`	thinning interval for monitoring purposes.
`n.adapt`	the number of iterations for adaptation. See `?rjags::adapt` for details.
`forecast`	a logical value indicating if forecast is to be performed (default is `FALSE`). See below for details.
`h`	a numeric value giving the number of years to forecast. Default is `h=5`.
`quiet`	if TRUE then messages generated during compilation will be suppressed, as well as the progress bar during adaptation.

Details

The model can be described mathematically as follows: If `family="poisson"`, then

$$d_{x,t,p} \sim \text{Poisson}(E^c_{x,t,p} m_{x,t,p}),$$

$$\log(m_{x,t,p}) = a_x + c_p + b_x k_t,$$

where $d_{x,t,p}$ represents the number of deaths at age x in year t of stratum p, while $E^c_{x,t,p}$ and $m_{x,t,p}$ represents respectively the corresponding central exposed to risk and central mortality rate at age x in year t of stratum p. Similarly, if `family="nb"`, then a negative binomial distribution is fitted, i.e.

$$d_{x,t,p} \sim \text{Negative-Binomial}(\phi, \frac{\phi}{\phi + E^c_{x,t,p} m_{x,t,p}}),$$

$$\log(m_{x,t,p}) = a_x + c_p + b_x k_t,$$

where ϕ is the overdispersion parameter. See Wong et al. (2018). But if `family="binomial"`, then

$$d_{x,t,p} \sim \text{Binomial}(E^0_{x,t,p}, q_{x,t,p}),$$

$$\text{logit}(q_{x,t,p}) = a_x + c_p + b_x k_t,$$

where $q_{x,t,p}$ represents the initial mortality rate at age x in year t of stratum p, while $E^0_{x,t,p} \approx E^c_{x,t,p} + \frac{1}{2} d_{x,t,p}$ is the corresponding initial exposed to risk. Constraints used are:

$$\sum_p c_p = 0, \sum_x b_x = 1, \sum_t k_t = 0.$$

If `forecast=TRUE`, then the following time series models are fitted on $k_{t,p}$ as follows:

$$k_{t,p} = \eta_1 + \eta_2 t + \rho(k_{t-1,p} - (\eta_1 + \eta_2(t-1))) + \epsilon_{t,p} \text{ for } p = 1, \ldots, P \text{ and } t = 1, \ldots, T,$$

where $\epsilon_{t,p} \sim N(0, \sigma_k^2)$ for $t = 1, \ldots, T$, while $\eta_1, \eta_2, \rho, \sigma_k^2$ are additional parameters to be estimated. Note that the forecasting models are inspired by Wong et al. (2023).

Value

A list with components:

`post_sample` An `mcmc` object containing the posterior samples generated.

`param` A vector of character strings describing the names of model parameters.

`death` The death data that was used.

`expo` The expo data that was used.

`family` The family function used.

`forecast` A logical value indicating if forecast has been performed.

`h` The forecast horizon used.

References

Jackie S. T. Wong, Jonathan J. Forster, and Peter W. F. Smith. (2018). Bayesian mortality forecasting with overdispersion, Insurance: Mathematics and Economics, Volume 2018, Issue 3, 206-221. doi:10.1016/j.insmatheco.2017.09.023

Jackie S. T. Wong, Jonathan J. Forster, and Peter W. F. Smith. (2023). Bayesian model comparison for mortality forecasting, Journal of the Royal Statistical Society Series C: Applied Statistics, Volume 72, Issue 3, 566–586. doi:10.1093/jrsssc/qlad021

Examples

```
#load and prepare mortality data
data("dxt_array_product");data("Ext_array_product")
death<-preparedata_fn(dxt_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)
expo<-preparedata_fn(Ext_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)

#fit the model (negative-binomial family)
#NOTE: This is a toy example, please run it longer in practice.
fit_M1A_result<-fit_M1A(death=death,expo=expo,n_iter=50,n.adapt=50,family="nb")
head(fit_M1A_result)
```

`fit_M1M` *A function to fit the stochastic mortality model M1M.*

Description

Carry out Bayesian estimation of the stochastic mortality **model M1M**.

Usage

```
fit_M1M(
  death,
  expo,
  n_iter = 10000,
  family = "nb",
```

```
  n.chain = 1,
  thin = 1,
  n.adapt = 1000,
  forecast = FALSE,
  h = 5,
  quiet = FALSE
)
```

Arguments

`death`	death data that has been formatted through the function `preparedata_fn`.
`expo`	expo data that has been formatted through the function `preparedata_fn`.
`n_iter`	number of iterations to run. Default is `n_iter=10000`.
`family`	a string of characters that defines the family function associated with the mortality model. "poisson" would assume that deaths follow a Poisson distribution and use a log link; "binomial" would assume that deaths follow a Binomial distribution and a logit link; "nb" (default) would assume that deaths follow a Negative-Binomial distribution and a log link.
`n.chain`	number of parallel chains for the model.
`thin`	thinning interval for monitoring purposes.
`n.adapt`	the number of iterations for adaptation. See `?rjags::adapt` for details.
`forecast`	a logical value indicating if forecast is to be performed (default is `FALSE`). See below for details.
`h`	a numeric value giving the number of years to forecast. Default is `h=5`.
`quiet`	if TRUE then messages generated during compilation will be suppressed, as well as the progress bar during adaptation.

Details

The model can be described mathematically as follows: If `family="poisson"`, then

$$d_{x,t,p} \sim \text{Poisson}(E^c_{x,t,p} m_{x,t,p}),$$

$$\log(m_{x,t,p}) = a_x c_p + b_x k_t,$$

where $d_{x,t,p}$ represents the number of deaths at age x in year t of stratum p, while $E^c_{x,t,p}$ and $m_{x,t,p}$ represents respectively the corresponding central exposed to risk and central mortality rate at age x in year t of stratum p. Similarly, if `family="nb"`, then a negative binomial distribution is fitted, i.e.

$$d_{x,t,p} \sim \text{Negative-Binomial}(\phi, \frac{\phi}{\phi + E^c_{x,t,p} m_{x,t,p}}),$$

$$\log(m_{x,t,p}) = a_x c_p + b_x k_t,$$

where ϕ is the overdispersion parameter. See Wong et al. (2018). But if `family="binomial"`, then

$$d_{x,t,p} \sim \text{Binomial}(E^0_{x,t,p}, q_{x,t,p}),$$

$$\text{logit}(q_{x,t,p}) = a_x c_p + b_x k_t,$$

where $q_{x,t,p}$ represents the initial mortality rate at age x in year t of stratum p, while $E^0_{x,t,p} \approx E^c_{x,t,p} + \frac{1}{2}d_{x,t,p}$ is the corresponding initial exposed to risk. Constraints used are:

$$\sum_p c_p = 1, \sum_x b_x = 1, \sum_t k_t = 0.$$

If `forecast=TRUE`, then the following time series models are fitted on $k_{t,p}$ as follows:

$$k_{t,p} = \eta_1 + \eta_2 t + \rho(k_{t-1,p} - (\eta_1 + \eta_2(t-1))) + \epsilon_{t,p} \text{ for } p = 1, \ldots, P \text{ and } t = 1, \ldots, T,$$

where $\epsilon_{t,p} \sim N(0, \sigma_k^2)$ for $t = 1, \ldots, T$, while $\eta_1, \eta_2, \rho, \sigma_k^2$ are additional parameters to be estimated. Note that the forecasting models are inspired by Wong et al. (2023).

Value

A list with components:

`post_sample`	An `mcmc.list` object containing the posterior samples generated.
`param`	A vector of character strings describing the names of model parameters.
`death`	The death data that was used.
`expo`	The expo data that was used.
`family`	The family function used.
`forecast`	A logical value indicating if forecast has been performed.
`h`	The forecast horizon used.

References

Jackie S. T. Wong, Jonathan J. Forster, and Peter W. F. Smith. (2018). Bayesian mortality forecasting with overdispersion, Insurance: Mathematics and Economics, Volume 2018, Issue 3, 206-221. doi:10.1016/j.insmatheco.2017.09.023

Jackie S. T. Wong, Jonathan J. Forster, and Peter W. F. Smith. (2023). Bayesian model comparison for mortality forecasting, Journal of the Royal Statistical Society Series C: Applied Statistics, Volume 72, Issue 3, 566–586. doi:10.1093/jrsssc/qlad021

Examples

```
#load and prepare mortality data
data("dxt_array_product");data("Ext_array_product")
death<-preparedata_fn(dxt_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)
expo<-preparedata_fn(Ext_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)

#fit the model (negative-binomial family)
#NOTE: This is a toy example, please run it longer in practice.
fit_M1M_result<-fit_M1M(death=death,expo=expo,n_iter=50,n.adapt=50)
head(fit_M1M_result)
```

`fit_M1U` *A function to fit the stochastic mortality model M1U.*

Description

Carry out Bayesian estimation of the stochastic mortality **model M1U**.

Usage

```
fit_M1U(
  death,
  expo,
  n_iter = 10000,
  family = "nb",
  n.chain = 1,
  thin = 1,
  n.adapt = 1000,
  forecast = FALSE,
  h = 5,
  quiet = FALSE
)
```

Arguments

`death`	death data that has been formatted through the function `preparedata_fn`.
`expo`	expo data that has been formatted through the function `preparedata_fn`.
`n_iter`	number of iterations to run. Default is `n_iter=10000`.
`family`	a string of characters that defines the family function associated with the mortality model. "poisson" would assume that deaths follow a Poisson distribution and use a log link; "binomial" would assume that deaths follow a Binomial distribution and a logit link; "nb" (default) would assume that deaths follow a Negative-Binomial distribution and a log link.
`n.chain`	number of parallel chains for the model.
`thin`	thinning interval for monitoring purposes.
`n.adapt`	the number of iterations for adaptation. See `?rjags::adapt` for details.
`forecast`	a logical value indicating if forecast is to be performed (default is `FALSE`). See below for details.
`h`	a numeric value giving the number of years to forecast. Default is `h=5`.
`quiet`	if TRUE then messages generated during compilation will be suppressed, as well as the progress bar during adaptation.

Details

The model can be described mathematically as follows: If `family="poisson"`, then

$$d_{x,t,p} \sim \text{Poisson}(E^c_{x,t,p} m_{x,t,p}),$$

$$\log(m_{x,t,p}) = a_{x,p} + b_x k_t,$$

where $d_{x,t,p}$ represents the number of deaths at age x in year t of stratum p, while $E^c_{x,t,p}$ and $m_{x,t,p}$ represents respectively the corresponding central exposed to risk and central mortality rate at age x in year t of stratum p. Similarly, if `family="nb"`, then a negative binomial distribution is fitted, i.e.

$$d_{x,t,p} \sim \text{Negative-Binomial}(\phi, \frac{\phi}{\phi + E^c_{x,t,p} m_{x,t,p}}),$$

$$\log(m_{x,t,p}) = a_{x,p} + b_x k_t,$$

where ϕ is the overdispersion parameter. See Wong et al. (2018). But if `family="binomial"`, then

$$d_{x,t,p} \sim \text{Binomial}(E^0_{x,t,p}, q_{x,t,p}),$$

$$\text{logit}(q_{x,t,p}) = a_{x,p} + b_x k_t,$$

where $q_{x,t,p}$ represents the initial mortality rate at age x in year t of stratum p, while $E^0_{x,t,p} \approx E^c_{x,t,p} + \frac{1}{2} d_{x,t,p}$ is the corresponding initial exposed to risk. Constraints used are:

$$\sum_x b_x = 1, \sum_t k_t = 0.$$

If `forecast=TRUE`, then the following time series models are fitted on $k_{t,p}$ as follows:

$$k_{t,p} = \eta_1 + \eta_2 t + \rho(k_{t-1,p} - (\eta_1 + \eta_2(t-1))) + \epsilon_{t,p} \text{ for } p = 1, \dots, P \text{ and } t = 1, \dots, T,$$

where $\epsilon_{t,p} \sim N(0, \sigma_k^2)$ for $t = 1, \dots, T$, while $\eta_1, \eta_2, \rho, \sigma_k^2$ are additional parameters to be estimated. Note that the forecasting models are inspired by Wong et al. (2023).

Value

A list with components:

`post_sample` An `mcmc.list` object containing the posterior samples generated.

`param` A vector of character strings describing the names of model parameters.

`death` The death data that was used.

`expo` The expo data that was used.

`family` The family function used.

`forecast` A logical value indicating if forecast has been performed.

`h` The forecast horizon used.

References

Jackie S. T. Wong, Jonathan J. Forster, and Peter W. F. Smith. (2018). Bayesian mortality forecasting with overdispersion, Insurance: Mathematics and Economics, Volume 2018, Issue 3, 206-221. doi:10.1016/j.insmatheco.2017.09.023

Jackie S. T. Wong, Jonathan J. Forster, and Peter W. F. Smith. (2023). Bayesian model comparison for mortality forecasting, Journal of the Royal Statistical Society Series C: Applied Statistics, Volume 72, Issue 3, 566–586. doi:10.1093/jrsssc/qlad021

Examples

```
#load and prepare mortality data
data("dxt_array_product");data("Ext_array_product")
death<-preparedata_fn(dxt_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)
expo<-preparedata_fn(Ext_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)

#fit the model (negative-binomial family)
#NOTE: This is a toy example, please run it longer in practice.
fit_M1U_result<-fit_M1U(death=death,expo=expo,n_iter=50,n.adapt=50)
head(fit_M1U_result)
```

`fit_M2A1` *A function to fit the stochastic mortality model M2A1.*

Description

Carry out Bayesian estimation of the stochastic mortality **model M2A1**.

Usage

```
fit_M2A1(
  death,
  expo,
  n_iter = 10000,
  family = "nb",
  n.chain = 1,
  thin = 1,
  n.adapt = 1000,
  forecast = FALSE,
  h = 5,
  quiet = FALSE
)
```

Arguments

`death`	death data that has been formatted through the function `preparedata_fn`.
`expo`	expo data that has been formatted through the function `preparedata_fn`.
`n_iter`	number of iterations to run. Default is `n_iter=10000`.

`family`	a string of characters that defines the family function associated with the mortality model. "poisson" would assume that deaths follow a Poisson distribution and use a log link; "binomial" would assume that deaths follow a Binomial distribution and a logit link; "nb" (default) would assume that deaths follow a Negative-Binomial distribution and a log link.
`n.chain`	number of parallel chains for the model.
`thin`	thinning interval for monitoring purposes.
`n.adapt`	the number of iterations for adaptation. See `?rjags::adapt` for details.
`forecast`	a logical value indicating if forecast is to be performed (default is `FALSE`). See below for details.
`h`	a numeric value giving the number of years to forecast. Default is `h=5`.
`quiet`	if TRUE then messages generated during compilation will be suppressed, as well as the progress bar during adaptation.

Details

The model can be described mathematically as follows: If `family="poisson"`, then

$$d_{x,t,p} \sim \text{Poisson}(E^c_{x,t,p} m_{x,t,p}),$$

$$\log(m_{x,t,p}) = a_x + (c_p + b_x)k_t,$$

where $d_{x,t,p}$ represents the number of deaths at age x in year t of stratum p, while $E^c_{x,t,p}$ and $m_{x,t,p}$ represents respectively the corresponding central exposed to risk and central mortality rate at age x in year t of stratum p. Similarly, if `family="nb"`, then a negative binomial distribution is fitted, i.e.

$$d_{x,t,p} \sim \text{Negative-Binomial}(\phi, \frac{\phi}{\phi + E^c_{x,t,p} m_{x,t,p}}),$$

$$\log(m_{x,t,p}) = a_x + (c_p + b_x)k_t,$$

where ϕ is the overdispersion parameter. See Wong et al. (2018). But if `family="binomial"`, then

$$d_{x,t,p} \sim \text{Binomial}(E^0_{x,t,p}, q_{x,t,p}),$$

$$\text{logit}(q_{x,t,p}) = a_x + (c_p + b_x)k_t,$$

where $q_{x,t,p}$ represents the initial mortality rate at age x in year t of stratum p, while $E^0_{x,t,p} \approx E^c_{x,t,p} + \frac{1}{2}d_{x,t,p}$ is the corresponding initial exposed to risk. Constraints used are:

$$\sum_p c_p = 0, \sum_x b_x = 1, \sum_t k_t = 0.$$

If `forecast=TRUE`, then the following time series models are fitted on $k_{t,p}$ as follows:

$$k_{t,p} = \eta_1 + \eta_2 t + \rho(k_{t-1,p} - (\eta_1 + \eta_2(t-1))) + \epsilon_{t,p} \text{ for } p = 1, \ldots, P \text{ and } t = 1, \ldots, T,$$

where $\epsilon_{t,p} \sim N(0, \sigma_k^2)$ for $t = 1, \ldots, T$, while $\eta_1, \eta_2, \rho, \sigma_k^2$ are additional parameters to be estimated. Note that the forecasting models are inspired by Wong et al. (2023).

Value

A list with components:

`post_sample` An `mcmc.list` object containing the posterior samples generated.

`param` A vector of character strings describing the names of model parameters.

`death` The death data that was used.

`expo` The expo data that was used.

`family` The family function used.

`forecast` A logical value indicating if forecast has been performed.

`h` The forecast horizon used.

References

Jackie S. T. Wong, Jonathan J. Forster, and Peter W. F. Smith. (2018). Bayesian mortality forecasting with overdispersion, Insurance: Mathematics and Economics, Volume 2018, Issue 3, 206-221. doi:10.1016/j.insmatheco.2017.09.023

Jackie S. T. Wong, Jonathan J. Forster, and Peter W. F. Smith. (2023). Bayesian model comparison for mortality forecasting, Journal of the Royal Statistical Society Series C: Applied Statistics, Volume 72, Issue 3, 566–586. doi:10.1093/jrsssc/qlad021

Examples

```
#load and prepare mortality data
data("dxt_array_product");data("Ext_array_product")
death<-preparedata_fn(dxt_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)
expo<-preparedata_fn(Ext_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)

#fit the model (poisson family)
#NOTE: This is a toy example, please run it longer in practice.
fit_M2A1_result<-fit_M2A1(death=death,expo=expo,n_iter=50,n.adapt=50,family="poisson")
head(fit_M2A1_result)
```

`fit_M2A2` *A function to fit the stochastic mortality model M2A2.*

Description

Carry out Bayesian estimation of the stochastic mortality **model M2A2**.

Usage

```
fit_M2A2(
  death,
  expo,
  n_iter = 10000,
  family = "nb",
```

```
  n.chain = 1,
  thin = 1,
  n.adapt = 1000,
  forecast = FALSE,
  h = 5,
  quiet = FALSE
)
```

Arguments

`death`	death data that has been formatted through the function `preparedata_fn`.
`expo`	expo data that has been formatted through the function `preparedata_fn`.
`n_iter`	number of iterations to run. Default is `n_iter=10000`.
`family`	a string of characters that defines the family function associated with the mortality model. "poisson" would assume that deaths follow a Poisson distribution and use a log link; "binomial" would assume that deaths follow a Binomial distribution and a logit link; "nb" (default) would assume that deaths follow a Negative-Binomial distribution and a log link.
`n.chain`	number of parallel chains for the model.
`thin`	thinning interval for monitoring purposes.
`n.adapt`	the number of iterations for adaptation. See `?rjags::adapt` for details.
`forecast`	a logical value indicating if forecast is to be performed (default is `FALSE`). See below for details.
`h`	a numeric value giving the number of years to forecast. Default is `h=5`.
`quiet`	if TRUE then messages generated during compilation will be suppressed, as well as the progress bar during adaptation.

Details

The model can be described mathematically as follows: If `family="poisson"`, then

$$d_{x,t,p} \sim \text{Poisson}(E^c_{x,t,p} m_{x,t,p}),$$

$$\log(m_{x,t,p}) = a_x + b_{x,p} k_t,$$

where $d_{x,t,p}$ represents the number of deaths at age x in year t of stratum p, while $E^c_{x,t,p}$ and $m_{x,t,p}$ represents respectively the corresponding central exposed to risk and central mortality rate at age x in year t of stratum p. Similarly, if `family="nb"`, then a negative binomial distribution is fitted, i.e.

$$d_{x,t,p} \sim \text{Negative-Binomial}(\phi, \frac{\phi}{\phi + E^c_{x,t,p} m_{x,t,p}}),$$

$$\log(m_{x,t,p}) = a_x + b_{x,p} k_t,$$

where ϕ is the overdispersion parameter. See Wong et al. (2018). But if `family="binomial"`, then

$$d_{x,t,p} \sim \text{Binomial}(E^0_{x,t,p}, q_{x,t,p}),$$

$$\text{logit}(q_{x,t,p}) = a_x + b_{x,p} k_t,$$

where $q_{x,t,p}$ represents the initial mortality rate at age x in year t of stratum p, while $E^0_{x,t,p} \approx E^c_{x,t,p} + \frac{1}{2}d_{x,t,p}$ is the corresponding initial exposed to risk. Constraints used are:

$$\sum_{x,p} b_{x,p} = 1, \sum_t k_t = 0.$$

If `forecast=TRUE`, then the following time series models are fitted on $k_{t,p}$ as follows:

$$k_{t,p} = \eta_1 + \eta_2 t + \rho(k_{t-1,p} - (\eta_1 + \eta_2(t-1))) + \epsilon_{t,p} \text{ for } p = 1, \dots, P \text{ and } t = 1, \dots, T,$$

where $\epsilon_{t,p} \sim N(0, \sigma_k^2)$ for $t = 1, \dots, T$, while $\eta_1, \eta_2, \rho, \sigma_k^2$ are additional parameters to be estimated. Note that the forecasting models are inspired by Wong et al. (2023).

Value

A list with components:

`post_sample`	An `mcmc.list` object containing the posterior samples generated.
`param`	A vector of character strings describing the names of model parameters.
`death`	The death data that was used.
`expo`	The expo data that was used.
`family`	The family function used.
`forecast`	A logical value indicating if forecast has been performed.
`h`	The forecast horizon used.

References

Jackie S. T. Wong, Jonathan J. Forster, and Peter W. F. Smith. (2018). Bayesian mortality forecasting with overdispersion, Insurance: Mathematics and Economics, Volume 2018, Issue 3, 206-221. doi:10.1016/j.insmatheco.2017.09.023

Jackie S. T. Wong, Jonathan J. Forster, and Peter W. F. Smith. (2023). Bayesian model comparison for mortality forecasting, Journal of the Royal Statistical Society Series C: Applied Statistics, Volume 72, Issue 3, 566–586. doi:10.1093/jrsssc/qlad021

Examples

```
#load and prepare mortality data
data("dxt_array_product");data("Ext_array_product")
death<-preparedata_fn(dxt_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)
expo<-preparedata_fn(Ext_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)

#fit the model (poisson family)
#NOTE: This is a toy example, please run it longer in practice.
fit_M2A2_result<-fit_M2A2(death=death,expo=expo,n_iter=50,n.adapt=50,family="poisson")
head(fit_M2A2_result)
```

`fit_M2Y1` *A function to fit the stochastic mortality model M2Y1.*

Description

Carry out Bayesian estimation of the stochastic mortality **model M2Y1**.

Usage

```
fit_M2Y1(
  death,
  expo,
  n_iter = 10000,
  family = "nb",
  n.chain = 1,
  thin = 1,
  n.adapt = 1000,
  forecast = FALSE,
  h = 5,
  quiet = FALSE
)
```

Arguments

`death`	death data that has been formatted through the function `preparedata_fn`.
`expo`	expo data that has been formatted through the function `preparedata_fn`.
`n_iter`	number of iterations to run. Default is `n_iter=10000`.
`family`	a string of characters that defines the family function associated with the mortality model. "poisson" would assume that deaths follow a Poisson distribution and use a log link; "binomial" would assume that deaths follow a Binomial distribution and a logit link; "nb" (default) would assume that deaths follow a Negative-Binomial distribution and a log link.
`n.chain`	number of parallel chains for the model.
`thin`	thinning interval for monitoring purposes.
`n.adapt`	the number of iterations for adaptation. See `?rjags::adapt` for details.
`forecast`	a logical value indicating if forecast is to be performed (default is `FALSE`). See below for details.
`h`	a numeric value giving the number of years to forecast. Default is `h=5`.
`quiet`	if TRUE then messages generated during compilation will be suppressed, as well as the progress bar during adaptation.

Details

The model can be described mathematically as follows: If `family="poisson"`, then

$$d_{x,t,p} \sim \text{Poisson}(E^c_{x,t,p} m_{x,t,p}),$$

$$\log(m_{x,t,p}) = a_x + b_x(k_t + c_p),$$

where $d_{x,t,p}$ represents the number of deaths at age x in year t of stratum p, while $E^c_{x,t,p}$ and $m_{x,t,p}$ represents respectively the corresponding central exposed to risk and central mortality rate at age x in year t of stratum p. Similarly, if `family="nb"`, then a negative binomial distribution is fitted, i.e.

$$d_{x,t,p} \sim \text{Negative-Binomial}(\phi, \frac{\phi}{\phi + E^c_{x,t,p} m_{x,t,p}}),$$

$$\log(m_{x,t,p}) = a_x + b_x(k_t + c_p),$$

where ϕ is the overdispersion parameter. See Wong et al. (2018). But if `family="binomial"`, then

$$d_{x,t,p} \sim \text{Binomial}(E^0_{x,t,p}, q_{x,t,p}),$$

$$\text{logit}(q_{x,t,p}) = a_x + b_x(k_t + c_p),$$

where $q_{x,t,p}$ represents the initial mortality rate at age x in year t of stratum p, while $E^0_{x,t,p} \approx E^c_{x,t,p} + \frac{1}{2} d_{x,t,p}$ is the corresponding initial exposed to risk. Constraints used are:

$$\sum_p c_p = 0, \sum_x b_x = 1, \sum_t k_t = 0.$$

If `forecast=TRUE`, then the following time series models are fitted on $k_{t,p}$ as follows:

$$k_{t,p} = \eta_1 + \eta_2 t + \rho(k_{t-1,p} - (\eta_1 + \eta_2(t-1))) + \epsilon_{t,p} \text{ for } p = 1, \ldots, P \text{ and } t = 1, \ldots, T,$$

where $\epsilon_{t,p} \sim N(0, \sigma_k^2)$ for $t = 1, \ldots, T$, while $\eta_1, \eta_2, \rho, \sigma_k^2$ are additional parameters to be estimated. Note that the forecasting models are inspired by Wong et al. (2023).

Value

A list with components:

`post_sample`	An `mcmc.list` object containing the posterior samples generated.
`param`	A vector of character strings describing the names of model parameters.
`death`	The death data that was used.
`expo`	The expo data that was used.
`family`	The family function used.
`forecast`	A logical value indicating if forecast has been performed.
`h`	The forecast horizon used.

References

Jackie S. T. Wong, Jonathan J. Forster, and Peter W. F. Smith. (2018). Bayesian mortality forecasting with overdispersion, Insurance: Mathematics and Economics, Volume 2018, Issue 3, 206-221. doi:10.1016/j.insmatheco.2017.09.023

Jackie S. T. Wong, Jonathan J. Forster, and Peter W. F. Smith. (2023). Bayesian model comparison for mortality forecasting, Journal of the Royal Statistical Society Series C: Applied Statistics, Volume 72, Issue 3, 566–586. doi:10.1093/jrsssc/qlad021

Examples

```
#load and prepare mortality data
data("dxt_array_product");data("Ext_array_product")
death<-preparedata_fn(dxt_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)
expo<-preparedata_fn(Ext_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)

#fit the model (poisson family)
#NOTE: This is a toy example, please run it longer in practice.
fit_M2Y1_result<-fit_M2Y1(death=death,expo=expo,n_iter=50,n.adapt=50,family="poisson")
head(fit_M2Y1_result)
```

`fit_M2Y2` *A function to fit the stochastic mortality model M2Y2.*

Description

Carry out Bayesian estimation of the stochastic mortality **model M2Y2**.

Usage

```
fit_M2Y2(
  death,
  expo,
  n_iter = 10000,
  family = "nb",
  n.chain = 1,
  thin = 1,
  n.adapt = 1000,
  forecast = FALSE,
  h = 5,
  quiet = FALSE
)
```

Arguments

`death`	death data that has been formatted through the function `preparedata_fn`.
`expo`	expo data that has been formatted through the function `preparedata_fn`.
`n_iter`	number of iterations to run. Default is `n_iter=10000`.

`family`	a string of characters that defines the family function associated with the mortality model. "poisson" would assume that deaths follow a Poisson distribution and use a log link; "binomial" would assume that deaths follow a Binomial distribution and a logit link; "nb" (default) would assume that deaths follow a Negative-Binomial distribution and a log link.
`n.chain`	number of parallel chains for the model.
`thin`	thinning interval for monitoring purposes.
`n.adapt`	the number of iterations for adaptation. See `?rjags::adapt` for details.
`forecast`	a logical value indicating if forecast is to be performed (default is `FALSE`). See below for details.
`h`	a numeric value giving the number of years to forecast. Default is `h=5`.
`quiet`	if TRUE then messages generated during compilation will be suppressed, as well as the progress bar during adaptation.

Details

The model can be described mathematically as follows: If `family="poisson"`, then

$$d_{x,t,p} \sim \text{Poisson}(E^c_{x,t,p} m_{x,t,p}),$$

$$\log(m_{x,t,p}) = a_x + b_x k_{t,p},$$

where $d_{x,t,p}$ represents the number of deaths at age x in year t of stratum p, while $E^c_{x,t,p}$ and $m_{x,t,p}$ represents respectively the corresponding central exposed to risk and central mortality rate at age x in year t of stratum p. Similarly, if `family="nb"`, then a negative binomial distribution is fitted, i.e.

$$d_{x,t,p} \sim \text{Negative-Binomial}(\phi, \frac{\phi}{\phi + E^c_{x,t,p} m_{x,t,p}}),$$

$$\log(m_{x,t,p}) = a_x + b_x k_{t,p},$$

where ϕ is the overdispersion parameter. See Wong et al. (2018). But if `family="binomial"`, then

$$d_{x,t,p} \sim \text{Binomial}(E^0_{x,t,p}, q_{x,t,p}),$$

$$\text{logit}(q_{x,t,p}) = a_x + b_x k_{t,p},$$

where $q_{x,t,p}$ represents the initial mortality rate at age x in year t of stratum p, while $E^0_{x,t,p} \approx E^c_{x,t,p} + \frac{1}{2} d_{x,t,p}$ is the corresponding initial exposed to risk. Constraints used are:

$$\sum_x b_x = 1, \sum_{t,p} k_{t,p} = 0.$$

If `forecast=TRUE`, then the following time series models are fitted on $k_{t,p}$ as follows:

$$k_{t,p} = \eta_1 + \eta_2 t + \rho(k_{t-1,p} - (\eta_1 + \eta_2(t-1))) + \epsilon_{t,p} \text{ for } p = 1, \ldots, P \text{ and } t = 1, \ldots, T,$$

where $\epsilon_{t,p} \sim N(0, \sigma_k^2)$ for $t = 1, \ldots, T$, while $\eta_1, \eta_2, \rho, \sigma_k^2$ are additional parameters to be estimated. Note that the forecasting models are inspired by Wong et al. (2023).

Value

A list with components:

`post_sample` An `mcmc.list` object containing the posterior samples generated.

`param` A vector of character strings describing the names of model parameters.

`death` The death data that was used.

`expo` The expo data that was used.

`family` The family function used.

`forecast` A logical value indicating if forecast has been performed.

`h` The forecast horizon used.

References

Jackie S. T. Wong, Jonathan J. Forster, and Peter W. F. Smith. (2018). Bayesian mortality forecasting with overdispersion, Insurance: Mathematics and Economics, Volume 2018, Issue 3, 206-221. doi:10.1016/j.insmatheco.2017.09.023

Jackie S. T. Wong, Jonathan J. Forster, and Peter W. F. Smith. (2023). Bayesian model comparison for mortality forecasting, Journal of the Royal Statistical Society Series C: Applied Statistics, Volume 72, Issue 3, 566–586. doi:10.1093/jrsssc/qlad021

Examples

```
#load and prepare mortality data
data("dxt_array_product");data("Ext_array_product")
death<-preparedata_fn(dxt_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)
expo<-preparedata_fn(Ext_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)

#fit the model (poisson family)
#NOTE: This is a toy example, please run it longer in practice.
fit_M2Y2_result<-fit_M2Y2(death=death,expo=expo,n_iter=50,n.adapt=50,family="poisson")
head(fit_M2Y2_result)
```

`fit_MLiLee` *A function to fit the stochastic mortality model MLiLee.*

Description

Carry out Bayesian estimation of the stochastic mortality **model MLiLee** (Li and Lee, 2005). Note that if the number of strata is one, results from this model are essentially the same as the Lee-Carter model, fit_LC().

Usage

```
fit_MLiLee(
  death,
  expo,
  n_iter = 10000,
  family = "nb",
  share_alpha = FALSE,
  n.chain = 1,
  thin = 1,
  n.adapt = 1000,
  forecast = FALSE,
  h = 5,
  quiet = FALSE
)
```

Arguments

`death`	death data that has been formatted through the function `preparedata_fn`.
`expo`	expo data that has been formatted through the function `preparedata_fn`.
`n_iter`	number of iterations to run. Default is `n_iter=10000`.
`family`	a string of characters that defines the family function associated with the mortality model. "poisson" would assume that deaths follow a Poisson distribution and use a log link; "binomial" would assume that deaths follow a Binomial distribution and a logit link; "nb" (default) would assume that deaths follow a Negative-Binomial distribution and a log link.
`share_alpha`	a logical value indicating if $a_{x,p}$ should be shared across all strata (see details below). Default is `FALSE`.
`n.chain`	number of parallel chains for the model.
`thin`	thinning interval for monitoring purposes.
`n.adapt`	the number of iterations for adaptation. See `?rjags::adapt` for details.
`forecast`	a logical value indicating if forecast is to be performed (default is `FALSE`). See below for details.
`h`	a numeric value giving the number of years to forecast. Default is `h=5`.
`quiet`	if TRUE then messages generated during compilation will be suppressed, as well as the progress bar during adaptation.

Details

The model can be described mathematically as follows: If `family="log"`, then

$$d_{x,t,p} \sim \text{Poisson}(E^c_{x,t,p} m_{x,t,p}),$$

$$\log(m_{x,t,p}) = a_{x,p} + b_{x,p} k_{t,p} + B_x K_t,$$

where $d_{x,t,p}$ represents the number of deaths at age x in year t of stratum p, while $E^c_{x,t,p}$ and $m_{x,t,p}$ represents respectively the corresponding central exposed to risk and central mortality rate at age x in year t of stratum p. Similarly, if `family="nb"`, then a negative binomial distribution is fitted, i.e.

$$d_{x,t,p} \sim \text{Negative-Binomial}(\phi, \frac{\phi}{\phi + E^c_{x,t,p} m_{x,t,p}}),$$

$$\log(m_{x,t,p}) = a_{x,p} + b_{x,p} k_{t,p} + B_x K_t,$$

where ϕ is the overdispersion parameter. See Wong et al. (2018). But if `family="binomial"`, then

$$d_{x,t,p} \sim \text{Binomial}(E^0_{x,t,p}, q_{x,t,p}),$$

$$\text{logit}(q_{x,t,p}) = a_{x,p} + b_{x,p} k_{t,p} + B_x K_t,$$

where $q_{x,t,p}$ represents the initial mortality rate at age x in year t of stratum p, while $E^0_{x,t,p} \approx E^c_{x,t,p} + \frac{1}{2} d_{x,t,p}$ is the corresponding initial exposed to risk. Constraints used are:

$$\sum_x b_x = 1, \sum_{t,p} k_{t,p} = 0.$$

If `share_alpha=TRUE`, then the additive age-specific parameter is the same across all strata p, i. e.

$$a_x + b_{x,p} k_{t,p} + B_x K_t.$$

If `forecast=TRUE`, then a time series model (an AR(1) with linear drift) will be fitted on both $k_{t,p}$ and K_t as follows:

$$k_{t,p} = \eta^k_1 + \eta^k_2 t + \rho_k (k_{t-1,p} - (\eta^k_1 + \eta^k_2 (t-1))) + \epsilon^k_{t,p} \text{ for } p = 1, \dots, P-1 \text{ and } t = 1, \dots, T,$$

and

$$K_t = \eta^K_1 + \eta^K_2 t + \rho^K (K_{t-1} - (\eta^K_1 + \eta^K_2 (t-1))) + \epsilon^K_t \text{ for } t = 1, \dots, T,$$

where $\epsilon^k_{t,p} \sim N(0, \sigma^2_k)$, $\epsilon^K_t \sim N(0, \sigma^2_K)$, while $\eta^k_1, \eta^k_2, \rho_k, \sigma^2_k, \eta^K_1, \eta^K_2, \rho_K, \sigma^2_K$ are additional parameters to be estimated. In principle, there are many other options for forecasting the mortality time trend. But currently, we assume that this serves as a general purpose forecasting model for simplicity.

Value

A list with components:

`post_sample` An `mcmc.list` object containing the posterior samples generated.

`param` A vector of character strings describing the names of model parameters.

`death` The death data that was used.

`expo` The expo data that was used.

`family` The family function used.

`forecast` A logical value indicating if forecast has been performed.

`h` The forecast horizon used.

References

Li N., & Lee R. (2005). Coherent mortality forecasts for a group of populations: an extension of the Lee-Carter method. Demography. 42(3):575-94. doi:10.1353/dem.2005.0021

Jackie S. T. Wong, Jonathan J. Forster, and Peter W. F. Smith. (2018). Bayesian mortality forecasting with overdispersion, Insurance: Mathematics and Economics, Volume 2018, Issue 3, 206-221. doi:10.1016/j.insmatheco.2017.09.023

Examples

```
#load and prepare mortality data
data("dxt_array_product");data("Ext_array_product")
death<-preparedata_fn(dxt_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)
expo<-preparedata_fn(Ext_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)

#fit the model (negative-binomial family)
#NOTE: This is a toy example, please run it longer in practice.
fit_MLiLee_result<-fit_MLiLee(death=death,expo=expo,n_iter=50,n.adapt=50)
head(fit_MLiLee_result)


#if sharing the alphas (poisson family)
fit_MLiLee_result2<-fit_MLiLee(death=death,expo=expo,n_iter=1000,family="poisson",share_alpha=TRUE)
head(fit_MLiLee_result2)

#if forecast (poisson family)
fit_MLiLee_result3<-fit_MLiLee(death=death,expo=expo,n_iter=1000,family="poisson",forecast=TRUE)
plot_rates_fn(fit_MLiLee_result3)
plot_param_fn(fit_MLiLee_result3)
```

`fit_PLAT` *A function to fit the stochastic mortality model "PLAT".*

Description

Carry out Bayesian estimation of the stochastic mortality model referred to as the **"PLAT"** model as proposed by Plat (2009).

Usage

```
fit_PLAT(
  death,
  expo,
  share_alpha = FALSE,
  share_gamma = FALSE,
  n_iter = 10000,
  family = "nb",
  n.chain = 1,
```

```
  thin = 1,
  n.adapt = 1000,
  forecast = FALSE,
  h = 5,
  quiet = FALSE
)
```

Arguments

`death`	death data that has been formatted through the function `preparedata_fn`.
`expo`	expo data that has been formatted through the function `preparedata_fn`.
`share_alpha`	a logical value indicating if $a_{x,p}$ should be shared across all strata (see details below). Default is `FALSE`.
`share_gamma`	a logical value indicating if the cohort parameter $\gamma_{c,p}$ should be shared across all strata (see details below). Default is `FALSE`.
`n_iter`	number of iterations to run. Default is `n_iter=10000`.
`family`	a string of characters that defines the family function associated with the mortality model. "poisson" would assume that deaths follow a Poisson distribution and use a log link; "binomial" would assume that deaths follow a Binomial distribution and a logit link; "nb" (default) would assume that deaths follow a Negative-Binomial distribution and a log link.
`n.chain`	number of parallel chains for the model.
`thin`	thinning interval for monitoring purposes.
`n.adapt`	the number of iterations for adaptation. See `?rjags::adapt` for details.
`forecast`	a logical value indicating if forecast is to be performed (default is `FALSE`). See below for details.
`h`	a numeric value giving the number of years to forecast. Default is `h=5`.
`quiet`	if TRUE then messages generated during compilation will be suppressed, as well as the progress bar during adaptation.

Details

The model can be described mathematically as follows: If `family="log"`, then

$$d_{x,t,p} \sim \text{Poisson}(E^c_{x,t,p} m_{x,t,p}),$$

$$\log(m_{x,t,p}) = a_{x,p} + k^1_{t,p} + k^2_{t,p}(\bar{x} - x) + k^3_{t,p}(\bar{x} - x)^+ + \gamma_{c,p},$$

where $c = t - x$ is the cohort index, $\bar{x}$ is the mean of x, $(\bar{x} - x)^+ = \max(\bar{x} - x, 0)$, $d_{x,t,p}$ represents the number of deaths at age x in year t of stratum p, while $E^c_{x,t,p}$ and $m_{x,t,p}$ represents respectively the corresponding central exposed to risk and central mortality rate at age x in year t of stratum p. Similarly, if `family="nb"`, then a negative binomial distribution is fitted, i.e.

$$d_{x,t,p} \sim \text{Negative-Binomial}(\phi, \frac{\phi}{\phi + E^c_{x,t,p} m_{x,t,p}}),$$

$$\log(m_{x,t,p}) = a_{x,p} + k^1_{t,p} + k^2_{t,p}(\bar{x} - x) + k^3_{t,p}(\bar{x} - x)^+ + \gamma_{c,p},$$

where ϕ is the overdispersion parameter. See Wong et al. (2018). But if `family="binomial"`, then

$$d_{x,t,p} \sim \text{Binomial}(E^0_{x,t,p}, q_{x,t,p}),$$

$$\text{logit}(q_{x,t,p}) = a_{x,p} + k^1_{t,p} + k^2_{t,p}(\bar{x} - x) + k^3_{t,p}(\bar{x} - x)^+ + \gamma_{c,p},$$

where $q_{x,t,p}$ represents the initial mortality rate at age x in year t of stratum p, while $E^0_{x,t,p} \approx E^c_{x,t,p} + \frac{1}{2}d_{x,t,p}$ is the corresponding initial exposed to risk. Constraints used are:

$$\sum_t k^1_{t,p} = \sum_t k^2_{t,p} = \sum_t k^3_{t,p} = 0, \sum_c \gamma_{c,p} = \sum_c c\gamma_{c,p} = \sum_c c^2\gamma_{c,p} = 0 \quad \text{for each stratum } p.$$

If `share_alpha=TRUE`, then the additive age-specific parameter is the same across all strata p, i. e.

$$a_x + k^1_{t,p} + k^2_{t,p}(\bar{x} - x) + k^3_{t,p}(\bar{x} - x)^+ + \gamma_{c,p}.$$

If `share_gamma=TRUE`, then the cohort parameter is the same across all strata p, i. e.

$$a_{x,p} + k^1_{t,p} + k^2_{t,p}(\bar{x} - x) + k^3_{t,p}(\bar{x} - x)^+ + \gamma_c.$$

If `forecast=TRUE`, then the following time series models are fitted on $k^1_{t,p}$ as follows:

$$k^1_{t,p} = \eta^1_1 + \eta^1_2 t + \rho(k^1_{t-1,p} - (\eta^1_1 + \eta^1_2(t-1))) + \epsilon^1_{t,p} \text{ for } p = 1, \ldots, P \text{ and } t = 1, \ldots, T,$$

where $\epsilon^1_{t,p} \sim N(0, (\sigma^1)^2_k)$ for $t = 1, \ldots, T$, while $\eta^1_1, \eta^1_2, \rho^1, (\sigma^1)^2_k$ are additional parameters to be estimated. Similarly for $k^2_{t,p}$ and $k^3_{t,p}$. Also,

$$\gamma_{c,p} = \gamma_{c-1,p} + \rho_\gamma(\gamma_{c-1,p} - \gamma_{c-2,p}) + \epsilon^\gamma_{c,p} \text{ for } p = 1, \ldots, P \text{ and } c = 3, \ldots, C,$$

$$\gamma_2 = \gamma_1 + \frac{1}{\sqrt{1 - \rho^2_\gamma}}\epsilon^\gamma_{2,p},$$

$$\gamma_1 = 100\epsilon^\gamma_{1,p}$$

where $\epsilon^\gamma_{c,p} \sim N(0, \sigma^2_\gamma)$ for $c = 1, \ldots, C$, while $\rho_\gamma, \sigma^2_\gamma$ are additional parameters to be estimated. Note that the forecasting models are inspired by Wong et al. (2023).

Value

A list with components:

`post_sample`	An `mcmc.list` object containing the posterior samples generated.
`param`	A vector of character strings describing the names of model parameters.
`death`	The death data that was used.
`expo`	The expo data that was used.
`family`	The family function used.
`forecast`	A logical value indicating if forecast has been performed.
`h`	The forecast horizon used.

References

Plat R. (2009). On Stochastic Mortality Modeling. Insurance: Mathematics and Economics, 45(3), 393–404. doi:10.1016/j.insmatheco.2009.08.006

Jackie S. T. Wong, Jonathan J. Forster, and Peter W. F. Smith. (2018). Bayesian mortality forecasting with overdispersion, Insurance: Mathematics and Economics, Volume 2018, Issue 3, 206-221. doi:10.1016/j.insmatheco.2017.09.023

Jackie S. T. Wong, Jonathan J. Forster, and Peter W. F. Smith. (2023). Bayesian model comparison for mortality forecasting, Journal of the Royal Statistical Society Series C: Applied Statistics, Volume 72, Issue 3, 566–586. doi:10.1093/jrsssc/qlad021

Examples

```
#load and prepare mortality data
data("dxt_array_product");data("Ext_array_product")
death<-preparedata_fn(dxt_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)
expo<-preparedata_fn(Ext_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)

#fit the model (negative-binomial family)
#NOTE: This is a toy example, please run it longer in practice.
fit_PLAT_result<-fit_PLAT(death=death,expo=expo,n_iter=50,n.adapt=50)
head(fit_PLAT_result)


#if sharing the cohorts only (poisson family)
fit_PLAT_result2<-fit_PLAT(death=death,expo=expo,n_iter=1000,family="poisson",share_gamma=TRUE)
head(fit_PLAT_result2)
```

`fit_RH` *A function to fit the stochastic mortality model by Renshaw and Haberman (2006).*

Description

Carry out Bayesian estimation of the stochastic mortality model, the **Renshaw-Haberman model** (Renshaw and Haberman, 2006). Note that this model is equivalent to "M2" under the formulation of Cairns et al. (2009).

Usage

```
fit_RH(
  death,
  expo,
  n_iter = 10000,
  family = "nb",
  share_alpha = FALSE,
  share_beta = FALSE,
```

```
  share_gamma = FALSE,
  n.chain = 1,
  thin = 1,
  n.adapt = 1000,
  forecast = FALSE,
  h = 5,
  quiet = FALSE
)
```

Arguments

`death`	death data that has been formatted through the function `preparedata_fn`.
`expo`	expo data that has been formatted through the function `preparedata_fn`.
`n_iter`	number of iterations to run. Default is `n_iter=10000`.
`family`	a string of characters that defines the family function associated with the mortality model. "poisson" would assume that deaths follow a Poisson distribution and use a log link; "binomial" would assume that deaths follow a Binomial distribution and a logit link; "nb" (default) would assume that deaths follow a Negative-Binomial distribution and a log link.
`share_alpha`	a logical value indicating if $a_{x,p}$ should be shared across all strata (see details below). Default is `FALSE`.
`share_beta`	a logical value indicating if $b_{x,p}$ should be shared across all strata (see details below). Default is `FALSE`.
`share_gamma`	a logical value indicating if the cohort parameter $\gamma_{x,p}$ should be shared across all strata (see details below). Default is `FALSE`.
`n.chain`	number of parallel chains for the model.
`thin`	thinning interval for monitoring purposes.
`n.adapt`	the number of iterations for adaptation. See `?rjags::adapt` for details.
`forecast`	a logical value indicating if forecast is to be performed (default is `FALSE`). See below for details.
`h`	a numeric value giving the number of years to forecast. Default is `h=5`.
`quiet`	if TRUE then messages generated during compilation will be suppressed, as well as the progress bar during adaptation.

Details

The model can be described mathematically as follows: If `family="poisson"`, then

$$d_{x,t,p} \sim \text{Poisson}(E^c_{x,t,p} m_{x,t,p}),$$

$$\log(m_{x,t,p}) = a_{x,p} + b_{x,p} k_{t,p} + \gamma_{c,p},$$

where $c = t - x$ is the cohort index, $d_{x,t,p}$ represents the number of deaths at age x in year t of stratum p, while $E^c_{x,t,p}$ and $m_{x,t,p}$ represents respectively the corresponding central exposed to risk and central mortality rate at age x in year t of stratum p. Similarly, if `family="nb"`, then a negative binomial distribution is fitted, i.e.

$$d_{x,t,p} \sim \text{Negative-Binomial}(\phi, \frac{\phi}{\phi + E^c_{x,t,p} m_{x,t,p}}),$$

$$\log(m_{x,t,p}) = a_{x,p} + b_{x,p} k_{t,p} + \gamma_{c,p},$$

where ϕ is the overdispersion parameter. See Wong et al. (2018). But if `family="binomial"`, then

$$d_{x,t,p} \sim \text{Binomial}(E^0_{x,t,p}, q_{x,t,p}),$$

$$\text{logit}(q_{x,t,p}) = a_{x,p} + b_{x,p} k_{t,p} + \gamma_{c,p},$$

where $q_{x,t,p}$ represents the initial mortality rate at age x in year t of stratum p, while $E^0_{x,t,p} \approx E^c_{x,t,p} + \frac{1}{2} d_{x,t,p}$ is the corresponding initial exposed to risk. Constraints used are:

$$\sum_x b_{x,p} = 1, \sum_t k_{t,p} = 0, \sum_c \gamma_{c,p} = \sum_c c\gamma_{c,p} = 0 \quad \text{for each stratum } p.$$

If `share_alpha=TRUE`, then the additive age-specific parameter is the same across all strata p, i. e.

$$a_x + b_{x,p} k_{t,p} + \gamma_{c,p}.$$

Similarly if `share_beta=TRUE`, then the multiplicative age-specific parameter is the same across all strata p, i. e.

$$a_{x,p} + b_x k_{t,p} + \gamma_{c,p}.$$

Similarly if `share_gamma=TRUE`, then the cohort parameter is the same across all strata p, i. e.

$$a_{x,p} + b_{x,p} k_{t,p} + \gamma_c.$$

If `forecast=TRUE`, then the following time series models are fitted on $k_{t,p}$ and $\gamma_{c,p}$ as follows:

$$k_{t,p} = \eta_1 + \eta_2 t + \rho(k_{t-1,p} - (\eta_1 + \eta_2(t-1))) + \epsilon_{t,p} \text{ for } p = 1, \ldots, P \text{ and } t = 1, \ldots, T,$$

and

$$\gamma_{c,p} = \gamma_{c-1,p} + \rho_\gamma(\gamma_{c-1,p} - \gamma_{c-2,p}) + \epsilon^\gamma_{c,p} \text{ for } p = 1, \ldots, P \text{ and } c = 3, \ldots, C,$$

$$\gamma_2 = \gamma_1 + \frac{1}{\sqrt{1-\rho^2_\gamma}} \epsilon^\gamma_{2,p},$$

$$\gamma_1 = 100\epsilon^\gamma_{1,p}$$

where $\epsilon_{t,p} \sim N(0, \sigma^2_k)$ for $t = 1, \ldots, T$, $\epsilon^\gamma_{c,p} \sim N(0, \sigma^2_\gamma)$ for $c = 1, \ldots, C$, while $\eta_1, \eta_2, \rho, \sigma^2_k, \rho_\gamma, \sigma^2_\gamma$ are additional parameters to be estimated. Note that the forecasting models are inspired by Wong et al. (2023).

Value

A list with components:

`post_sample`	An `mcmc.list` object containing the posterior samples generated.
`param`	A vector of character strings describing the names of model parameters.
`death`	The death data that was used.
`expo`	The expo data that was used.
`family`	The family function used.
`forecast`	A logical value indicating if forecast has been performed.
`h`	The forecast horizon used.

References

Cairns, A. J. G., Blake, D., Dowd, K., Coughlan, G. D., Epstein, D., Ong, A., and Balevich, I. (2009). A Quantitative Comparison of Stochastic Mortality Models Using Data From England and Wales and the United States. North American Actuarial Journal, 13(1), 1–35. doi:10.1080/10920277.2009.10597538

Renshaw, A. and S. Haberman (2006). A cohort-based extension to the Lee-Carter model for mortality reduction factors. Insurance: Mathematics and Economics, 38(3), 556-570. doi:10.1016/j.insmatheco.2005.12.001

Jackie S. T. Wong, Jonathan J. Forster, and Peter W. F. Smith. (2018). Bayesian mortality forecasting with overdispersion, Insurance: Mathematics and Economics, Volume 2018, Issue 3, 206-221. doi:10.1016/j.insmatheco.2017.09.023

Jackie S. T. Wong, Jonathan J. Forster, and Peter W. F. Smith. (2023). Bayesian model comparison for mortality forecasting, Journal of the Royal Statistical Society Series C: Applied Statistics, Volume 72, Issue 3, 566–586. doi:10.1093/jrsssc/qlad021

Examples

```
#load and prepare mortality data
data("dxt_array_product");data("Ext_array_product")
death<-preparedata_fn(dxt_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)
expo<-preparedata_fn(Ext_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)

#fit the model (negative-binomial family)
#NOTE: This is a toy example, please run it longer in practice.
fit_RH_result<-fit_RH(death=death,expo=expo,n_iter=50,n.adapt=50)
head(fit_RH_result)


#if sharing the cohorts only (poisson family)
fit_RH_result2<-fit_RH(death=death,expo=expo,n_iter=1000,family="poisson",share_gamma=TRUE)
head(fit_RH_result2)

#if sharing all alphas, betas, and cohorts (poisson family)
fit_RH_result3<-fit_RH(death=death,expo=expo,n_iter=1000,family="poisson",
share_alpha=TRUE,share_beta=TRUE,share_gamma=TRUE)
head(fit_RH_result3)

#if forecast (negative-binomial family)
fit_RH_result4<-fit_RH(death=death,expo=expo,n_iter=1000,forecast=TRUE)
plot_rates_fn(fit_RH_result4)
plot_param_fn(fit_RH_result4)
```

`plot_deaths_fn` *A function to plot the fitted and forecast number of deaths, accompanied by credible intervals, from posterior samples generated for stochastic mortality models*

Description

Plot the median fitted and forecast number of deaths, accompanied by credible intervals (user-specified level), using posterior samples stored in "fit_result" object.

Usage

```
plot_deaths_fn(
  result,
  expo_forecast = NULL,
  pred_int = 0.95,
  plot_type = "age",
  plot_ages = NULL,
  plot_years = NULL,
  legends = TRUE
)
```

Arguments

`result`	object of type either "fit_result" or "BayesMoFo".
`expo_forecast`	An optional 3-dimensional array (of dimensions $p \times A \times h$) containing exposure data for the forecast period. If not provided, the exposure data from the most recent year will be used for forecasting.
`pred_int`	A numeric value (between 0 and 1) specifying the credible level of uncertainty bands. Default is `pred_int=0.95` (95% intervals).
`plot_type`	A character string (`c("age","time")`) to indicate whether to plot by age (default) or by time/year.
`plot_ages`	A numeric vector specifying which range of ages to plot for visualisation. If not specified, use whatever ages that were used to fit the model (i.e. `fit_result$death$ages`). One panel will be constructed per age when `plot_type="time"`, with a maximum of nine panels. If exceeded, only the first nine ages will be plotted.
`plot_years`	A numeric vector specifying which range of years to plot for visualisation. If not specified, use whatever years that were used to fit the model (i.e. `fit_result$death$years`). One panel will be constructed per year when `plot_type="age"`, with a maximum of nine panels. If exceeded, only the first nine years will be plotted.
`legends`	A logical value to indicate if legends of the plots should be shown (default) or suppressed (e.g. to aid visibility).

Value

A plot illustrating the median fitted and forecast number of deaths, accompanied by credible intervals.

Examples

```
#load and prepare data
data("dxt_array_product");data("Ext_array_product")
death<-preparedata_fn(dxt_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)
```

```
expo<-preparedata_fn(Ext_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)

#fit any mortality model
runBayesMoFo_result<-runBayesMoFo(death=death,expo=expo,models="APCI",n_iter=1000,forecast=TRUE)

#default plot
plot_deaths_fn(runBayesMoFo_result)

#plot by age and changing pre-specified arguments
plot_deaths_fn(runBayesMoFo_result,pred_int=0.8,plot_ages=40:60,plot_years=c(2017,2020))

#plot by time/year
plot_deaths_fn(runBayesMoFo_result,plot_type="time",plot_ages=c(40,50,60))
```

`plot_param_fn` *A function to plot the fitted parameters of stochastic mortality models, accompanied by credible intervals*

Description

Plot the fitted parameters, accompanied by credible intervals (user-specified level), using posterior samples stored in "fit_result" object.

Usage

```
plot_param_fn(result, pred_int = 0.95, legends = TRUE)
```

Arguments

`result`	object of type either "fit_result" or "BayesMoFo".
`pred_int`	A numeric value (between 0 and 1) specifying the credible level of uncertainty bands. Default is `pred_int=0.95` (95% intervals).
`legends`	A logical value to indicate if legends of the plots should be shown (default) or suppressed (e.g. to aid visibility).

Value

A plot illustrating the median fitted and forecast parameters, accompanied by credible intervals.

Examples

```
#load and prepare data
data("dxt_array_product");data("Ext_array_product")
death<-preparedata_fn(dxt_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)
expo<-preparedata_fn(Ext_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)

#fit any mortality model
runBayesMoFo_result<-runBayesMoFo(death=death,expo=expo,n_iter=1000,models="APCI",
```

```
  family="poisson",forecast=TRUE)

#default plot
plot_param_fn(runBayesMoFo_result)

#with 80% credible intervals
plot_param_fn(runBayesMoFo_result,pred_int=0.8)
```

`plot_rates_fn` *A function to plot the fitted mortality rates, accompanied by credible intervals, from posterior samples generated for stochastic mortality models*

Description

Plot the fitted mortality rates, accompanied by credible intervals (user-specified level), using posterior samples stored in "fit_result" object.

Usage

```
plot_rates_fn(
  result,
  pred_int = 0.95,
  plot_type = "age",
  plot_ages = NULL,
  plot_years = NULL,
  legends = TRUE
)
```

Arguments

`result`	object of type either "fit_result" or "BayesMoFo".
`pred_int`	A numeric value (between 0 and 1) specifying the credible level of uncertainty bands. Default is `pred_int=0.95` (95% intervals).
`plot_type`	A character string (`c("age","time")`) to indicate whether to plot by age (default) or by time/year.
`plot_ages`	A numeric vector specifying which range of ages to plot for visualisation. If not specified, use whatever ages that were used to fit the model (i.e. `fit_result$death$ages`). One panel will be constructed per age when `plot_type="time"`, with a maximum of nine panels. If exceeded, only the first nine ages will be plotted.
`plot_years`	A numeric vector specifying which range of years to plot for visualisation. If not specified, use whatever years that were used to fit the model (i.e. `fit_result$death$years`). One panel will be constructed per year when `plot_type="age"`, with a maximum of nine panels. If exceeded, only the first nine years will be plotted.
`legends`	A logical value to indicate if legends of the plots should be shown (default) or suppressed (e.g. to aid visibility).

Value

A plot illustrating the median fitted and forecast mortality rates, accompanied by credible intervals.

Examples

```
#load and prepare data
data("dxt_array_product");data("Ext_array_product")
death<-preparedata_fn(dxt_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)
expo<-preparedata_fn(Ext_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)

#fit any mortality model
runBayesMoFo_result<-runBayesMoFo(death=death,expo=expo,models="APCI",n_iter=1000,forecast=TRUE)

#default plot
plot_rates_fn(runBayesMoFo_result)

#plot by age and changing pre-specified arguments
plot_rates_fn(runBayesMoFo_result,pred_int=0.8,plot_ages=40:60,plot_years=c(2017,2020))

#plot by time/year
plot_rates_fn(runBayesMoFo_result,plot_type="time",plot_ages=c(40,50,60))
```

`predict_deaths_fn` *A function to compute the fitted and forecast number of deaths, accompanied by credible intervals, from posterior samples generated for stochastic mortality models*

Description

Return the median fitted and forecast number of deaths, accompanied by credible intervals (user-specified level), using posterior samples stored in "fit_result" object.

Usage

```
predict_deaths_fn(result, expo_forecast = NULL, pred_int = 0.95)
```

Arguments

`result`	object of type either "fit_result" or "BayesMoFo".
`expo_forecast`	An optional 3-dimensional array (of dimensions $p \times A \times h$) containing exposure data for the forecast period. If not provided, the exposure data from the most recent year will be used for forecasting.
`pred_int`	A numeric value (between 0 and 1) specifying the credible level of uncertainty bands. Default is `pred_int=0.95` (95% intervals).

Value

An array containing the lower, median, and upper quantiles of the number of deaths for both the fitted and forecast periods.

Examples

```
#load and prepare data
data("dxt_array_product");data("Ext_array_product")
death<-preparedata_fn(dxt_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)
expo<-preparedata_fn(Ext_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)

#fit any mortality model
runBayesMoFo_result<-runBayesMoFo(death=death,expo=expo,models="APCI",n_iter=1000)

#default
predict_deaths_fn(runBayesMoFo_result)

#changing pre-specified arguments
predict_deaths_fn(runBayesMoFo_result,pred_int=0.8)
```

`preparedata_fn` *A function to prepare mortality data with stratification (e.g. by product, sex/gender, country, cause of death etc.)*

Description

Construct a 3-dimensional data array (dim 1: strata, dim 2: ages, dim 3: years) from raw data frames/arrays suitable for fitting various Bayesian Stochastic Mortality Models.

Usage

```
preparedata_fn(
  data_df_array,
  data_matrix = FALSE,
  strat_name = NULL,
  ages = NULL,
  years = NULL,
  round = TRUE
)
```

Arguments

`data_df_array`	data (normally death counts or exposures) to be provided either as a **data frame** or a **3-dimensional array**. If providing data frame, it should be structured as (column 1: ages, column 2: years, column 3: death/expo data, column 4: strata); if providing array, it should be as (dim 1: strata, dim 2: ages, dim 3: years). If only one stratum (AP data), either remove column 4 or ensure it contains only

	one stratum label. Alternatively, one can also provide a **2-dimensional matrix** (age×year) but please set `data_matrix=TRUE`.
`data_matrix`	a logical value indicating if a 2-dimensional matrix (age×year) is supplied.
`strat_name`	a vector of strings indicating names of each stratum.
`ages`	a numeric vector indicating which ages to use.
`years`	a numeric vector indicating which years to use.
`round`	a logical value indicating whether data entries should be rounded.

Value

A list with components:

`data` A 3-dimensional data array (dim 1: strata, dim 2: ages, dim 3: years).

`strat_name` A vector of strings describing the names of each stratum.

`ages` A numeric vector describing the ages used.

`years` A numeric vector describing the years used.

`n_strat` A numeric value describing the number of strata.

`n_ages` A numeric value describing the number of ages.

`n_years` A numeric value describing the number of years.

Examples

```
##################
##if p>1 (more than 1 strata)
##################

#Input: data.frame
data("data_summarised")
head(data_summarised)

#prepare death data
death<-preparedata_fn(data_summarised[,c("Age","Year","Claim","Product")],
strat_name = c("ACI","DB","SCI"),ages=35:65)
#prepare exposure data
expo<-preparedata_fn(data_summarised[,c("Age","Year","Exposure","Product")],
strat_name = c("ACI","DB","SCI"),ages=35:65)

#visualise data
str(death);str(expo)

#works also if data.frame only contains only 1 stratum
death<-preparedata_fn(data_summarised[,
c("Age","Year","Claim","Product")][data_summarised$Product=="ACI",],ages=35:65)

expo<-preparedata_fn(data_summarised[,
c("Age","Year","Exposure","Product")][data_summarised$Product=="ACI",],ages=35:65)

str(death);str(expo)
```

```
#Input: 3D data array
data("dxt_array_product");data("Ext_array_product")
death<-preparedata_fn(dxt_array_product,ages=35:65)
expo<-preparedata_fn(Ext_array_product,ages=35:65)

str(death);str(expo)

##################
##if p=1 (only 1 stratum)
##################

#specifying only one of the strats from the data.frame
death<-preparedata_fn(data_summarised[,c("Age","Year","Claim","Product")],
  strat_name = "ACI",ages=35:65)
expo<-preparedata_fn(data_summarised[,c("Age","Year","Exposure","Product")],
  strat_name = "ACI",ages=35:65)
str(death);str(expo)

#if data.frame only contains 1 strat (4 columns)
death<-preparedata_fn(data_summarised[,c("Age","Year","Claim","Product")]
  [data_summarised$Product=="ACI",],ages=35:65)
expo<-preparedata_fn(data_summarised[,c("Age","Year","Exposure","Product")]
  [data_summarised$Product=="ACI",],ages=35:65)
str(death);str(expo)

#if data.frame only contains 1 strat (3 columns)
death<-preparedata_fn(data_summarised[,c("Age","Year","Claim")]
  [data_summarised$Product=="ACI",],ages=35:65)
expo<-preparedata_fn(data_summarised[,c("Age","Year","Exposure")]
  [data_summarised$Product=="ACI",],ages=35:65)
str(death);str(expo)

#Input: 3D data array
death<-preparedata_fn(dxt_array_product,strat_name="ACI",ages=35:65)
expo<-preparedata_fn(Ext_array_product,strat_name="ACI",ages=35:65)
str(death);str(expo)

#Input: 2D matrix
death<-preparedata_fn(dxt_array_product["ACI",,],data_matrix=TRUE,ages=35:65)
expo<-preparedata_fn(Ext_array_product["ACI",,],data_matrix=TRUE,ages=35:65)
str(death);str(expo)
```

`quantile_fn` *A function to calculate sample quantiles*

Description

Compute quantiles from posterior samples.

Usage

```
quantile_fn(x, probs)
```

Arguments

`x`	a numeric vector.
`probs`	a numeric vector specifying which quantiles are to be computed.

Value

A numeric vector giving the samples quantiles.

Examples

```
#generate random samples
x<-rnorm(1000)

#compute the 5th, 50th, 95th percentiles
quantile_fn(x,probs=c(0.05,0.5,0.95))
```

`runBayesMoFo` *A function to fit mortality models.*

Description

Carry out Bayesian estimation a selection of stochastic mortality models considered in the paper. DIC (Spiegelhalter et al., 2002) is used to perform model selection in determining the best/worst model for the specified (stratified) mortality data.

Usage

```
runBayesMoFo(
  death,
  expo,
  models = NULL,
  family = "nb",
  forecast = FALSE,
  h = 5,
  n_iter = 1000,
  n.chain = 1,
  n.adapt = 1000,
  thin = 1,
  quiet = FALSE
)
```

Arguments

`death`	death data that has been formatted through the function `preparedata_fn`.
`expo`	expo data that has been formatted through the function `preparedata_fn`.
`models`	a vector of strings specifying the models to run. If not specified, a standard set of models is run. If we set `models="all"`, all the possible models considered will be run.
`family`	a string of characters that defines the family function associated with the mortality model. "poisson" would assume that deaths follow a Poisson distribution and use a log family; "binomial" would assume that deaths follow a Binomial distribution and a logit family; "nb" (default) would assume that deaths follow a Negative-Binomial distribution and a log family.
`forecast`	a logical value indicating if forecast is to be performed (default is `FALSE`). See below for details.
`h`	a numeric value giving the number of years to forecast. Default is `h=5`.
`n_iter`	number of iterations to run. Default is `n_iter=1000`.
`n.chain`	number of parallel chains for the model. Default is `n.chain=1`.
`n.adapt`	the number of iterations for adaptation. See `?rjags::adapt` for details.
`thin`	thinning interval for monitoring purposes.
`quiet`	if TRUE then messages generated during compilation will be suppressed, as well as the progress bar during adaptation.

Details

The standard set of models used (25 in total) is as follows:
`M1A`, `M1M`, `M1U`, `M2A1`, `M2A2`, `M2Y1`, `M2Y2`,
`MLiLee`, `MLiLee_sharealpha`,
`CBD_M1`,
`CBD_M2`, `CBD_M2_sharegamma`,
`CBD_M3`, `CBD_M3_sharegamma`,
`CBD_M5`,
`CBD_M6`, `CBD_M6_sharegamma`,
`CBD_M7`, `CBD_M7_sharegamma`,
`CBD_M8`, `CBD_M8_sharegamma`,
`APCI`, `APCI_sharegamma`,
`PLAT`, `PLAT_sharegamma`.

The full list of mortality models fitted (44 in total) is as follows:
`M1A`, `M1M`, `M1U`, `M2A1`, `M2A2`, `M2Y1`, `M2Y2`,
`MLiLee`, `MLiLee_sharealpha`,
`CBD_M1`, `CBD_M1_sharealpha`, `CBD_M1_sharebeta`, `CBD_M1_shareall`,
`CBD_M2`, `CBD_M2_sharealpha`, `CBD_M2_sharebeta`, `CBD_M2_sharegamma`, `CBD_M2_sharealpha_sharebeta`,
`CBD_M2_sharealpha_sharegamma`, `CBD_M2_sharebeta_sharegamma`, `CBD_M2_shareall`,
`CBD_M3`, `CBD_M3_sharealpha`, `CBD_M3_sharegamma`, `CBD_M3_shareall`,
`CBD_M5`,
`CBD_M6`, `CBD_M6_sharegamma`,
`CBD_M7`, `CBD_M7_sharegamma`,

`CBD_M8`, `CBD_M8_sharegamma`,
`APCI`, `APCI_sharealpha`, `APCI_sharebeta`, `APCI_sharegamma`, `APCI_sharealpha_sharebeta`,
`APCI_sharealpha_sharegamma`, `APCI_sharebeta_sharegamma`, `APCI_shareall`,
`PLAT`, `PLAT_sharealpha`, `PLAT_sharegamma`, `PLAT_shareall`.

Value

A list with components:

`result` A list containing 2 lists, respectively called "best" (`$result$best`) and "worst" (`$result$best`). Both return the similar output as `fit_result`, with the former giving those of the best model while the latter giving those of the worst model.

`DIC` A table containing the numeric values of the DIC of all mortality models fitted.

`best_model` A character string indicating the best model (lowest DIC).

`worst_model` A character string indicating the worst model (highest DIC). If only one model was specified, then this is the same as `best_model`.

`BayesMoFo_obj` A logical value indicating whether the result has been generated using the functionrunBayesMoFo (default=TRUE).

References

Spiegelhalter, David J., Best, Nicola G., Carlin, Bradley P., and van der Linde, Angelika. (2002). "Bayesian measures of model complexity and fit (with discussion)". Journal of the Royal Statistical Society, Series B. 64 (4): 583–639.doi:10.1111/14679868.00353

Examples

```
#load and prepare mortality data
data("dxt_array_product");data("Ext_array_product")
death<-preparedata_fn(dxt_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)
expo<-preparedata_fn(Ext_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)

##automatically find the best model from a standard set
runBayesMoFo_result<-runBayesMoFo(death=death,expo=expo,n_iter=50,n.adapt=50)
head(runBayesMoFo_result$result$best)
runBayesMoFo_result$DIC
runBayesMoFo_result$best_model

##if fit all the models
runBayesMoFo_result<-runBayesMoFo(death=death,expo=expo,models="all",
  n_iter=50,n.adapt=50)

# fit a subset of the models and forecast for 10 years
runBayesMoFo_result<-runBayesMoFo(death=death,expo=expo,models=c("APCI","LC","PLAT"),
  n_iter=1000,n.adapt=1000,n.chain=2,forecast=TRUE,h=10)

##plot the best model
plot_rates_fn(runBayesMoFo_result)
plot_rates_fn(runBayesMoFo_result,plot_type="time",plot_ages=c(40,50,60))
plot_param_fn(runBayesMoFo_result)
```

```
##convergence diagnostics plots

#trace and density plots of death rates
converge_diag_rates_fn(runBayesMoFo_result)

#trace and density plots of parameters
converge_diag_param_fn(runBayesMoFo_result)

#ACF plots of death rates
converge_diag_rates_fn(runBayesMoFo_result, trace = FALSE, density = FALSE, acf_plot = TRUE)

#ACF plots of parameters
converge_diag_param_fn(runBayesMoFo_result, trace = FALSE, density = FALSE, acf_plot = TRUE)

#Some MCMC diagnostics (Gelman, Geweke, Heidel diagnostics etc.)
converge_diag_fn(runBayesMoFo_result)
```

`summary_fn` *A function to summarise the fitted mortality rates and parameters of stochastic mortality models*

Description

Provide summaries (means, standard errors, percentiles) of the fitted mortality rates and parameters, derived using posterior samples stored in "fit_result" object.

Usage

```
summary_fn(result, pred_int = 0.95)
```

Arguments

`result`	object of type either "fit_result" or "BayesMoFo".
`pred_int`	A numeric value (between 0 and 1) specifying the credible level of uncertainty bands. Default is `pred_int=0.95` (95% intervals).

Value

A list with components: rates_summary=list(mean=rates_mean,std=rates_std),rates_pn=list(lower=rates_lower,median=rate

`rates_summary` A list containing 2 components, respectively called "mean" (`$rates_summary$mean`) and "std" (`$rates_summary$std`). Both return a 3-dimensional data array (dim 1: strata, dim 2: ages, dim 3: years), with the former giving posterior means of fitted mortality rates while the latter giving standard errors.

`rates_pn` A list containing 3 components, respectively called "lower" (`$rates_pn$lower`), "median" (`$rates_pn$median`), and "upper" (`$rates_pn$upper`). All return a 3-dimensional data array (dim 1: strata, dim 2: ages, dim 3: years), representing the respective percentiles for the fitted mortality rates.

`param_summary` A list containing 2 components, respectively called "mean" (`$param_summary$mean`) and "std" (`$param_summary$std`). Both return a 3-dimensional data array (dim 1: strata, dim 2: ages, dim 3: years), with the former giving posterior means of fitted parameters while the latter giving standard errors.

`param_pn` A 2-dimensional matrix containing percentiles of fitted parameters.

Examples

```
#load and prepare data
data("dxt_array_product");data("Ext_array_product")
death<-preparedata_fn(dxt_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)
expo<-preparedata_fn(Ext_array_product,strat_name = c("ACI","DB","SCI"),ages=35:65)

#fit any mortality model
runBayesMoFo_result<-runBayesMoFo(death=death,expo=expo,n_iter=1000,models="APCI")

#default summary
summary_runBayesMoFo<-summary_fn(runBayesMoFo_result)

#mean of fitted mortality rates
summary_runBayesMoFo$rates_summary$mean

#standard errors of fitted mortality rates
summary_runBayesMoFo$rates_summary$std

#97.5th percentile of fitted mortality rates
summary_runBayesMoFo$rates_pn$upper

#mean of fitted parameters
summary_runBayesMoFo$param_summary$mean

#standard errors of fitted parameters
summary_runBayesMoFo$param_summary$std

#97.5th percentile of fitted parameters
summary_runBayesMoFo$param_pn[,"upper"]
```

`uk_deathscausedata` *Sample mortality data stratified by cause of death*

Description

UK causes of deaths data from the Human Mortality Database

Usage

```
data("uk_deathscausedata")
```

Format

A data.frame with 1600 rows and 5 columns (col 1: Age, col 2: Year, col 3: Deaths, col 4: Exposures, col 5: Cause)/

Ages Numeric, ranging between 15 and 90.

Years Numeric. Years at claims, spanning years 2001-2020.

Deaths Numeric.

Exposures Numeric.

Cause Character. cause of deaths as coded on the HMD

Examples

```
##Load death data
data("uk_deathscausedata")
str(uk_deathscausedata)
head(uk_deathscausedata)
```

`uk_mortalitydata` *Sample mortality data*

Description

UK mortality data from the Human Mortality Database

Usage

```
data("uk_mortalitydata")
```

Format

A data.frame with 11100 rows and 4 columns (col 1: Age, col 2: Year, col 3: Deaths, col 4: Exposures)/

Ages Numeric, ranging between 0-110.

Years Numeric. Years at claims, spanning years 1922-2021.

Deaths Numeric.

Exposures Numeric.

Examples

```
##Load death data
data("uk_mortalitydata")
str(uk_mortalitydata)
head(uk_mortalitydata)
```

Index

∗ **APCI**
  fit_APCI, 13
∗ **APC**
  fit_CBD_M3, 16
∗ **CBD_M3**
  fit_CBD_M3, 16
∗ **CBD_M5**
  fit_CBD_M5, 19
∗ **CBD_M6**
  fit_CBD_M6, 21
∗ **CBD_M7**
  fit_CBD_M7, 24
∗ **CBD_M8**
  fit_CBD_M8, 27
∗ **DIC**
  DIC_fn, 9
  runBayesMoFo, 66
∗ **Lee-Carter**
  fit_LC, 30
∗ **Li-Lee**
  fit_MLiLee, 49
∗ **M1A**
  fit_M1A, 33
∗ **M1M**
  fit_M1M, 35
∗ **M1U**
  fit_M1U, 38
∗ **M2A1**
  fit_M2A1, 40
∗ **M2A2**
  fit_M2A2, 42
∗ **M2Y1**
  fit_M2Y1, 45
∗ **M2Y2**
  fit_M2Y2, 47
∗ **PLAT**
  fit_PLAT, 52
∗ **PSRF**
  converge_diag_fn, 3
∗ **Renshaw-Haberman**
  fit_RH, 55
∗ **autocorrelations**
  converge_diag_param_fn, 5
  converge_diag_rates_fn, 6
∗ **bayesian**
  converge_diag_fn, 3
  converge_diag_param_fn, 5
  converge_diag_rates_fn, 6
  DIC_fn, 9
  fit_APCI, 13
  fit_CBD_M3, 16
  fit_CBD_M5, 19
  fit_CBD_M6, 21
  fit_CBD_M7, 24
  fit_CBD_M8, 27
  fit_LC, 30
  fit_M1A, 33
  fit_M1M, 35
  fit_M1U, 38
  fit_M2A1, 40
  fit_M2A2, 42
  fit_M2Y1, 45
  fit_M2Y2, 47
  fit_MLiLee, 49
  fit_PLAT, 52
  fit_RH, 55
  runBayesMoFo, 66
∗ **convergence**
  converge_diag_fn, 3
∗ **credible intervals**
  plot_deaths_fn, 58
  plot_param_fn, 60
  plot_rates_fn, 61
  predict_deaths_fn, 62
  summary_fn, 69
∗ **data formatting**
  preparedata_fn, 63
∗ **data preprocessing**

preparedata_fn, 63
∗ datasets
data_summarised, 8
Ext_array_country, 10
Ext_array_product, 11
Ext_array_sex, 12
uk_deathscausedata, 70
uk_mortalitydata, 71
∗ death rates
converge_diag_rates_fn, 6
summary_fn, 69
∗ density plots
converge_diag_param_fn, 5
converge_diag_rates_fn, 6
∗ diagnostics
converge_diag_fn, 3
converge_diag_param_fn, 5
converge_diag_rates_fn, 6
∗ effective sample sizes
converge_diag_param_fn, 5
converge_diag_rates_fn, 6
∗ estimation
fit_APCI, 13
fit_CBD_M3, 16
fit_CBD_M5, 19
fit_CBD_M6, 21
fit_CBD_M7, 24
fit_CBD_M8, 27
fit_LC, 30
fit_M1A, 33
fit_M1M, 35
fit_M1U, 38
fit_M2A1, 40
fit_M2A2, 42
fit_M2Y1, 45
fit_M2Y2, 47
fit_MLiLee, 49
fit_PLAT, 52
fit_RH, 55
runBayesMoFo, 66
∗ fitted death rates
plot_rates_fn, 61
∗ fitted deaths
plot_deaths_fn, 58
predict_deaths_fn, 62
∗ fitted parameters
plot_param_fn, 60
∗ forecast death rates
plot_rates_fn, 61
∗ forecast deaths
plot_deaths_fn, 58
predict_deaths_fn, 62
∗ forecast parameters
plot_param_fn, 60
∗ forecasting
predict_deaths_fn, 62
∗ graphics
plot_deaths_fn, 58
plot_param_fn, 60
plot_rates_fn, 61
∗ models
fit_APCI, 13
fit_CBD_M3, 16
fit_CBD_M5, 19
fit_CBD_M6, 21
fit_CBD_M7, 24
fit_CBD_M8, 27
fit_LC, 30
fit_M1A, 33
fit_M1M, 35
fit_M1U, 38
fit_M2A1, 40
fit_M2A2, 42
fit_M2Y1, 45
fit_M2Y2, 47
fit_MLiLee, 49
fit_PLAT, 52
fit_RH, 55
predict_deaths_fn, 62
∗ model
DIC_fn, 9
runBayesMoFo, 66
∗ mortality data
preparedata_fn, 63
∗ package
BayesMoFo, 3
∗ parameter estimation
fit_APCI, 13
fit_CBD_M3, 16
fit_CBD_M5, 19
fit_CBD_M6, 21
fit_CBD_M7, 24
fit_CBD_M8, 27
fit_LC, 30
fit_M1A, 33
fit_M1M, 35

fit_M1U, 38
fit_M2A1, 40
fit_M2A2, 42
fit_M2Y1, 45
fit_M2Y2, 47
fit_MLiLee, 49
fit_PLAT, 52
fit_RH, 55
∗ **parameters**
converge_diag_param_fn, 5
summary_fn, 69
∗ **plots**
plot_deaths_fn, 58
plot_param_fn, 60
plot_rates_fn, 61
∗ **posterior samples**
converge_diag_param_fn, 5
converge_diag_rates_fn, 6
∗ **quantiles**
quantile_fn, 65
∗ **selection**
DIC_fn, 9
runBayesMoFo, 66
∗ **statistics**
summary_fn, 69
∗ **stochastic mortality models**
fit_APCI, 13
fit_CBD_M3, 16
fit_CBD_M5, 19
fit_CBD_M6, 21
fit_CBD_M7, 24
fit_CBD_M8, 27
fit_LC, 30
fit_M1A, 33
fit_M1M, 35
fit_M1U, 38
fit_M2A1, 40
fit_M2A2, 42
fit_M2Y1, 45
fit_M2Y2, 47
fit_MLiLee, 49
fit_PLAT, 52
fit_RH, 55
runBayesMoFo, 66
∗ **stratification**
preparedata_fn, 63
∗ **summary**
summary_fn, 69
∗ **trace plots**
converge_diag_param_fn, 5
converge_diag_rates_fn, 6
∗ **utilities**
preparedata_fn, 63
∗ **visualization**
converge_diag_fn, 3
converge_diag_param_fn, 5
converge_diag_rates_fn, 6
plot_deaths_fn, 58
plot_param_fn, 60
plot_rates_fn, 61

BayesMoFo, 3
BayesMoFo-package (BayesMoFo), 3

converge_diag_fn, 3
converge_diag_param_fn, 5
converge_diag_rates_fn, 6

data_summarised, 8
DIC_fn, 9
dxt_array_country (Ext_array_country), 10
dxt_array_product (Ext_array_product), 11
dxt_array_sex (Ext_array_sex), 12

Ext_array_country, 10
Ext_array_product, 11
Ext_array_sex, 12

fit_APCI, 13
fit_CBD_M3, 16
fit_CBD_M5, 19
fit_CBD_M6, 21
fit_CBD_M7, 24
fit_CBD_M8, 27
fit_LC, 30
fit_M1A, 33
fit_M1M, 35
fit_M1U, 38
fit_M2A1, 40
fit_M2A2, 42
fit_M2Y1, 45
fit_M2Y2, 47
fit_MLiLee, 49
fit_PLAT, 52
fit_RH, 55

plot_deaths_fn, 58
plot_param_fn, 60
plot_rates_fn, 61
predict_deaths_fn, 62
preparedata_fn, 63

quantile_fn, 65

runBayesMoFo, 66

summary_fn, 69

uk_deathscausedata, 70
uk_mortalitydata, 71