Journal of Statistical Software

July 2025, Volume 113, Issue 9. doi: 10.18637/jss.v113.i09

BayesMortalityPlus: A Package in R for Bayesian Mortality Modeling

Lucas M. F. Silva
Universidade Federal
do Rio de Janeiro

Luiz F. V. Figueiredo
Universidade Federal
do Rio de Janeiro

Viviana G. R. Lobo
Universidade Federal
do Rio de Janeiro

Thais C. O. Fonseca
Universidade Federal
do Rio de Janeiro

Mariane B. Alves
Universidade Federal
do Rio de Janeiro

Abstract

The **BayesMortalityPlus** package provides a framework for modeling and predicting mortality data. The package includes tools for the construction of life tables based on Heligman-Pollard laws, and also on dynamic linear smoothers. Flexibility is available in terms of modeling so that the response variable may be modeled as Poisson, binomial or Gaussian. If temporal data is available, the package provides a Bayesian implementation for the well-known Lee-Carter model that allows for estimation, projection of mortality over time, and assessment of uncertainty of any linear or nonlinear function of parameters such as life expectancy. Illustrations are considered to show the capability of the proposed package to model mortality data.

Keywords: Bayesian mortality graduation, Heligman-Pollard model, dynamic linear models, Bayesian Lee-Carter, R.

1. Introduction

Models used to characterize mortality data through the Bayesian paradigm have become more popular and called the attention of actuaries, statisticians, and other researchers in recent years. In the actuarial context, it is essential to understand the mortality behavior so that smoothed death probabilities over ages can be used for pricing life insurance and annuities. From a demographic point of view, this is an essential tool for understanding the changes of

patterns in a population. Thus, applying mathematical formulations, such as mortality laws, smoothing models, and improvement techniques is useful to understand the mortality curves of populations or portfolios. Statistical methodologies that consider Bayesian graduation have been more attractive since they allow for the incorporation of prior knowledge through the specification of prior distributions. Besides, graduation is particularly important at advanced ages, for which exposure numbers are small and data are sparse (see Dodd, Forster, Bijak, and Smith 2018).

Mortality graduation models have become more sophisticated over time. Kimeldorf and Jones (1967) propose the use of mortality smoothing and the constructions of life tables via Bayesian graduation, and Carlin (1992) proposes the use of Markov chain Monte Carlo (MCMC) techniques to fit mortality curves. Dellaportas, Smith, and Stavropoulos (2001) suggest estimating the Heligman-Pollard (HP) laws proposed by Heligman and Pollard (1980) using a non-linear logistic and log-normal model that accounts for uncertainty in model parameters. Njenga and Sherris (2020) use the Bayesian vector auto-regressive (BVAR) model for the parameters of the HP model by considering temporal evolution of parameters in the HP function. Dodd *et al.* (2018) and Hilton, Dodd, Forster, and Smith (2019) provide a methodology for mortality estimation based on generalized additive models (GAMs – see Wood 2006) at the youngest ages and use a simpler parametric model at older ages that depend on mortality laws well-established in the literature.

Packages and functions for fitting mortality curves have been available in R environment (R Core Team 2025) for several years, through the Comprehensive R Archive Network (CRAN). The **MortalityLaws** package exploits optimization methods for fitting a wide range of point estimates for mortality laws (Pascariu 2025). The **demography** package developed by Hyndman (2023) provides functions for demographic analysis, such as life table calculations, fertility rates, and functional data analysis of mortality rates. In the context of Bayesian computation, R packages have been proposed such as the **HPbayes** package that provides the eight parameters of the Heligman-Pollard mortality model using a Bayesian Melding procedure with importance sampling (Sharrow 2012). However, the **HPbayes** package is no longer available in the active CRAN repository.

The Heligman-Pollard law considers a specific mathematical function to model mortality rates, as a mixture of infant, young adult, and adult survival functions. However, other flexible smoothing approaches could be considered to model mortality, such as splines techniques (Currie, Durban, and Eilers 2004; Camarda, Eilers, and Gampe 2016; Camarda 2019; Tang, Dodd, and Forster 2021). In this context, Camarda (2012) proposes a package in R called **MortalitySmooth**, that provides a framework for smoothing count data assumed to be Poisson-distributed in both one- and two-dimensional settings through P-splines. In addition to the proposal of a function in **BayesMortalityPlus** to model mortality curves via the Heligman-Pollard law, in this article we propose a smoother based on dynamic linear models (DLM) (West and Harrison 1997) that is flexible as splines and has an interpretable parameter for controlling smoothness in the mortality graduation across ages. Dynamic linear or state-space models are a large class of models controlled by indexed parameters – the states – for which a stochastic evolution is assigned. Typical applications assume time-indexed states, in the context of time series modeling (see West and Harrison 1997; Petris, Petrone, and Campagnoli 2009). Our proposal is to consider the ages of individuals exposed to risk as indexers of the states, thus defining a stochastic evolution of mortality patterns along ages. Our proposal takes into account the association in mortality rates along ages and provides

graduated mortality curves with a tuning parameter used for calibrating the smoothness of the graduated mortality curve. Section 3 details our proposal of using state-space models to graduate age-indexed mortality.

A common interest in studies on mortality laws is the analysis of the evolution of these laws through time and the prediction of future mortality. Among several methods, the Lee-Carter model (Lee and Carter 1992) is a stochastic demographic model that considers temporal dependence in the data, whereas the Heligman-Pollard and the state-space models as described in Section 3 are designed for cross-sectional data. Lee and Carter (1992) was a pioneer work in the mortality modeling of a single population over time. The method is based on a factor model with a latent factor varying across time (a time-indexed state parameter). Several extensions have been proposed to the Lee-Carter model. Li, Lee, and Tuljapurkar (2004) present an extension of the Lee-Carter model which allows for mortality prediction when the time series is observed in unequal intervals of time. The paper discusses the effects of parameter estimation and prediction uncertainty when the data is limited. The package **demography**, previously mentioned, implements the original Lee-Carter model and other variants presented in Lee and Miller (2001), Booth, Maindonald, and Smith (2002), and Hyndman and Shahid Ullah (2007). The package **StMoMo** developed by Villegas, Kaishev, and Millossovich (2018) fits the Lee-Carter model amongst a handful of other mortality models via generalized nonlinear models, using the existent **gnm** R package (Turner and Firth 2023). From a Bayesian point of view, several papers have dealt with mortality modeling, such as Czado, Delwarde, and Denuit (2005) and Pedroza (2006). In this context, the package **StanMoMo** (Barigou and Goffard 2023) models a variety of popular stochastic models with the help of Stan software via **rstan** (Guo, Gabry, Goodrich, Johnson, Weber, and Badr 2025). However, it does require some degree of knowledge of Stan tools to perform mortality graduation. As we can see, several proposal packages for mortality modeling using R software are available. However, few proposals for mortality modeling are ready for use in other programming languages. The package **leecarter** (Chen 2023) available in Python (Van Rossum *et al.* 2011) fits the original Lee-Cartel model, whereas **LifeTables** in the Financial toolbox (Inc. 2023) in MATLAB language (The MathWorks Inc. 2021) fits the usual Heligman-Pollard model.

Although several packages to study mortality data are available, there are some issues that our proposed package **BayesMortalityPlus** seeks to solve. Firstly, our package provides a user-friendly interface, as well as instructions and simple examples for running each function available for mortality modeling and prediction. The package provides examples from the Human Mortality Database (HMD 2022), and allows the user to supply external data if desired. Moreover, we perform a full Bayesian inference procedure for several smoothing and prediction models: the HP laws following the specifications described in Dellaportas *et al.* (2001), the Lee-Carter model, as described by Pedroza (2006) and a dynamic linear model with latent states indexed by ages, proposed here for mortality graduation, with analytical solution for Bayesian information updating across ages, as detailed in West and Harrison (1997, Chapter 4). This approach makes it possible to simplify the modeling process for the user, as well as to provide estimation and credibility intervals with adequate uncertainty measurement, any model parameters, and nonlinear functions of those, such as probabilities of death, life expectancy, and an easy visualization through graphic tools. We also provide a specific function for closing life tables that is not available in any of the packages mentioned previously. It is based on Dodd *et al.* (2018), providing mortality probabilities for advanced adult ages, for which exposures are usually reduced.

In brief, we review the statistical framework underlying the **BayesMortalityPlus** package and show its ability to model mortality rates via the Heligman-Pollard laws as suggested by Dellaportas *et al.* (2001), the dynamic linear model, and the Lee-Carter model via the Bayesian framework. For the Heligman-Pollard and Lee-Carter models, obtaining analytical posterior distributions of the involved parameters is not feasible, and MCMC methods (see Gamerman and Lopes 2006) are adopted to estimate the mortality curve and to perform prediction. The package was coded in R and is available on CRAN `https://CRAN.R-project.org/package=BayesMortalityPlus`.

The remaining text is organized as follows. Section 2 presents the Bayesian Heligman-Pollard model, beginning with a brief review of the original Heligman-Pollard law and its properties. The Bayesian inferential and computational procedures based on Monte Carlo Markov chain techniques are addressed. Section 3 presents the Bayesian graduation via Dynamic Linear Model where we propose to model the mortality rates through age-indexed sates. Section 4 provides tools for the Bayesian graduation of cross-sectional mortality data through **BayesMortalityPlus** package. Section 4.1 shows an illustrative example modeling the mortality curve via the HP model. The functions available in the package are presented and applied to purposes such as computing life expectations, closing life tables using different methods, and producing plots. In Section 4.2, the same cross-sectional example is considered using dynamic linear models with age-varying latent components to smooth log-mortality curves. Closure of the mortality tables and computation of life expectations are also considered. Section 5 reviews the Bayesian Lee-Carter model and some functions supplied in the package are employed. Section 6 concludes the paper with a final discussion and some remarks.

2. Bayesian graduation via Heligman-Pollard model

The methods considered in the **BayesMortalityPlus** package assume that the process of mortality tables graduation is based on a probabilistic approach that allows the computation of point estimates for mortality rates and life expectations, as well as the measurement of the associated uncertainty. We consider the data (x, E_x, D_x) in a fixed period of time, where x denotes the age of the individuals and assumes an integer value, D_x denotes the number of deaths at age x and E_x denotes the total exposure of individuals aged x. The central mortality rate is defined as $m_x = D_x/E_x$ (Bowers 1986) and the probability of death is given by $q_x = 1 - \exp(-m_x)$.

An usual approach considered for mortality graduation uses the well-known Heligman-Pollard law (Heligman and Pollard 1980). The HP model is a parametric function that captures the main characteristics of mortality tables, specified in terms of parameters that aim to have a demographic interpretation of the ages' effect on the mortality rates. It is written as

$$\frac{q_x}{1-q_x} = A^{(x+B)^C} + D\exp\left[-E\left\{\log\left(\frac{x}{F}\right)\right\}^2\right] + GH^x. \tag{1}$$

Equation 1 provides a mathematical formulation that takes into account three terms, each representing a mortality component over the age domain as illustrated in Figure 1. The first term reflects the fall in mortality during the early childhood years through a rapidly declining exponential curve. The second term, similar to the log-normal curve, reflects accident mortality being called the *accident hump in the demographic literature.* The last term reflects

Term	Interpretation	Parameters
$A^{(x+B)^C}$	Infant mortality	A measures the level of the mortality. B is an age displacement for the mortality of an infant (age 1). textbf$C$ measures the decline of the mortality rate throughout childhood. The domain of these three components lies on the interval (0, 1)
$D\exp\left[-E\left\{\log\left(\frac{x}{F}\right)\right\}^2\right]$	Accident hump	D represents the severity, E represents the spread and F the location of the *accident hump*. These three parameters have the following domains: $D \in (0,1)$, $E \in (0,\infty)$ and $F \in (15,110)$.
GH^x	Advanced age mortality	G represents the base level of senescent mortality while H reflects the rate of increase of that mortality. Their respective domains are: $G \in (0,1)$ and $H \in (0,\infty)$.

Table 1: Description of the parameters of the Heligman Pollard model.

the near geometric rise in mortality experienced in advanced ages, through the Gompertz exponential formula as described in Heligman and Pollard (1980). Table 1 summarizes the interpretation of each term of the HP model. Several methods have been proposed to estimate the parameters in Equation 1. The first method suggested by Heligman and Pollard (1980) considers weighted least squares with weights $w_x = 1/\hat{q}_x^2$. This proposal could be problematic due to the over-parameterization of the model and numerical instabilities. Sharrow (2012) considers the Bayesian Melding with incremental mixture importance sampling techniques implemented in the **HPbayes** package. Dellaportas *et al.* (2001) suggest Bayesian inference using MCMC to estimate the parameters. For more details on MCMC algorithms see Gamerman and Lopes (2006).

Following the proposal based on Dellaportas *et al.* (2001), we assume that the death odds are modeled through the log-normal distribution and that all individuals of the same age die independently with the same probability and a constant parameter of variation σ^2 for all ages. Therefore, the model can be written as

$$\log\left(\frac{q_x}{1-q_x}\right) = \log\left(A^{(x+B)^C} + D\exp\left[-E\left\{\log\left(\frac{x}{F}\right)\right\}^2\right] + GH^x\right) + \varepsilon_x, \tag{2}$$

where $\varepsilon_x \sim N(0,\sigma^2)$ are independent for all age x. Equation 2 can be rewritten in a general form as $\log(y_x) = \log(f_x) + \varepsilon_x$, with f_x being a parametric function. Thus $\mathsf{E}(y_x) = f_x\exp(\sigma^2/2)$ and $\mathsf{VAR}(y_x) = [\exp(\sigma^2) - 1]\exp(\sigma^2)f_x^2$. Here the Markov chain Monte Carlo techniques require the updating of parameters that depend on the function f_x and the parameter σ^2. See Dellaportas *et al.* (2001) for a more detailed discussion.

Although the proposal in Dellaportas *et al.* (2001) considers modeling the odds via a log-normal distribution, several papers make other probabilistic assumptions about the mortality

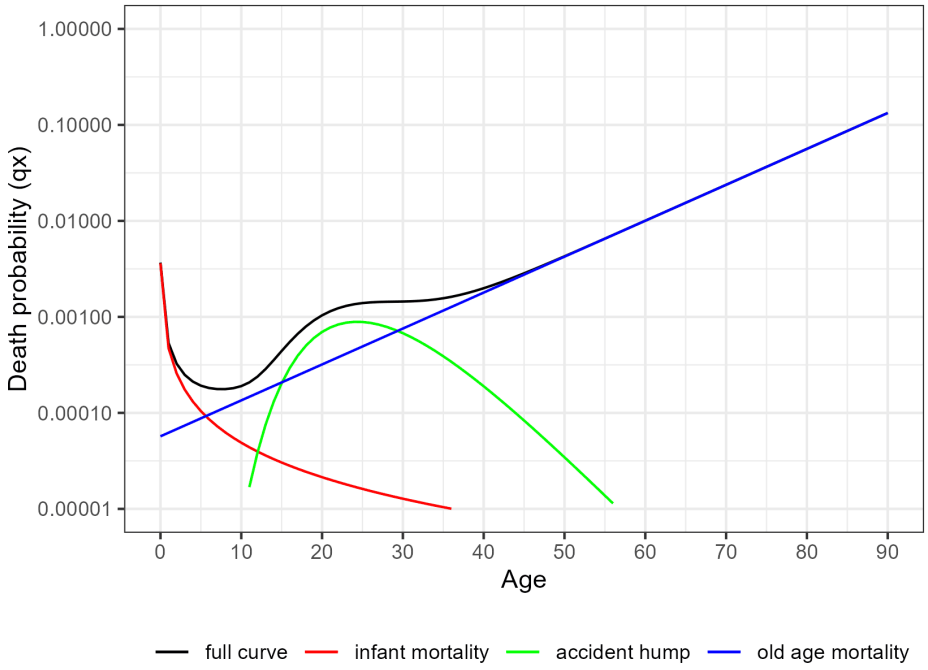


Figure 1: An illustration of the Heligman-Pollard curve: Progress with age of the probability of dying and its three components.

law (Czado *et al.* 2005; Renshaw, Haberman, and Hatzopoulos 1996; Li 2013). In particular, we are interested in allowing the exposure to be related to the model uncertainty since lower exposure is usually associated with higher variability in the data. Therefore we consider modeling the mortality via Poisson and binomial models as suggested by Dellaportas *et al.* (2001).

The binomial model assumes that the death counts at age x, D_x, follow a binomial distribution with the size parameter being the exposure in age x, E_x, with death probability at age x given by q_x, that is, $D_x \sim Binomial(E_x, q_x)$. On the other hand, the Poisson model considers that D_x represents the death counts for the age x following a Poisson distribution with rate $E_x \cdot q_x$ for each age, that is, $D_x \sim Poisson(E_x \cdot q_x)$. In this case, the exposure is an offset. For these two sampling distributions, we consider an alternative representation for the HP curve by adding an extra parameter as follows

$$q_x = A^{(x+B)^C} + D \exp\left[-E\left\{\log\left(\frac{x}{F}\right)\right\}^2\right] + \frac{GH^x}{1 + KGH^x},$$

where the parameter K is considered in order to allow for changes in the concavity of the curve at its final portion, resulting in a more flexible approach for capturing mortality trends at advanced ages. These alternative formulations have the advantage that the uncertainty relative to the mortality data changes according to the exposure at each age.

In **BayesMortalityPlus** package, the user can estimate the parameters of the HP curve through the function `hp` for the log-normal, ninomial, and Poisson models. **BayesMortalityPlus** can be installed with the code:

```
R> install.packages("BayesMortalityPlus")
```

The package is loaded within R as follows:

```
R> library("BayesMortalityPlus")
```

The function reproduces the inference procedure presented by Dellaportas *et al.* (2001) as follows

```
hp(x, Ex, Dx, model = c("binomial", "lognormal", "poisson"),
  M = NULL, bn = NULL, thin = 10, m = rep(NA, 8),
  v = rep(NA, 8), inits = NULL, K = NULL, sigma2 = NULL,
  prop.control = NULL, reduced_model = FALSE)
```

- The arguments `x`, `Ex`, and `Dx` represent the vector of the ages, exposures by age, and deaths by age, respectively.
- The argument `model` defines the mortality model chosen by the user. Setting `model = "poisson"` assumes that deaths follow the Poisson distribution, setting `model = "binomial"` assumes that deaths follow the binomial distribution, and setting `model = "lognormal"` assumes that the odds follow the log-normal distribution.
- The arguments `m` and `v` can be used to specify means and variances, respectively, for the prior distributions of each parameter, with `inits` specifying the initial values for the parameters in the algorithm. The `K` argument specifies the extra parameter K for the binomial and the Poisson models. `sigma2` is responsible for the initial value for the variance estimated for the log-normal distribution. Also, the argument `prop.control` tunes the acceptance rate of the MCMC algorithm. As default, the total number of iterations is `M = 50000`, the burn-in period is half of the total iterations, `bn = round(M/2)` and thinning given by `thin = 10`. Details on the specification of the MCMC algorithm can be seen in Gamerman and Lopes (2006).
- The argument `reduced_model` allows the user to fit a truncated version of the HP curve, which will be discussed in Section 4.1. For this specification, we set `M = 30000` iterations.

The package makes the posterior distribution samples available for the user to make inferences about any transformations of the parameters. Therefore, it is simple to obtain a Monte Carlo sample of the probability of death q_x for any age x, from which any desired quantiles can be obtained. Thus, the user is able to compute predictive intervals for q_x and survival probabilities ($p_x = 1 - q_x$), which can be used to quantify the life expectancy for any required age, with associated credibility intervals.

As previously seen, Heligman and Pollard (1980) propose variations to the basic mortality curve to address limitations of the standard formulation. The authors argue that the original equation becomes a straight line at older ages, and in cases where an accident hump is absent, the curve may exhibit curvature at older ages. To address this, they introduced some alternative expressions for the mortality curve, one of them incorporating the parameter K. Estimation of K can be challenging due to limited data at older ages, leading to the common practice of fixing it at sensible values or using alternative methods with age limitations (Congdon 1993). Despite these challenges, selecting appropriate K values is crucial for achieving

realistic closures of life tables. A higher value of K makes the curve more concave, indicating smaller additional increases in the logarithm of mortality per year of age. Negative K values imply convex curves with greater gains per year and no upper limit. In the **BayesMortalityPlus** package the argument `K` is available for choice by the user in the `hp()` function, otherwise, if `K = NULL`, the choice is determined through the maximum likelihood estimation.

For the binomial and the Poisson models, the HP formula provides estimates for the central mortality rate m_x. Then, under the assumption of uniform distribution for the deaths over an age interval x, we can compute the death probability q_x at age x through the usual relation $q_x = 1 - \exp(-m_x)$. For the log-normal model, these probabilities can be obtained through $q_x = \mu_x/(1 + \mu_x)$, where μ_x denotes the HP curve at age x. Finally, we obtain the point estimation for the death probabilities through the posterior median distribution of q_x.

In resume, the Heligman-Pollard model stands out as a comprehensive summary of various parametric mortality laws, offering the advantage of incorporating a component to represent young adult mortality, a notable feature in recent decades. With nine/eight parameters for three components, the HP model is both flexible and interpretable, capable of approximating diverse mortality patterns in human populations. Alternative methods for mortality graduation, such as non-parametric smoothing through penalized splines, generalized additive models, and dynamic linear models (Hyndman and Shahid Ullah 2007; Currie *et al.* 2004; Hilton *et al.* 2019), often compromise parameter estimates with demographic interpretations. If the goal is to interpret population patterns based on these parameters, the HP model is recommended; otherwise, alternatives like dynamic linear models (see Section 3) provide smooth adaptability to data.

3. Bayesian graduation via dynamic linear model

In this section we propose the use of dynamic linear models based on age-indexed states, aiming to obtain smoothed mortality curves for cross-sectional data. Traditionally, dynamic linear models (DLM) (West and Harrison 1997) are applied in time series analysis, addressing intrinsically auto-correlated observations gathered through time. Such models are specified hierarchically. At the first hierarchical level, an observational equation describes the probabilistic model assigned to the response variable, with dynamic mean response given by a linear predictor governed by states which are latent parameters indexed by the observational unit. At the second hierarchical level, a system equation assigns a stochastic Markovian evolution to the latent states across observational units. Polynomial trend models (West and Harrison 1997; Petris *et al.* 2009) are a particular case of this structure which is flexible enough to produce smoothed fits across observational units, capturing linear or non-linear trend patterns, with levels, slopes and concavities that can vary throughout the fitting interval, across observational units.

The use of DLMs in mortality studies is common practice in modeling temporal dependence in mortality patterns and predicting future mortality. For instance, the well-known Lee-Carter model (Lee and Carter 1992) considers a dynamical model to estimate temporal improvement for each age and can be used for predicting mortality in future years. Other proposals are Li *et al.* (2004) and Neves and Migon (2007).

Although DLMs are usually applied in the context of time-indexed observations, a point to be highlighted in our formulation is that the dynamic parameters will be indexed by ages x,

since our aim, in the present section, is to obtain smooth graduated curves through ages, for a fixed time period. We propose the use of a second-order polynomial trend model (see Petris *et al.* 2009, Section 3.2.1) to produce graduated mortality tables, imposing smoothness of the estimated mortality curve in the transition between ages. As will be seen in the applications presented in Section 4, the proposed model is able to accommodate non-linear patterns that are typical in log-mortality curves, across ages. Let D_x and E_x, respectively, denote the death counts and exposure at age x and define $Y_x = \log\left(\frac{D_x}{E_x}\right)$. We consider the following model:

$$\begin{aligned} Y_x &= \mu_x + v_x, \quad v_x \sim N(0, V) \\ \mu_x &= \mu_{x-1} + \beta_{x-1} + w_{x,1}, \quad w_{x,1} \sim N(0, \sigma^2_{\mu,x}) \qquad (3) \\ \beta_x &= \beta_{x-1} + w_{x,2}, \quad w_{x,2} \sim N(0, \sigma^2_{\beta,x}), \qquad (4) \end{aligned}$$

with random errors v_x, $w_{x,1}$ and $w_{x,2}$ assumed mutually and sequentially independent. The state μ_x denotes the dynamic level of the log-mortality, with stochastic evolution guided by Equation 3, and β_x denotes the local slope of the log-mortality curve at age x, allowing for different gradients through ages, since β_x evolves according to the random walk described in Equation 4. The evolution structures in Equations 3 and 4 formally accommodate the autocorrelation inherent to mortality patterns across ages.

The model may be rewritten in the general DLM form as

$$\begin{aligned} Y_x &= \boldsymbol{F}_x\boldsymbol{\theta}_x + v_x, \quad v_x \sim N(0, V) \\ \boldsymbol{\theta}_x &= \boldsymbol{G}_x\boldsymbol{\theta}_{x-1} + \boldsymbol{w}_x, \quad \boldsymbol{w}_x \sim N_2(\boldsymbol{0}, \boldsymbol{W}_x), \qquad (5) \end{aligned}$$

where $N_2(\cdot, \cdot)$ denotes a bivariate Gaussian density and

$$\boldsymbol{\theta}_x = \begin{bmatrix} \mu_x \\ \beta_x \end{bmatrix}, \quad \boldsymbol{G}_x = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad \boldsymbol{W}_x = \begin{bmatrix} \sigma^2_{\mu,x} & 0 \\ 0 & \sigma^2_{\beta,x} \end{bmatrix}, \quad \boldsymbol{F}_x = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad x = 1, 2, \ldots.$$

Let I_0 denote all the information available to the analyst, external to the observed data. Following a Bayesian approach, the model is completed with subject information about the latent states at age 0, which can be vague and expressed as $\boldsymbol{\theta}_0|I_0 \sim N_2(\boldsymbol{m}_0, \boldsymbol{C}_0)$, with $\boldsymbol{m}_0, \boldsymbol{C}_0$ chosen by the analyst. For instance, $\boldsymbol{C}_0$ can be specified as a diagonal matrix with large variances indicating no prior knowledge about the latent states (μ_0, β_0). It is clear from Equations 3, 4 and 5 that the smoothness of the graduated mortality curves strongly depends on the magnitude of the evolutional errors' variances, $\sigma^2_{\mu,x}$ and $\sigma^2_{\beta,x}$, which define the main diagonal of $\boldsymbol{W}_x$. The estimation of evolution variances introduces computational challenges for updating inference about the model. Thus, $\boldsymbol{W}_x$ is specified in the **BayesMortalityPlus** package via discounting strategies, as discussed in West and Harrison (1997, Chapter 6). Assume that the available information up to age $x-1$ is $I_{x-1} = \{I_0, Y_1, \ldots, Y_{x-1}\}$ and $I_x = \{I_{x-1}, I_x\}$. As detailed in West and Harrison (1997, Chapter 4), for each age $x-1$, the posterior covariance $\boldsymbol{C}_{x-1} = \mathsf{COV}[\boldsymbol{\theta}_{x-1}|I_{x-1}]$ is obtained via sequential recurrence relationships and the Bayes theorem. The prior covariance for age x is given by: $\boldsymbol{R}_x = \boldsymbol{P}_x + \boldsymbol{W}_x$, where $\boldsymbol{P}_x = \boldsymbol{G}_x\boldsymbol{C}_{x-1}\boldsymbol{G}'_x = \mathsf{COV}[\boldsymbol{G}_x\boldsymbol{\theta}_{x-1}|I_{x-1}]$. Therefore, $\boldsymbol{W}_x$ is an inflation factor added to $\boldsymbol{P}_x$. That is, $\boldsymbol{W}_x$ is the added uncertainty when transitioning from the posterior distribution on $\boldsymbol{\theta}$ with respect to information up to age $x-1$ to the prior distribution with respect to age x. Alternatively, uncertainty could be inflated by taking $\boldsymbol{R}_x = \boldsymbol{P}_x/\delta_x$, where $0 < \delta_x \leq 1$ is a discount factor. These two forms of uncertainty inflation become equivalent if

$$\boldsymbol{W}_x = \frac{1-\delta_x}{\delta_x}\boldsymbol{P}_x. \qquad (6)$$

The specification of δ_x is a choice of the user. As can be easily observed in Equation 6, $\delta_x \approx 1$ implies evolution variances close to zero, thus resulting in smoother trajectories for the level and slope of the log-mortality curve. Values of δ_x further from 1 yield fits that are more adaptive to the observed data. In the extreme case where $\delta_x = 1$, the evolution variances $\sigma^2_{\mu,x}$ and $\sigma^2_{\beta,x}$ are zero, implying that the evolution errors are null with probability 1. This results in a linear model with constant intercept and slope for all ages x. It is possible to perform sensitivity analysis regarding the specification of the discount factor, for instance by fitting a model for a grid of $\delta_x = \delta$ values and using model selection metrics to define the discount to be chosen. It is also possible to adopt different discount factors for different regions of the ages domain, shifting δ_x away from 1 for age intervals where the log-mortality rates show greater variation in level and slope, and fixing values close or equal to 1 for age intervals where death rates exhibit more stable linear patterns. For instance, a user could specify discounts δ_x deviating from 1 for younger ages where mortality laws can exhibit varying slopes and concavities along ages, a discount $\delta_x \approx 1$ for smooth graduation on intermediate ages and $\delta_x = 1$ for advanced ages, resulting in a straight line fit for this region of the mortality rate, where it can be reasonable to assume monotonically increasing log-mortality.

The **BayesMortalityPlus** package accommodates the possibility of specifying different discount factors for various age intervals, thereby producing more flexible fitted curves in age ranges where log-mortality exhibits greater variation in level and slope, and tighter fits, approaching a straight line (or reducing to a straight line in the extreme case where $\delta_x = 1$), for age ranges where it is believed that the underlying mortality law behaves approximately linearly and the adjusted log-mortality curve should not be influenced by fluctuations in observed log mortality rates.

Details on general forms of the DLMs and especially on the particular case of polynomial trend models, adopted here, are found in West and Harrison (1997, Chaps 4, 7) and Petris *et al.* (2009, Section 3.2.1). It is worth noting that a simpler first-order polynomial model, that is, a model with only a dynamic level μ_x could be able to capture several dynamic mortality patterns over a range of ages, but the resulting point predictive function for the following h ages, extrapolating the fitting range, would be a constant function of h. The use of the additional age-varying slope β_x results in a DLM that generates a predictive curve for h future ages given by a straight line with slope given by the gradient of the log-mortality curve in the final portion of the ages fitting interval, thus capturing the potentially increasing mortality risk for advanced ages. The concern with the form of the predictive function associated with the adopted model is justified by the fact that, for advanced ages, it is usual that the databases present a shortage of exposure. Therefore the mortality tables are typically adjusted using information up to a certain age x_* and from that point on, extrapolations are necessary. When using the DLM structure in **BayesMortalityPlus**, the predictive function of the second-order polynomial DLM is used in the extrapolation process of the mortality curve.

The user can estimate the parameters of the DLM through the function `dlm`. The function implements the inference procedure based on West and Harrison (1997) as follows

```
dlm(y, Ft = matrix(c(1, 0), nrow = 1), Gt = matrix(c(1, 0, 1, 1), 2),
  delta = 0.85,
  prior = list(m0 = rep(0, nrow(Gt)), C0 = diag(100, nrow(Gt))),
  prior.sig2 = list(a = 0.01, b = 0.01), M = 2000,
  ages = 0:(length(y) - 1))
```

- `y` represents the vector of log mortality rates.
- The arguments `Ft`, `Gt`, and `delta` represent the structural elements for the specification of the observational and system equations, and the discount factor (default = 0.85, for all ages), respectively. The discount factor can vary in the age domain, so the user can specify a single value for all ages, as it is in default, or specify a vector with a discount factor value for each age.
- The arguments `prior` and `prior.sig2` can be used to specify prior information, both as a `list` object. Argument `prior` receives the prior mean vector and covariance matrix and `prior.sig2` receives the prior parameters of the Inverse Gamma distribution for the estimated variance of the process.
- The argument `ages` allows the user to define the vector of ages associated with `y` in case the age interval does not equal the default graduation `0:(length(y) - 1)`.

Assuming that $\phi = V^{-1}$ follows a Gamma prior distribution, and that the states (μ_x, β_x), conditionally on ϕ, follow a bivariate normal prior distribution, both the posterior distribution of the model parameters and the predictive distribution for y_x are obtained analytically, as described in West and Harrison (Chapter 4 1997), making it trivial to generate samples from these distributions, which are used in **BayesMortalityPlus** to calculate point and interval estimates of log-mortality y_x and any nonlinear functions of these quantities. Thus, inference about the probability of death q_x at age x are conducted through the relationship $q_x = 1 - \exp(-\exp(y_x))$, applied to points from the predictive sample of y_x.

4. Static graduation with BayesMortalityPlus

In this section, we present the functions available in **BayesMortalityPlus** that can be used in the construction of life tables based on the Heligman-Pollard law and the dynamic linear models, respectively, as described in Section 2 and Section 3. The main functions for smoothing are `hp` and `dlm`. All analyses presented in the manuscript were performed on a Windows (64-bit) quad-core 2.4 Ghz, 8GB RAM, using R version 4.3.1 and Rstudio version 2023.09.0+463. Slight differences in the results may occur due to the use of different system settings. We also explore the posterior summaries and methodologies for extrapolation. Data from the United States and Portugal, which are extracted from the Human Mortality Database (HMD 2022), are contained in the object `data`, stratified by sex (as well as total population).

In the following, we present the Bayesian graduation by selecting the total population from the United States over the past forty years. We estimate the mortality curves for the specific years 1980, 1990, 2000, 2010, 2019:

```
R> data("USA")
R> library("dplyr")
```

We load the **dplyr** package to extract information from the database in a simple way through the `select`, `filter` and `mutate` commands (see more details in Wickham, François, Henry, Müller, and Vaughan 2023). Notice that other ways to manipulate the data could be applied. In this example, the vector `[1:81]` means that the ages $x = 0, \ldots, 80$ are selected, so that

the exposures (E_x) and death counts (D_x) for the years considered in the study are specified and filtered up to 80 years old for model fitting.

```
R> data <- select(USA, Year, Age, Ex.Total, Dx.Total) %>%
+    filter(Year %in% c(1980, 1990, 2000, 2010, 2019) & Age %in% 0:80) %>%
+    mutate(mx = Dx.Total / Ex.Total, Age = as.numeric(Age))
```

We also provide code to plot the raw mortality rates ($q_x = D_x/E_x$) over years via function `ggplot` available from package **ggplot2** (Wickham 2016). The `comma` function from **scales** package (Wickham, Pedersen, and Seidel 2025) is used for the labels on the y-axis.

```
R> library("ggplot2")
R> x.labs <- c("USA 1980", "USA 1990", "USA 2000", "USA 2010", "USA 2019")
R> ggplot(data) +
+    scale_y_continuous(trans = "log10", breaks = 10^-seq(0, 5),
+      limits = 10^-c(5, 0), labels = scales::comma) +
+    scale_x_continuous(breaks = seq(0, 100, by = 10)) +
+    theme_bw() + theme(legend.position = "bottom") +
+    labs(x = "Age", y = "Raw Mortality Rate", title = NULL) +
+    geom_point(aes(x = Age, y = mx, col = as.factor(Year))) +
+    scale_color_manual(name = NULL, values = c(rainbow(5)), label = x.labs)
```

4.1. Heligman-Pollard model

The function `hp` returns an object of class 'HP', which is an HP curve fit to the input data settled by the user. In this illustration, we consider vague or non-informative prior distributions as well as a seed to guarantee some level of reproducibility, although different system specifications may cause slight differences in parameter values and computational time. Notice that the user could provide their own prior information if desired. The MCMC scheme is the default one. The HP model under a log-normal setting for the respective years can be defined using the following code:

```
R> set.seed(111)
R> fit_1980 <- hp(0:80, data$Ex.Total[data$Year == 1980],
+    data$Dx.Total[data$Year == 1980], model = "lognormal")

Simulating [==================================] 100% in 42s

R> set.seed(112)
R> fit_1990 <- hp(0:80, data$Ex.Total[data$Year == 1990],
+    data$Dx.Total[data$Year == 1990], model = "lognormal")

Simulating [==================================] 100% in 47s

R> set.seed(113)
R> fit_2000 <- hp(0:80, data$Ex.Total[data$Year == 2000],
+    data$Dx.Total[data$Year == 2000], model = "lognormal")

Simulating [==================================] 100% in 43s
```

```
R> set.seed(114)
R> fit_2010 <- hp(0:80, data$Ex.Total[data$Year == 2010],
+    data$Dx.Total[data$Year == 2010], model = "lognormal")

Simulating [==================================] 100% in 41s

R> set.seed(115)
R> fit_2019 <- hp(0:80, data$Ex.Total[data$Year == 2019],
+    data$Dx.Total[data$Year == 2019], model = "lognormal")

Simulating [==================================] 100% in 36s
```

The `summary` function in R provides a summary table with the estimation of the parameters and the acceptance rate of the MCMC algorithm. As an example, the posterior summary for the 1980-year fit is available using the code:

```
R> summary(fit_1980)

       mean       sd      2.5%     50.0%     97.5% Accept %
A  0.001026 0.000053  0.000927  0.001023  0.001135     22.1
B  0.026546 0.006079  0.015898  0.025967  0.039927     22.1
C  0.125732 0.005171  0.115856  0.125582  0.135841     22.1
D  0.000922 0.000033  0.000858  0.000922  0.000991     22.1
E 11.474677 0.690612 10.223645 11.449133 12.900590     22.1
F 21.091245 0.157078 20.785356 21.084695 21.403583     22.1
G  0.000074 0.000003  0.000068  0.000074  0.000079     22.1
H  1.090758 0.000688  1.089471  1.090745  1.092192     22.1
```

The `fitted` function can provide a summary with the point estimate of death probabilities generated by the model for specific ages, as well as their predictive credible intervals via the composition sampling technique (see Chapter 5, Banerjee, Carlin, and Gelfand 2004):

```
fitted(fit, age = NULL, Ex = NULL, prob = 0.95)
```

The argument `fit` represents a fitted curve by the `hp` function via **BayesMortalityPlus** package, the argument `age` represents the range age considered for the death probabilities estimation, the parameter `Ex` represents a vector of the exposures that are considered for the binomial and the Poisson modeling. By default, the arguments `age` and `Ex` are the same considered in the fitting function. Notice that if we take into account a different age used before, the exposure `Ex` for that age should be set by the user. See that for the log-normal model this specification is not available. Additionally, the user can specify the probability of the predictive credible interval through the argument `prob`.

For illustration, consider data from the year 1980 and setting `age = c(0, 20, 40, 60, 80)`:

```
R> set.seed(116)
R> fitted(fit_1980, age = c(0, 20, 40, 60, 80))
```

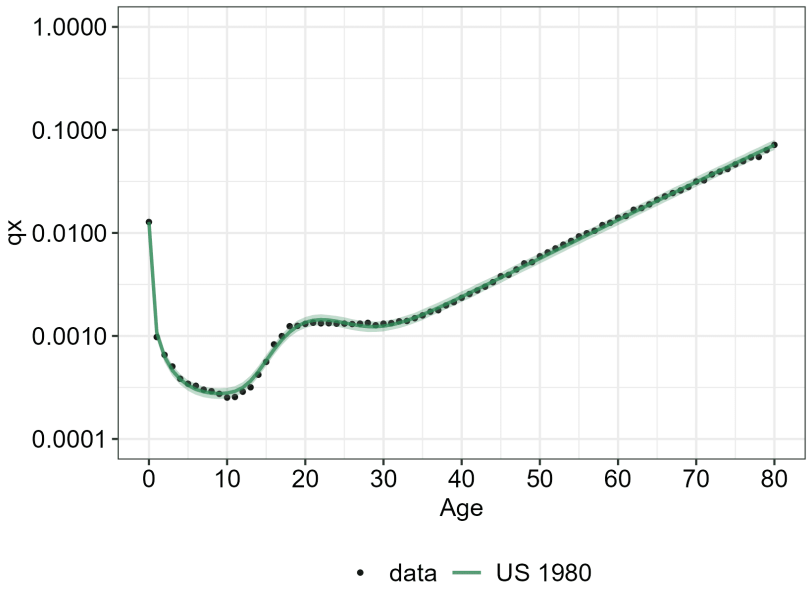


Figure 2: Posterior summaries via HP: Median mortality curve and 95% predictive credible interval in log-scale. The United States, total population, ages 0-80 and year 1980. The black dots represent the raw mortality rates.

```
  age   qx.fitted    qx.lower    qx.upper
1   0 0.012793731 0.010895801 0.014977969
2  20 0.001352868 0.001204037 0.001524133
3  40 0.002400773 0.002146262 0.002685540
4  60 0.013351369 0.011964270 0.014897479
5  80 0.071342288 0.063927059 0.079605888
```

where `qx.lower` and `qx.upper` represent the lower limit and upper limit of the predictive credible intervals, respectively. Figure 2 presents the fitted death probabilities with the 95% predictive credible interval. Figure 2 was obtained with the code:

```
R> plot(fit_1980, labels = "US 1980", plotIC = TRUE, plotData = TRUE)
```

Note that the log-normal model assumes constant error variance across ages, posing challenges in scenarios of lower exposure at risk for certain ages. This leads to less precise uncertainty in the fitted curve and wider credible intervals. In such contexts, considering binomial/Poisson models could be more appropriate. Figure 3 illustrates the differences between the credible intervals of the log-normal and the Poisson models fitted to the example data, considering a 99% credible interval and zooming in the 0-40 age interval. The log-normal model presents a wider credible interval overall, while the Poisson model takes into account the lower volume of data present in early ages. However, it should be noted that the point estimates produced by the two models are practically identical across the age range.

```
R> set.seed(117)
R> fit_1980_poi <- hp(0:80, data$Ex.Total[data$Year == 1980],
+    data$Dx.Total[data$Year == 1980], model = "poisson")
```

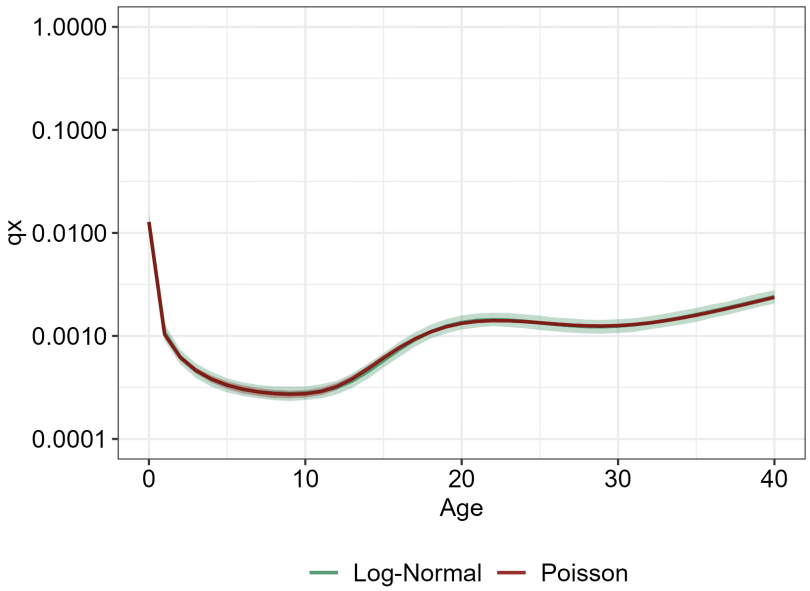


Figure 3: Posterior summaries via HP: Zoomed in median mortality curve and 99% predictive credible interval in log-scale for the log-normal and Poisson models. The United States, total population, ages 0-80 and year 1980.

```
Simulating [==================================] 100% in 50s

R> plot(list(fit_1980, fit_1980_poi), labels = c("Log-Normal", "Poisson"),
+    plotData = FALSE, age = 0:40, prob = 0.99,
+    colors = c("seagreen", "darkred"))
```

One way to evaluate the behavior of mortality for the United States population over the years and ages is by analyzing the estimate of the HP parameters, obtained via `summary` function, for the five years selected, as shown in Table 2. **BayesMortalityPlus** provides tools to check the convergence of the generated Markov chains obtained in the estimation procedure (for more details, see `plot_chain`). To investigate the mortality improvement over five years, we consider assessing the significance of the parameters through the credible intervals criterion. We consider that there is a significant difference when the credible intervals are disjoint. Notice that parameter A decreases significantly from 1980 to 1990 and from 1990 to 2000, and then there is no significant difference, but the point estimate continues to decline. Parameters B and C show similar interpretations as parameter A, both increasing over time until 2010, but in 2019 their estimates decrease to values close to the ones in 2000. It indicates that the changes are not statistically significant in a shorter temporal window. In summary, the level of mortality in the first years of life decreased significantly over the years, except for the last year of the analysis.

For the second term of the Heligman-Pollard law, see that parameter D decreases significantly over the years up to 2000, analogous to the estimates of parameter A. In 2019, there is a significant increase in its estimate. That means that the level of mortality in the accident hump decreased until 2000, persisted in this level in 2010, and increased in 2019. Parameter E indicates that 1980 and 2000 were the years in which the mortality in the accident hump

	US 1980	US 1990	US 2000	US 2010	US 2019
A	0.001026	0.000779	0.000546	0.000511	0.000398
	(0.000927; 0.001135)	(0.000648; 0.000915)	(0.000477; 0.000625)	(0.000409; 0.000638)	(0.000329; 0.000479)
B	0.026546	0.039092	0.053842	0.090452	0.055666
	(0.015898; 0.039927)	(0.017992; 0.068840)	(0.032378; 0.080154)	(0.046277; 0.149125)	(0.025771; 0.098442)
C	0.125732	0.133525	0.143113	0.165191	0.141152
	(0.115856; 0.135841)	(0.115996; 0.151385)	(0.128136; 0.158270)	(0.141631; 0.190495)	(0.118657; 0.165019)
D	0.000922	0.000772	0.000599	0.000591	0.000806
	(0.000858; 0.000991)	(0.000698; 0.000846)	(0.000553; 0.000648)	(0.000531; 0.000655)	(0.000725; 0.000888)
E	11.474677	6.448776	11.843087	8.785818	4.448380
	(10.223645; 12.900590)	(5.159864; 7.986869)	(10.114352; 13.879869)	(7.149351; 10.817768)	(3.575357; 5.398248)
F	21.091245	22.853152	20.930475	23.729958	29.036959
	(20.785356; 21.403583)	(21.961918; 23.816729)	(20.540865; 21.347651)	(23.014240; 24.530911)	(27.573791; 30.879372)
G	0.000074	0.000063	0.000060	0.000048	0.000044
	(0.000068; 0.000079)	(0.000054; 0.000073)	(0.000056; 0.000064)	(0.000043; 0.000053)	(0.000037; 0.000050)
H	1.090758	1.091188	1.090201	1.090971	1.091519
	(1.089471; 1.092192)	(1.088823; 1.093914)	(1.088962; 1.091458)	(1.089130; 1.092987)	(1.089212; 1.094211)

Table 2: Posterior summaries: Median and 95% credibility interval for the log-normal for years 1980, 1990, 2000, 2010 and 2019.

was most severe. The estimate of parameter F is around 21 to 24 years until 2010 and in 2019 its estimate becomes almost 29 years, indicating a large shift of the accident hump to older ages. For the last term of the HP function, parameter G shows a decreasing behavior over the years, which means that the level of mortality in adulthood is decreasing over the years. This reduction was significant in the period from 2000 to 2010. On the other hand, parameter H remains almost constant in all fits.

To facilitate comparison among the five fitted models, we call the function `plot` to visualize the behavior of the mortality curves. Figure 4 shows the fitted mortality curves for the United States population.

```
R> fits <- list(fit_1980, fit_1990, fit_2000, fit_2010, fit_2019)
R> labels <- c("US 1980", "US 1990", "US 2000", "US 2010", "US 2019")
R> plot(fits, labels = labels, plotData = FALSE, plotIC = FALSE)
```

As seen in Figure 4, there is a similar behavior of the adjusted tables in the first four years of the analysis, except for some level changes that occur between 1980 and 1990 in ages $x = 25$ and $x = 34$ and also between 2000 and 2010, in ages $x = 24$ and $x = 32$. This indicates consistency in the mortality pattern in the US population until 2010. On the other hand, this pattern is missed in 2019, where it can be observed that the accident hump is longer than in previous years, indicating that the causes of death that make up the accident hump are lasting longer than they used to.

Informative prior incorporation

One of the advantages of the Bayesian approach implemented in the **BayesMortalityPlus** package is the ability to incorporate informative prior information into the modeling process. To illustrate the incorporation of prior information, consider the following illustrative example: we generate artificial data based on the total population (in the year 2010 for the USA) assuming a noisy pattern at younger ages and overall lower volume of information. Figure 5

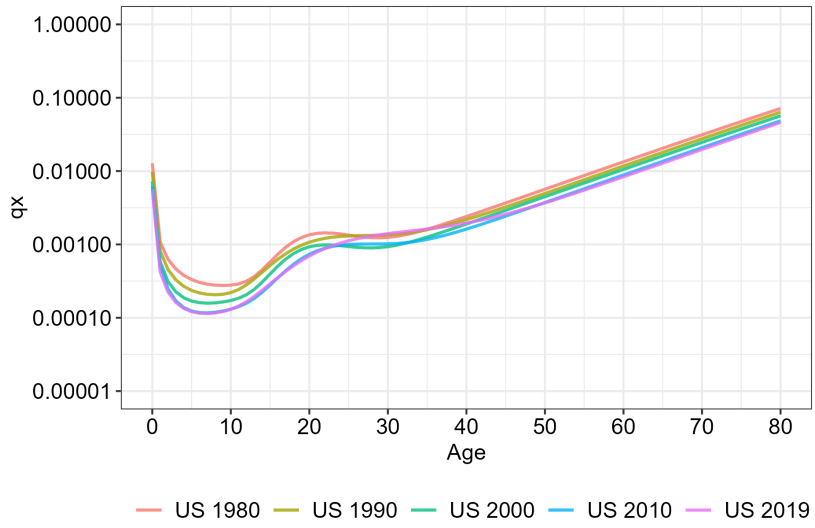


Figure 4: Posterior summaries via HP: Median mortality curve in log-scale. The United States, total population, ages 0-80 and years 1980, 1990, 2000, 2010, 2019.

(left panel) shows the artificially generated mortality data (green dots), which were obtained by reducing the exposure and death counts by 50% and then adding a truncated white noise (to avoid negative death counts) via **truncnorm** package (see `rtruncnorm` function, Mersmann, Trautmann, Steuer, and Bornkamp 2023). The specified white noise had a stronger effect on younger ages to simulate a common occurrence in certain types of mortality data. The behavior at younger ages directly affects the estimation of parameters A, B, C, and even D, E, and F, since the likelihood sometimes does not contain enough information about these parameters.

```
R> library("truncnorm")
R> USA2010 <- USA %>% filter(Year == 2010, Age <= 90)
R> set.seed(1)
R> sigma2 <- c(rep(10, 31), rep(1, 91 - 31))
R> Ex <- USA2010$Ex.Total / 50
R> Dx <- USA2010$Dx.Total / 50
R> set.seed(14)
R> Dx <- rtruncnorm(91, a = 0, mean = Dx,
+    sd = sqrt(Ex * (Dx / Ex) * (1 - Dx / Ex)) * sqrt(sigma2))
```

This data behavior is typical in the insurance context, where lower mortality incidence and lower demand for life insurance at younger ages result in more scarce and variable information, making the results less reliable. One way to address this issue is by incorporating prior information about this population, such as past experiences, inputs from experts, and reference life tables that reflect the behavior of this population. In this example, we consider past experience from the year 2009 for the mortality rate in the USA (see red dots in Figure 5 – left panel), as it exhibits well-behaved mortality and provides a sufficient volume of data.

Assuming that the expected raw mortality rates for the year 2010 should retain a similar shape, the means and standard deviations of the posterior distribution from the past year are

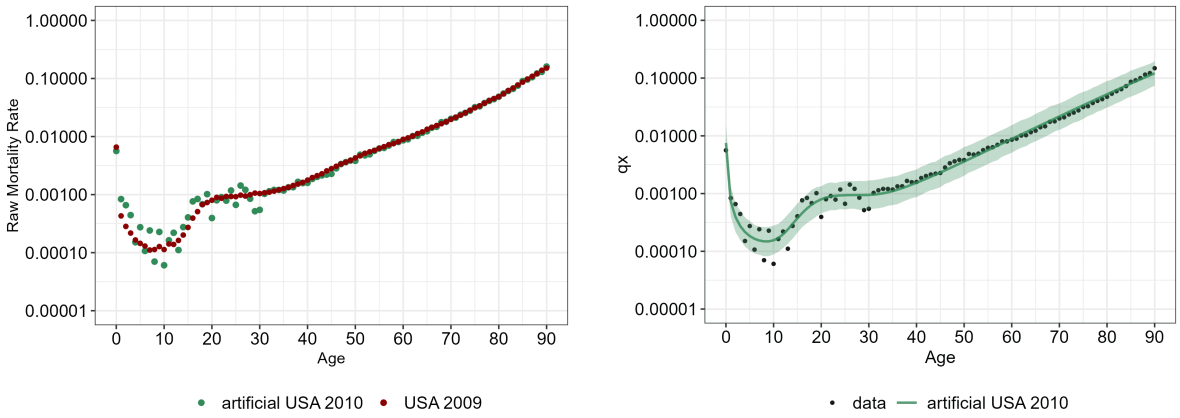


Figure 5: Posterior summaries via HP: Raw mortality rates motivation for past experience incorporation through informative prior distributions (left) and resulting fitted curve with 95% credible intervals (right) for the United States. Total population, ages 0-90 years old and year 2010 – artificial dataset.

good candidates to construct informative prior distributions for the year 2010. In this case, we fit the HP curve for the past experience to obtain the posterior distribution (regarding 2009). The obtained posterior means and standard deviations are then used as prior hyperparameters in order to produce inference based on the 2010 experience.

```
R> USA2009 <- USA %>% filter(Year == 2009, Age <= 90)
R> set.seed(14)
R> hp_2009 <- hp(0:90, Ex = USA2009$Ex.Total[1:91],
+    Dx = USA2009$Dx.Total[1:91], model = "lognormal", M = 50000)

Simulating [==================================] 100% in 48s

R> prior <- summary(hp_2009)$mean
R> prior_sd <- summary(hp_2009)$sd
```

As shown in Section 2, the arguments `m = rep(NA, 8)` and `v = rep(NA, 8)` are vectors that specify the means and variances for the prior distributions of the parameters and can be used to incorporate prior information for the model. In order to illustrate this specification, we consider incorporating prior information only for parameters B and D.

```
R> set.seed(14)
R> hp_2010 <- hp(0:90, Ex = Ex, Dx = Dx, model = "lognormal", M = 50000,
+    m = c(NA, prior[2], NA, prior[4], NA, NA, NA, NA),
+    v = c(NA, prior_sd[2]^3, NA, prior_sd[4]^3, NA, NA, NA, NA))

Simulating [==================================] 100% in 45s
```

The right panel of Figure 5 presents the estimation curve considering the past experience (the year 2009) as prior. The utilization of past experience demonstrated to be a suitable fit for the artificial mortality dataset. The two panels of Figure 5 are produced by:

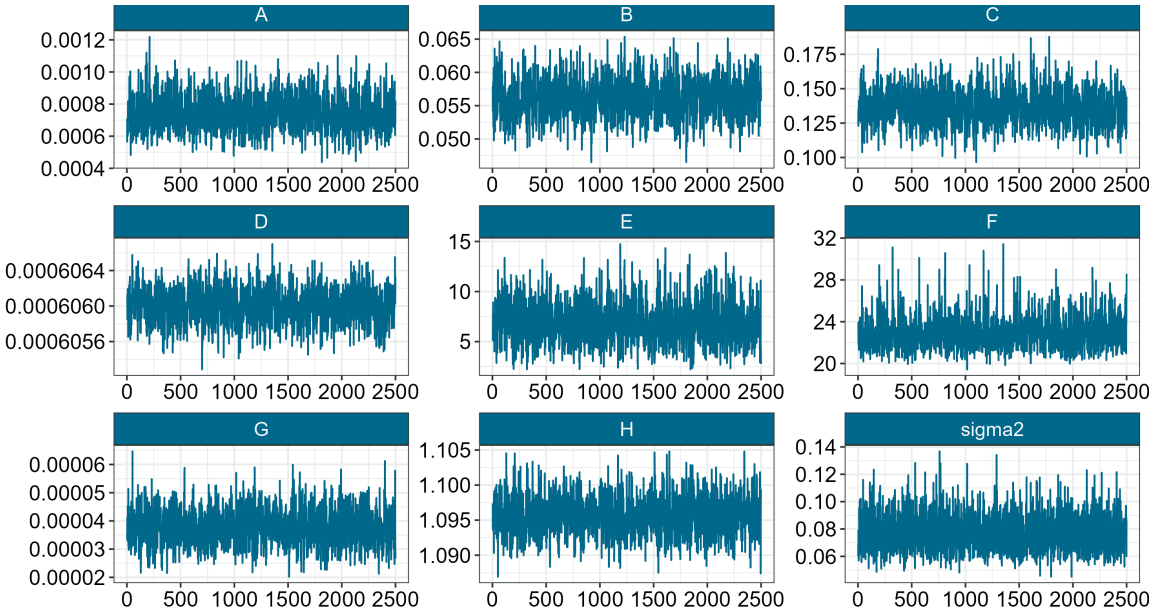


Figure 6: Posterior summaries via HP: Traces of the chains for the estimation of the parameters after incorporating prior information. The United States, total population, ages 0-90 and year 2010 – artificial dataset.

```
R> ggplot() +
+    scale_y_continuous(trans = "log10", breaks = 10^-seq(0, 5),
+      limits = 10^-c(5, 0), labels = scales::comma) +
+    scale_x_continuous(breaks = seq(0, 100, by = 10)) + theme_bw() +
+    theme(legend.position = "bottom", legend.text = element_text(size = 12),
+      axis.text = element_text(color = 'black', size = 12)) +
+    labs(x = "Age", y = "Raw Mortality Rate", title = NULL, color = "") +
+    geom_point(data = data.frame(x = 0:90, y = Dx / Ex),
+      aes(x = x, y = y, col = "artificial USA 2010")) +
+    geom_point(data = data.frame(x = 0:90,
+        y = (USA2009$Dx.Total / USA2009$Ex.Total)[1:91]),
+      aes(x = x, y = y, col = "USA 2009"), pch = 16) +
+    scale_color_manual(name = NULL, values = c("seagreen", "red4"),
+      label = c("artificial USA 2010", "USA 2009"))
R> plot(hp_2010, labels = "artificial USA 2010")
```

To visualize and evaluate the convergence of parameter estimation, the `plot_chain` function will plot the trace of chains, discarding the burn-in period and considering thinning for eliminating serial autocorrelation. Details regarding the default values or specification of these quantities, as well as the total number of iterations considered in the MCMC algorithm, can be examined by referring to the arguments `bn`, `thin`, and `M` in the fitting function `hp`.

```
R> plot_chain(hp_2010)
```

The resulting Figure 6 confirms that the parameters have converged to a reasonable value, and all other estimations align with expectations.

Life expectancy

The estimate of life expectancy is obtained via function `expectancy` through the curtate life expectancy as follows

$$e_x = \sum_{k=1}^{\omega} {}_kp_x,$$

where ω is the maximum age available in the life table, and ${}_kp_x$ is the probability that someone aged (x) will attain age $x + k$. Under the assumption of age-independent mortality, we can write ${}_kp_x$ as a cumulative product in terms of the survival probability ${}_kp_x = \prod_{i=0}^{k-1} p_{x+i} = p_x \times p_{x+1} \times p_{x+2} \times \ldots \times p_{x+k-1}$. Then,

$$e_x = \sum_{k=1}^{\omega} \left(\prod_{i=0}^{k-1} p_{x+i} \right).$$

The function is called in the package as follows:

```
expectancy(fit, Ex = NULL, age = NULL, graph = TRUE, max_age = 110,
  prob = 0.95)
```

The output from function `expectancy` is a `data.frame` with the resulting life expectancies and credible intervals for each age and, if `graph = TRUE`, a plot showing the life expectancy evolution throughout the age interval:

- `fit` represents the fitted curve by Heligman-Pollard or dynamic linear model.
- Arguments `Ex` and `prob` are exposure and probability, necessary to calculate the predictive intervals for the expectancy.
- By default, `Ex` is set to `NULL` which indicates that the exposure available for the life expectancy is the same as used in the fitted curve. It is important to note that argument `Ex` is used by the HP binomial and HP Poisson models to associate the uncertainty with the amount of available information.
- Argument `max_age` (default = 110) represents the maximum age to calculate the life expectancy. If necessary, the `expectancy()` function will extrapolate the fitted HP curve until it reaches the maximum age argument. In these cases, it is important to attend to `Ex` argument: if it is set to `NULL`, the function will repeat the last informed exposure to match the age interval.

The user can obtain the residual life expectancy for specific ages, with age 0 meaning life expectancy at birth. Note that the resulting numeric estimates are rounded internally and some minor differences may occur. For illustration, consider data from the year 1980 and setting `age = c(0, 20, 40, 60, 80)`, with the following code:

```
R> set.seed(120)
R> expectancy(fit_1980, age = c(0, 20, 40, 60, 80), graph = FALSE)
```

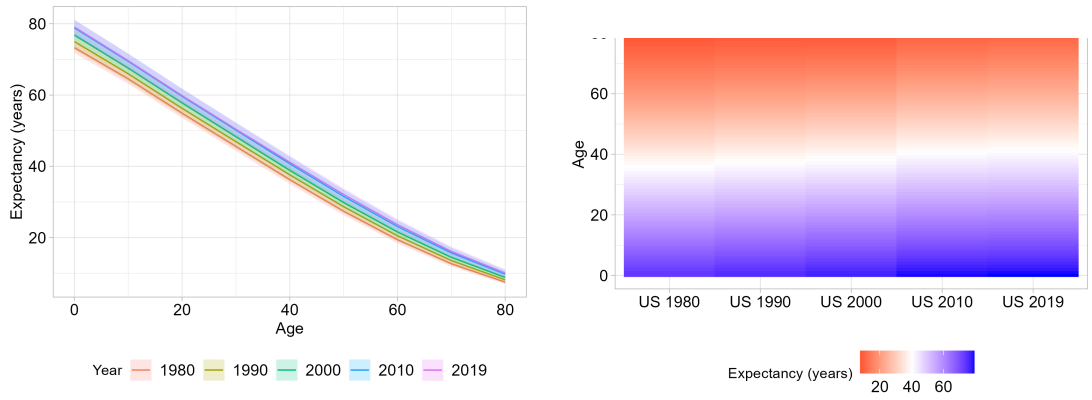


Figure 7: Posterior summaries via HP: Life expectancy graduation for the United States. Total population, ages 0-80 and years 1980, 1990, 2000, 2010, and 2019.

```
   age expectancy ci.lower ci.upper
1    0      73.31    71.76    74.82
21  20      54.91    53.57    56.24
41  40      36.26    35.04    37.47
61  60      19.42    18.43    20.41
81  80       7.51     6.93     8.11
```

Figure 7 illustrates the behavior of the posterior distribution for the life expectancy considering some years. For all ages, we see an increase in life expectancy. Notice that the increase is not significant in close years but is remarkable when the decades are considered. In 2019, the life expectancy at birth is superior by more than five years to that in 1980. In terms of point estimates, the most considerable difference occurs between 2000 and 2010, while the smallest difference is between 2010 and 2019. We provide a code for the life expectancy visualization below. In addition, **BayesMortalityPlus** package allows graphical visualization of the behavior of life expectancy over the years via `Heatmap` function as follows:

```
R> set.seed(121)
R> ex1 <- expectancy(fit_1980, age = seq(0, 80, by = 10), graph = FALSE) %>%
+    cbind(year = rep(1980, 9))
R> set.seed(122)
R> ex2 <- expectancy(fit_1990, age = seq(0, 80, by = 10), graph = FALSE) %>%
+    cbind(year = rep(1990, 9))
R> set.seed(123)
R> ex3 <- expectancy(fit_2000, age = seq(0, 80, by = 10), graph = FALSE) %>%
+    cbind(year = rep(2000, 9))
R> set.seed(124)
R> ex4 <- expectancy(fit_2010, age = seq(0, 80, by = 10), graph = FALSE) %>%
+    cbind(year = rep(2010, 9))
R> set.seed(125)
R> ex5 <- expectancy(fit_2019, age = seq(0, 80, by = 10), graph = FALSE) %>%
+    cbind(year = rep(2019, 9))
R> aux.ex <- rbind(ex1, ex2, ex3, ex4, ex5)
```

```
R> ggplot(data = aux.ex) + theme_light() +
+    theme(axis.title.x = element_text(color = 'black', size = 12),
+      axis.title.y = element_text(color = 'black', size = 12),
+      axis.text = element_text(color = 'black', size = 12),
+      legend.text = element_text(size = 12),
+      legend.position = "bottom") +
+    geom_line(aes(x = age, y = expectancy, color = as.factor(year))) +
+    geom_ribbon(aes(x = age, ymin = ci.lower, ymax = ci.upper,
+      fill = as.factor(year)), alpha = 0.2) +
+    labs(color = "Year", fill = "Year", x = "Age", y = "Expectancy (years)")
R> Heatmap(fits, x_lab = labels, age = 0:80)
```

Modeling adult ages

Assume a scenario in which interest lies in modeling only the mortality rate for adults. Some issues could result in this scenario such as poor quality of the data on the mortality of infants and young ages or their non-existence. For example, consider an insurance life product for employees of a company. In this case, we would not access data for children and young people due to the fact that they are not legally allowed to work. In this case, the younger ages are not modeled as the first term of the HP curve is not available for analysis. The argument `reduced_model` is available in function `hp` to deal with such situations. For illustration, consider the data for the year 2010 from age 18 (see Figure 8).

```
R> red_ex <- filter(data, Year == "2010", Age %in% 18:80)$Ex.Total
R> red_dx <- filter(data, Year == "2010", Age %in% 18:80)$Dx.Total
R> set.seed(126)
R> hp.fit2 <- hp(x = 18:80, Ex = red_ex, Dx = red_dx,
+    model = "lognormal", reduced_model = TRUE)

Simulating [==================================] 100% in 30s

R> plot(hp.fit2, plotIC = FALSE, labels = "HP fitted")
```

This approach allows fast computations and convergence in the MCMC algorithm for two reasons. Firstly, the dimension of the parameters is reduced, the acceptance rate raises and consequently, the algorithm converges fast. Furthermore, identifiability issues in the parameters A, B, and C could occur if they are taken into account in the inference procedure. These parameters are not estimated when considering the `reduce_model` argument. For instance, if we consider the 15-60 years old age range, the output returns to a specific warning:

```
R> data_aux <- data %>% filter(Age %in% 15:60, Year == 2010)
R> set.seed(127)
R> fit <- hp(15:60, data_aux$Ex.Total,
+    data_aux$Dx.Total, model = "lognormal")

Warning in hp(15:60, data_aux$Ex.Total, data_aux$Dx.Total,
  model = "lognormal") :
```

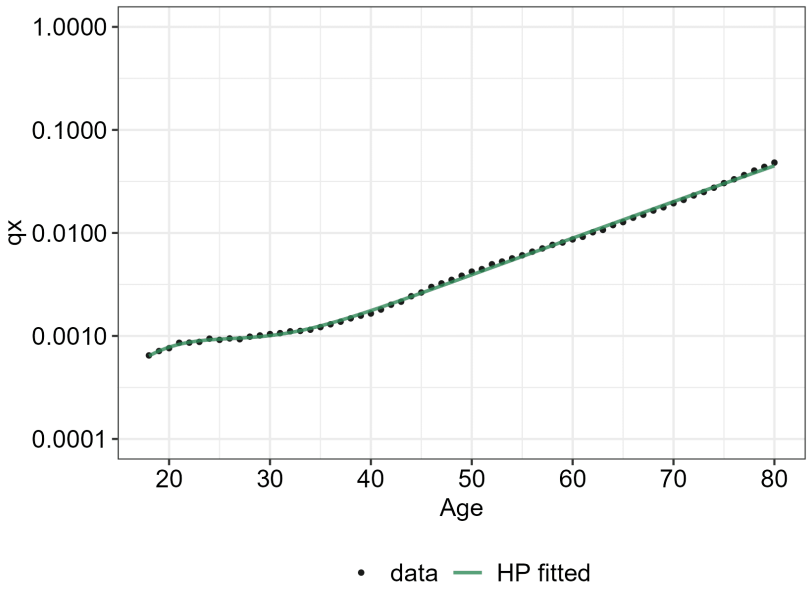


Figure 8: Posterior summaries via HP: Median mortality curve in log-scale via the reduced model. The United States, total population, ages 18-80 and year 2010. Black dots represent the raw mortality rates.

```
  Lower age >= 15. We recommend to use reduced_model = TRUE.
Simulating [==================================] 100% in  1m
Warning message:
In hp_lognormal(x = x, Ex = Ex, Dx = Dx, M = M, bn = bn, thin = thin,  :
  MCMC may have had some issues with the parameter(s): A, B, C.
Check the 'plot_chain' function output to visualize the parameters chain. It
might be helpful to assign informative prior distribution for these parameter
s. See ?hp.
```

Some scenarios can be studied. For the first scenario, if we consider the assumption that young-adult mortality begins at age 15 and ends at age 50, the modeling should incorporate the `reduced_model` argument in the `hp` function. It is important to note that the HP model is capable of capturing the behavior within this age range, particularly emphasizing the second term of the HP curve – the accident hump, which initiates at young-adult ages (refer to Figure 9).

```
R> data_aux <- data %>% filter(Age %in% 15:50, Year == 2010)
R> set.seed(128)
R> fit <- hp(15:50, data_aux$Ex.Total,
+    data_aux$Dx.Total, model = "lognormal", reduced_model = TRUE)

Simulating [==================================] 100% in 34s

R> plot(fit)
R> summary(fit)
```

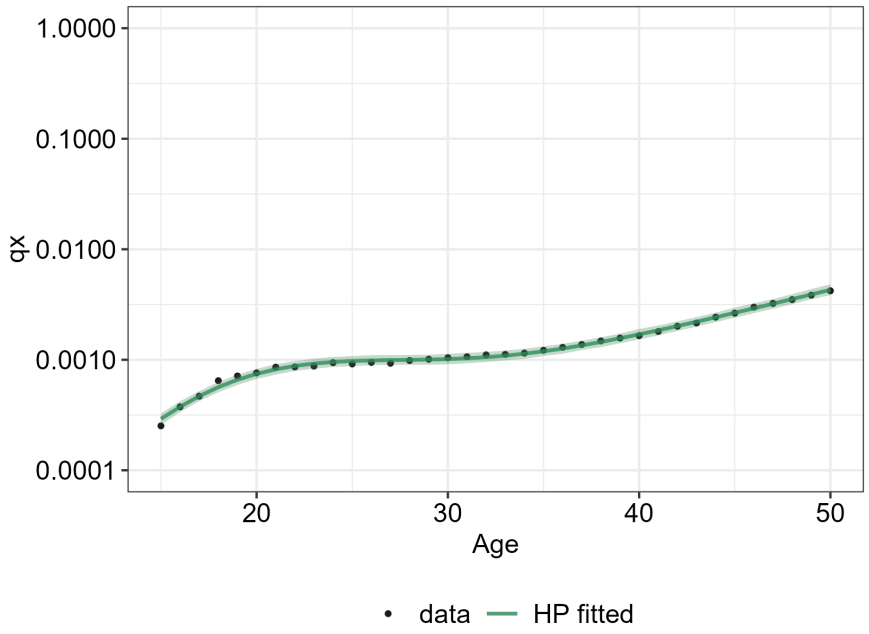


Figure 9: Scenario 1: HP curve for young-adult mortality begins at age 15 and ends at age 50.

```
       mean       sd      2.5%     50.0%     97.5% Accept %
A  0.000000 0.000000  0.000000  0.000000  0.000000      0.0
B  0.000000 0.000000  0.000000  0.000000  0.000000      0.0
C  0.000000 0.000000  0.000000  0.000000  0.000000      0.0
D  0.000625 0.000038  0.000556  0.000622  0.000704     31.1
E  7.016653 0.915445  5.329168  6.947915  8.926799     31.1
F 23.527901 0.364176 22.886080 23.501890 24.311139     31.1
G  0.000032 0.000007  0.000018  0.000032  0.000046     31.1
H  1.103566 0.005542  1.094127  1.103129  1.116395     31.1
```

The second scenario considers that the young-adult mortality begins at age 25 and ends at age 50. Note that the initial age of 25 years is a pivotal age as it represents the term for accidental deaths in the HP modeling. In this case, the HP model may be less flexible than other established approaches in the literature. Prior beliefs should be taken into account here.

```
R> data_aux <- data %>% filter(Age %in% 25:50, Year == 2010)
R> set.seed(129)
R> fit <- hp(25:50, data_aux$Ex.Total,
+    data_aux$Dx.Total, model = "lognormal", reduced_model = TRUE)

Simulating [==================================] 100% in 32s
Warning message:
In hp_lognormal_red(x = x, Ex = Ex, Dx = Dx, M = M, bn = bn, thin = thin,  :
  MCMC may have had some issues with the parameter(s): D, E, F, G.
Check the 'plot_chain' function output to visualize the parameters chain. It
might be helpful to assign informative prior distribution for these
parameters. See ?hp.
```

Warnings show us that each component parameter (D, E, F, G) is necessary to provide informative prior beliefs based on the mean and the variance of the prior distributions via arguments `m` and `v`, respectively. After the choice of the priors, the HP model is capable of capturing the behavior within this age range, particularly emphasizing the second term of the HP curve – the accident hump, which initiates at young-adult ages. Figure 10 shows the fit.

```
R> set.seed(130)
R> fit <- hp(25:50, data_aux$Ex.Total, data_aux$Dx.Total,
+    model = "lognormal",
+    m = c(NA, NA, NA, .0006, 7, 22, .0003, NA),
+    v = c(NA, NA, NA, 1e-5, .9, .3, 1e-7, NA), reduced_model = TRUE)

Simulating [==================================] 100% in 36s

R> plot(fit)
R> summary(fit)

       mean       sd      2.5%     50.0%     97.5% Accept %
A  0.000000 0.000000  0.000000  0.000000  0.000000      0.0
B  0.000000 0.000000  0.000000  0.000000  0.000000      0.0
C  0.000000 0.000000  0.000000  0.000000  0.000000      0.0
D  0.000587 0.000026  0.000534  0.000587  0.000641     27.9
E  5.084311 0.679830  3.853178  5.033740  6.431372     27.9
F 23.218069 0.531269 22.144443 23.232539 24.173043     27.9
G  0.000032 0.000004  0.000024  0.000031  0.000041     27.9
H  1.102970 0.003160  1.096401  1.102983  1.109065     27.9
```

Mortality measurement at advanced ages and extrapolation

The life tables are composed of the mortality probability q_x, associated with each age x. The estimates of mortality at advanced ages are difficult to compute due to the fact that there is a small number of survivors in this age group. In this context, to achieve a robust fit at advanced ages, following Hustead (2005) we consider four methodologies to accommodate the mortality pattern at the end of the mortality graduation. The synopsis of the function `hp_close` is given by:

```
hp_close(fit, method = c("hp", "plateau", "linear", "gompertz"),
  x0 = max(fit$data$x), max_age = 120, k = 7,
  weights = seq(from = 0, to = 1, length.out = 2 * k + 1),
  new_Ex = NULL, new_Dx = NULL)
```

This function receives an object of the class 'HP' adjusted by the function `hp` and fits a closing method to expand the data of the life table to a maximum age argument `max_age` (default = 120) input by the user. The user can adopt alternative approaches for closing tables using the argument `method`. The package provides four closing methods: `method = "hp"`, `method = "plateau"`, `method = "linear"` and `method = "gompertz"`. Notice that `method = "linear"` can only be used with 'HP' objects following the `model = "lognormal"` option of the HP model. Also,

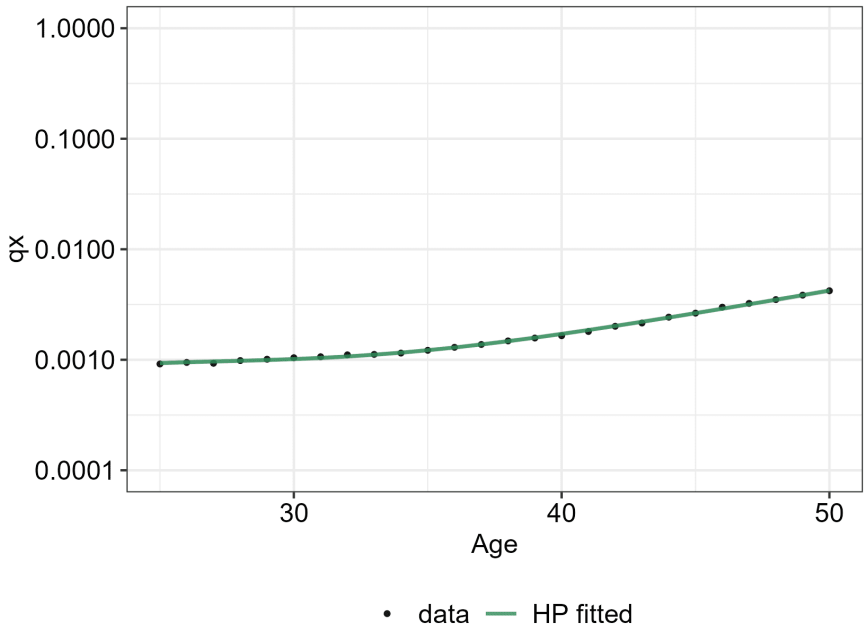


Figure 10: Scenario 2: HP curve for young-adult mortality begins at age 25 and ends at age 50.

- The parameters `x0` and `k` define the ages where a mixture procedure is applied. The age range defined is from `x0`−`k` to `x0`+`k`. In this interval, a weighted mean is calculated for the original fitted model and the closing model with the parameter `weights` specifying the weights of each model for each age in this mixture.
- The parameters `new_Ex` and `new_Dx` represent the data that was not fitted by the original HP curve, the exposure, and the death count after the `x0` argument. These are optional arguments used in the `method = "linear"` and `method = "gompertz"`.

For illustration of the closing methods available in the **BayesMortalityPlus** package, we consider the object `fit_2019` and ages 80–120 (default extrapolation).

The HP method The argument `method = "hp"` extrapolates the fitted Heligman and Pollard (1980) curve to the `max_age` argument informed by the user. We expect the mean of the posterior distribution to be similar to the values obtained by the `fitted` function, as well as the predictive intervals. Since it is just an extrapolation of the HP curve, the mixture is not applied. The results of the HP method are plotted in Figure 11 with 95% predictive credible interval, for the year 2019.

```
R> set.seed(131)
R> hp.close1 <- hp_close(fit_2019, method = "hp")
R> plot(hp.close1, labels = "hp method")
```

The plateau (less-than-one) method The `method = "plateau"` considers that the death probability q_x of the last age fitted by the HP model is kept constant until it reaches the maximum age. No mixture is applied. More detailed discussion about this mortality pattern at

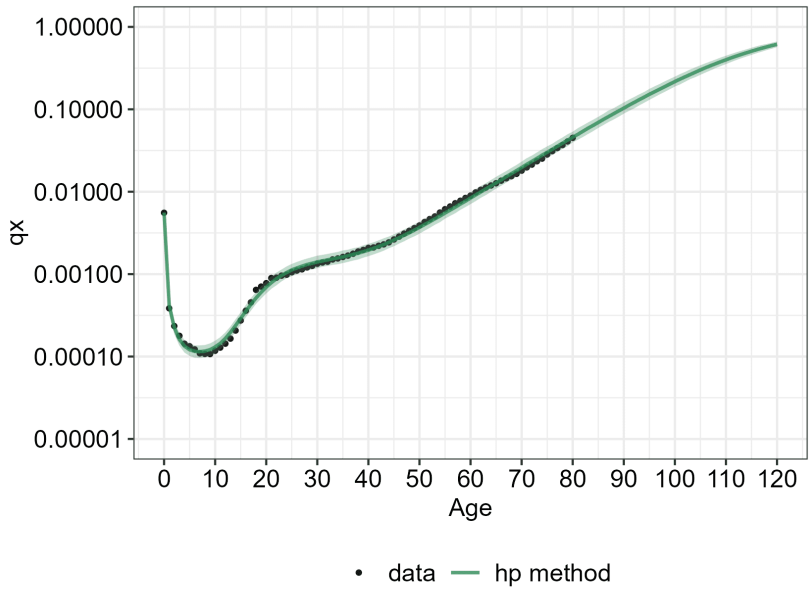


Figure 11: Posterior summaries via HP: Mortality graduation with HP curve extrapolation in log-scale. The United States, total population, ages 0–120 and year 2019.

the oldest ages can be seen in Lai (2012). The results of the plateau method are plotted in Figure 12 with 95% predictive credible interval, for the year 2019.

```
R> set.seed(132)
R> hp.close2 <- hp_close(fit_2019, method = "plateau")
R> plot(hp.close2, labels = "Plateau method")
```

The linear method The `method = "linear"` is only available for the log-normal model. This method fits a linear regression starting at age $x_0 - k$ (more details on that later) until the last age with available data and is specified as:

$$\log(q_x) = \beta_0 + \beta_1 x + \varepsilon_x, \quad \varepsilon_x \sim N(0, \sigma_0^2).$$

After fitting the linear regression, predictive samples are generated for the death probabilities q_x starting at age $x_0 - k$, dividing the graduation into three parts: the first one is the fitted curve given by the HP model, followed by the mixture interval given by the `k` argument, ending with the fitted linear regression. The results of the linear method are plotted in Figure 13 with 95% predictive credible interval, for the year 2019.

```
R> set.seed(133)
R> hp.close3 <- hp_close(fit_2019, method = "linear")
R> plot(hp.close3, labels = "Linear method")
```

The Gompertz method Details about the `method = "gompertz"` are available in Dodd *et al.* (2018) and Gavrilov and Gavrilova (2011). Dodd *et al.* (2018) consider the Gompertz

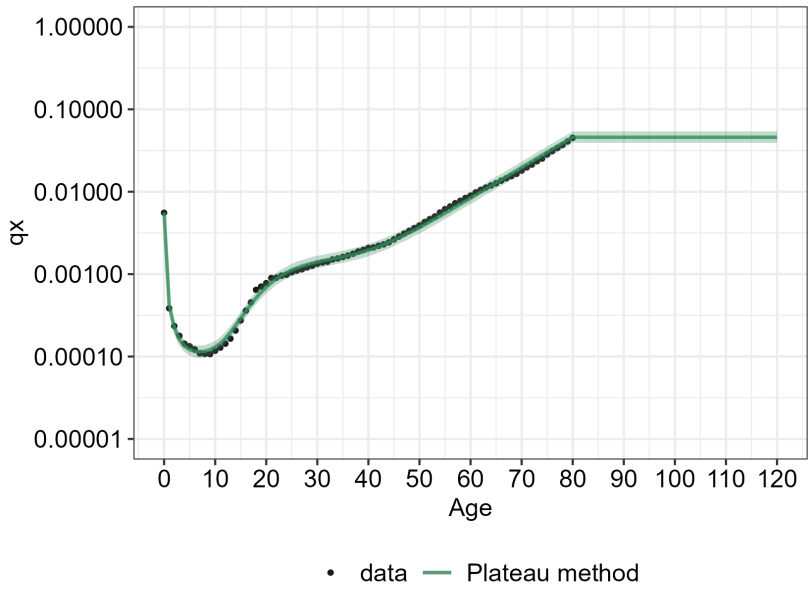


Figure 12: Posterior summaries via HP: Mortality graduation with plateau extrapolation in log-scale. The United States, total population, ages 0–120 and year 2019.

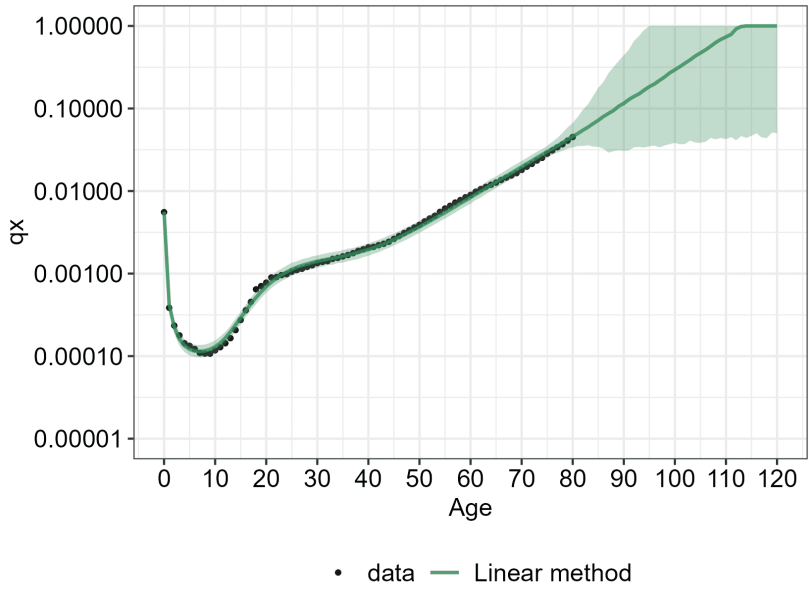


Figure 13: Posterior summaries via HP: Mortality graduation with linear extrapolation in log-scale. The United States, total population, ages 0–120 and year 2019.

curve to close the English life tables between 2010–2012 and conclude that the better the quality of mortality data at advanced ages, the more the behavior of the mortality curve approaches the Gompertz function.

This method fits the Gompertz curve developed by Gompertz (1825) through the sampling

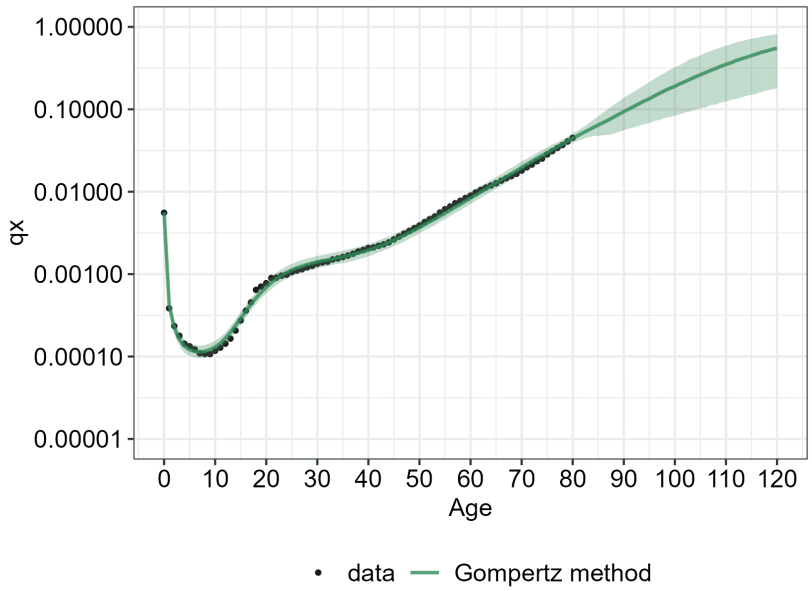


Figure 14: Posterior summaries via HP: Mortality graduation with Gompertz extrapolation in log-scale. The United States, total population, ages 0–120 and year 2019.

importance resampling method (SIR) as

$$\mu(x) = Ae^{Bx}, \quad q_x = 1 - e^{-\mu(x)},$$

where $\mu(x)$ represents the mortality force that grows exponentially with the increase of age x, depending on parameters A and B. Parameter $A \in (0, 1)$ reflects the general level of mortality, while $B \in (0, \infty)$ controls the rate at which the force of mortality increases with age. Notice that if we assume $C = e^B$, then $\mu(x) = AC^x$ that is equivalent to the third term of the eight-parameter HP curve previously seen in Section 2. The results of the Gompertz method are plotted in Figure 14 with 95% predictive credible interval, for the year 2019.

```
R> set.seed(134)
R> hp.close4 <- hp_close(fit_2019, method = "gompertz")
R> plot(hp.close4, labels = "Gompertz method")
```

This approach and the previous linear one are closing methods that apply a mixture between the fitted HP curve and the fitted closure curve. To illustrate the mixing procedure, consider the Gompertz extrapolation example made previously. The mixing arguments `x0` and `k` are both at their default values (in this example, 80 and 7, respectively). Consequently, the three age groups are as follows: 0 to 72, representing the original adjusted HP model; 73 to 87, constituting the mixing age group; and 88 to 120, representing the final age group adjusted solely by the closing method. The default last age in the function is 120. As no new data was introduced, the closing method is adjusted using ages from 73 $(x_0 - k)$ up to the last available age, which in this case is 80. Figure 15 shows us how these mixing age interval arguments impact the composition of the final product.

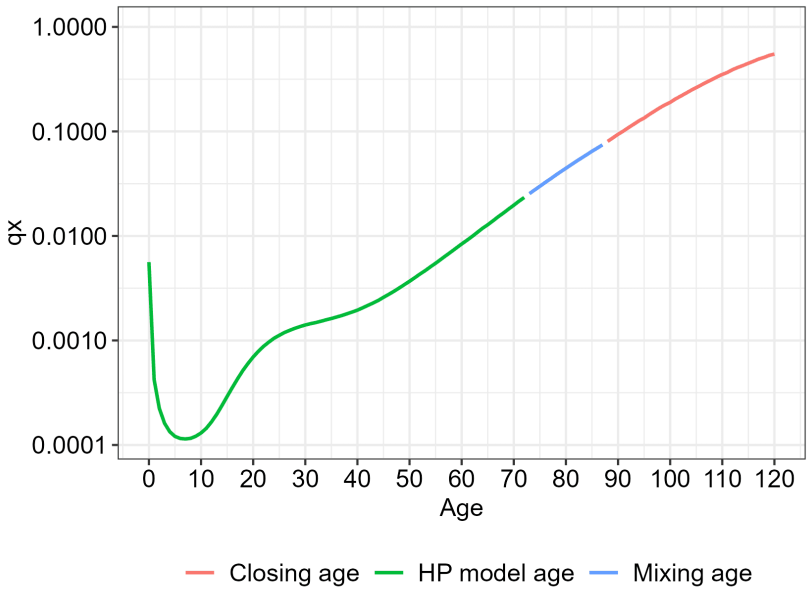


Figure 15: Posterior summaries via HP: Mortality graduation with Gompertz extrapolation in log-scale. The United States, total population, ages 0–120 and year 2019. The age groups are colored to illustrate the mixing procedure.

```
R> df <- fitted(hp.close4) %>%
+    select(age, qx.fitted) %>%
+    mutate(age_group = rep(c("HP model age", "Mixing age", "Closing age"),
+      c(73, 15, 33)))
R> ggplot(df) +
+    theme_bw() +
+    theme(plot.title = ggplot2::element_text(lineheight = 1.2),
+      axis.title.x = ggplot2::element_text(color = 'black', size = 12),
+      axis.title.y = ggplot2::element_text(color = 'black', size = 12),
+      axis.text = ggplot2::element_text(color = 'black', size = 12),
+      legend.text = ggplot2::element_text(size = 12),
+      legend.position = "bottom") +
+    scale_x_continuous(name = "Age", breaks = seq(0, 120, by = 10)) +
+    scale_y_continuous(name = "qx", trans = "log10", labels = scales::comma,
+      limits = c(NA, 1)) +
+    geom_line(aes(x = age, y = qx.fitted, col = age_group),
+      linewidth = 0.8) +
+    labs(col = NULL)
```

For these methods, there is also the option of providing new information about the data to guarantee a better fit for advanced ages. The `new_Ex` and `new_Dx` arguments represent the data that was left out by the original estimated HP curve, the exposure and death count after the `x0` argument. These arguments must have the same length. Figure 16 illustrates this option by comparing the previous Gompertz extrapolation and another one with new data added.

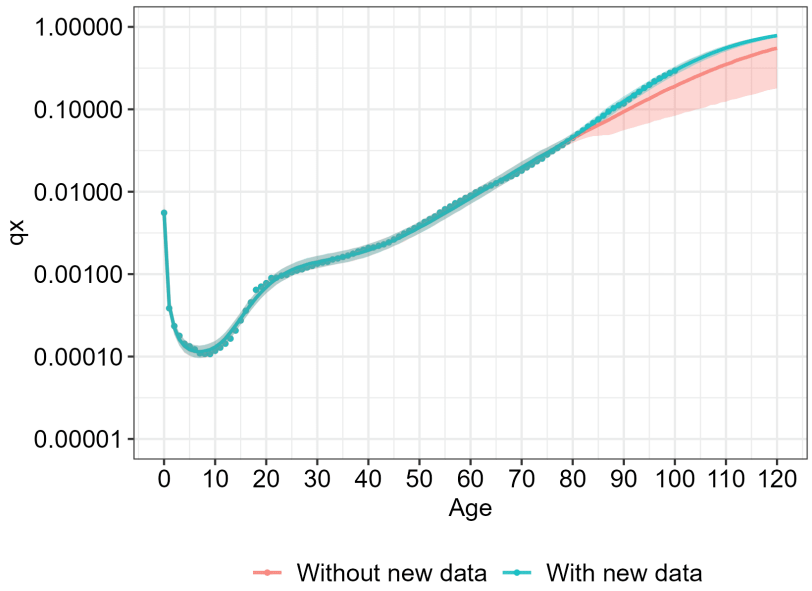


Figure 16: Posterior summaries via HP: Comparison between Gompertz extrapolations with and without new information in log-scale. The United States, total population, ages 0–120 and year 2019.

```
R> new_Ex <- filter(USA, Year == 2019)$Ex.Total[82:90]
R> new_Dx <- filter(USA, Year == 2019)$Dx.Total[82:90]
R> set.seed(135)
R> hp.close4.alt <- hp_close(fit_2019, method = "gompertz", new_Ex = new_Ex,
+    new_Dx = new_Dx)
R> plot(list(hp.close4, hp.close4.alt),
+    labels = c("Without new data", "With new data"))
```

After choosing the closing method, the object '`ClosedHP`' will be generated to save the new life table. This allows the new complete graduation to be used as an argument in other functions within the package.

In summary, the Gompertz model is employed for extrapolation by describing an exponential increase in mortality with age, assuming the continuation of the Gompertz trend at older ages. Conversely, the linear method assumes a linear age-mortality relationship, extrapolating the observed linear trend to older ages, potentially oversimplifying the mortality pattern. The pattern method relies on a consistent mortality pattern beyond the observed age range based on a specific model, with the plateau method using a rate less than one for extrapolation when mortality rates are very low and decreasing. The accuracy of these methods depends on the reliability of assumptions about mortality patterns at extreme ages.

4.2. Mortality graduation via dynamical linear smoothers

The function `dlm` returns an object of class '`DLM`', which is a dynamic linear model with input data settled by the user. We consider the same dataset for which we have applied the Heligman-Pollard model, with the data being transformed into log mortality to reproduce

the results of the graduation mortality curves. To fit a dynamical linear model under the log mortality for five different years, consider the following code:

```
R> y <- data %>%
+    mutate(logmx = log(mx)) %>%
+    select(Year, Age, logmx) %>%
+    pivot_wider(names_from = Year, values_from = logmx, id_cols = Age)
R> set.seed(135)
R> dlm_1980 <- dlm(y[[2]], delta = 0.85)
R> set.seed(136)
R> dlm_1990 <- dlm(y[[3]], delta = 0.85)
R> set.seed(137)
R> dlm_2000 <- dlm(y[[4]], delta = 0.85)
R> set.seed(138)
R> dlm_2010 <- dlm(y[[5]], delta = 0.85)
R> set.seed(139)
R> dlm_2019 <- dlm(y[[6]], delta = 0.85)
```

Posterior summaries for the five fitted models are allowed via the `summary` function. For illustration, we exhibit posterior summaries for the year 1980 and some ages, as follows:

```
R> head(summary(dlm_1980))
```

```
           mean      sd     2.5%    50.0%    97.5%
sigma2  1.69894 0.28331  1.23994  1.66997  2.31219
mu[0]  -4.35655 0.13103 -4.61839 -4.35769 -4.09541
mu[1]  -6.93392 0.12905 -7.18145 -6.93520 -6.68165
mu[2]  -7.33474 0.12998 -7.59899 -7.33591 -7.07436
mu[3]  -7.60868 0.11833 -7.84572 -7.60926 -7.38168
mu[4]  -7.87296 0.10181 -8.07472 -7.87133 -7.67453
```

Note that `mu[]` represents the posterior mean of the log mortality rate for each age in the study. The user can also call the `plot_chain` function to visualize the traces of the generated chains for the estimated parameters, as seen before. As an illustration, see in Figure 17 the traces for the posterior chains (based on data from 1980) for the mean and variance of the log mortality for ages 0, 40 and 80, respectively. Analogously, the user can resort to the same function to plot the posterior chains under the HP model.

```
R> params <- c("sigma2", "mu[0]", "mu[40]", "mu[80]")
R> plot_chain(dlm_1980, param = params)
```

Figure 18 shows the mortality curve fit via DLM graduation for the five years considered in the analysis, plotted through the function `plot`.

```
R> fits <- list(dlm_1980, dlm_1990, dlm_2000, dlm_2010, dlm_2019)
R> plot(fits, labels = labels, plotData = FALSE, plotIC = FALSE)
```

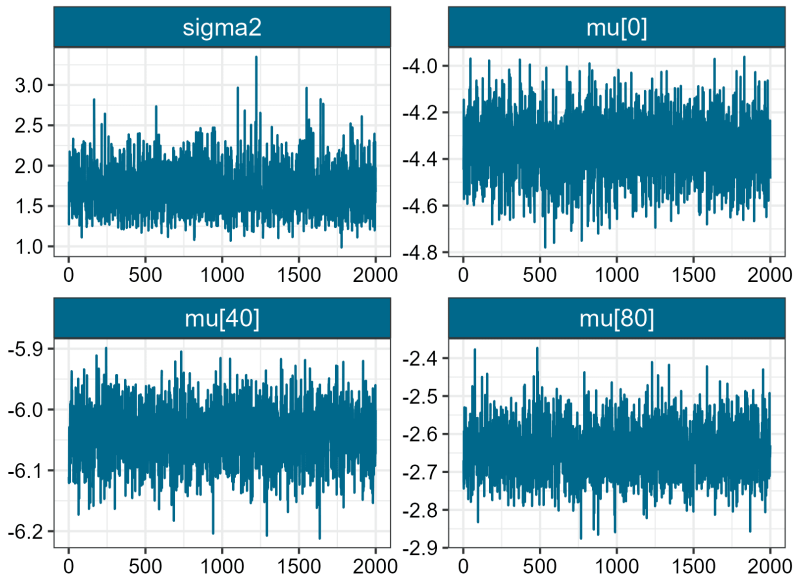


Figure 17: Posterior summaries via DLM: Traces of the chains for the mean and variance of the process. The United States, total population, ages 0, 40 and 80 for the mean chains and year 1980.

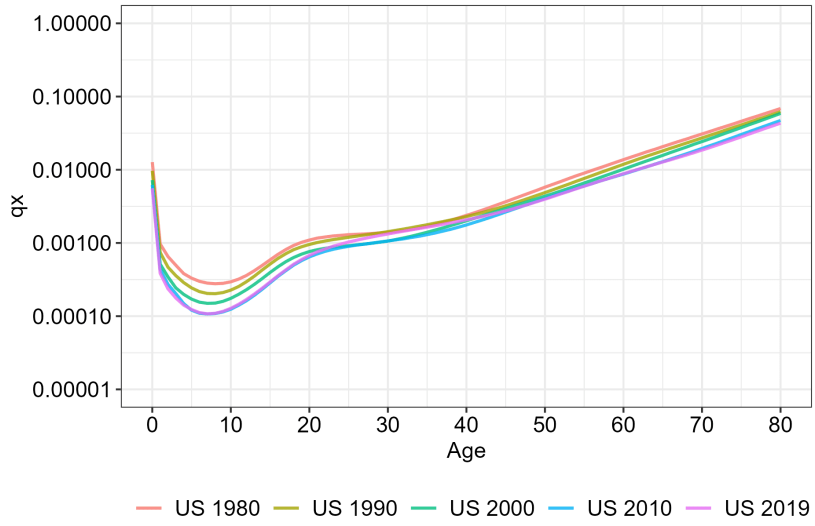


Figure 18: Posterior summaries via DLM: Median mortality curve in log-scale. The United States, total population, ages 0-80 and years 1980, 1990, 2000, 2010, 2019.

Predictive credible intervals for the mortality curves can be addressed using the `fitted` function, already mentioned in Section 4.1. Figure 19 presents the fitted mortality curve via DLM with the 95% credible interval for the year 1980 and is obtained with the code:

```
R> plot(dlm_1980, labels = "US 1980", plotIC = TRUE, plotData = TRUE)
```

As shown in Section 4.1, other posterior measures of the behavior of the population can be computed. For example, life expectations for the ages informed by the user, as seen below:

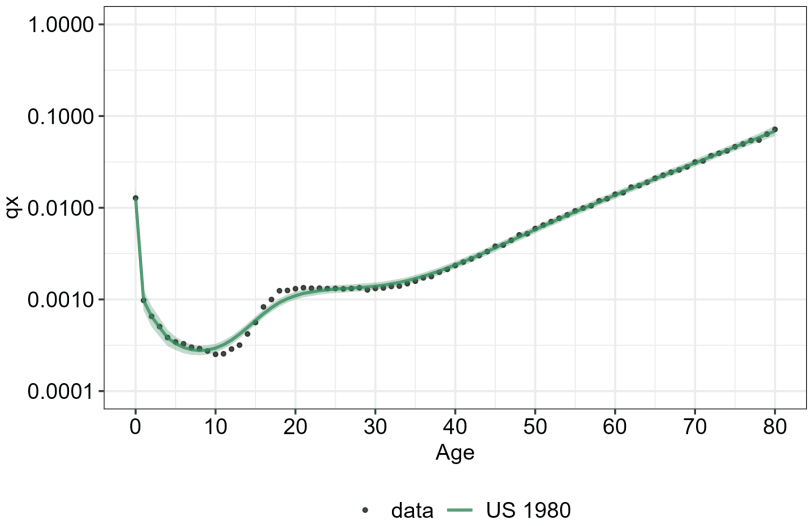


Figure 19: Posterior summaries via DLM: Median mortality curve and predictive credible interval 95% in log-scale. The United States, total population, ages 0-80 and year 1980. The black dots represent the raw mortality rates.

```
R> set.seed(141)
R> expectancy(dlm_1980, age = c(0, 20, 40, 60, 80), graph = FALSE)
```

```
   age expectancy ci.lower ci.upper
1    0      73.42    71.15    79.02
21  20      55.00    53.07    60.40
41  40      36.34    34.48    41.77
61  60      19.58    17.77    25.35
81  80       7.70     5.16    17.27
```

Opposed to the Heligman-Pollard model, the DLM approach does not assume a parametric structure in terms of mortality laws, resulting in more flexibility in the table graduation. We can model the adult ages as seen previously in Section 4.1, through the `dlm` function, taking into account a range of the adult ages of interest. For illustration purposes, an age range from 18 to 80 is considered. The results are plotted in Figure 20.

```
R> red_y <- filter(y, Age %in% 18:80)[[5]]
R> set.seed(142)
R> dlm.fit2 <- dlm(red_y, delta = 0.95, ages = 18:80)
R> plot(dlm.fit2, plotIC = FALSE, labels = "DLM fitted")
```

For the advanced ages modeling, the `"plateau"`, `"linear"` and `"gompertz"` methods are available for the '`DLM`' object through the `dlm_close` function. Usage and interaction with other functions is the same as seen in Section 4.1 resulting in a '`ClosedDLM`' object, with the exception of `"new_Ex"` and `"new_Dx"` arguments that are not used in the DLM methods, as it models the log-mortality directly, replaced by the `"new_data"` argument. Consider the object `dlm_2019` and ages 80–100 for illustration:

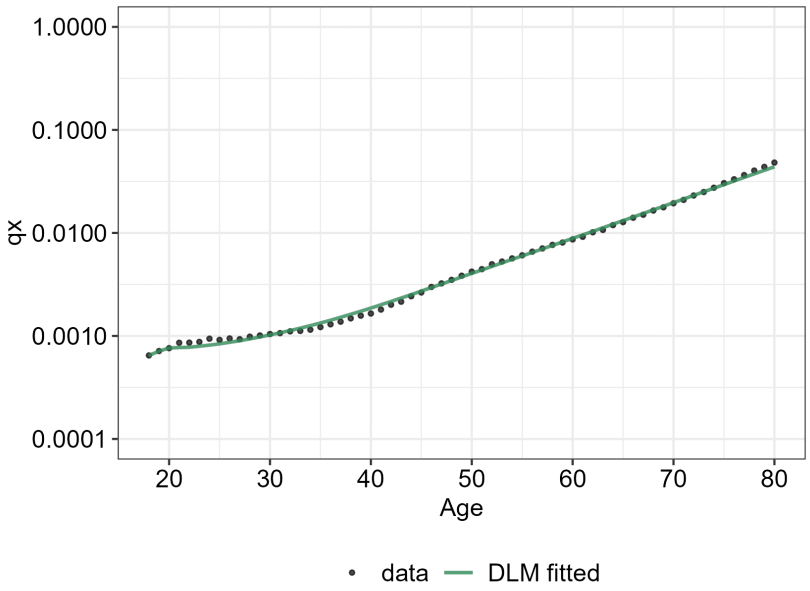


Figure 20: Posterior summaries via DLM: Median mortality curve in log-scale to adult ages. The United States, total population, ages 18-80 and year 2010. Black dots represent the raw mortality rates.

```
R> new_data <- log(new_Dx / new_Ex)
R> set.seed(143)
R> dlm.close1 <- dlm_close(dlm_2019, method = "plateau", max_age = 100)
R> set.seed(144)
R> dlm.close2 <- dlm_close(dlm_2019, method = "linear", max_age = 100,
+    new_data = new_data)
R> set.seed(145)
R> dlm.close3 <- dlm_close(dlm_2019, method = "gompertz", max_age = 100,
+    new_data = new_data)
```

Here, we bring attention to the fact that, due to the model nature, it is possible to fit a reasonable advanced age curve without the closing methods. As long as the advanced age data available has some degree of reliability, we encourage the user to try fitting the simpler `dlm` function, as seen in Figure 21.

```
R> new_Ex <- filter(USA, Year == 2019)$Ex.Total[1:101]
R> new_Dx <- filter(USA, Year == 2019)$Dx.Total[1:101]
R> new_y <- log(new_Dx / new_Ex)
R> set.seed(146)
R> dlm.fit3 <- dlm(new_y, delta = 0.85)
R> plot(list(dlm.fit3, dlm.close1, dlm.close2, dlm.close3), plotIC = FALSE,
+    plotData = FALSE, age = 70:100,
+    labels = c("DLM fitted", "Plateau", "Linear", "Gompertz"),
+    linetype = c("twodash", "solid", "solid", "solid")) + guides(
+      colour = guide_legend(override.aes = list(linetype = c(6, 1, 1, 1)))
+    )
```

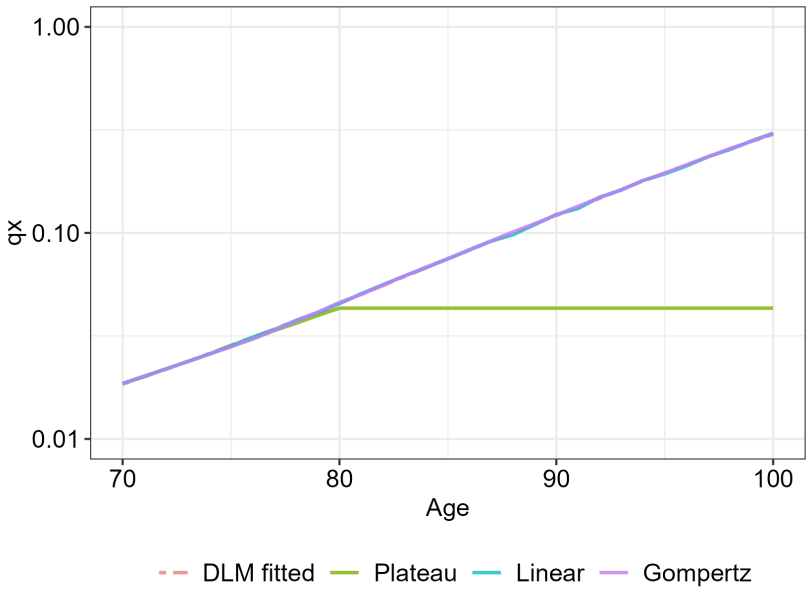


Figure 21: Posterior summaries via DLM: Mortality graduation with different closing methods in log-scale. The United States, total population, ages 70–100 and year 2019.

Extrapolation for dynamic linear smoothers

We consider k-steps-ahead predictive distributions to extrapolate the fitted mortality curve by a dynamic linear model. According to Petris *et al.* (2009, Section 2.8), for the DLM, the k-steps-ahead predictive distributions, $k = 1, 2, ...$ are obtained as a by-product of the Kalman filter as follows:

$$\begin{aligned}\pi(\theta_{x+k}|y_{1:x}) &= \int \pi(\theta_{x+k}|\theta_{x+k-1})\pi(\theta_{x+k-1}|y_{1:x})d\theta_{x+k-1},\\ \pi(y_{x+k}|y_{1:x}) &= \int \pi(y_{x+k}|\theta_{x+k})\pi(\theta_{x+k}|y_{1:x})d\theta_{x+k},\end{aligned}$$

where $\pi(.)$ are the filtered densities, $\pi(\theta_{x+k}|y_{1:x})$ denotes the k-steps-ahead prior distribution of the state θ and $\pi(y_{x+k}|y_{1:x})$ the k-steps-ahead forecast distribution of the observation. Considering extrapolation for k ages ahead, we obtain the h-step-ahead prediction distributions, $h = 1, 2, \ldots, k$, conditional on information up to the maximum age used in the model fitting. The prediction can be obtained by the basic `predict()` function as follows:

```
predict(object, h, prob = 0.95)
```

This function receives an object of the class '`DLM`' adjusted by the function `dlm` and returns a `data.frame` with the death probability prediction and credible intervals, with credibility level specified by the argument `prob` for the ages in the prediction horizon (argument `h`). Consider the ages 80–100 for the `dlm_2019` object predictions:

```
R> set.seed(147)
R> dlm.fit4 <- predict(dlm_2019, h = 20, prob = 0.95)
R> head(dlm.fit4)
```

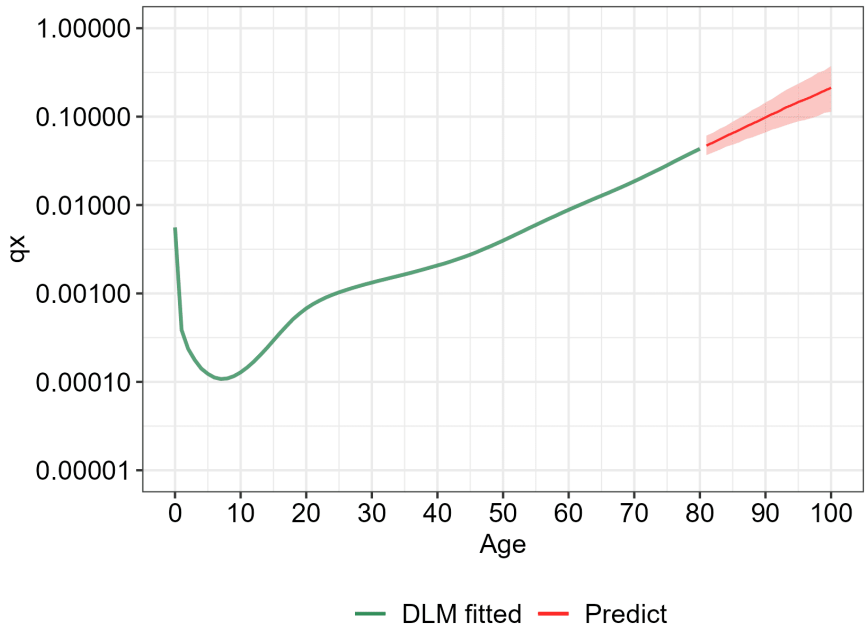


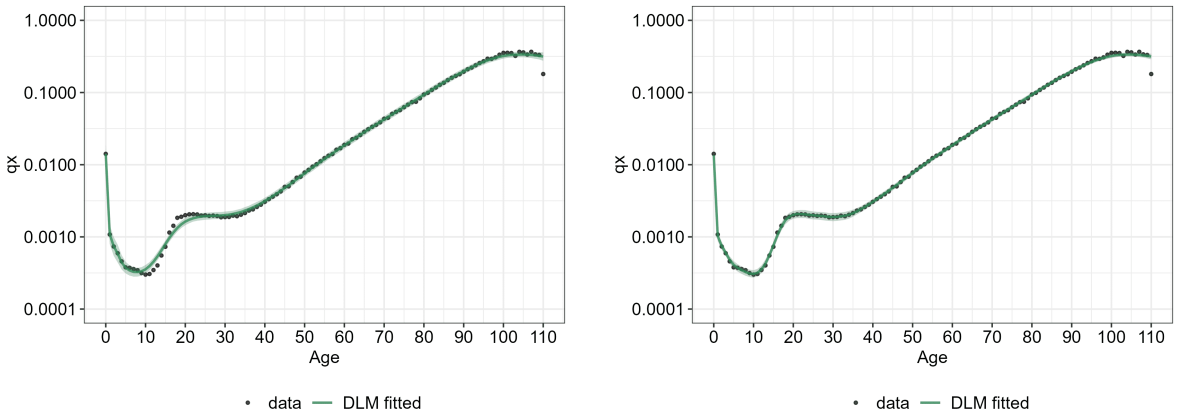
Figure 22: Posterior summaries via DLM: Prediction for the United States. Total population, ages 0–100 and year 2019.

```
  age  qx.fitted   qx.lower   qx.upper
1  81 0.04700581 0.03598107 0.06230630
2  82 0.05086165 0.03748048 0.06870590
3  83 0.05537742 0.03860046 0.07924328
4  84 0.06055539 0.04028375 0.08899759
5  85 0.06518669 0.04027860 0.10467863
6  86 0.07011525 0.04003360 0.12150102
```

Figure 22 illustrates the previous `dlm_2019` object predictions, as well as the credible interval associated with the point estimations, obtained with:

```
R> plot(dlm_2019, plotIC = FALSE, plotData = FALSE) +
+    geom_line(data = dlm.fit4,
+      aes(x = age, y = qx.fitted, col = "Predict")) +
+    geom_ribbon(data = dlm.fit4,
+      aes(x = age, ymin = qx.lower, ymax = qx.upper, fill = "Predict"),
+      alpha = 0.4) +
+    scale_color_manual(values = c("seagreen", "red"),
+      label = c("DLM fitted", "Predict")) +
+    guides(fill = "none", linetype = "none") + labs(colour = "")
```

The extrapolation via `predict` function is used in the life expectancy computation when the maximum age specified was not fitted. It replaces the extrapolation of the HP model method found in Section 4.1 to match the `max_age` argument.



(a) Static discount factor per age

(b) Different discount factor per age

Figure 23: Posterior summaries via DLM: (a) the mortality curve fitted considering static discount factor per age equal 0.85 and, (b) the mortality curve fitted considering different discount factor per age, 0.99 for ages 0–5, 0.60 for range 6–35 and 0.85 after 35 years old.

Different discount factors per age

To add flexibility to the DLM, allowing users to control the smoothness of the fit at each age interval, we consider to adopt different discount factors δ_x per age. In this case, the discount factor (argument `delta` in the `dlm` function) can vary across age domains. For instance, at younger ages, there may be higher volatility in the fitted curve, achieved by adopting discounts deviating from 1, while at adult and advanced ages, the fitted curve can be smoother by adopting discounts closer to 1.

We illustrate the flexibility offered by adopting different discount factors per age. For this, we analyse the mortality rates of the US male population in 1980. In the first scenario, a DLM is adjusted using a unique discount factor for all ages, set as 0.85 (default value in the `dlm` function). For the second scenario, we consider varying the discount factor per age, where we apply a discount factor of $\delta_{1,x} = 0.80$ for ages 0–35 years old and $\delta_{2,x} = 0.90$ for ages beyond 35.

```
R> data <- select(USA, Year, Age, Ex.Male, Dx.Male) %>%
+    filter(Year == 1980 & Age %in% 0:110) %>%
+    mutate(logmx = log(Dx.Male / Ex.Male)) %>%
+    select(Age, logmx)
R> set.seed(148)
R> dlm_1980_1 = dlm(data$logmx, delta = 0.85)
R> set.seed(149)
R> dlm_1980_2 = dlm(data$logmx, delta = rep(c(0.99, 0.6, 0.85), c(5, 31, 75)))
R> plot(dlm_1980_1)
R> plot(dlm_1980_2)
```

As we can see, the mortality rates after the age of 100 exhibit a decreasing trend as age advances. In such cases, the data for advanced ages is not reliable. Therefore, it is preferable to consider a discount factor approaching 1 for advanced ages. This ensures that the data from earlier age groups, which are more reliable, exert a more significant influence on the behavior of the mortality curve.

Figure 23 displays both scenarios for discount factors. Panel (a) illustrates the graduation considering a single discount factor. It is noticeable that the fitted mortality curve exhibits a decreasing trend in the older ages. Conversely, in Panel (b), the fitted mortality curve displays a more acceptable pattern with an increasing mortality rate in the older ages. Additionally, within the age range of 0–35, the fitted mortality curve (shown with posterior mean and a 95% predictive credible interval) is better for the second scenario (varying the discount factor) compared to the first scenario (static discount factor).

5. Bayesian Lee-Carter model

The methods presented in Sections 2 and 3 provide means for smoothing mortality rates over ages for cross-sectional data, but there are contexts in which mortality data are also available over years. Thus one can be interested in recognizing the evolution of mortality laws, as time passes. Dynamic linear models, as described in Section 3, can be naturally used to accommodate time-indexed observations. Several methods have emerged to enhance mortality forecasts based on Lee and Carter's work (Lee and Carter 1992). Li *et al.* (2004) extended the model to handle unequal observation times, while alternative approaches, incorporating the Poisson log-bilinear formulation, were proposed by Brouhns, Denuit, and Vermunt (2002) and Renshaw and Haberman (2003a,b). In a Bayesian framework, Czado *et al.* (2005) and Pedroza (2006) introduced methods for future mortality forecasting, enabling simultaneous parameter estimation with uncertainty consideration. Czado *et al.* (2005) applied MCMC techniques for the log-bilinear Poisson, and Pedroza (2006) notably extended the Lee-Carter model to address differential across-age variability. Pedroza's Bayesian approach, specifying the Lee-Carter model as a Dynamic Linear Model (West and Harrison 1997), considers Forward Filtering Backward Sampling (Carter and Kohn 1994; Frühwirth-Schnatter 1994) for state parameter estimation and Gibbs sampler to produce inference on the remaining parameters and predictive posterior distribution.

We consider a non-linear dynamic formulation indexed by age and time, as follows. Let $D_{x,t}$ denote the number of deaths at age x and calendar period t; and $E_{x,t}$ denote the population exposed to risk at age x and time t. The basic Lee and Carter (Lee and Carter 1992) model seeks to describe the age-time surface of log mortality rates $y_{x,t} = \log \frac{D_{x,t}}{E_{x,t}}$ as:

$$y_{x,t} = \alpha_x + \beta_x \kappa_t + \varepsilon_{x,t}, \quad \varepsilon_{x,t} \overset{iid}{\sim} N(0, \sigma_\varepsilon^2), \tag{7}$$

where α_x denotes the general log-mortality pattern for age x; β_x denotes an age-specific change rate in log-mortality. This term is essential for understanding how the mortality rate for a specific age responds to changes in the temporal trends common to all ages. If β_x is positive, it suggests that the mortality rate for age x tends to increase as general temporal trends rise. Otherwise, a negative β_x indicates that the mortality rate for age x decreases as temporal trends increase; $\varepsilon_{x,t}$ are sequentially independent and homoscedastic random errors and κ_t is an unobservable vector of time-indexed states, representing the global level of mortality at time period t, $t = 1, 2, \ldots, T$; $x = 1, 2, \ldots$. Pedroza (2006) follows Lee and Carter (1992) suggestion that κ_t, $t = 1, 2, \ldots, T$ evolve according to a random walk with drift, proposing the following state space representation for the temporal evolution of the states:

$$\kappa_t = \theta + \kappa_{t-1} + \omega_t, \quad \omega_t \overset{iid}{\sim} N(0, \sigma_\omega^2), \tag{8}$$

where ω_t are independent and homoscedastic evolution random errors, which are independent of the observational errors $\varepsilon_{x,t}$.

Unlike a traditional regression model, all quantities on the right side of Equation 7 are unobservable. In order to ensure identifiability, Lee and Carter (1992) impose the constraints $\sum_t \kappa_t = 0$ and $\sum_x \beta_x = 1$ and the estimation process uses single value decomposition to find a least squares solution, with κ_t reestimated using Box-Jenkins methodology. We follow Pedroza (2006) in its fully Bayesian approach for the fit of the model given by Equations 7 and 8, which enables simultaneous estimation of all the parameters while accounting for the uncertainty in the estimation process. Details on the MCMC algorithm can be found at Pedroza (2006).

5.1. Dynamical graduation with BayesMortalityPlus

The **BayesMortalityPlus** package provides an R implementation of the Bayesian Lee Carter (BLC) model proposed by Pedroza (2006). The BLC models are constructed using the `blc()` function. The brief of this function is given by:

```
blc(Y, prior = NULL, init = NULL, M = 5000, bn = 4000, thin = 1)
```

The `blc` function prompts as input a `matrix` type dataset with the log mortality rates, where the columns represent years and lines represent ages, to create an object of the type '`BLC`' acting the Bayesian Lee-Carter model.

- `Y` represents the matrix of log mortality rates containing the log ratio between deaths and exposures in a matrix format with ages on the rows and years on the columns.
- The argument `prior` (default = `NULL`) specifies the prior information about mean and variance, while `init` (default = `NULL`) specifies the initial values of each parameter to be estimated by the model.
- The arguments `M` (default = 5000), `bn` (default = 4000) and `thin` (default = 1) control the number of iterations, the warm-up interval and the thinning for the chains estimation, respectively.

In order to illustrate the modeling of the Bayesian Lee-Carter and other features available on **BayesMortalityPlus** for mortality and life expectancy forecasting consider total mortality data from Portugal, for ages 18 to 80 and the period from 2000 to 2015, obtained from HMD (2022).

```
R> data("PT")
R> Y <- PT
R> head(Y[,1:4])
```

```
        2000      2001      2002      2003
18 -7.159943 -7.145267 -7.480066 -7.498491
19 -7.092145 -7.216402 -7.331667 -7.380096
20 -7.077723 -7.101210 -7.244965 -7.287915
21 -7.035949 -7.020229 -7.370222 -7.133571
22 -6.845233 -6.956609 -7.363061 -7.184341
23 -7.043766 -7.150159 -7.083607 -7.302066
```

The usage of the `blc` function to fit the Bayesian Lee-Carter model results in an object of class 'BLC'. In this example, we consider the default settings to fit the BLC model with the code:

```
R> set.seed(150)
R> fit.blc <- blc(Y)

Simulating [==================================] 100% in  1m
```

The `fitted` function returns the fitted log mortality estimates for each year. For the 'BLC' object, the function returns an object of type `list` containing log mortality means (`$mean`) as well as credible intervals (`$lower` and `$upper`). The `plot` function is called to visualize the evolution of the fitted log mortality estimates through the years (Figure 24). The output from the function `fitted` for ages 18–23 and years 2000–2003 and call to the `plot` function are shown as follows:

```
R> set.seed(151)
R> head(fitted(fit.blc)[[1]][, 1:4])

           2000         2001         2002         2003
18 0.0007216572 0.0006838988 0.0006460932 0.0006198745
19 0.0007628780 0.0007264195 0.0006897155 0.0006642326
20 0.0008569172 0.0008135852 0.0007701043 0.0007399184
21 0.0008255496 0.0007881429 0.0007503962 0.0007241120
22 0.0009396375 0.0008903787 0.0008409458 0.0008067569
23 0.0008568085 0.0008173607 0.0007776398 0.0007499492

R> plot(fit.blc, parameter = "fitted", ages = 18:80)
```

The evolution of the mortality graduation can be seen through the α, β and κ parameters. Figure 25 depicts the fitted parameters of the BLC model using the `plot` method for the 'BLC' class.

```
R> plot(fit.blc, parameter = "all", ages = 18:80)
```

From Figure 25 we see that the α_x parameter displays the general mortality pattern present in the data. Notice that the parameter κ_t which represents the global level of mortality in period t is decreasing through the years. This behavior impacts the interpretation of parameter β_x directly. In this case, β_x is also called *improvement* and reflects the rate at which mortality is decreasing over the years. To capture the mortality improvements percentage over the time interval in the study, the users can apply the function called `improvement`. The `improvement` function is responsible to denote positive changes in mortality rates, typically measured as the first difference in log mortality rates between consecutive periods. The function helps assess whether mortality outcomes have improved or non improved over time, offering valuable insights into the impact of healthcare and societal factors on overall mortality trends.

```
R> set.seed(152)
R> head(improvement(fit.blc, prob = 0.95))
```

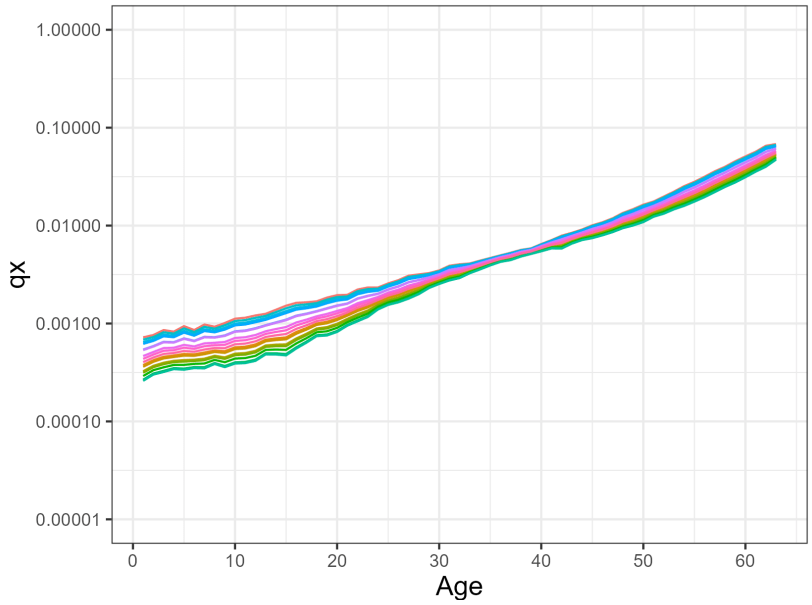


Figure 24: Posterior summaries via BLC: Median mortality curves in log-scale. Portugal, total population, ages 18–80 and years 2000–2015.

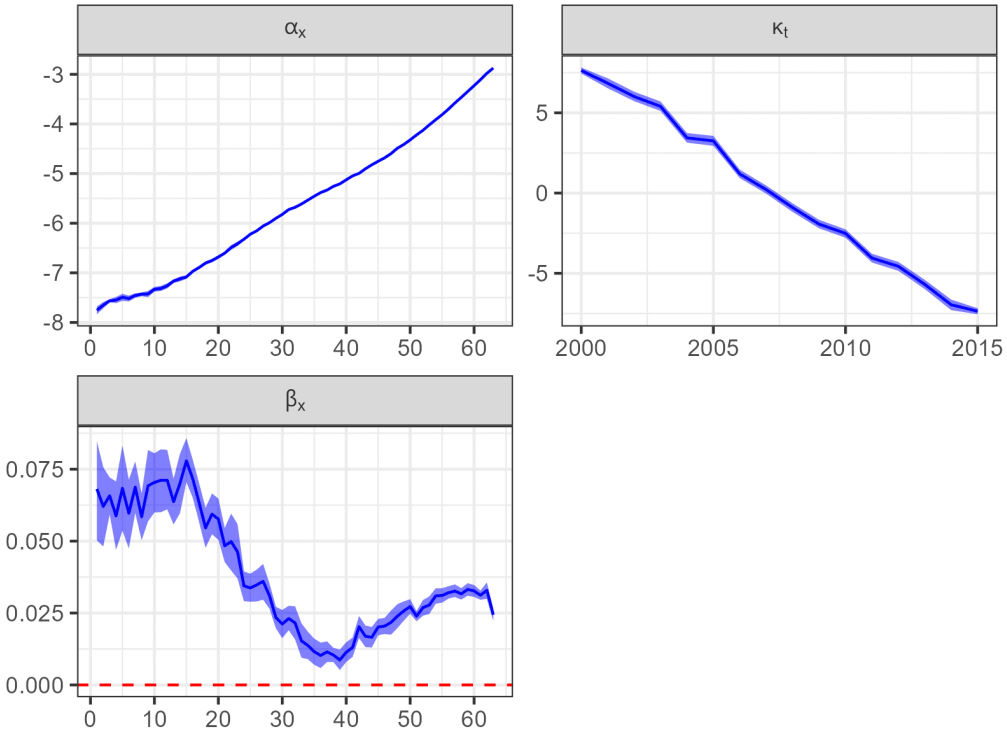


Figure 25: Posterior summaries via BLC: Mean and variance for each parameter. Portugal, total population, ages 18–80 and years 2000–2015.

```
  improvement  lower.lim  upper.lim
1  0.06581328 0.04618991 0.08617572
2  0.06013670 0.04489126 0.07558807
3  0.06364831 0.05614345 0.07051277
4  0.05704613 0.04359967 0.07128233
5  0.06605560 0.04962923 0.08240440
6  0.05790090 0.04416148 0.07131315
```

The `expectancy` and `Heatmap` methods are available to compute life expectations and their uncertainty via credible intervals for each age and year. For instance, we consider specific ages using the argument `"at"` and the output can be obtained with the commands:

```
R> set.seed(153)
R> expectancy(fit.blc, at = c(1, 21, 41, 61))$expectancy[, 1:4]
```

```
     2000   2001   2002   2003
18 55.885 56.036 56.192 56.302
38 36.983 37.081 37.184 37.258
58 19.024 19.091 19.162 19.213
78  2.657  2.664  2.672  2.678
```

Forecast for fitted BLC models

In the package **BayesMortalityPlus** the forecasting of the BLC mortality model for n-years ahead is implemented via the `predict` function. Following Pedroza (2006) the predictive steps can be incorporated into the Gibbs sampler. The posterior predictive distribution $(y_{n+1} \mid Y^n)$ for future observations can be expressed as

$$p(y_{n+1} \mid Y^n) = \int p(y_{n+1} \mid \phi, Y^n) p(\phi \mid Y^n) d\phi = \int p(y_{n+1} \mid \phi) p(\phi \mid Y^n) d\phi$$

where ϕ represents the model parameters. We assume that y_{n+1} and Y^n are conditionally independent given ϕ.

The prediction can be obtained by the basic `predict` function, specifying the years ahead to be forecasted by the `h` argument, resulting in an object of class '`PredBLC`'. The output considers 10-years-ahead ($h = 10$) for the Portugal mortality experience:

```
R> set.seed(154)
R> fit.blc2 <- predict(fit.blc, h = 10)
R> print(fit.blc2)
Forecast of a Bayesian Lee-Carter model (h = 10)
```

The functions `fitted` and `expectancy` are also available for the '`PredBLC`' object. See below:

```
R> set.seed(155)
R> head(fitted(fit.blc2)$mean[, 1:4])
```

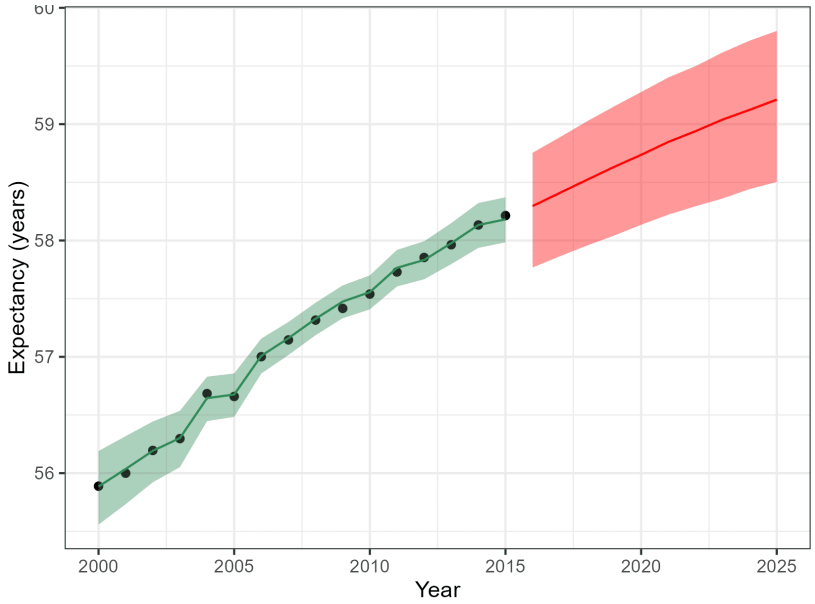


Figure 26: Posterior summaries via BLC: Fitted and predicted life expectancy at age 18 by year. Portugal, total population, ages 18–80 and years 2000–2015. The black dots represent the observed life expectancy.

```
             [,1]         [,2]         [,3]         [,4]
[1,] 0.0002430999 0.0002277473 0.0002132104 0.0001985425
[2,] 0.0002840746 0.0002652627 0.0002509471 0.0002342569
[3,] 0.0002998441 0.0002811439 0.0002632052 0.0002439636
[4,] 0.0003236503 0.0003039402 0.0002865571 0.0002701634
[5,] 0.0003176779 0.0002933812 0.0002748522 0.0002561061
[6,] 0.0003298484 0.0003124827 0.0002930122 0.0002761380
```

```
R> set.seed(156)
R> expectancy(fit.blc2, at = c(1, 21, 41, 61))$expectancy[, 1:4]
```

```
       [,1]   [,2]   [,3]   [,4]
[1,] 58.296 58.409 58.521 58.632
[2,] 38.702 38.789 38.878 38.967
[3,] 20.221 20.282 20.344 20.406
[4,]  2.783  2.789  2.795  2.802
```

To evaluate the fit and predictive performance of the BLC model, Figure 26) illustrates the life expectancy at age 18 by year, including the prediction horizon and observed values (which, in this case, correspond to the life expectancy calculated from age 18 to 80). The plot can be generated using the following code:

```
R> set.seed(157)
R> ex.fitted <- expectancy(fit.blc, at = 1)
R> ex.pred <- expectancy(fit.blc2, at = 1)
```

```
R> ex.obs <- apply(Y, 2, function(x) sum(cumprod(1 - (1 - exp(-exp(x))))))
R> ggplot(NULL) + theme_bw() +
+    theme(legend.text = element_text(size = 12)) +
+    geom_point(data = data.frame(x = 2000:2015, y = ex.obs),
+      aes(x = x, y = y), color = "black") +
+    geom_line(data = data.frame(x = 2000:2015, y = ex.fitted$expectancy),
+      aes(x = x, y = y), color = "seagreen") +
+    geom_ribbon(data = data.frame(x = 2000:2015, l = ex.fitted$lower,
+        u = ex.fitted$upper),
+      aes(x = x, ymax = u, ymin = l), fill = "seagreen", alpha = .4) +
+    geom_line(data = data.frame(x = 2016:2025, y = ex.pred$expectancy),
+      aes(x = x, y = y), color = "red") +
+    geom_ribbon(data = data.frame(x = 2016:2025, l = ex.pred$lower,
+        u = ex.pred$upper),
+      aes(x = x, ymax = u, ymin = l), fill = "red", alpha = .4) +
+    labs(y = "Expectancy (years)", x = "Year")
```

6. Conclusions

In this paper, we present an R package called **BayesMortalityPlus** for mortality modeling using a Bayesian approach. The package allows for Bayesian inference for several models used for mortality table graduation as well as mortality prediction for future years. The tools available in the proposed package provide model fitting, visualization of parameters and linear and non-linear functions of parameters, and uncertainty quantification via credible intervals or complete posterior distributions. Examples are provided to illustrate the features of all models. For the Heligman-Pollard law of mortality, Bayesian model fitting is available for three probability distributions: Poisson, binomial and log-normal. The interpretable parameters in the HP model can be visualized through graphs and summaries of the resulting posterior distributions. As an alternative to spline fitting for mortality graduation, the `dlm` function takes into account the autocorrelation in the mortality across ages and provides estimation of mortality curves and extrapolation for older ages. Opposed to the HP model, DLM fit does not depend on a specific law of mortality for the data. Smoothness is controlled by discount factors, which are common practice in the context of time series modeling via dynamic models, and offer flexibility to the model of death probabilities. Lastly, the Bayesian version of the well-known Lee-Carter model is implemented via MCMC methods and the `predict` function allows for prediction in future time steps. Furthermore, point estimates, as well as uncertainty measurements, can be computed for improvement parameters which are often the main interest in studies of longevity and pricing of products of long term in insurance modeling.

Acknowledgments

We are grateful to LabMA/UFRJ (Laboratório de Matemática Aplicada of the Universidade Federal do Rio de Janeiro) based in Brazil for financial support and to its members for the very enriching discussions.

References

Banerjee S, Carlin BP, Gelfand AE (2004). *Hierarchical Modeling and Analysis for Spatial Data*. Monographs on Statistics and Applied Probability 101. Chapman & Hall/CRC.

Barigou K, Goffard PO (2023). ***StanMoMo**: Bayesian Mortality Modelling with Stan*. doi:10.32614/CRAN.package.stanmomo. R package version 1.2.0.

Booth H, Maindonald JH, Smith L (2002). "Applying Lee-Carter under Conditions of Variable Mortality Decline." *Population Studies*, **56**, 325–336. doi:10.1080/00324720215935.

Bowers NL (1986). *Actuarial Mathematics*. Society of Actuaries. ISBN 9780938959106.

Brouhns N, Denuit M, Vermunt JK (2002). "A Poisson Log-Bilinear Regression Approach to the Construction of Projected Lifetables." *Insurance: Mathematics and Economics*, **31**(3), 373–393. doi:10.1016/s0167-6687(02)00185-3.

Camarda CG (2012). "**MortalitySmooth**: An R Package for Smoothing Poisson Counts with P-Splines." *Journal of Statistical Software*, **50**(1), 1–24. doi:10.18637/jss.v050.i01.

Camarda CG (2019). "Smooth Constrained Mortality Forecasting." *Demographic Research*, **41**(38), 1091–1130. doi:10.4054/demres.2019.41.38.

Camarda CG, Eilers PHC, Gampe J (2016). "Sums of Smooth Exponentials to Decompose Complex Series of Counts." *Statistical Modelling*, **16**(4), 279–296. doi:10.1177/1471082x16641796.

Carlin BP (1992). "A Simple Monte Carlo Approach to Bayesian Graduation." In *Transactions of the Society of Actuaries*. Citeseer.

Carter C, Kohn R (1994). "On Gibbs Sampling for State Space Models." *Biometrika*, **81**(3), 541?553. doi:10.2307/2337125.

Chen A (2023). ***leecarter**: Lee-Carter Model*. Python package version 1.0.2, URL https://pypi.org/project/leecarter/.

Congdon P (1993). "Statistical Graduation in Local Demographic Analysis and Projection." *Journal of the Royal Statistical Society A*, **156**(2), 237–270. doi:10.2307/2982731.

Currie ID, Durban M, Eilers PHC (2004). "Smoothing and Forecasting Mortality Rates." *Statistical Modelling*, **4**(4), 279–298. doi:10.1191/1471082x04st080oa.

Czado C, Delwarde A, Denuit M (2005). "Bayesian Poisson Log-Bilinear Mortality Projections." *Insurance: Mathematics and Economics*, **36**(3), 260–284. ISSN 0167-6687. doi:10.1016/j.insmatheco.2005.01.001.

Dellaportas P, Smith AFM, Stavropoulos P (2001). "Bayesian Analysis of Mortality Data." *Journal of the Royal Statistical Society A*, **164**(2), 275–291. doi:10.1111/1467-985x.00202.

Dodd E, Forster J, Bijak J, Smith P (2018). "Smoothing Mortality Data: The English Life Table, 2010–12." *Journal of the Royal Statistical Society A*, **181**(3), 717–735. doi:10.1111/rssa.12309.

Frühwirth-Schnatter S (1994). "Data Augmentation and Dynamic Linear Models." *Journal of Time Series Analysis*, **15**(2), 183–202. doi:10.1111/j.1467-9892.1994.tb00184.x.

Gamerman D, Lopes HF (2006). *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference*. Texts in Statistical Science, 2nd edition. Taylor & Francis. doi:10.1201/9781482296426.

Gavrilov LA, Gavrilova NS (2011). "Mortality Measurement at Advanced Ages: A Study of the Social Security Administration Death Master File." *North American Actuarial Journal*, **15**(3), 432–447. doi:10.1080/10920277.2011.10597629.

Gompertz B (1825). "On the Nature of the Function Expressive of the Law of Human Mortality, and on a New Mode of Determining the Value of Life Contingencies. In a Letter to Francis Baily, Esq. FRS &c." *Philosophical Transactions of the Royal Society of London*, **115**, 513–583. doi:10.1098/rspl.1815.0271.

Guo J, Gabry J, Goodrich B, Johnson A, Weber S, Badr HS (2025). "**rstan**: R Interface to Stan." doi:10.32614/CRAN.package.rstan. R package version 2.32.7.

Heligman L, Pollard JH (1980). "The Age Pattern of Mortality." *Journal of the Institute of Actuaries*, **107**(1), 49–80. doi:10.1017/s0020268100040257.

Hilton J, Dodd E, Forster JJ, Smith PWF (2019). "Projecting UK Mortality by Using Bayesian Generalized Additive Models." *Journal of the Royal Statistical Society C*, **68**(1), 29–49. doi:10.1111/rssc.12299.

HMD (2022). "Human Mortality Database. Max Planck Institute for Demographic Research (Germany), University of California, Berkeley (USA), and French Institute for Demographic Studies (France)." Accessed: 2022-07-28, URL https://www.mortality.org/.

Hustead EC (2005). "Ending the Mortality Table." In *Living to 100 and Beyond Symposium*.

Hyndman RJ (2023). ***demography**: Forecasting Mortality, Fertility, Migration and Population Data*. doi:10.32614/CRAN.package.demography. R package version 2.0.

Hyndman RJ, Shahid Ullah M (2007). "Robust Forecasting of Mortality and Fertility Rates: A Functional Data Approach." *Computational Statistics & Data Analysis*, **51**(10), 4942–4956. ISSN 0167-9473. doi:10.1016/j.csda.2006.07.028.

Inc TM (2023). *MATLAB – Financial Toolbox (R2023b)*. Natick. URL https://www.mathworks.com/help/finance/.

Kimeldorf G, Jones DA (1967). "Bayesian Graduation." *Transactions of the Society of Actuaries*, **19**(54 part 1), 66–112. doi:10.1159/000211609.

Lai CD (2012). "Human Mortality Curves That Decelerate to a Plateau." *Mathematical and Computer Modelling*, **55**(3-4), 1118–1128. doi:10.1016/j.mcm.2011.09.036.

Lee R, Miller T (2001). "Evaluating the Performance of the Lee-Carter Method for Forecasting Mortality." *Demography*, **38**, 537–549. doi:10.2307/3088317.

Lee RD, Carter LR (1992). "Modeling and Forecasting U.S. Mortality." *Journal of the American Statistical Association*, **87**(419), 659–671. doi:10.2307/2290201.

Li J (2013). "A Poisson Common Factor Model for Projecting Mortality and Life Expectancy Jointly for Females and Males." *Population Studies*, **67**(1), 111–126. doi:10.1080/00324728.2012.689.

Li N, Lee RD, Tuljapurkar S (2004). "Using the Lee-Carter Method to Forecast Mortality for Populations with Limited Data." *International Statistical Review*, **72**, 19–36. doi:10.1111/j.1751-5823.2004.tb00221.x.

Mersmann O, Trautmann H, Steuer D, Bornkamp B (2023). ***truncnorm**: Truncated Normal Distribution*. doi:10.32614/CRAN.package.truncnorm. R package version 1.0-9.

Neves C, Migon HS (2007). "Bayesian Graduation of Mortality Rates: An Application to Reserve Evaluation." *Insurance: Mathematics and Economics*, **40**, 424?434. doi:10.1016/j.insmatheco.2006.06.005.

Njenga CN, Sherris M (2020). "Modeling Mortality with a Bayesian Vector Autoregression." *Insurance: Mathematics and Economics*, **94**, 40–57. doi:10.1016/j.insmatheco.2020.05.011.

Pascariu MD (2025). ***MortalityLaws**: Parametric Mortality Models, Life Tables and HMD*. doi:10.32614/CRAN.package.mortalitylaws. R package version 2.1.3.

Pedroza C (2006). "A Bayesian Forecasting Model: Predicting U.S. Male Mortality." *Biostatistics*, **7**(4), 530–550. ISSN 1465-4644. doi:10.1093/biostatistics/kxj024.

Petris G, Petrone S, Campagnoli P (2009). "Dynamic Linear Models." In *Dynamic Linear Models with R*, pp. 31–84. Springer-Verlag.

R Core Team (2025). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. doi:10.32614/R.manuals. URL https://www.R-project.org/.

Renshaw A, Haberman S (2003a). "Lee-Carter Mortality Forecasting: A Parallel Generalized Linear Modelling Approach for England and Wales Mortality Projections." *Journal of the Royal Statistical Society C*, **52**(1), 119–137. doi:10.1111/1467-9876.00393.

Renshaw AE, Haberman S (2003b). "On the Forecasting of Mortality Reduction Factors." *Insurance: Mathematics and Economics*, **32**(3), 379–401. doi:10.1016/s0167-6687(03)00118-5.

Renshaw AE, Haberman S, Hatzopoulos P (1996). "The Modeling of Recent Mortality Trends in United Kingdom Male Assured Lives." *British Actuarial Journal*, **2**. doi:10.1017/s1357321700003470.

Sharrow DJ (2012). ***HPbayes**: Heligman Pollard Mortality Model Parameter Estimation Using Bayesian Melding with Incremental Mixture Importance Sampling*. R package version 0.1, URL https://CRAN.R-project.org/package=HPbayes.

Tang K, Dodd E, Forster J (2021). "Joint Modelling of Male and Female Mortality Rates Using Adaptive P-Splines." *Annals of Actuarial Science*, **16**, 1–17. doi:10.1017/s1748499521000105.

The MathWorks Inc (2021). *MATLAB – The Language of Technical Computing, Version R2021a*. Natick. URL https://www.mathworks.com/products/matlab/.

Turner H, Firth D (2023). ***gnm**: Generalized Nonlinear Models*. doi:10.32614/CRAN.package.gnm. R package version 1.1-5.

Van Rossum G, *et al.* (2011). *Python Programming Language*. URL https://www.python.org/.

Villegas AM, Kaishev VK, Millossovich P (2018). "**StMoMo**: An R Package for Stochastic Mortality Modeling." *Journal of Statistical Software*, **84**(3), 1–38. doi:10.18637/jss.v084.i03.

West M, Harrison J (1997). *Bayesian Forecasting and Dynamic Models*. 2nd edition. Springer-Verlag.

Wickham H (2016). ***ggplot2**: Elegant Graphics for Data Analysis*. Springer-Verlag, New York. doi:10.1007/978-0-387-98141-3.

Wickham H, François R, Henry L, Müller K, Vaughan D (2023). ***dplyr**: A Grammar of Data Manipulation*. doi:10.32614/CRAN.package.dplyr. R package version 1.1.4.

Wickham H, Pedersen TL, Seidel D (2025). ***scales**: Scale Functions for Visualization*. doi:10.32614/CRAN.package.scales. R package version 1.4.0.

Wood SN (2006). *Generalized Additive Models: An Introduction with R*. Chapman & Hall/CRC, Boca Raton.

Affiliation:

Viviana G. R. Lobo
Departmento de Métodos Estatísticos
and
Laboratório de Matemática Aplicada
Instituto de Matemática
Universidade Federal do Rio de Janeiro
Av. Athos da Silveira Ramos, Centro de Tecnologia, Bloco C, CEP 21941-909, Brazil
E-mail: viviana@dme.ufrj.br
URL: https://sites.google.com/a/dme.ufrj.br/viviana/

Journal of Statistical Software https://www.jstatsoft.org/
published by the Foundation for Open Access Statistics https://www.foastat.org/
July 2025, Volume 113, Issue 9 *Submitted:* 2023-09-14
doi:10.18637/jss.v113.i09 *Accepted:* 2024-08-20