Package: paramDemo (via r-universe)

September 9, 2024

Type Package

Title Parametric and Non-Parametric Demographic Functions and Applications

Version 1.0.1

Date 2024-02-05

Author Fernando Colchero

Maintainer Fernando Colchero <`fernando_colchero@eva.mpg.de`>

Depends R (>= 2.10)

Description Functions to extract demographic information from parametric mortality and Fertility models, summary statistics (e.g. ageing rates, life expectancy, lifespan equality, etc.), life table and product limit estimator calculation from census data.

License GPL

LazyLoad yes

LazyData yes

Repository https://fercol.r-universe.dev

RemoteUrl https://github.com/fercol/paramdemo

RemoteRef HEAD

RemoteSha 64716b88ee94848d940e8f7add0004e346aa098b

Contents

`paramDemo-package` *Parametric and Non-Parametric Demographic Functions and Applications*

Description

Functions for parametric and non-parametric calculations of survival, mortality and fertility

Details

Package:	paramDemo
Type:	Package
Version:	1.0.1
Date:	2024-02-05
License:	GNU General Public Licence
LazyLoad:	yes

This package contains functions for calculations of parametric mortality, survival and fertility models, as well as non-parametric methods for life tables and product limit estimators.

Author(s)

Fernando Colchero <`fernando_colchero@eva.mpg.de`>

`CalcAgeingRateMort` *Calculating actuarial and reproduction aging rates.*

Description

`CalcAgeingRateMort` and `CalcAgeingRateFert` are used to calculate actuarial and reproductive ageing rates from parametric models of mortality and fertility.

Usage

```
CalcAgeingRateMort(theta, x, model = "GO", shape = "simple",
                   checkTheta = TRUE)

CalcAgeingRateFert(beta, x, modelFert = "quadratic", ageMatur = 0,
                   checkBeta = TRUE)
```

Arguments

`theta`	Numerical vector of age-specific mortality parameters (see details).
`beta`	Numerical vector of age-specific fertility parameters (see details.)
`x`	Numerical vector of ages at which to calculate mortality.
`model`	The underlying mortality model to be used. `"EX"` = exponential,`"GO"` = Gompertz, `"WE"` = Weibull and `"LO"` = logistic (see `details`).
`shape`	The overall shape of the model. Values are: `simple` = no extra parameters added; `Makeham` = a constant parameter is added to the mortality; and `bathtub` = a Gompertz declining mortality for early ages and a constant parameter are added to the mortality model (see `details`).
`modelFert`	Age-specific fertility model. Options are “`quadratic`” (default), “`PeristeraKostaki`”, “`ColcheroMuller`”, “`Hadwiger`”, “`gamma`”, “`beta`”, “`skewNormal`”, “`gammaMixture`”, “`HadwigerMixture`”, “`skewSymmetric`”, “`skewLogistic`” (see details)
`ageMatur`	Numerical value for the age at sexual maturity.
`checkTheta`	Logical to verify that the `theta` parameters conform with the mortality model's specification (see details).
`checkBeta`	Logical to verify that the `beta` parameters conform with the fertility model's specification (see details).

Details

The function `CalcAgeingRateMort` uses parametric functions to calculate the actuarial (i.e., mortality) rate of ageing. The function follows the conventions from package BaSTA (Colchero and Clark 2012, Colchero *et al.* 2012, Colchero *et al.* 2021) to select the parametric model of mortality. The mortality function describes how the risk of mortality changes with age, and is defined as

$$\mu(x|\theta) = \lim_{\Delta x \to 0} \frac{\Pr[x < X < x + \Delta x | X > x]}{\Delta x},$$

where X is a random variable for ages at death, $x \geq 0$ are ages and θ is the vector of mortality parameters. (For further details on the mortality and survival models see `CalcMort`).

Given a vector of ages $x_1, x_2, \ldots, x_n$ specified by the user with argument `x`, the function calculates ageing rates at age x_i as

$$\frac{d}{dx} \ln[\mu(x)]|_{x=x_i},$$

for $i = 1, 2, \ldots, n$.

Similarly, function `CalcAgeingRatesFert` calculate reproductive ageing rates from parametric models of age-specific fertility, $g(x)$. It uses a numerical approximation to

$$\frac{d}{dx}\ln[g(x)]|_{x=x_i},$$

for $i = 1, 2, \ldots, n$.

Value

The functions output a matrix with the ages at which ageing rates were calculated, the estimated actuarial rate of ageing, and the level of survival at that age.

Author(s)

Fernando Colchero <`fernando_colchero@eva.mpg.de`>

See Also

`CalcSurv` to calculate age-specific survival, `CalcMort` to calculate age-specific mortality, `CalcFert` to calculate age-specific fertility.

`CalcAgeMaxFert` to calculate the age at maximum fertility from parametric models of age-specific fertility. `CalcRemainLifeExp` to calculate remaining life expectancy from parametric models of age-specific mortality.

Examples

```
# Calculate actuarial ageing rate from Gompertz model:
arm <- CalcAgeingRateMort(theta = c(b0 = -5, b1 = 0.1), x = 10)

# Calculate reproductive ageing rate from quadratic model:
arf <- CalcAgeingRateFert(beta = c(b0 = 2, b1 = 0.0025, b2 = 2), x = 10,
                          modelFert = "quadratic")
```

`CalcAgeMaxFert` *Calculating Age at Maximum Fertility and Maximum Fertility.*

Description

`CalcAgeMaxFert` is used to calculate the age at maximum fertility, and the maximum fertility level from parametric models of fertility.

Usage

```
CalcAgeMaxFert (beta, modelFert = "quadratic", ageMatur = 0,
                              maxAge = 100)
```

Arguments

`beta`	Numerical vector of age-specific fertility parameters (see details).
`modelFert`	Age-specific fertility model. Options are "`quadratic`" (default), "`PeristeraKostaki`", "`ColcheroMuller`", "`Hadwiger`", "`gamma`", "`beta`", "`skewNormal`", "`gammaMixture`", "`HadwigerMixture`", "`skewSymmetric`", "`skewLogistic`" (see details)
`ageMatur`	Numerical value for the age at sexual maturity.
`maxAge`	Numerical value for the maximum possible age of reproduction.

Details

For a given function g of age-specific fertility, `CalcAgeMaxFert` uses either an analytical solution to

$$\frac{dg}{dx} = 0$$

or a numerical approximation. For details of the parametric models available see `CalcFert`.

Value

The function outputs a vector with the calculated or estimated age at maximum fertility, the maximum level of fertility, the error in the estimation and the number of iterations (for numerical approximations), and the age at maturity as specified by the user.

Author(s)

Fernando Colchero <`fernando_colchero@eva.mpg.de`>

See Also

`CalcSurv` to calculate age-specific survival, `CalcMort` to calculate age-specific mortality, `CalcFert` to calculate age-specific fertility.

`CalcAgeingRateMort` to calculate ageing rates from parametric models of age-specific mortality. `CalcRemainLifeExp` to calculate remaining life expectancy from parametric models of age-specific mortality.

Examples

```
# Calculate age at maximum fertility from quadratic model:
maxg <- CalcAgeMaxFert(beta = c(b0 = 0.5, b1 = 0.01, b2 = 10))

# Calculate age at maximum fertility from gamma model:
maxg <- CalcAgeMaxFert(beta = c(b0 = 13, b1 = 2, b2 = 0.15),
                     modelFert = "gamma")
```

`CalcDemo` *Calculating Parametric Age-Specific Mortality, Survival, and Fertility*

Description

`CalcDemo` is used to calculate age-specific demographic rates and summary statistics from parametric models of survival and fertility.

Usage

```
CalcDemo(theta = NULL, beta = NULL, x = NULL, dx = NULL,
         model = "GO",  shape = "simple", modelFert = "quadratic",
         type = "both", minSx = 0.01,  summarStats = TRUE,
         ageMatur = 0, maxAge = NULL, agesAR = NULL,
         SxValsAR = NULL)
```

Arguments

`theta`	Numerical vector of age-specific mortality parameters (see details).
`beta`	Numerical vector of age-specific fertility parameters (see details).
`x`	Numerical vector of ages. If `NULL` (default), ages are constructed from age 0 to the age when the survival function reaches 0.0001.
`dx`	Numerical value for age increments. If `NULL`, the default value is 0.01.
`model`	Mortality model, options are "`EX`" = exponential (i.e., constant mortality with age), "`GO`" = Gompertz (default), "`WE`" = Weibull, or "`LO`" = logistic.
`shape`	The overall shape of the mortality model. Values are: "`simple`" = no extra parameters added; "`Makeham`" = a constant parameter is added to the mortality; and "`bathtub`" = a Gompertz declining mortality for early ages and a constant parameter are added to the mortality model (see `details`).
`modelFert`	Age-specific fertility model. Options are "`quadratic`" (default), "`PeristeraKostaki`", "`ColcheroMuller`", "`Hadwiger`", "`gamma`", "`beta`", "`skewNormal`", "`gammaMixture`", "`HadwigerMixture`", "`skewSymmetric`", "`skewLogistic`" (see details)
`type`	Character string specifying the demographic rates to be calculated. Options are "`both`" (default), "`survival`", "`fertility`".
`minSx`	Numerical value specifying the lower bound of the survival function to find the maximum age. The default is 0.01.
`summarStats`	Logical specifying whether summary statistics such as life expectancy, lifespan equality and ageing rates at the specified ages should be calculated.
`ageMatur`	Numerical value specifying the age at sexual maturity, used as the lower bound for the age-specific fertility. Default is 0.
`maxAge`	Numerical values for the maximum age for survival and fertility demographic rates. If `NULL` (default), the maximum age is calculated when the survival function reaches `minSx`.

`agesAR`	Numerical vector of ages at which ageing rates should be calculated. If `NULL` (default) they are calculated at the ages when the survival function reaches 0.5, 0.2, and 0.05.
`SxValsAR`	Alternative to `agesAR`, numerical value for the levels of survival at which ageing rates should be calculated. If `NULL` (default) they are calculated at the ages when the survival function reaches 0.5, 0.2, and 0.05.

Details

1) Age-specific mortality and survival models:

The function `CalcDemo` uses parametric functions to calculate age-specific survival, defined as the probability of surviving to a given age. The function follows the conventions from package BaSTA (Colchero and Clark 2012, Colchero *et al.* 2012, Colchero *et al.* 2021). The mortality function describes how the risk of mortality changes with age, and is defined as

$$\mu(x|\theta) = \lim_{\Delta x \rightarrow 0} \frac{\Pr[x < X < x + \Delta x | X > x]}{\Delta x},$$

where X is a random variable for ages at death, $x \geq 0$ are ages and θ is the vector of mortality parameters. From the mortality function, the survival function is then given by

$$S(x|\boldsymbol{\theta}) = \exp[-\int_0^x \mu(t|\boldsymbol{\theta}) dt].$$

`1.1)Argument` "`model`":

The `model` argument allows the user to choose between four basic mortality functions, namely

(a) `model` = "`EX`": The exponential model (Cox and Oakes 1974), with constant mortality with age, specified as

$$\mu_b(x|\boldsymbol{\theta}) = b,$$

where $b > 0$, with survival

$$S_b(x|\boldsymbol{\theta}) = \exp[-bx].$$

(b) `model` = "`GO`": The Gompertz mortality model (Gompertz 1925, Pletcher 1999), calculated as

$$\mu_b(x|\boldsymbol{\theta}) = \exp(b_0 + b_1 x),$$

where $-\infty < b_0, b_1 < \infty$, with survival

$$S_b(x|\boldsymbol{\theta}) = \exp\left[\frac{e^{b_0}}{b_1}\left(1 - e^{b_1 x}\right)\right].$$

(c) `model` = "`WE`": The Weibull mortality model (Pinder III *et al.* 1978) calculated as

$$\mu_b(x|\boldsymbol{\theta}) = b_0 b_1^{b_0} x^{b_0 - 1},$$

where $b_0, b_1 > 0$, with survival

$$S_b(x|\boldsymbol{\theta}) = \exp\left[-(b_1 x)^{b_0}\right].$$

(d) `model` = “`LO`”: The logistic mortality model (Pletcher 1999), calculated as

$$\mu_b(x|\boldsymbol{\theta}) = \frac{\exp(b_0 + b_1 x)}{1 + b_2 \frac{e^{b_0}}{b_1}\left(e^{b_1 x} - 1\right)},$$

where $b_0, b_1, b_2 > 0$, with survival

$$S_b(x|\boldsymbol{\theta}) = \left[1 + b_2 \frac{e^{b_0}}{b_1}\left(e^{b_1 x} - 1\right)\right]^{-1/b_2}.$$

`1.2)Argument` “shape”:

The `shape` argument allows the user to extend these models in order to explore more complex mortality shapes.

(a) `shape` = “`simple`”: (default) Leaves the model as defined above, with mortality given by

$$\mu(x|\boldsymbol{\theta}) = \mu_b(x|\boldsymbol{\theta})$$

and survival

$$S(x|\boldsymbol{\theta}) = S_b(x|\boldsymbol{\theta}.$$

(b) `shape` = “`Makeham`”: A constant is added to the mortality, such that the mortality is given by

$$\mu(x|\boldsymbol{\theta}) = c + \mu_b(x|\boldsymbol{\theta}_1),$$

where $\boldsymbol{\theta} = [c, \boldsymbol{\theta}_1]$, and with survival

$$S(x|\boldsymbol{\theta}) = e^{-cx} S_b(x|\boldsymbol{\theta}_1)$$

The most common models with this shape is the Gompertz-Makeham model (Gompertz 1825, Makeham 1866).

(c) `shape` = “`bathtub`”: produces a concave shapes in mortality by adding a declining Gompertz term and a constant parameter to the basic mortality model, where the mortality function is

$$\mu(x|\boldsymbol{\theta}) = \exp(a_0 - a_1 x) + c + \mu_b(x|\boldsymbol{\theta}_1),$$

where $a_0 \in \mathbb{R}$, $a_1, c \geq 0$ and $\boldsymbol{\theta}_1 \subset \boldsymbol{\theta}$ are specified based on argument `model`, and with survival

$$S(x|\theta) = \exp\left[\frac{e^{a_0}}{a_1}\left(e^{a_1 x} - 1\right) - cx\right] S_b(x|\theta_1).$$

The most widely use “`bathtub`” shaped model is the **Siler mortality model** (Siler 1979), which provides considerably good fits to mammalian data. The arguments for the Siler model are:

```
CalcDemo(..., model = "GO", shape = "bathtub", ...)
```

2) Specifying theta parameters:

Argument `theta` requires a numerical vector of parameters. For instance, for a Gompertz model with simple shape and with parameter vector $\boldsymbol{\theta}^\top = [b_0, b_1]$ where $b_0 = -5$ and $b_1 = 0.1$, the argument should be specified as

`CalcDemo(theta = c(b0 = -5, b1 = 0.1), ...)`

Note that in this example the parameter names are specified directly, this is required when `checkTheta = FALSE`. Although assigning the names to each parameter is not necessary when `checkTheta = TRUE`, it is advisable to ensure that the right values are assigned to the right parameter.

If argument `checkTheta` is set to `TRUE`, then the vector of `theta` parameters is verified for consistency with the requirements of the model and shape selected.

3) Age-specific fertility models:

The age-specific fertility models correspond to the expected number of offspring produced by adults of a given age. Therefore, for a random variable Y_x with realizations y_x for the number of offspring produced by adults of age x, we have that $E(Y_x) = g(x|\boldsymbol{\beta})$, where $g : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ is a non-negative smooth fertility function and $\boldsymbol{\beta}$ is a vector of parameters to be estimated. The functional forms of function g fall in two main categories, namely polynomial and distributional models.

3.a) Polynomial:

Of the models available in `paramDemo`, the “`quadratic`”, “`PeristeraKostaki`” (Peristera and Kostaki 2007), and “`ColcheroMuller`” (Colchero *et al.* 2021, Muller *et al.* 2020) fall within the polynomial category. Both, the “`PeristeraKostaki`” and “`ColcheroMuller`” are non-symmetric around the age at maximum fertility.

3.b) Distributional:

The distributional models are of the form

$$g(x|\boldsymbol{\beta}) = R f(x|\boldsymbol{\beta}_1),$$

where $f : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ is a probability density function, $R > 0$ is a parameter for the total fertility rate, and $\boldsymbol{\beta}_1 \subset \boldsymbol{\beta}$ is a vector of parameters. The “`Hadwiger`” (Hadwiger 1940), “`gamma`” (Hoem *et al.* 1981), “`beta`” (Hoem *et al.* 1981), “`skewNormal`” (Mazzuco and Scarpa 2011, 2015), “`gammaMixture`” (Hoem *et al.* 1981), “`HadwigerMixture`” (Chandola *et al.* 1991), “`skewSymmetric`” (Mazzuco and Scarpa 2011, 2015), and “`skewLogistic`” (Asili *et al.* 2014) all fall in this category. Notably, the “`gammaMixture`”, “`HadwigerMixture`”, “`skewSymmetric`”, “`skewLogistic`” are appropriate when fertility might be bimodal (Hoem *et al.* 1981, Chandola *et al.* 1999, Mazzuco and Scarpa 2011, 2015, Asili *et al.* 2014).

4) Specifiying beta parameters:

Argument `beta` requires a vector of parameters. For instance, for a quadratic model with parameter vector $\boldsymbol{\beta}^\top = [b_0, b_1, b_2]$ where $b_0 = 0.5$, $b_1 = 0.01$ and $b_2 = 10$, the argument should be specified as

`CalcDemo(..., beta = c(b0 = 0.5, b1 = 0.01, b2 = 10), ...)`

Note that in this example the parameter names are specified directly, this is required when `checkBeta = FALSE`. Although assigning the names to each parameter is not necessary when `checkBeta = TRUE`, it is advisable to ensure that the right values are assigned to the right parameter.

If argument `checkBeta` is set to `TRUE`, then the vector of `beta` parameters is verified for consistency with the requirements of the fertility model selected.

5) Summary statistics:

`5.1) Ageing rates:`

Given a vector of ages $x_1, x_2, \ldots, x_n$ specified by the user with argument `agesAR`, the function calculates ageing rates at age x_i as

$$\frac{d}{dx}\ln[\mu(x)]|_{x=x_i},$$

for $i = 1, 2, \ldots, n$.

`5.2) Ages at different levels of survival:`

The function calculates the ages when the survival function reaches specific values(default at 0.2, 0.1, 0.05), commonly used as population measures of old age. Thus, for a given value $s = S(x)$, the function computes the inverse function

$$x = S^{-1}(s).$$

`5.3) Remaining life expectancy:`

The function calculates the life expectancy at birth as

$$e_0 = \int_0^\infty S(t)dt.$$

`5.4) Measures of inequality and equality:`

The function calculates different measures of inequality and equality in the distribution of ages at death that results from the parametric model:

- `Lifespan inequality:` (Demetrius 1974, Keyfitz and Caswell 2005) given by

$$H = -\frac{\int_0^\infty S(x)\ln[S(x)]dx}{e_0}$$

- `Lifespan equality:` (Colchero *et al.* 2016, Colchero *et al.* 2021) given by

$$\varepsilon = -\ln H.$$

- `Gini coefficient:` (Gini 1912, Shkolnikov *et al.* 2003) given by

$$G = 1 - \frac{1}{e_0}\int_0^\infty [l(x)]^2 dx$$

- `Coefficient of variation:` given by

$$CV = \frac{\sqrt{\sigma^2}}{e_0},$$

where σ^2 is the variance in ages at death.

Value

`CalcDemo` returns an object of class “`paramDemo`” with output consisting of two lists, one for survival and one for fertility. The survival list can be called as `object$surv` while the fertility list is in `object$fert`. Both include the following outputs:

`functs`	data.frame including the follwing columns for survival: `age`, `mort` for the mortality function, `surv` for the survival, `pdf` for the probability density function of ages at death, and `cumhaz` for the cumulative hazards. For fertility it includes columns for `age` and `fert` for the age-specific fertility.
`summStats`	list with elements: `calculated` indicating whether summary statistics were calculated, `ageingRates` with the resulting actuarial ageing rates, and `summStatsMort` with ages at which the survival function reaches different levels (default 0.5, 0.2, 0.05) (e.g., `AgeSx=0.5`), with the life expectancy `lifeExp`, `lifespIneq` for the lifespan inequality (see Demetrius 1974), `lifespEqual` with the lifespan equality as in Colchero *et al.* (2021), `Gini` for the Gini coefficient, and `CoeffVar` for the coefficient of variation in the distribution of ages at death. For fertility, the list includes a logical `calculated` to indicate if summary statistics were calculated, and `summStatsFert` with the age at maximum fertility calculations (see `CalcAgeMaxFert`).
`settings`	list with the details of the parameters and the models used.
`analyzed`	logical indicating whether `surv` or `fert` were calculated.

Author(s)

Fernando Colchero <`fernando_colchero@eva.mpg.de`>

References

Asili S., Rezaei S., Najjar, L. (2014) Using Skew-Logistic Probability Density Function as a Model for Age-Specific Fertility Rate Pattern. *BioMed Research International*, 2014, 790294.

Azzalini, A. (1985) A Class of Distributions Which Includes the Normal Ones. *Scandinavian Journal of Statistics*, 2, 171-178.

Chandola, T., Coleman D.A., Hiorns R.W. (1999) Recent European fertility patterns: Fitting curves to ‘distorted’ distributions. *Population Studies*, 53, 317-329.

Colchero, F. (*In prep.*) Inference on age-specific fertility in ecology and evolution. Learning from other disciplines and improving the state of the art.

Colchero, F., Eckardt, W., Stoinski, T. (2021) Evidence of demographic buffering in an endangered great ape: Social buffering on immature survival and the role of refined sex-age classes on population growth rate. *Journal of Animal Ecology*, 90, 1701-1713.

Colchero, F., J.S. Clark (2012) Bayesian inference on age-specific survival from capture-recapture data for censored and truncated data. *Journal of Animal Ecology*. 81, 139-149.

Colchero, F., O.R. Jones, M. Rebke. (2012) BaSTA: an R package for Bayesian estimation of age-specific survival from incomplete mark-recapture/recovery data with covariates. *Method in Ecology and Evolution*. 3, 466-470.

Colchero, F., *et al.* (2021) The long lives of primates and the “invariant rate of aging” hypothesis. *Nature Communications* 12:3666

Cox, D. R., and Oakes D. (1984) *Analysis of Survival Data*. Chapman and Hall, London.

Demetrius, L. (1974) Demographic parameters and natural selection. *PNAS* 71, 4645-4647.

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. *Philosophical Transactions of the Royal Society of London*, 115, 513-583.

Hadwiger, H. (1940) Eine analytische Reproduktionssunktion fur biologische Gesamtheiten. *Scandinavian Actuarial Journal*, 1940, 101-113.

Hoem, J.M., Madien, D., Nielsen, J.L., Ohlsen, E.M., Hansen, H.O., Rennermalm, B. (1981) Experiments in modelling recent Danish fertility curves. *Demography*, 18, 231-244.

Keyfitz, N., Caswell, H. (2005) Applied Mathematical Demography. (Springer-Verlag).

Makeham, W. M. (1866) On the law of mortality *Journal of the Institute of Actuaries* 13, 1-34.

Mazzuco, S., Scarpa, B. (2011) Fitting age-specific fertility rates by a skew- symmetric probability density function. (Working paper 10), University of Padua.

Mazzuco, S., Scarpa, B. (2015) Fitting age-specific fertility rates by a flexible generalized skew normal probability density function. *Journal of the Royal Statistical Society: Series A*, 178, 187-203.

Muller M. N., Blurton Jones N. G, Colchero F., Thompson M. E., Enigk D. K. (2020) Sexual dimorphism in chimpanzee (*Pan troglodytes schweinfurthii*) and human age-specific fertility. *Journal of human evolution*, 144, 102795.

Peristera P., Kostaki A. (2007) Modeling fertility in modern populations. *Demographic Research*, 16, 141-194.

Pinder III, J.E., Wiener, J.G. and Smith, M.H. (1978) The Weibull distribution: a new method of summarizing survivorship data. *Ecology*, 59, 175-179.

Shkolnikov, V., Andreev, E., Begun, A. Z. (2003) Gini coefficient as a life table function. *Demographic Research* 8, 305-358.

Siler W. A (1979) competing-risk model for animal mortality. *Ecology* 60, 750-757.

See Also

`CalcSurv` to calculate age-specific survival, `CalcMort` to calculate age-specific mortality, `CalcFert` to calculate age-specific fertility.

`CalcAgeingRateMort` to calculate ageing rates from parametric models of age-specific mortality. `CalcRemainLifeExp` to calculate remaining life expectancy from parametric models of age-specific mortality. `CalcAgeMaxFert` to calculate the age at maximum fertility from parametric models of age-specific fertility.

`CalcLifeHist` to calculate life history variables from parametric demographic models.

Examples

```
# Create paramDemo object from Gompertz mortality and
# quadratic fertility:
dem <- CalcDemo(theta = c(b0 = -5, b1 = 0.1),
                 beta = c(b0 = 0.5, b1 = 0.01, b2 = 10),
                 summarStats = TRUE, agesAR = c(5, 10))
```

```
# Plot demographic object:
plot(dem)
```

`CalcDiscrDemo` *Calculating Demographic Rates on Discrete Age Intervals from Parametric Age-Specific Mortality*

Description

`CalcDiscrDemo` is used to calculate survival and fertility functions (i.e., `lx`, `px`, and `qx`) on discrete age intervals from an object of class “`paramDemo`” created with function `CalcDemo`.

Usage

```
CalcDiscrDemo(demo, dx = 1)
```

Arguments

`demo`	Object of class “`paramDemo`” created with function `CalcDemo`
`dx`	Numeric value for the age increments. Default is `dx = 1`

Details

`CalcDiscrDemo` takes the continuous survival functions from an object of class “`paramDemo`” and calculates the basic survival probabilities and, if available, fertility rates on discrete age intervals.

Let $x \geq 0$ and $\omega \in \mathbb{N}$ be the next integer of the age when the survival function $S(x)$ reaches the minimum. Given the age increments, Δx, specified with argument `dx`, the function creates a partition of $[0, \omega]$ such that $[x_i, x_i + \Delta x) \subset [0, \omega]$ for $i = 1, \ldots, n$. At each age interval it calculates the discrete survival, `lx`, as

$$l_{x_i} = S(x_i).$$

The age- or stage-specific survival probability, `px`, is calculated as

$$p_{x_i} = \frac{S(x_i + \Delta x)}{S(x_i)},$$

while the age- or stage-specific mortality probability, `qx`, is $q_{x_i} = 1 - p_{x_i}$.

If fertility was also calculated with function `CalcDemo`, then the function includes the discrete age- or stage-specific fertility, `bx`, as

$$b_{x_i} = b\left(x_i + \frac{\Delta x}{2}\right)\Delta x,$$

where $b(x)$ is the continuous fertility function.

Value

codeCalcDiscrDemo returns an object of class “`discrDemo`” that consist of a matrix with the following columns:

`age`	Partition of the full age subset into discrete intervals
`lx`	Survival or cumulative survival
`px`	The age- or stage-specific survival probability
`qx`	The age- or stage-specific mortality probability
`bx`	If available, the age- or stage-specific fertility

Author(s)

Fernando Colchero <`fernando_colchero@eva.mpg.de`>

See Also

`CalcDemo` to create an object of class “`paramDemo`”.

Examples

```
# Create paramDemo object from Gompertz mortality and
# quadratic fertility:
dem <- CalcDemo(theta = c(b0 = -5, b1 = 0.1),
                beta = c(b0 = 0.5, b1 = 0.01, b2 = 10),
                summarStats = TRUE, agesAR = c(5, 10))

# Create discrete demographic object:
demDisc <- CalcDiscrDemo(dem, dx = 1)
```

`CalcFert` *Calculating Parametric Age-Specific Fertility.*

Description

`CalcFert` is used to calculate age-specific fertility from different parametric models.

Usage

```
CalcFert(beta, x, modelFert = "quadratic", checkBeta = TRUE)
```

Arguments

`beta`	Numerical vector of age-specific fertility parameters (see details).
`x`	Numerical vector of ages at which to calculate fertility.
`modelFert`	Age-specific fertility model. Options are "`quadratic`" (default), "`PeristeraKostaki`", "`ColcheroMuller`", "`Hadwiger`", "`gamma`", "`beta`", "`skewNormal`", "`gammaMixture`", "`HadwigerMixture`", "`skewSymmetric`", "`skewLogistic`" (see details)
`checkBeta`	Logical to verify that the `beta` parameters conform with the model's specification (see details).

Details

1) Age-specific fertility models:

The age-specific fertility models correspond to the expected number of offspring produced by adults of a given age. Therefore, for a random variable Y_x with realizations y_x for the number of offspring produced by adults of age x, we have that $E(Y_x) = g(x|\beta)$, where $g : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ is a non-negative smooth fertility function and β is a vector of parameters to be estimated. The functional forms of function g fall in two main categories, namely polynomial and distributional models.

2.a) Polynomial:

Of the models available in `paramDemo`, the "`quadratic`", "`PeristeraKostaki`" (Peristera and Kostaki 2007), and "`ColcheroMuller`" (Colchero *et al.* 2021, Muller *et al.* 2020) fall within the polynomial category. Both, the "`PeristeraKostaki`" and "`ColcheroMuller`" are non-symmetric around the age at maximum fertility.

2.b) Distributional:

The distributional models are of the form

$$g(x|\beta) = Rf(x|\beta_1),$$

where $f : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ is a probability density function, $R > 0$ is a parameter for the total fertility rate, and $\beta_1 \in \beta$ is a vector of parameters. The "`Hadwiger`" (Hadwiger 1940), "`gamma`" (Hoem *et al.* 1981), "`beta`" (Hoem *et al.* 1981), "`skewNormal`" (Mazzuco and Scarpa 2011, 2015), "`gammaMixture`" (Hoem *et al.* 1981), "`HadwigerMixture`" (Chandola *et al.* 1991), "`skewSymmetric`" (Mazzuco and Scarpa 2011, 2015), and "`skewLogistic`" (Asili *et al.* 2014) all fall in this category. Notably, the "`gammaMixture`", "`HadwigerMixture`", "`skewSymmetric`", "`skewLogistic`" are appropriate when fertility might be bimodal (Hoem *et al.* 1981, Chandola *et al.* 1999, Mazzuco and Scarpa 2011, 2015, Asili *et al.* 2014).

2) Specifying beta parameters:

Argument `beta` requires a vector of parameters. For instance, for a quadratic model with parameter vector $\beta^\top = [0.5, 0.01, 10]$, the argument should be specified as `theta = c(b0 = 0.5, b1 = 0.01, b2 = 10)`. Note that in this example the parameter names are specified directly, this is required when `checkBeta = FALSE`. Although assigning the names to each parameter is not necessary when `checkBeta = TRUE`, it is advisable to ensure that the right values are assigned to the right parameter.

If argument `checkBeta` is set to `TRUE`, then the vector of `beta` parameters is verified for consistency with the requirements of the model and shape selected.

Author(s)

Fernando Colchero <`fernando_colchero@eva.mpg.de`>

References

Asili, S., Rezaei, S. & Najjar, L. (2014) Using Skew-Logistic Probability Density Function as a Model for Age-Specific Fertility Rate Pattern. *BioMed Research International*, 2014, 790294.

Azzalini, A. (1985) A Class of Distributions Which Includes the Normal Ones. *Scandinavian Journal of Statistics*, 2, 171-178.

Chandola, T., Coleman, D.A. & Hiorns, R.W. (1999) Recent European fertility patterns: Fitting curves to 'distorted' distributions. *Population Studies*, 53, 317-329.

Colchero, F. (*In prep.*) Inference on age-specific fertility in ecology and evolution. Learning from other disciplines and improving the state of the art.

Colchero, F., Eckardt, W. & Stoinski, T. (2021) Evidence of demographic buffering in an endangered great ape: Social buffering on immature survival and the role of refined sex-age classes on population growth rate. *Journal of Animal Ecology*, 90, 1701-1713.

Hadwiger, H. (1940) Eine analytische Reproduktionssunktion fur biologische Gesamtheiten. *Scandinavian Actuarial Journal*, 1940, 101-113.

Hoem, J.M., Madien, D., Nielsen, J.L., Ohlsen, E.M., Hansen, H.O. & Rennermalm, B. (1981) Experiments in modelling recent Danish fertility curves. *Demography*, 18, 231-244.

Mazzuco, S. & Scarpa, B. (2011) Fitting age-specific fertility rates by a skew- symmetric probability density function. (Working paper 10), University of Padua.

Mazzuco, S. & Scarpa, B. (2015) Fitting age-specific fertility rates by a flexible generalized skew normal probability density function. *Journal of the Royal Statistical Society: Series A*, 178, 187-203.

Muller, M. N., Blurton Jones, N. G, Colchero, F., Thompson, M. E., Enigk, D. K. (2020) Sexual dimorphism in chimpanzee (*Pan troglodytes schweinfurthii*) and human age-specific fertility. *Journal of human evolution*, 144, 102795.

Peristera, P. & Kostaki, A. (2007) Modeling fertility in modern populations. *Demographic Research*, 16, 141-194.

See Also

`CalcMort` to calculate age-specific mortality, `CalcSurv` to calculate age-specific survival

Examples

```
# Age specific fertility based on quadratic model (default):
fert <- CalcFert(beta = c(b0 = 0.5, b1 = 0.01, b2 = 10), x = 10)

# Age specific fertility based on gamma model:
fert <- CalcFert(beta = c(b0 = 13, b1 = 2, b2 = 0.15), x = 10,
                 modelFert = "gamma")
```

`CalcLifeHist` *Calculating life history variables from parametric models of age-specific demographic rates.*

Description

`CalcLifeHist` uses parametric models of survival and fertility to calculate age-specific demographic rates, stable age structure, reproductive values, sensitivity and elasticities of the stable population growth rate to demographic rates, as well as different life history variables such as generation time or population entropy.

Usage

```
CalcLifeHist(theta = NULL, beta = NULL, dx = NULL,
             model = "GO", shape = "simple",
             modelFert = "quadratic", ageMatur = 0,
             maxAge = NULL, lambdaMethod = "matrix")
```

Arguments

`theta`	Numerical vector of age-specific mortality parameters (see details).
`beta`	Numerical vector of age-specific fertility parameters (see details).
`dx`	Numerical value for the age-interval increments, default is 1.
`model`	Mortality model, options are “`EX`” = exponential (i.e., constant mortality with age), “`GO`” = Gompertz (default), “`WE`” = Weibull, or “`LO`” = logistic.
`shape`	The overall shape of the mortality model. Values are: “`simple`” = no extra parameters added; “`Makeham`” = a constant parameter is added to the mortality; and “`bathtub`” = a Gompertz declining mortality for early ages and a constant parameter are added to the mortality model (see `details`).
`modelFert`	Age-specific fertility model. Options are “`quadratic`” (default), “`PeristeraKostaki`”, “`ColcheroMuller`”, “`Hadwiger`”, “`gamma`”, “`beta`”, “`skewNormal`”, “`gammaMixture`”, “`HadwigerMixture`”, “`skewSymmetric`”, “`skewLogistic`” (see details)
`ageMatur`	Numerical value for the age at maturity.
`maxAge`	Numerical value for the maximum age. If `NULL` (default), it is calculated from the survival (see details).
`lambdaMethod`	Character string specifying the method to calculate the intrinsic population growth rate. Values are “`matrix`” for methods based on population projection matrices, or “`Lotka`” for numerical approximation of the root of the Lotka equation (see details)

Details

1) Age-specific mortality and survival models:

The function `CalcDemo` uses parametric functions to calculate age-specific survival, defined as the probability of surviving to a given age. The function follows the conventions from package BaSTA

(Colchero and Clark 2012, Colchero *et al.* 2012, Colchero *et al.* 2021). The mortality function describes how the risk of mortality changes with age, and is defined as

$$\mu(x|\theta) = \lim_{\Delta x \to 0} \frac{\Pr[x < X < x + \Delta x | X > x]}{\Delta x},$$

where X is a random variable for ages at death, $x \geq 0$ are ages and θ is the vector of mortality parameters. From the mortality function, the survival function is then given by

$$S(x|\boldsymbol{\theta}) = \exp[-\int_0^x \mu(t|\boldsymbol{\theta})dt].$$

`1.1)Argument` “`model`”:

The `model` argument allows the user to choose between four basic mortality functions, namely

(a) `model` = “`EX`”: The exponential model (Cox and Oakes 1974), with constant mortality with age, specified as

$$\mu_b(x|\boldsymbol{\theta}) = b,$$

where $b > 0$, with survival

$$S_b(x|\boldsymbol{\theta}) = \exp[-bx].$$

(b) `model` = “`GO`”: The Gompertz mortality model (Gompertz 1925, Pletcher 1999), calculated as

$$\mu_b(x|\boldsymbol{\theta}) = \exp(b_0 + b_1 x),$$

where $-\infty < b_0, b_1 < \infty$, with survival

$$S_b(x|\boldsymbol{\theta}) = \exp\left[\frac{e^{b_0}}{b_1}\left(1 - e^{b_1 x}\right)\right].$$

(c) `model` = “`WE`”: The Weibull mortality model (Pinder III *et al.* 1978) calculated as

$$\mu_b(x|\boldsymbol{\theta}) = b_0 b_1^{b_0} x^{b_0 - 1},$$

where $b_0, b_1 > 0$, with survival

$$S_b(x|\boldsymbol{\theta}) = \exp\left[-(b_1 x)^{b_0}\right].$$

(d) `model` = “`LO`”: The logistic mortality model (Pletcher 1999), calculated as

$$\mu_b(x|\boldsymbol{\theta}) = \frac{\exp(b_0 + b_1 x)}{1 + b_2 \frac{e^{b_0}}{b_1}\left(e^{b_1 x} - 1\right)},$$

where $b_0, b_1, b_2 > 0$, with survival

$$S_b(x|\boldsymbol{\theta}) = \left[1 + b_2 \frac{e^{b_0}}{b_1}\left(e^{b_1 x} - 1\right)\right]^{-1/b_2}.$$

`1.2)Argument` “shape”:

The `shape` argument allows the user to extend these models in order to explore more complex mortality shapes.

(a) `shape` = “`simple`”: (default) Leaves the model as defined above, with mortality given by

$$\mu(x|\boldsymbol{\theta}) = \mu_b(x|\boldsymbol{\theta})$$

and survival

$$S(x|\boldsymbol{\theta}) = S_b(x|\boldsymbol{\theta}.$$

(b) `shape` = “`Makeham`”: A constant is added to the mortality, such that the mortality is given by

$$\mu(x|\boldsymbol{\theta}) = c + \mu_b(x|\boldsymbol{\theta}_1),$$

where $\boldsymbol{\theta} = [c, \boldsymbol{\theta}_1]$, and with survival

$$S(x|\boldsymbol{\theta}) = e^{-cx} S_b(x|\boldsymbol{\theta}_1)$$

The most common models with this shape is the Gompertz-Makeham model (Gompertz 1825, Makeham 1866).

(c) `shape` = “`bathtub`”: produces a concave shapes in mortality by adding a declining Gompertz term and a constant parameter to the basic mortality model, where the mortality function is

$$\mu(x|\boldsymbol{\theta}) = \exp(a_0 - a_1 x) + c + \mu_b(x|\boldsymbol{\theta}_1),$$

where $a_0 \in \mathbb{R}$, $a_1, c \geq 0$ and $\boldsymbol{\theta}_1 \subset \boldsymbol{\theta}$ are specified based on argument `model`, and with survival

$$S(x|\theta) = \exp\left[\frac{e^{a_0}}{a_1}\left(e^{a_1 x} - 1\right) - cx\right] S_b(x|\theta_1).$$

The most widely use “`bathtub`” shaped model is the **Siler mortality model** (Siler 1979), which provides considerably good fits to mammalian data. The arguments for the Siler model are:

```
CalcLifeHist(..., model = "GO", shape = "bathtub", ...)
```

2) Specifying theta parameters:

Argument `theta` requires a numerical vector of parameters. For instance, for a Gompertz model with simple shape and with parameter vector $\boldsymbol{\theta}^\top = [b_0, b_1]$ where $b_0 = -5$ and $b_1 = 0.1$, the argument should be specified as

`CalcDemo(theta = c(b0 = -5, b1 = 0.1), ...)`

Note that in this example the parameter names are specified directly, this is required when `checkTheta = FALSE`. Although assigning the names to each parameter is not necessary when `checkTheta = TRUE`, it is advisable to ensure that the right values are assigned to the right parameter.

If argument `checkTheta` is set to `TRUE`, then the vector of `theta` parameters is verified for consistency with the requirements of the model and shape selected.

3) Age-specific fertility models:

The age-specific fertility models correspond to the expected number of offspring produced by adults of a given age. Therefore, for a random variable Y_x with realizations y_x for the number of offspring produced by adults of age x, we have that $E(Y_x) = g(x|\boldsymbol{\beta})$, where $g : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a non-negative smooth fertility function and $\boldsymbol{\beta}$ is a vector of parameters to be estimated. The functional forms of function g fall in two main categories, namely polynomial and distributional models.

3.a) Polynomial:

Of the models available in `paramDemo`, the "`quadratic`", "`PeristeraKostaki`" (Peristera and Kostaki 2007), and "`ColcheroMuller`" (Colchero *et al.* 2021, Muller *et al.* 2020) fall within the polynomial category. Both, the "`PeristeraKostaki`" and "`ColcheroMuller`" are non-symmetric around the age at maximum fertility.

3.b) Distributional:

The distributional models are of the form

$$g(x|\boldsymbol{\beta}) = Rf(x|\boldsymbol{\beta}_1),$$

where $f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a probability density function, $R > 0$ is a parameter for the total fertility rate, and $\boldsymbol{\beta}_1 \subset \boldsymbol{\beta}$ is a vector of parameters. The "`Hadwiger`" (Hadwiger 1940), "`gamma`" (Hoem *et al.* 1981), "`beta`" (Hoem *et al.* 1981), "`skewNormal`" (Mazzuco and Scarpa 2011, 2015), "`gammaMixture`" (Hoem *et al.* 1981), "`HadwigerMixture`" (Chandola *et al.* 1991), "`skewSymmetric`" (Mazzuco and Scarpa 2011, 2015), and "`skewLogistic`" (Asili *et al.* 2014) all fall in this category. Notably, the "`gammaMixture`", "`HadwigerMixture`", "`skewSymmetric`", "`skewLogistic`" are appropriate when fertility might be bimodal (Hoem *et al.* 1981, Chandola *et al.* 1999, Mazzuco and Scarpa 2011, 2015, Asili *et al.* 2014).

4) Specifiying beta parameters:

Argument `beta` requires a vector of parameters. For instance, for a quadratic model with parameter vector $\boldsymbol{\beta}^\top = [b_0, b_1, b_2]$ where $b_0 = 0.5$, $b_1 = 0.01$ and $b_2 = 10$, the argument should be specified as

```
CalcLifeHist(..., beta = c(b0 = 0.5, b1 = 0.01, b2 = 10), ...)
```

Note that in this example the parameter names are specified directly, this is required when `checkBeta = FALSE`. Although assigning the names to each parameter is not necessary when `checkBeta = TRUE`, it is advisable to ensure that the right values are assigned to the right parameter.

If argument `checkBeta` is set to `TRUE`, then the vector of `beta` parameters is verified for consistency with the requirements of the fertility model selected.

5) Methods to estimate lambda, r, stable age structure and reproductive values:

If argument `lambdaMethod` is set to "`matrix`", the function calculates the assymptotic population growth rate λ as the dominant eigen-value of the resulting population projection matrix $\boldsymbol{A}$ (Caswell and Wegner 1978, Kroon *et al.* 1986, Caswell 2001). The intrinsing population growth rate is then $r = \log \lambda$, while the stable age-structure, $\boldsymbol{\omega}^\top = [\omega_0, \omega_1, \ldots, \omega_\tau]$, where τ is the maximum age of the population, is given by the corresponding right eigen vector, standardized by its norm. The reproductive value is the corresponding left eigen vector, $\boldsymbol{\nu}^\top = [\nu_0, \nu_1, \ldots, \nu_\tau]$, standardized such that $\nu_0 = 1$.

If argument `lambdaMethod` is set to "`Lotka`", the function estimates the intrinsic population growth rate, r, by finding the real root of Lotka' (1913) equation, given by

$$\int_0^\tau e^{-rx} S(x) m(x) dx = 1.$$

The stable population structure is then calculated as

$$\omega_x = \frac{e^{-rx}S(x)}{\sum_{k=0}^{\tau} e^{-rk}S(k)},$$

while the reproductive value is given by

$$\nu_x = \frac{\int_x^{\tau} e^{-r(k-x)}S(k)m(k)dk}{S(x)}.$$

The function then calculates the sensitivities of lambda to age-specific survival as

$$s_{p,x} = \frac{\nu_x \omega_{x+1}}{<\boldsymbol{\nu\omega}>}, \quad \text{for } x < \tau$$

where the denominator is the inner product, and the sensitivity of lambda to the reproductive value at x is given by

$$s_{m,x} = \frac{\nu_1 \omega_{x+1}}{<\boldsymbol{\nu\omega}>}, \quad \text{for } x < \tau$$

Elasticities are calculated as

$$e_{p,x} = \frac{p_x}{\lambda} s_{p,x} \text{ and } e_{m,x} = \frac{m_x}{\lambda} s_{m,x}.$$

6) Calculation of life history variables

The first life history variables are the intrinsic population growth rate r and the stable population growth rate $\lambda = e^r$, described in the previous section.

The function also provides a number of life history variables described by Tuljapurkar *et al.* (2009). For instance, it provides the generation time for the stable population as

$$T_s = \int_0^{\tau} xe^{-rx}S(x)m(x)dx,$$

as well as the cohort generation time, given by

$$T_c = \frac{\int_0^{\tau} xS(x)m(x)dx}{R_0},$$

where R_0 is the net reproductive rate, given by

$$R_0 = \int_0^{\tau} S(x)m(x)dx.$$

It provides the demographic dispersion given by

$$\sigma_d^2 = \frac{\int_0^{\tau} (x - T_c)^2 S(x)m(x)dx}{R_0}$$

which measures the dispersion of reproduction across ages.

Finally, it calculates the population entropy as proposed by Demetrius (1974), given by

$$H = -\frac{\int_0^{\tau} \rho(x)\log\rho(x)dx}{T_s}$$

where $\rho(x) = e^{-rx}S(x)m(x)$.

Value

The function produces an object of class "`PDlifeHist`", which consists of a list with the following elements:

`demoTab`	A data frame with the resulting individual ages, the survival $S(x)$, the age-specific survival probability p_x, the age-specific mortality probability q_x, the age-specific fertility rate m_x, the stable age structure $omega_x$, the age specific reproductive value nu_x, and the sensitivities and elasticities.
`lifeHist`	A data frame with the life history variables described in details, including a description.
`settings`	A list with the information of the models and parameters as provided by the user.

Author(s)

Fernando Colchero `<fernando_colchero@eva.mpg.de>`

References

Asili S., Rezaei S., Najjar, L. (2014) Using Skew-Logistic Probability Density Function as a Model for Age-Specific Fertility Rate Pattern. *BioMed Research International*, 2014, 790294.

Azzalini, A. (1985) A Class of Distributions Which Includes the Normal Ones. *Scandinavian Journal of Statistics*, 2, 171-178.

Caswell, H. & Werner, P. A. (1978) Transient Behavior and Life History Analysis of Teasel (Dipsacus Sylvestris Huds.). Ecology 59, 53-66.

Caswell, H. (2001) Matrix Population Models: Construction, Analysis and Interpretation. (Sinauer Associates).

Chandola, T., Coleman D.A., Hiorns R.W. (1999) Recent European fertility patterns: Fitting curves to 'distorted' distributions. *Population Studies*, 53, 317-329.

Colchero, F. (*In prep.*) Inference on age-specific fertility in ecology and evolution. Learning from other disciplines and improving the state of the art.

Colchero, F., Eckardt, W., Stoinski, T. (2021) Evidence of demographic buffering in an endangered great ape: Social buffering on immature survival and the role of refined sex-age classes on population growth rate. *Journal of Animal Ecology*, 90, 1701-1713.

Colchero, F., J.S. Clark (2012) Bayesian inference on age-specific survival from capture-recapture data for censored and truncated data. *Journal of Animal Ecology*. 81, 139-149.

Colchero, F., O.R. Jones, M. Rebke. (2012) BaSTA: an R package for Bayesian estimation of age-specific survival from incomplete mark-recapture/recovery data with covariates. *Method in Ecology and Evolution*. 3, 466-470.

Colchero, F., *et al.* (2021) The long lives of primates and the "invariant rate of aging" hypothesis. *Nature Communications* 12:3666

Cox, D. R., and Oakes D. (1984) *Analysis of Survival Data*. Chapman and Hall, London.

Demetrius, L. (1974) Demographic parameters and natural selection. *PNAS* 71, 4645-4647.

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. *Philosophical Transactions of the Royal Society of London*, 115, 513-583.

Hadwiger, H. (1940) Eine analytische Reproduktionssunktion fur biologische Gesamtheiten. *Scandinavian Actuarial Journal*, 1940, 101-113.

Hoem, J.M., Madien, D., Nielsen, J.L., Ohlsen, E.M., Hansen, H.O., Rennermalm, B. (1981) Experiments in modelling recent Danish fertility curves. *Demography*, 18, 231-244.

Keyfitz, N., Caswell, H. (2005) Applied Mathematical Demography. (Springer-Verlag).

Kroon, H. de, Plaisier, A., Groenendael, J. van & Ecology, H. C. (1986) Elasticity: the relative contribution of demographic parameters to population growth rate. Ecology 67, 1427-1431.

Lotka, A. J. (1913) A natural population norm. II. Journal of the Washington Academy of Sciences 3, 289-293.

Makeham, W. M. (1866) On the law of mortality *Journal of the Institute of Actuaries* 13, 1-34.

Mazzuco, S., Scarpa, B. (2011) Fitting age-specific fertility rates by a skew- symmetric probability density function. (Working paper 10), University of Padua.

Mazzuco, S., Scarpa, B. (2015) Fitting age-specific fertility rates by a flexible generalized skew normal probability density function. *Journal of the Royal Statistical Society: Series A*, 178, 187-203.

Muller M. N., Blurton Jones N. G, Colchero F., Thompson M. E., Enigk D. K. (2020) Sexual dimorphism in chimpanzee (*Pan troglodytes schweinfurthii*) and human age-specific fertility. *Journal of human evolution*, 144, 102795.

Peristera P., Kostaki A. (2007) Modeling fertility in modern populations. *Demographic Research*, 16, 141-194.

Pinder III, J.E., Wiener, J.G. and Smith, M.H. (1978) The Weibull distribution: a new method of summarizing survivorship data. *Ecology*, 59, 175-179.

Shkolnikov, V., Andreev, E., Begun, A. Z. (2003) Gini coefficient as a life table function. *Demographic Research* 8, 305-358.

Siler W. A (1979) competing-risk model for animal mortality. *Ecology* 60, 750-757.

See Also

`CalcDemo` to calculate parametric demographic functions.

Examples

```
# Calculate life histories on the defalult models:
test <- CalcLifeHist(theta = c(b0 = -3, b1 = 0.1),
                     beta = c(b0 = 1, b1 = 0.01, b2 = 5),
                     ageMatur = 2)

# Print results:
test

# Plot results:
plot(test, type = "all")
```

`CalcLifeTable` *Calculating a Life Table from Data.*

Description

`CalcLifeTable` uses non-parametric methods to calculate life tables and confidence intervals.

Usage

```
CalcLifeTable(ageLast, ageFirst = NULL, departType, dx = 1,
                         calcCIs = FALSE, nboot = 1000, alpha = 0.05)
```

Arguments

`ageLast`	Numerical vector with the ages at last detection (i.e., death and censoring) (see `details`)
`ageFirst`	Numerical vector of ages at first detection (i.e., truncation). If `NULL` then all values are set to 0 (see `details`).
`departType`	Character string vector for the type of departure (i.e., last detection), with values “`D`” for death and “`C`” for censoring (see `details`).
`dx`	Age interval size, default set at 1 (see `details`)
`calcCIs`	Logical indicating whether confidence intervals should be calculated
`nboot`	Number of bootstrap iterations
`alpha`	Alpha level. Default is 0.05 for 95% CIs

Details

1) Data structure:

`CalcLifeTable` allows to construct life tables for data that includes the following types of records:

- **Uncensored**: individuals with known ages at death;
- **right-censored**: individuals last seen alive;
- **left-truncated**: individuals born before the start of the study and are truncated at the age of entry.

The data required are the ages at last detection (i.e., *uncensored* or *right-censored*) passed through argument “`ageLast`”, the type of departure via argument “`departType`”, which takes two values, namely “`D`” for death, and “`C`” for censored (i.e., right-censored).

In addition, if there is *left-truncation*, it takes the ages at entry to the study by means of argument “`ageFirst`”. If all the individuals were born during the study, the value of “`ageFirst`” can be left as `NULL`, which will make them all equal to 0.

2) Computing life tables

To calculate life tables, the function uses conventional formal demgraphic methods as depicted by Preston *et al.* (2001). Argument “`ageLast`” provides a vector of ages at last detection, $\boldsymbol{x}^\top =$

$[x_1, x_2, \ldots, x_n]$, while argument "`ageFirst`" provides a vector of ages at first detection $\boldsymbol{y}^\top = [y_1, y_2, \ldots, y_n]$. From argument "`departType`" the function produces an indicator vector for censoring $\boldsymbol{v} = \{v_i\}_{i \in \mathbb{N}_n}$ where $v_i = 1$ if individual i is censored and 0 otherwise.

The function creates a partition of the interval of ages between 0 and $\max(x)$, for age intervals $[x, x + \Delta x)$ where Δx is specified by argument "`dx`". As default `dx = 1`. At each age interval, the function calculates the following variables:

- `Nx`: which corresponds to number of individuals that entered the interval, but considering the proportion of time they were present within the interval as a function of left-truncation. It is given by

  $$N_x = \sum_{i \in I_x} \lambda_{i,x},$$

  where I_x is the subset of individuals recorded within the interval, and $\lambda_{i,x}$ is the proportion of time during the age interval each individual was present in the study. For individuals that entered the study before x then $\lambda_{i,x} = 1$, while for those that were truncated within the interval $\lambda_{i,x} = (x + \Delta x - y_i)/\Delta x$.
- `Dx`: the number of individuals dying in the interval.
- `qx`: the age-specific mortality probability, calculated as $q_x = D_x/N_x$.
- `px`: the age-specific survival probability, given by $p_x = 1 - q_x$.
- `lx`: the life table survival calculated as

  $$l_x = \prod_{j=0}^{x-1} p_j$$

  where $l_0 = 1$.
- `ax`: Proportion of the interval lived by those that died in the interval, given by

  $$a_x = \frac{\sum_{i \in J_x} \delta_{i,x}}{D_x}$$

  where J_x is the subset of individuals that died within the interval, and $\delta_{i,x}$ is the proportion lived by those individuals from the start of the interval to their deaths, this is $\delta_{i,x} = (x_i - x)/\Delta x$.
- `Lx`: The number of individual years lived within the interval, given by

  $$L_x = l_x(1 - a_x q_x)$$
- `Tx`: The total number of individual years lived after age x, given by $T_x = \sum_{j=x}^{\infty} L_j \Delta x$
- `ex`: the remaining life expectancy at the beginning of each age interval, calculated as $e_x = T_x/l_x$.

3) Calculating confidence intervals If argument "`calcCIs`" is set to `TRUE`, the function uses a non-parametric bootstrap by sampling with replacement the data. Argument `nboot` specifies the number of bootstrap steps, with default `nboot = 2000`. From each re-sampled dataset, it uses function `CalcLifeTable` to construct the corresponding life table and stores the values of l_x, q_x, p_x, and e_x from each iteration. From these, it calculates quantiles at the given alpha level.

Value

`CalcLifeTable` returns an object of class “`paramDemoLT`” with output consisting of a list with the life table and, if indicated by the user, with the confidence interval information. The life table in matrix format includes the following columns:

`Ages`	Ages with increments given by `dx`
`Nx`	Number of individuals entering the interval, does not need to be an integer since it considers truncation
`Dx`	Number of individuals that died within the age interval
`lx`	Survival (i.e., cumulative survival)
`px`	Age-specific survival probability
`qx`	Age-specific mortality probability
`Lx`	Number of individual years lived within the interval
`Tx`	Number of individual years lived after age x
`ex`	Remaining life expectancy at each age

If argument `calcCIs` = `TRUE`, the function also returns a list containing the following components:

`lx`	Matrix with Ages and mean and upper and lower CIs for the life table survival
`qx`	Matrix with Ages and mean and upper and lower CIs for the age-specific mortality probability
`px`	Matrix with Ages and mean and upper and lower CIs for the age-specific survival probability
`ex`	Matrix with Ages and mean and upper and lower CIs for the remaining life expectancy
`Settings`	Numerical vector including whether CIs were calculated, the number of bootstrap iterations, “`nboot`” and the alpha level “`alpha`”

Author(s)

Fernando Colchero <`fernando_colchero@eva.mpg.de`>

References

Preston, S.H., Heuveline, P. and Guillot, M. (2001) Demography: Measuring and Modeling Population Processes. Blackwell, Oxford.

See Also

`CalcProductLimitEst` to calculate product limit estimators.

Examples

```
# Simulate age at death data from Gompertz model:
ages <- SampleRandAge(n = 100, theta = c(b0 = -5, b1 = 0.1))

# Calculate life table:
lt <- CalcLifeTable(ageLast = ages, departType = rep("D", 100))

# Calculate life table with 95% CIs:
ltCIs <- CalcLifeTable(ageLast = ages, departType = rep("D", 100),
                       calcCIs = TRUE, nboot = 100)
```

`CalcMort` *Calculating Parametric Age-Specific Mortality and survival.*

Description

`CalcMort` and `CalcSurv` are used to calculate the age-specific mortality and survival functions from different parametric models.

Usage

```
CalcMort(theta, x, model = "GO", shape = "simple",
          checkTheta = TRUE)

CalcSurv(theta, x, model = "GO", shape = "simple",
          checkTheta = TRUE)
```

Arguments

`theta`	Numerical vector of age-specific mortality parameters (see details).
`x`	Numerical vector of ages at which to calculate mortality.
`model`	The underlying mortality model to be used. `"EX"` = exponential,`"GO"` = Gompertz, `"WE"` = Weibull and `"LO"` = logistic (see `details`).
`shape`	The overall shape of the model. Values are: `simple` = no extra parameters added; `Makeham` = a constant parameter is added to the mortality; and `bathtub` = a Gompertz declining mortality for early ages and a constant parameter are added to the mortality model (see `details`).
`checkTheta`	Logical to verify that the `beta` parameters conform with the model's specification (see details).

Details

1) Age-specific mortality and survival models:

The function `CalcMort` uses parametric functions to calculate age-specific survival, defined as the probability of surviving to a given age. The function follows the conventions from package BaSTA (Colchero and Clark 2012, Colchero *et al.* 2012, Colchero *et al.* 2021). The mortality function or hazard rate describes how the risk of mortality changes with age, and is defined as

$$\mu(x|\theta) = \lim_{\Delta x \to 0} \frac{\Pr[x < X < x + \Delta x | X > x]}{\Delta x},$$

where X is a random variable for ages at death, $x \geq 0$ are ages and θ is the vector of mortality parameters. From the mortality function, the survival function is then given by

$$S(x|\boldsymbol{\theta}) = \Pr[X > x] = \exp[-\int_0^x \mu(t|\boldsymbol{\theta})dt],$$

with the cumulative distribution function $F(x|\boldsymbol{\theta}) = 1 - S(x|\boldsymbol{\theta})$. The probability density function of ages at death is

$$f(x|\boldsymbol{\theta}) = \mu(x|\boldsymbol{\theta})S(x|\boldsymbol{\theta}),$$

for $x \geq 0$.

`1.1)Argument` “`model`”:

The `model` argument allows the user to choose between four basic mortality functions, namely

(a) `model` = “`EX`”: The exponential model (Cox and Oakes 1974), with constant mortality with age, specified as

$$\mu_b(x|\boldsymbol{\theta}) = b,$$

where $b > 0$, with survival

$$S_b(x|\boldsymbol{\theta}) = \exp[-bx].$$

(b) `model` = “`GO`”: The Gompertz mortality model (Gompertz 1925, Pletcher 1999), calculated as

$$\mu_b(x|\boldsymbol{\theta}) = \exp(b_0 + b_1 x),$$

where $-\infty < b_0, b_1 < \infty$, with survival

$$S_b(x|\boldsymbol{\theta}) = \exp\left[\frac{e^{b_0}}{b_1}\left(1 - e^{b_1 x}\right)\right].$$

(c) `model` = “`WE`”: The Weibull mortality model (Pinder III *et al.* 1978) calculated as

$$\mu_b(x|\boldsymbol{\theta}) = b_0 b_1^{b_0} x^{b_0 - 1},$$

where $b_0, b_1 > 0$, with survival

$$S_b(x|\boldsymbol{\theta}) = \exp\left[-(b_1 x)^{b_0}\right].$$

(d) `model` = “`LO`”: The logistic mortality model (Pletcher 1999), calculated as

$$\mu_b(x|\boldsymbol{\theta}) = \frac{\exp(b_0 + b_1 x)}{1 + b_2 \frac{e^{b_0}}{b_1}\left(e^{b_1 x} - 1\right)},$$

where $b_0, b_1, b_2 > 0$, with survival

$$S_b(x|\boldsymbol{\theta}) = \left[1 + b_2 \frac{e^{b_0}}{b_1}\left(e^{b_1 x} - 1\right)\right]^{-1/b_2}.$$

`1.2)Argument` “shape”:

The `shape` argument allows the user to extend these models in order to explore more complex mortality shapes.

(a) `shape` = “`simple`”: (default) Leaves the model as defined above, with mortality given by

$$\mu(x|\boldsymbol{\theta}) = \mu_b(x|\boldsymbol{\theta})$$

and survival

$$S(x|\boldsymbol{\theta}) = S_b(x|\boldsymbol{\theta}.$$

(b) `shape` = “`Makeham`”: A constant is added to the mortality, such that the mortality is given by

$$\mu(x|\boldsymbol{\theta}) = c + \mu_b(x|\boldsymbol{\theta}_1),$$

where $\boldsymbol{\theta} = [c, \boldsymbol{\theta}_1]$, and with survival

$$S(x|\boldsymbol{\theta}) = e^{-cx} S_b(x|\boldsymbol{\theta}_1)$$

The most common models with this shape is the Gompertz-Makeham model (Gompertz 1825, Makeham 1866).

(c) `shape` = “`bathtub`”: produces a concave shapes in mortality by adding a declining Gompertz term and a constant parameter to the basic mortality model, where the mortality function is

$$\mu(x|\boldsymbol{\theta}) = \exp(a_0 - a_1 x) + c + \mu_b(x|\boldsymbol{\theta}_1),$$

where $a_0 \in \mathbb{R}$, $a_1, c \geq 0$ and $\boldsymbol{\theta}_1 \subset \boldsymbol{\theta}$ are specified based on argument `model`, and with survival

$$S(x|\theta) = \exp\left[\frac{e^{a_0}}{a_1}\left(e^{a_1 x} - 1\right) - cx\right] S_b(x|\theta_1).$$

The most widely use “`bathtub`” shaped model is the **Siler mortality model** (Siler 1979), which provides considerably good fits to mammalian data. The arguments for the Siler model are:

```
CalcMort(..., model = "GO", shape = "bathtub", ...)
```

2) Specifying theta parameters:

Argument `theta` requires a numerical vector of parameters. For instance, for a Gompertz model with simple shape and with parameter vector $\boldsymbol{\theta}^\top = [b_0, b_1]$ where $b_0 = -5$ and $b_1 = 0.1$, the argument should be specified as

`CalcMort(theta = c(b0 = -5, b1 = 0.1), ...)`

Note that in this example the parameter names are specified directly, this is required when `checkTheta = FALSE`. Although assigning the names to each parameter is not necessary when `checkTheta = TRUE`, it is advisable to ensure that the right values are assigned to the right parameter.

If argument `checkTheta` is set to `TRUE`, then the vector of `theta` parameters is verified for consistency with the requirements of the model and shape selected.

Author(s)

Fernando Colchero <`fernando_colchero@eva.mpg.de`>

References

Colchero, F. and J.S. Clark (2012) Bayesian inference on age-specific survival from capture-recapture data for censored and truncated data. *Journal of Animal Ecology*. 81, 139-149.

Colchero, F., O.R. Jones and M. Rebke. (2012) BaSTA: an R package for Bayesian estimation of age-specific survival from incomplete mark-recapture/recovery data with covariates. *Method in Ecology and Evolution*. 3, 466-470.

Colchero, F., *et al.* (2021) The long lives of primates and the "invariant rate of aging" hypothesis. *Nature Communications* 12:3666

Cox, D. R., and Oakes D. (1984) *Analysis of Survival Data*. Chapman and Hall, London.

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. *Philosophical Transactions of the Royal Society of London*, 115, 513-583.

Makeham, W. M. On the law of mortality (1866). *Journal of the Institute of Actuaries* 13, 1-34.

Pinder III, J.E., Wiener, J.G. and Smith, M.H. (1978) The Weibull distribution: a new method of summarizing survivorship data. *Ecology*, 59, 175-179.

Siler, W. A (1979) competing-risk model for animal mortality. *Ecology* 60, 750-757.

See Also

`CalcSurv` to calculate age-specific survival, `CalcFert` to calculate age-specific fertility.

Examples

```
# Gompertz age specific mortality (default):
mort <- CalcMort(theta = c(b0 = -5, b1 = 0.1), x = 10)

# Siler age specific mortality:
mort <- CalcMort(theta = c(a0 = -1, a1 = 1, c = 0.0001,
                          b0 = -6, b1 = 0.15), x = 10,
                 model = "GO", shape = "bathtub")

# Gompertz age specific survival (default):
surv <- CalcSurv(theta = c(b0 = -5, b1 = 0.1), x = 10)

# Siler age specific survival:
surv <- CalcSurv(theta = c(a0 = -1, a1 = 1, c = 0.0001,
                          b0 = -6, b1 = 0.15), x = 10,
                 model = "GO", shape = "bathtub")
```

`CalcProductLimitEst` *Calculating the product limit estimator from Data.*

Description

`CalcProductLimitEst` uses non-parametric methods to calculate the product limit estimator and confidence intervals.

Usage

```
CalcProductLimitEst(ageLast, ageFirst = NULL, departType,
                    calcCIs = FALSE, nboot = 1000, alpha = 0.05)
```

Arguments

`ageLast`	Numerical vector with the ages at last detection (i.e., death and censoring) (see `details`)
`ageFirst`	Numerical vector of ages at first detection (i.e., truncation). If `NULL` then all values are set to 0 (see `details`).
`departType`	Character string vector for the type of departure (i.e., last detection), with values "`D`" for death and "`C`" for censoring (see `details`).
`calcCIs`	Logical indicating whether confidence intervals should be calculated
`nboot`	Number of bootstrap iterations
`alpha`	Alpha level. Default is 0.05 for 95% CIs

Details

1) Data structure:

The function allows to calculate product limit estimator (Wann *et al.* 1987) for data that includes the following types of records:

- **Uncensored**: individuals with known ages at death;
- **right-censored**: individuals last seen alive;
- **left-truncated**: individuals born before the start of the study and are truncated at the age of entry.

The data required are the ages at last detection (i.e., *uncensored* or *right-censored*) passed through argument "`ageLast`", the type of departure via argument "`departType`", which takes two values, namely "`D`" for death, and "`C`" for censored (i.e., right-censored).

In addition, if there is *left-truncation*, it takes the ages at entry to the study by means of argument "`ageFirst`". If all the individuals were born during the study, the value of "`ageFirst`" can be left as `NULL`, which will make them all equal to 0.

Value

`CalcProductLimitEst` returns an object of class “`paramDemoPLE`” with consists of a data frame and, if indicated by the user, the confidence interval of the PLE. The data frame includes the following numerical columns:

`Ages`	ages when individuals died
`ple`	product limit estimator
`Lower`	If indicated with argument `calcCIs`, the lower $\alpha/2$ confidence bound
`Lower`	If indicated with argument `calcCIs`, the upper $1 - \alpha/2$ confidence bound

Author(s)

Fernando Colchero `<fernando_colchero@eva.mpg.de>`

See Also

`CalcLifeTable` to calculate life tables.

Examples

```
# Simulate age at death data from Gompertz model:
ages <- SampleRandAge(n = 100, theta = c(b0 = -5, b1 = 0.1))

# Calculate life table:
lt <- CalcLifeTable(ageLast = ages, departType = rep("D", 100))
```

`CalcRemainLifeExp` *Calculating Remaining Life Expectancy.*

Description

`CalcRemainingLifeExp` calculates remaining life expectancy from parametric models of age-specific mortality.

Usage

```
CalcRemainLifeExp (theta, x = NULL, dx = NULL, xmax = NULL,
                                  atAllAges = FALSE, model = "GO",
                                  shape = "simple", checkTheta = TRUE)
```

Arguments

`theta`	Numerical vector of age-specific mortality parameters (see details).
`x`	Numerical vector of ages at which to calculate mortality.
`dx`	Numerical value for the width of age intervals used for integration. If `NULL`, default is `dx = 0.01`
`xmax`	Numerical value for the maximum age. If `NULL`, the value is obtain as the age when the surviva
`atAllAges`	Logical to indicate whether ageing rates should be provided between ages 0 and `xmax`
`model`	The underlying mortality model to be used. `"EX"` = exponential,`"GO"` = Gompertz, `"WE"` = Weibull and `"LO"` = logistic (see `details`).
`shape`	The overall shape of the model. Values are: `simple` = no extra parameters added; `Makeham` = a constant parameter is added to the mortality; and `bathtub` = a Gompertz declining mortality for early ages and a constant parameter are added to the mortality model (see `details`).
`checkTheta`	Logical to verify that the `beta` parameters conform with the model's specification (see details).

Details

The function `CalcAgeingRate` uses parametric functions to calculate the actuarial (i.e., survival) rate of ageing. The function follows the conventions from package BaSTA (Colchero and Clark 2012, Colchero *et al.* 2012, Colchero *et al.* 2021) to select the parametric model of mortality. The mortality function describes how the risk of mortality changes with age, and is defined as

$$\mu(x|\theta) = \lim_{\Delta x \to 0} \frac{\Pr[x < X < x + \Delta x | X > x]}{\Delta x},$$

where X is a random variable for ages at death, $x \geq 0$ are ages and θ is the vector of mortality parameters. From the mortality function, the survival function is then given by

$$S(x|\theta) = \exp[-\int_0^x \mu(t|\theta) dt].$$

(For further details on the mortality and survival models see `CalcMort`).

Given a vector of ages $x_1, x_2, \ldots, x_n$ specified by the user with argument `x`, the function calculates the remaining life expectancy at age x_i as

$$e_{x_i} = \frac{\int_{x_i}^{\infty} S(t) dt}{S(x_i)}$$

for $i = 1, 2, \ldots, n$.

Value

The function outputs a matrix with the ages from which remaining life expectancies were calculated, and the values for the remaining life expectancy.

Author(s)

Fernando Colchero <`fernando_colchero@eva.mpg.de`>

See Also

`CalcSurv` to calculate age-specific survival, `CalcMort` to calculate age-specific mortality, `CalcFert` to calculate age-specific fertility.

`CalcAgeMaxFert` to calculate the age at maximum fertility from parametric models of age-specific fertility. `CalcAgeingRateMort` to calculate ageing rates from parametric models of age-specific mortality.

Examples

```
# Calculate ageing rate from Gompertz model:
rle <- CalcRemainLifeExp(theta = c(b0 = -5, b1 = 0.1), x = 10)

# Calculate ageing rate from Siler model:
rle <- CalcRemainLifeExp(theta = c(a0 = -1, a1 = 1, c = 0.0001,
                                   b0 = -6, b1 = 0.15), x = 10,
                              model = "GO", shape = "bathtub")
```

`plot.paramDemo` *Functions to plot parametric demographic functions*

Description

Draw a plot of demographic functions produced with function `CalcDemo`

Usage

```
## S3 method for class 'paramDemo'
plot(x, demofun = "all", ...)
```

Arguments

`x`	Object of class "`paramDemo`"
`demofun`	Character string for the demographic function to be ploted
`...`	Additional arguments passed to function `plot`

Author(s)

Fernando Colchero <`fernando_colchero@eva.mpg.de`>

See Also

`CalcDemo` to calculate parametric demographic functions.

Examples

```
# Create paramDemo object from Gompertz mortality and
# quadratic fertility:
dem <- CalcDemo(theta = c(b0 = -5, b1 = 0.1),
                 beta = c(b0 = 0.5, b1 = 0.01, b2 = 10),
                 summarStats = TRUE, agesAR = c(5, 10))

# Plot demographic object:
plot(dem)
```

`plot.paramDemoLT` *Functions to plot life tables and product limit estimators*

Description

Draw a plot of demographic rates from a life table produced with function `CalcLifeTable` or of product limit estimator produced with function `CalcProductLimitEst`

Usage

```
## S3 method for class 'paramDemoLT'
plot(x, demorate = "lx", inclCIs = FALSE, ...)

## S3 method for class 'paramDemoPLE'
plot(x, inclCIs = FALSE, ...)
```

Arguments

`x`	Object of class `paramDemoLT` or `paramDemoPLE` produced with functions `CalcLifeTable` or `CalcProductLimitEst`
`demorate`	Demographic rate to be plotted, choices are "`lx`" for the survival, "`px`" for the age-specific survival probability, "`qx`" for the age-specific mortality probability, and "`ex`" for the remaining life expectancy
`inclCIs`	Logical indicating whether confidence intervals should be ploted in case they were calculated with function `CalcLifeTable` or `CalcProductLimitEst`
`...`	Additional arguments passed to function `plot`

Author(s)

Fernando Colchero <`fernando_colchero@eva.mpg.de`>

See Also

`CalcLifeTable` to calculate life tables. `CalcProductLimitEst` to calculate product limit estimators.

Examples

```
# Simulate age at death data from Gompertz model:
ages <- SampleRandAge(n = 100, theta = c(b0 = -5, b1 = 0.1))

# Calculate life table:
lt <- CalcLifeTable(ageLast = ages, departType = rep("D", 100))

# Plot life table:
plot(lt)
```

`plot.PDlifeHist` *Summarizing and plotting life history variables from parametric demographic models.*

Description

These functions are all generic methods for class `PDlifeHist`.

Usage

```
## S3 method for class 'PDlifeHist'
plot(x, type = "rates", ...)

## S3 method for class 'PDlifeHist'
summary(object, ...)

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

Arguments

`x`	Object of class “`PDlifeHist`” obtained from function `CalcLifeHist`
`object`	Object of class “`PDlifeHist`” obtained from function `CalcLifeHist`
`type`	Character string specifying the type of plot to be drawn, options are “`rates`”, “`sensitivities`”, “`all`”
`...`	Additional arguments passed to functions `plot`, `print`, and `summary`

Details

For objects of class “`PDlifeHist`”, function `print` displays on the console the data frame of life history variables produced by function `CalcLifeHist`.

Function `summary` prints to the console, first the settings as specified by the user (i.e., `theta` and `beta` parameters, and the types of models used), and then, as function `print`, the table of life history variables.

Function `plot` with argument `type` = “`rates`” produces plots of the age-specific demographic rates (i.e., p_x and m_x) and the stable age structure and reproductive values (i.e., ω_x and ν_x). If argument `type` = “`sensitivity`”, then it produces plots of the sensitivities and elasticities of λ to p_x and m_x.

Author(s)

Fernando Colchero <`fernando_colchero@eva.mpg.de`>

See Also

`CalcLifeHist` to calculate life history variables from parametric demographic models.

Examples

```
# Calculate life histories on the defalult models:
test <- CalcLifeHist(theta = c(b0 = -3, b1 = 0.1),
                     beta = c(b0 = 1, b1 = 0.01, b2 = 5),
                     ageMatur = 2)

# Print results:
test

# Summary:
summary(test)

# Plot results:
plot(test, type = "all")
```

`SampleRandAge` *Sampling Ages at Death from a Parametric Mortality model*

Description

`SampleRandAge` is used to randomly sample ages at death from a parametric mortality model

Usage

```
SampleRandAge(n, theta, dx = 0.001, model = "GO",
                 shape = "simple",  minSx = 1e-04)
```

Arguments

`n`	Number of individual ages at death to sample
`theta`	Numerical vector of age-specific mortality parameters (see details).
`dx`	Numeric value for the age increments (see details).
`model`	The underlying mortality model to be used. `"EX"` = exponential,`"GO"` = Gompertz, `"WE"` = Weibull and `"LO"` = logistic (see `details`).
`shape`	The overall shape of the model. Values are: `simple` = no extra parameters added; `Makeham` = a constant parameter is added to the mortality; and `bathtub` = a Gompertz declining mortality for early ages and a constant parameter are added to the mortality model (see `details`).
`minSx`	Minimum value for the survival function to set the maximum age (see details)

Details

`SampleRandAge` arguments “`model`” and “`shape`”, to calculate the survival function, $S(x)$ for $x \geq 0$, at discrete age intervals $[x, x + \Delta x)$ where Δx is specfied with argument “`dx`”. It then calculates the CDF of ages at death, $F(x)$, and uses inverse sampling to sample n ages at death by random sampling from a uniform distribution bound as

$$u_i \sim Unif[F(x_m), F(x_M)],$$

where $x_m = min(x)$ and $x_M = max(x)$. It then finds the corresponding ages at death as

$$x_i = F^{-1}(u_i).$$

In most cases, this is acheived numerically and thus the lower the value for argument “`dx`” the higher the precision.

Author(s)

Fernando Colchero `<fernando_colchero@eva.mpg.de>`

Examples

```
# Simulate age at death data from Gompertz model:
ages <- SampleRandAge(n = 100, theta = c(b0 = -5, b1 = 0.1))

# Simulate age at death data from Siler model:
ages <- SampleRandAge(n = 100, theta = c(a0 = 0, a1 = 1, c = 0.001,
                      b0 = -5, b1 = 0.1), model = "GO",
                      shape = "bathtub")
```

Index