

Package ‘MortalityLaws’

October 7, 2026

Type Package

Title Parametric Mortality Models, Life Tables and HMD

Version 3.0.0

Maintainer Marius D. Pascariu <mpascariu@outlook.com>

Description Fit the most popular human mortality 'laws', and construct full and abridged life tables given various input indices. A mortality law is a parametric function that describes the dying-out process of individuals in a population during a significant portion of their life spans. For a comprehensive review of the most important mortality laws see Tabeau (2001) <[doi:10.1007/0-306-47562-6_1](https://doi.org/10.1007/0-306-47562-6_1)>. Practical functions for downloading data from various human mortality databases are provided as well.

License MIT + file LICENSE

LazyData TRUE

Depends R (>= 4.5.0)

Imports pbapply (>= 1.7-4), httr (>= 1.4.8)

Suggests testthat (>= 3.3.2), knitr, rmarkdown

URL <https://mpascariu.github.io/MortalityLaws/>,
<https://github.com/mpascariu/MortalityLaws>

BugReports <https://github.com/mpascariu/MortalityLaws/issues>

Encoding UTF-8

VignetteBuilder knitr

Config/testthat/edition 3

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Marius D. Pascariu [aut, cre, cph] (ORCID:
<<https://orcid.org/0000-0002-2568-6489>>),
Vladimir Canudas-Romo [ctb]

Repository CRAN

Date/Publication 2026-10-07 18:00:08 UTC

Contents

ahmd	2
AIC.MortalityLaw	3
availableHMD	4
availableLaws	5
availableLF	8
convertFx	9
deviance.MortalityLaw	11
df.residual.MortalityLaw	11
dispersion.MortalityLaw	12
LawTable	13
LifeTable	17
logLik.MortalityLaw	23
MortalityLaw	24
MortalityLaws	27
plot.convertFx	28
plot.LifeTable	29
plot.MortalityLaw	30
predict.MortalityLaw	31
ReadAHMD	32
ReadCHMD	34
ReadHMD	36
ReadJMD	39
summary.MortalityLaw	41
Index	42

ahmd	<i>MortalityLaws Test Data</i>
------	--------------------------------

Description

Dataset containing altered death rates (mx), death counts (Dx) and exposures (Ex) for the female population living in England & Wales in four different years: 1850, 1900, 1950 and 2010. This dataset is provided for testing purposes only. Download the actual data free of charge from <https://www.mortality.org>. Once a username and a password are created on the website, the function `ReadHMD` can be used for downloading.

Usage

ahmd

Format

A list with the components mx, Dx and Ex. Each component is a `data.frame` with 111 rows (ages 0 to 110) and one column per year: 1850, 1900, 1950 and 2010.

Source[Human Mortality Database](#)**See Also**[ReadHMD](#)**Examples**

```
head(ahmd$mx)
```

`AIC.MortalityLaw`*Extract the AIC of a Fitted Mortality Law*

Description

Returns the Akaike information criterion of a "MortalityLaw" fit, $2 * k - 2 * \logLik$, with k the number of fitted parameters. Like the log-likelihood it is defined only for the likelihood-based objectives and is NaN otherwise. Use it to compare candidate laws fitted to the same data.

Usage

```
## S3 method for class 'MortalityLaw'
AIC(object, ...)
```

Arguments

`object` An object of class "MortalityLaw".
`...` Further arguments passed to or from other methods.

Value

The AIC value for a single fit, or a named vector of AIC values for a multiple fit.

See Also[MortalityLaw](#); [logLik.MortalityLaw](#).**Examples**

```
x <- 45:75
M1 <- MortalityLaw(x = x, Dx = ahmd$Dx[as.character(x), "1950"],
                  Ex = ahmd$Ex[as.character(x), "1950"],
                  law = "makeham", opt.method = "poissonL")
AIC(M1)
```

`availableHMD`*Check Data Availability in HMD*

Description

Returns information about the data available in the Human Mortality Database (HMD), including the range of years covered by the life tables for each country or region.

Usage

```
availableHMD(link = "https://www.mortality.org/Data/DataAvailability")
```

Arguments

`link` URL to the HMD available data. Default: "https://www.mortality.org/Data/DataAvailability"

Details

The function scrapes the availability table published on the HMD site, so it needs no account. It is a thin companion to [ReadHMD](#), useful for checking what exists before a download; every failure (no connection, a non-200 status, a body that is not a table) is reported with a `message()` and returns `NULL` rather than raising an error.

Value

A data frame with one row per country or region, or `NULL` when the website cannot be reached or the response carries no table.

Author(s)

Marius D. Pascariu

See Also

[ReadHMD](#)

Examples

```
## Not run:  
availableHMD()  
  
## End(Not run)
```

availableLaws

*Check the Available Mortality Laws***Description**

The law catalogue. It lists every parametric model that [MortalityLaw](#) can fit, with the formula and the code to pass through law, and says where each model applies. Use it to choose a law before fitting; there is no need to know the functional form, only its code and the age range it is meant for. For a comprehensive review of the mortality laws themselves, Tabeau (2001) is a good starting point.

Usage

```
availableLaws(law = NULL)
```

Arguments

law	Optional. Default: NULL. One can extract details about a certain model by specifying its codename.
-----	--

Details

The TYPE column says where on the lifespan the law belongs, read off the legend in the second component of the result: a law covering the whole range (6) is fitted over all ages, while a law for old age (5) is fitted from the adult ages up. The FIT column says whether the law describes a hazard, $\mu[x]$, or a death probability, $q[x]$; [MortalityLaw](#) and [LawTable](#) handle both. The SCALE_X column flags the laws whose age vector is rescaled during fitting for numerical stability, which matters when the fitted coefficients are reused outside the fitted age range (see [LawTable](#)).

A law is normally reached through the law argument of [MortalityLaw](#), but each one is also a function that can be called by name for a plain hazard or death probability curve; those help pages exist for reference and are kept out of the help index. A law that is not in the catalogue can still be fitted by passing it as a function through the custom.law argument; see the examples on the [MortalityLaw](#) page.

A few laws carry a caveat worth knowing before choosing them, all documented in the model catalogue and their catalogue entries:

- "scholey": its truncation parameter is identified only on day- or week-level data over the first year of life; on single-year ages it collapses and the fit reduces to "scholey_shifted_power", with a warning.
- "opperman": the published middle term has a free sign; the package uses the negative branch the log-scale engine permits, which is the branch mortality data occupy.
- "steffensen": the formula is attributed to Steffensen (1930), but the attribution is not verified against the paywalled source.
- "HP", "HP2", "HP3", "HP4" and "kostaki": high-parameter models; fit them with `opt.method = "LF2"`.

- "gompertz_logquad" and "makeham_logquad": the sign of the quadratic term is fixed to the decelerating branch that mortality data occupy; the accelerating branch cannot be fitted.
- "beard_makeham" and "perks": the same four-parameter Perks-Beard logistic written two ways, so they fit identical curves; "beard", "kannisto" and "kannisto_makeham" are its two- and three-parameter cases. Pick one of them, not several.
- "demoivre": the 1725 baseline, kept for completeness. Its hazard is defined only below a limiting age, so it must not be extrapolated and the fit warns every time.
- "weibull": not defined at birth, so age 0 is reported as missing and takes no part in the fit; start the fit at age 1.

Two entries in the reference list are background for the infant laws rather than the source of a catalogue code: Harper (1936), and de Beer and Janssen (2016), whose infancy term is the fitted `pareto_2`.

Value

The output is of the "availableLaws" class with the following components:

table	Table with mortality models and codes to be used in MortalityLaw , the model formula, the lifespan section (TYPE), the code (CODE), whether the law describes $\mu[x]$ or $q[x]$ (FIT) and whether fitting rescales the ages (SCALE_X).
legend	Table with details about the section of the mortality curve.

Author(s)

Marius D. Pascariu

References

1. De Moivre, A. (1725). *Annuities on Lives: or, the Valuation of Annuities upon any Number of Lives*. London: William Pearson.
2. 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.
3. Makeham, W. (1860). On the Law of Mortality and Construction of Annuity Tables. The Assurance Magazine and Journal of the Institute of Actuaries, 8(6), 301-310. [doi:10.1017/S204616580000126X](#)
4. Thiele, T. (1871). On a Mathematical Formula to express the Rate of Mortality throughout the whole of Life, tested by a Series of Observations made use of by the Danish Life Insurance Company of 1871. Journal of the Institute of Actuaries and Assurance Magazine, 16(5), 313-329. [doi:10.1017/S2046167400043688](#)
5. Lomax, K. S. (1954). Business Failures: Another Example of the Analysis of Failure Data. Journal of the American Statistical Association, 49(268), 847-852. [doi:10.1080/01621459.1954.10501239](#)
6. Vaupel, J. W. and Yashin, A. I. (1983). The Deviant Dynamics of Death in Heterogeneous Populations. IIASA Research Report RR-83-1. Laxenburg, Austria.
7. de Beer, J. and Janssen, F. (2016). A new parametric model to assess delay and compression of mortality. Population Health Metrics, 14(1), 46. [doi:10.1186/s1296301601131](#)

8. Scholey, J. (2019). The Age-Trajectory of Infant Mortality in the United States: Parametric Models and Generative Mechanisms. PAA Annual Conference, Austin.
9. Oppermann, L. H. F. (1870). On the graduation of life tables, with special application to the rate of mortality in infancy and childhood. The Insurance Record Minutes from a meeting in the Institute of Actuaries, 42.
10. Wittstein, T. and D. Bumsted. (1883). **The Mathematical Law of Mortality**. Journal of the Institute of Actuaries and Assurance Magazine, 24(3), 153-173.
11. Steffensen, J. (1930). Infantile mortality from an actuarial point of view. Skandinavisk Aktuariatidskrift 13, 272-286. doi:10.1080/03461238.1930.10416902
12. Perks, W. (1932). On Some Experiments in the Graduation of Mortality Statistics. Journal of the Institute of Actuaries, 63(1), 12-57. doi:10.1017/S0020268100046680
13. Harper, F. S. (1936). An actuarial study of infant mortality. Scandinavian Actuarial Journal 1936 (3-4), 234-270. doi:10.1080/03461238.1936.10405113
14. Weibull, W. (1951). A statistical distribution function of wide applicability. Journal of applied mechanics 18, 293-297. doi:10.1115/1.4010337
15. Beard, R. E. (1971). **Some aspects of theories of mortality, cause of death analysis, forecasting and stochastic processes**. Biological aspects of demography 999, 57-68.
16. Vaupel, J., Manton, K.G., and Stallard, E. (1979). The impact of heterogeneity in individual frailty on the dynamics of mortality. Demography 16(3): 439-454. doi:10.2307/2061224
17. Siler, W. (1979), A Competing-Risk Model for Animal Mortality. Ecology, 60: 750-757. doi:10.2307/1936612
18. Heligman, L., & Pollard, J. (1980). The age pattern of mortality. Journal of the Institute of Actuaries, 107(1), 49-80. doi:10.1017/S0020268100040257
19. Rogers A and Planck F (1983). **MODEL: A General Program for Estimating Parametrized Model Schedules of Fertility, Mortality, Migration, and Marital and Labor Force Status Transitions**. IIASA Working Paper. IIASA, Laxenburg, Austria: WP-83-102
20. Martinelle S. (1987). A generalized Perks formula for old-age mortality. Stockholm, Sweden, Statistiska centralbyran, 1987. 55 p. (R&D Report, Research-Methods-Development, U/STM No. 38)
21. Forfar, D. O., McCutcheon, J. J. and Wilkie, A. D. (1988). On graduation by mathematical formula. Journal of the Institute of Actuaries, 115(1), 1-149.
22. Carriere J.F. (1992). Parametric models for life tables. Transactions of the Society of Actuaries. Vol.44
23. Kostaki A. (1992). A nine-parameter version of the Heligman-Pollard formula. Mathematical Population Studies. Vol. 3 277-288. doi:10.1080/08898489209525346
24. Thatcher AR, Kannisto V and Vaupel JW (1998). The force of mortality at ages 80 to 120. Odense Monographs on Population Aging Vol. 5, Odense University Press, 1998. 104, 20 p. Odense, Denmark
25. Tabeau E. (2001). A Review of Demographic Forecasting Models for Mortality. In: Tabeau E., van den Berg Jeths A., Heathcote C. (eds) Forecasting Mortality in Developed Countries. European Studies of Population, vol 9. Springer, Dordrecht. doi:10.1007/0306475626_1
26. Finkelstein M. (2012) Discussing the Strehler-Mildvan model of mortality Demographic Research, Vol. 26(9), 191-206. doi:10.4054/DemRes.2012.26.9

See Also

[MortalityLaw](#) to fit a law; [LawTable](#) to build a life table from fitted coefficients; [availableLF](#) for the loss functions.

[MortalityLaw](#)

Examples

```
availableLaws()
```

availableLF	<i>Check Available Loss Functions</i>
-------------	---------------------------------------

Description

Returns information about the loss functions implemented for use with the optimisation procedure in the [MortalityLaw](#) function.

Usage

```
availableLF()
```

Details

The two likelihoods ("poissonL", "binomialL") are the only objectives that yield a log-likelihood, an AIC and a BIC; the six loss functions ("LF1" to "LF6") leave those measures undefined (NaN) and are compared on the deviance or the loss itself. "LF2", the squared log-ratio, is the default: it is scale-free and robust, and it has been observed to return reliable estimates for the high-parameter laws such as Heligman-Pollard. There is no universally best choice, so it is worth trying more than one.

Value

A list of class availableLF with the components:

table	Table with loss functions and codes to be used in MortalityLaw .
legend	Table with details about the abbreviation used.

Author(s)

Marius D. Pascariu

See Also

[MortalityLaw](#)

Examples

```
availableLF()
```

convertFx

Convert Life Table Indicators

Description

Easily convert between different life table indicators (e.g., from death rates `mx` to death probabilities `qx`, or from survivorship `lx` to life expectancy `ex`). The function wraps [LifeTable](#) internally, so the conversion relies on the same constant-force-of-mortality (CFM) assumption and life-table methodology used throughout the package.

Usage

```
convertFx(
  x,
  data,
  from = c("mx", "qx", "dx", "lx", "ex"),
  to = c("mx", "qx", "dx", "lx", "Lx", "Tx", "ex"),
  ...
)
```

Arguments

<code>x</code>	Numeric vector of ages at the beginning of each age interval. For a full life table, use single-year ages (e.g., <code>0:110</code>). For an abridged life table, use the lower bound of each interval (e.g., <code>c(0, 1, 5, 10, ..., 110)</code>).
<code>data</code>	A numeric vector, matrix, or data frame containing the mortality indicator to be converted. Each row should correspond to an age, each column to a separate population or time period.
<code>from</code>	The type of indicator supplied in <code>data</code> . One of: <code>"mx"</code> , <code>"qx"</code> , <code>"dx"</code> , <code>"lx"</code> , or <code>"ex"</code> .
<code>to</code>	The desired output indicator. One of: <code>"mx"</code> , <code>"qx"</code> , <code>"dx"</code> , <code>"lx"</code> , <code>"Lx"</code> , <code>"Tx"</code> , or <code>"ex"</code> .
<code>...</code>	Further arguments passed to LifeTable that may effect the results, such as <code>sex</code> , <code>lx0</code> , <code>ax</code> , or the closing arguments <code>close</code> , <code>omega</code> and <code>fit_from</code> . When <code>omega</code> extends the table beyond the input's open age, the result carries the extended ages (vector names, or matrix row names) rather than the input ages.

Details

This function provides a convenient interface for converting a single mortality indicator into another, without having to call [LifeTable](#) directly and extract the desired column.

The supported **input** types (`from`) are: `mx`, `qx`, `dx`, `lx`, and `ex`.

The supported **output** types (`to`) are: `mx`, `qx`, `dx`, `lx`, `Lx`, `Tx`, and `ex`.

There are 35 possible from-to combinations ($5 \text{ inputs} \times 7 \text{ outputs}$). Conversions that need a single life-table identity, such as `mx` to `qx` or `dx` to `lx`, are computed directly from that relation. All the

other conversions are obtained from the full life-table computation; for example, converting `mx` to `ex` will internally compute `qx`, `lx`, `dx`, `Lx`, and `Tx` in sequence. A `ex` input is converted by building the life table that reproduces the supplied curve (see [LifeTable](#)).

When data is a vector, the function returns a named vector. When data is a `matrix` or `data.frame` with multiple columns, the function applies the conversion column-wise and returns a matrix with the same row and column names as the input.

Value

A numeric vector or matrix of class "convertFx" containing the converted life table indicator. If the input was a named object, the output retains those names. The result carries the ages it is indexed by (`x`), the conversion (`from`, `to`) and the input curve (`input`) as attributes, which is what [plot.convertFx](#) draws. It behaves like the underlying numeric vector or matrix in every other respect; subsetting returns the bare values.

Author(s)

Marius D. Pascariu

See Also

[LifeTable](#) for the underlying life-table construction; [LawTable](#) for generating life tables from parametric mortality laws; [plot.convertFx](#) for plotting a conversion.

Examples

```
# ---- Basic conversions ----

x  <- 0:110
mx <- ahmd$mx

# Convert death rates to death probabilities
qx <- convertFx(x, data = mx, from = "mx", to = "qx")

# Convert death rates to death distribution
dx <- convertFx(x, data = mx, from = "mx", to = "dx")

# Convert death rates to survivorship
lx <- convertFx(x, data = mx, from = "mx", to = "lx")

# Convert death rates to life expectancy
ex <- convertFx(x, data = mx, from = "mx", to = "ex")
```

deviance.MortalityLaw *Extract the Deviance of a Fitted Mortality Law*

Description

Returns the deviance of a "MortalityLaw" fit. For a fit entered from death counts and exposures (Dx, Ex) it is the Poisson deviance, the quantity `opt.method = "poissonL"` minimises. For a fit entered from rates (mx or qx) there is no count likelihood, so it is the sum of squared log-residuals.

Usage

```
## S3 method for class 'MortalityLaw'
deviance(object, ...)
```

Arguments

`object` An object of class "MortalityLaw".
`...` Further arguments passed to or from other methods.

Value

The deviance for a single fit, or a named vector of deviances for a multiple fit.

See Also

[MortalityLaw](#); [dispersion](#).

Examples

```
x <- 45:75
M1 <- MortalityLaw(x = x, Dx = ahmd$Dx[as.character(x), "1950"],
                  Ex = ahmd$Ex[as.character(x), "1950"],
                  law = "makeham", opt.method = "poissonL")
deviance(M1)
```

df.residual.MortalityLaw

Extract the Residual Degrees of Freedom of a Fitted Mortality Law

Description

Returns the residual degrees of freedom of a "MortalityLaw" fit, the number of fitted ages minus the number of estimated parameters.

Usage

```
## S3 method for class 'MortalityLaw'
df.residual(object, ...)
```

Arguments

object An object of class "MortalityLaw".
 ... Further arguments passed to or from other methods.

Value

The residual degrees of freedom for a single fit, or a named vector for a multiple fit.

See Also

[MortalityLaw](#); [dispersion](#).

Examples

```
x <- 45:75
M1 <- MortalityLaw(x = x, Dx = ahmd$Dx[as.character(x), "1950"],
                   Ex = ahmd$Ex[as.character(x), "1950"],
                   law = "makeham", opt.method = "poissonL")
df.residual(M1)
```

dispersion.MortalityLaw

Dispersion of a Fitted Mortality Law

Description

Returns the dispersion of the fit, a scalar measure of how far the fitted values spread around the data. For the count cases it is the Pearson chi-square divided by the residual degrees of freedom, the GLM dispersion (about 1 for a correctly specified Poisson model); for the rate cases it is the mean squared log-residual. The value is also reported by [summary.MortalityLaw](#).

Usage

```
dispersion(object, ...)

## S3 method for class 'MortalityLaw'
dispersion(object, ...)
```

Arguments

object An object of class "MortalityLaw".
 ... Further arguments passed to or from other methods.

Value

The dispersion for a single fit, or a named vector of dispersions for a multiple fit.

See Also

[MortalityLaw](#); [deviance.MortalityLaw](#).

Examples

```
x <- 45:75
M1 <- MortalityLaw(x = x, Dx = ahmd$Dx[as.character(x), "1950"],
                  Ex = ahmd$Ex[as.character(x), "1950"],
                  law = "makeham", opt.method = "poissonL")
dispersion(M1)
```

LawTable

Compute Life Tables from Parameters of a Mortality Law

Description

Generate a complete life table directly from the fitted parameters of a parametric mortality model. This function evaluates the mortality law at the given ages and passes the resulting death rates (m_x) or death probabilities (q_x) to [LifeTable](#) for further computation of all standard life-table columns (l_x , dx , L_x , T_x , ex , etc.).

Usage

```
LawTable(x, par, law, sex = NULL, lx0 = 1e+05, ax = "andreev_kingkade")
```

Arguments

- | | |
|-----|--|
| x | Numeric vector of ages at the beginning of each age interval. For a full life table, use single-year ages (e.g., 0:110). For an abridged life table, use the lower bound of each interval (e.g., c(0, 1, 5, 10, ..., 110)). |
| par | The parameters of the mortality model. Can be: <ul style="list-style-type: none"> • A numeric vector containing the coefficients (for a single life table). • A numeric matrix or data.frame where each row corresponds to a separate set of parameters (producing multiple life tables). Column names must match the parameter names of the chosen law, as described in the details below. |
| law | The name of the mortality law to be used (e.g., "gompertz", "makeham"). Run availableLaws to see all options. |
| sex | Sex of the population. Options are NULL (default), "male", "female", or "total". When specified, the first two entries of the ax column are adjusted using Coale-Demeny coefficients, producing more accurate life-table values at the youngest ages. The adjustment differs slightly between males and females. |

<code>lx0</code>	Radix, the starting population (or probability scale) at age 0. Default is 100,000. All subsequent life-table columns (<code>lx</code> , <code>dx</code> , <code>Lx</code> , <code>Tx</code>) are scaled accordingly.
<code>ax</code>	The average number of person-years lived in each age interval by those who die in it, given either as values or as the method that produces them. Accepts two forms: • a numeric scalar or vector: a scalar is applied to all intervals, a vector must have the same length as <code>x</code>. A common assumption is <code>ax = 0.5</code>, which places deaths at the midpoint of each interval; • a method name, one of "andreev_kingkade" (the default), "cfm", "preston" or "coale_demeny" (see below). "andreev_kingkade" follows the rule the Human Mortality Database applies to its period life tables (Methods Protocol, version 6, section 7.1): the Andreev-Kingkade (2015) formula sets the value for the first year of life from <code>m0</code>, every other closed interval uses half its length ($n/2$), and the open interval uses $1/mx$. It is the most accurate of the four methods and the one that reproduces the published HMD life tables most closely. Because it produces a numeric <code>ax</code> before the rates are converted, the <code>mx</code> to <code>qx</code> step uses the exact identity $qx = n*mx / (1 + (n - ax)*mx)$, which is the identity the protocol uses (equation 74). The Andreev-Kingkade value is a property of a one-year first interval that starts at birth, so when the first interval is wider or the table starts above age 0 the interval keeps the ordinary midpoint value instead. The other three methods share the same basis, the standard lifetable identity $ax = n + 1/m - n/q$ under a constant force of mortality (Preston, Heuveline and Guillot 2001, eq. 3.15), and differ in how they treat the first two intervals: • "cfm": the identity alone, for every interval; • "preston": the identity, with the first two intervals replaced by the Coale-Demeny West separation factors published by Preston et al. (2001), table 3.3, when sex is given; • "coale_demeny": the identity, with the first two intervals replaced by the original 1983 Coale-Demeny rule expressed in <code>qx</code>, reproduced by the PAS software, when sex is given. "preston" and "coale_demeny" agree exactly once <code>mx[1]</code> reaches 0.107, and differ by at most a few thousandths of a year below it. Both coincide with "cfm" when sex is NULL. "cfm" and the Coale-Demeny variants adjust <code>ax</code> after the rates have been converted, so they leave the constant force of mortality conversion in place; only "andreev_kingkade" changes the conversion itself. A value supplied for the open age interval is kept as given. Everybody alive there dies in that interval, so its own rate implies what the average time lived in it should be ($1/mx$); a value that differs is reported and left alone, which leaves the <code>ax</code> and <code>mx</code> columns of the table disagreeing at that age, or the model-implied value when <code>close</code> is set. The one exception is a table entered from <code>ex</code>, where the curve being inverted fixes that interval. See Details.

Details

This function is designed to work with models that have been fitted externally (e.g., via [MortalityLaw](#) or by hand). The `par` argument must contain the estimated coefficients of the mortality law, and

law must be one of the valid codes listed by [availableLaws](#).

The `ax` argument is passed straight to [LifeTable](#); its default, "andreev_kingkade", is the rule the Human Mortality Database applies to its period life tables (Methods Protocol, version 6; see [LifeTable](#) for the alternatives). Because a law is evaluated on a possibly scaled age vector, the first interval is only treated as a one-year interval with the Andreev-Kingkade rule when the ages passed in `x` really start one year apart.

Important caveat: age scaling during fitting

Several mortality laws (e.g., Gompertz, Makeham) internally *scale* the age vector during optimisation to ensure numerical stability. If the model was fitted using `MortalityLaw` over an age range $[a, b]$, the published coefficients correspond to the *scaled* ages, not the original ages. Consequently, `LawTable` will only produce valid life tables for ages $\geq a$ (the lower bound of the fitting range). Attempting to use the same coefficients at younger ages will yield incorrect results (e.g., life expectancy at age 25 will equal that at age 45).

To determine which models apply age scaling, run:

```
A <- availableLaws()$table
A[, c("CODE", "SCALE_X")]
```

Models with `SCALE_X = TRUE` rescale the age vector internally. When using `LawTable` with such a model, make sure the `x` argument starts from the same lower age bound used during fitting.

For models that do *not* scale (e.g., Heligman-Pollard "HP"), this limitation does not apply, and `LawTable` can be used for any age range.

Matching the coefficients to the model parameters

The coefficients supplied in `par` are matched to the parameters of the chosen law **by name and in any order**. For a matrix or a data.frame the column names must therefore be the parameter names of that law (e.g. `c("A", "B", "C")` for "makeham"); the row names are used as the labels of the resulting life tables and must be supplied. An unnamed vector keeps the positional convention, i.e. the coefficients are read in the order in which the parameters are documented for the law.

Value

An object of class "LifeTable" containing the following components:

<code>lt</code>	A data.frame with the complete life table, including columns for age interval (<code>x.int</code>), exact age (<code>x</code>), death rate (<code>mx</code>), death probability (<code>qx</code>), person-years lived by decedents (<code>ax</code>), survivorship (<code>lx</code>), death distribution (<code>dx</code>), person-years lived (<code>Lx</code>), total person-years remaining (<code>Tx</code>), and life expectancy (<code>ex</code>).
<code>call</code>	The matched function call.
<code>process_date</code>	Timestamp of when the life table was computed.

Author(s)

Marius D. Pascariu

See Also

[LifeTable](#) for constructing life tables from raw mortality data; [MortalityLaw](#) for fitting parametric mortality models; [availableLaws](#) for the list of implemented laws and their scaling behaviour.

Examples

```
# Example 1 --- Makeham --- multiple life tables from a matrix of parameters
```

```
x1 <- 45:100
L1 <- "makeham"
C1 <- matrix(
  c(0.00717, 0.07789, 0.00363,
    0.01018, 0.07229, 0.00001,
    0.00298, 0.09585, 0.00002,
    0.00067, 0.11572, 0.00078),
  nrow = 4,
  dimnames = list(1:4, c("A", "B", "C")))
)
```

```
LawTable(x = x1, par = C1, law = L1)
```

```
# ---- Important note on age scaling ----
```

```
# The Makeham model applies internal age scaling during fitting.
# If the coefficients above were estimated over ages 45-100, the life
# table produced by LawTable is valid only from age 45 onward.
```

```
# ---- Example 1B: correct usage ----
```

```
LawTable(x = x1, par = c(0.00717, 0.07789, 0.00363), law = L1)
```

```
# ---- Example 1C: incorrect usage ----
```

```
# The code below uses the same coefficients but starts at age 25.
# Because the model was fitted on scaled ages (starting at 45),
# the life table at age 25 will be meaningless (e.g., e25 equals e45).
LawTable(x = 25:100, par = c(0.00717, 0.07789, 0.00363), law = L1)
```

```
# ---- How to check which laws apply scaling ----
```

```
A <- availableLaws()$table
A[, c("CODE", "SCALE_X")]
```

```
# Example 2 --- Heligman-Pollard (no scaling) ---
```

```
x2 <- 0:110
L2 <- "HP"
C2 <- c(0.00223, 0.01461, 0.12292, 0.00091,
        2.75201, 29.01877, 0.00002, 1.11411)
```

```
LawTable(x = x2, par = C2, law = L2)
```

```
# Because "HP" does NOT scale the age vector, the output is valid for
# any starting age. Compare:
LawTable(x = 3:110, par = C2, law = L2)
# Note that e3 = 70.31 in both tables, confirming consistency.
```

LifeTable

*Compute Life Tables from Mortality Data***Description**

Construct either a full (single-year age intervals) or an abridged (wider age intervals) life table from a variety of input data types. The function accepts:

- Death counts and mid-interval population estimates (D_x , E_x)
- Age-specific death rates (m_x)
- Death probabilities (q_x)
- Survivorship curve (l_x)
- Distribution of deaths (dx)
- Remaining life expectancy (ex)

Exactly one of these input options must be provided; supplying more than one is an error. The input can be a numeric vector, `matrix`, or `data.frame`. When a `matrix` or `data.frame` with multiple columns is supplied, the function computes one life table per column.

Usage

```
LifeTable(x, Dx = NULL, Ex = NULL,
          mx = NULL,
          qx = NULL,
          lx = NULL,
          dx = NULL,
          ex = NULL,
          sex = NULL,
          lx0 = 1e5,
          ax = "andreev_kingkade",
          close = NULL,
          omega = NULL,
          fit_from = NULL)
```

Arguments

<code>x</code>	Numeric vector of ages at the beginning of each age interval. For a full life table, use single-year ages (e.g., <code>0:110</code>). For an abridged life table, use the lower bound of each interval (e.g., <code>c(0, 1, 5, 10, ..., 110)</code>).
<code>Dx</code>	Death counts. Each element represents the total number of deaths during the calendar year to persons aged x to $x + n$ (where n is the length of the age interval). Must be provided together with <code>Ex</code> .
<code>Ex</code>	Exposure-to-risk in the period. This is usually approximated by the mid-year population aged x to $x + n$. Must be provided together with <code>Dx</code> .
<code>mx</code>	Age-specific death rate in the age interval $[x, x+n)$. Defined as D_x / E_x .

q_x	Probability of dying within the age interval $[x, x+n)$.
l_x	Probability of surviving to exact age x (if $l_{x0} = 1$), or the number of survivors at exact age x (if $l_{x0} > 1$). When l_x is the sole input, the values are re-scaled to the chosen radix l_{x0} .
dx	Number of deaths in the life-table population occurring in the age interval $[x, x+n)$. When dx is the sole input, the values are re-scaled to sum to l_{x0} .
e_x	Remaining life expectancy at age x , in years. When e_x is the sole input the function builds the life table that reproduces the supplied curve: the survivorship, death probabilities and death rates are recovered from e_x and the a_x convention (see Details). A curve that rises with age at the youngest ages is allowed; a missing value anywhere in the curve, or a curve that is not a feasible life table (falling faster than the interval width, or below its own a_x), is an error naming the affected ages. A missing e_x is never repaired the way a missing rate is.
sex	Sex of the population. Options are NULL (default), "male", "female", or "total". When specified, the first two entries of the a_x column are adjusted using Coale-Demeny coefficients, producing more accurate life-table values at the youngest ages. The adjustment differs slightly between males and females.
l_{x0}	Radix, the starting population (or probability scale) at age 0. Default is 100,000. All subsequent life-table columns (l_x , dx , L_x , T_x) are scaled accordingly.
a_x	The average number of person-years lived in each age interval by those who die in it, given either as values or as the method that produces them. Accepts two forms: • a numeric scalar or vector: a scalar is applied to all intervals, a vector must have the same length as x. A common assumption is $a_x = 0.5$, which places deaths at the midpoint of each interval; • a method name, one of "andreev_kingkade" (the default), "cfm", "preston" or "coale_demeny" (see below).

"andreev_kingkade" follows the rule the Human Mortality Database applies to its period life tables (Methods Protocol, version 6, section 7.1): the Andreev-Kingkade (2015) formula sets the value for the first year of life from m_0 , every other closed interval uses half its length ($n/2$), and the open interval uses $1/m_x$. It is the most accurate of the four methods and the one that reproduces the published HMD life tables most closely. Because it produces a numeric a_x before the rates are converted, the m_x to q_x step uses the exact identity $q_x = n \cdot m_x / (1 + (n - a_x) \cdot m_x)$, which is the identity the protocol uses (equation 74). The Andreev-Kingkade value is a property of a one-year first interval that starts at birth, so when the first interval is wider or the table starts above age 0 the interval keeps the ordinary midpoint value instead.

The other three methods share the same basis, the standard lifetable identity $a_x = n + 1/m - n/q$ under a constant force of mortality (Preston, Heuveline and Guillot 2001, eq. 3.15), and differ in how they treat the first two intervals:

- "cfm": the identity alone, for every interval;
- "preston": the identity, with the first two intervals replaced by the Coale-Demeny West separation factors published by Preston et al. (2001), table 3.3, when sex is given;

- "coale_demeny": the identity, with the first two intervals replaced by the original 1983 Coale-Demeny rule expressed in qx, reproduced by the PAS software, when sex is given.

"preston" and "coale_demeny" agree exactly once $mx[1]$ reaches 0.107, and differ by at most a few thousandths of a year below it. Both coincide with "cfm" when sex is NULL. "cfm" and the Coale-Demeny variants adjust ax after the rates have been converted, so they leave the constant force of mortality conversion in place; only "andreev_kingkade" changes the conversion itself.

A value supplied for the open age interval is kept as given. Everybody alive there dies in that interval, so its own rate implies what the average time lived in it should be ($1/mx$); a value that differs is reported and left alone, which leaves the ax and mx columns of the table disagreeing at that age, or the model-implied value when close is set. The one exception is a table entered from ex, where the curve being inverted fixes that interval. See Details.

close	The method used to close the open age interval, named by the mortality-law code that implements it. NULL (the default) keeps the standard reciprocal close, $mx[N] = (\text{the observed rate})$. A code from availableLaws (e.g. "kannisto") closes the table instead with the model-implied average force of mortality over the open interval: the law is fitted to the closed intervals and the observed $mx[N]$ is replaced by $1/ex[N]$. This corrects the bias of the reciprocal close when the open interval begins at a young age, and it does not change the age grid. The standard row identities ($qx[N] = 1$, $ax[N] = ex[N] = 1/mx[N]$) still hold on the corrected rate. The same code is used when omega extends the table; when omega is set and close is NULL, the extrapolation defaults to "kannisto".
omega	The age at which to close the table when it should be extended beyond the input's open age. NULL (the default) keeps the table closed at the input's own open age. When omega is greater than the last age in x, the death rates from the open age up to omega are obtained by extrapolating the close law (see also fit_from) and the table is closed at omega. The input's own open interval is replaced by the extrapolated values, so a user-supplied ax is re-estimated on the extended grid. An omega that does not exceed the last age in x leaves the table unchanged, with a warning. Ignored by the in-place close, which operates on the input's own open age.
fit_from	The age from which the closing law is fitted. NULL (the default) uses 60 when the input reaches age 85, and the last 20 years of the input otherwise. Ignored when neither close nor omega is set.

Details

A life table (also called a mortality table or actuarial table) summarises the mortality experience of a population. For each age (or age interval) it reports:

- Death rates (mx) and death probabilities (qx)
- Survivorship (lx)
- Distribution of deaths (dx)
- Person-years lived (Lx) and total person-years remaining (Tx)
- Life expectancy (ex)

The life table is constructed sequentially: from the input data the function derives m_x , then q_x , then l_x , dx , L_x , T_x , and finally e_x . The conversion between m_x and q_x follows the a_x method. The default, $a_x = \text{"andreev_kingkade"}$, is the rule the Human Mortality Database applies to its period life tables (Methods Protocol, version 6): the Andreev-Kingkade (2015) formula for the first year of life and half the interval width elsewhere, converted through the exact interval identity so that the m_x , q_x and a_x columns are mutually consistent. The constant-force-of-mortality (CFM) assumption, $a_x = n + 1/m - n/q$, is available with $a_x = \text{"cfm"}$. When sex is given, the first two values of the a_x column can instead be adjusted using the Coale-Demeny method, which accounts for the different infant mortality patterns between males and females; two published parameterisations of that adjustment are available through $a_x = \text{"preston"}$, expressed in terms of m_x , and $a_x = \text{"coale_demeny"}$, expressed in terms of q_x and retrieved from m_x by the PAS inversion. The latter reproduces the coefficients used by the Coale-Demeny 1983 regional model tables and by the PAS software.

The e_x input runs the table backwards. The survivorship ratio of each closed interval follows from the identity

$$e_x l_x - e_{x+n} l_{x+n} = n L_x = a_x l_x + (n - a_x) l_{x+n}$$

, so that $r_x = l_{x+n}/l_x = (e_x - a_x)/(e_{x+n} + n - a_x)$. When a_x is supplied as a numeric vector the recovery is a single sweep. When it is not, each interval is solved so that the a_x the method in force would assign to the recovered interval reproduces the ratio it came from; the recovered table is then identical to the one the same e_x curve describes under that method. The open interval closes by the standard rule, $a_N = e_N$ and $m_N = 1/e_N$. A curve of life expectancy is allowed to rise with age at the youngest ages (life expectancy at birth is lower than at age one when infant mortality is high), so a rise is not an error; a curve that falls by more than the interval width, or below its own a_x , is not a feasible life table and is reported by age. Because e_x alone does not pin a_x , pass a_x alongside e_x when the a_x convention matters. One nuance follows from the forward build: when a_x is "preston" or "coale_demeny" and a sex is given, the forward table adjusts a_x in the first two intervals after converting the rates, so its stored m_x there does not satisfy the exact identity on its own a_x . The inverse returns the identity-consistent m_x , so in that one family the recovered m_x in the first two rows can differ from the forward table's at the third decimal while e_x , q_x and a_x still reproduce exactly.

Value

An object of class "LifeTable" containing the following components:

<code>lt</code>	A data.frame with the complete life table, including columns for age interval (<code>x.int</code>), exact age (<code>x</code>), death rate (<code>m_x</code>), death probability (<code>q_x</code>), person-years lived by decedents (<code>ax</code>), survivorship (<code>l_x</code>), death distribution (<code>dx</code>), person-years lived (<code>L_x</code>), total person-years remaining (<code>T_x</code>), and life expectancy (<code>e_x</code>).
<code>call</code>	The matched function call.
<code>process_date</code>	Timestamp of when the life table was computed.

Author(s)

Marius D. Pascariu

References

Coale, A. J., Demeny, P., and Vaughan, B. (1983). *Regional Model Life Tables and Stable Populations*. 2nd ed. New York: Academic Press.

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

When ax is supplied by the user as a numeric vector the conversion between mx and qx uses the exact interval identity $qx = nx * mx / (1 + (nx - ax) * mx)$ (and its inverse) instead of the CFM approximation, so that the mx , qx and ax columns of the result are mutually consistent. The "andreev_kingkade" method does the same, because it builds a numeric ax before the conversion. An ax that an interval cannot support ($ax * mx > 1$) is replaced with the implied average, $1/mx$, and the affected ages are reported.

The open (closing) age interval follows its own rule: $ax[N] = 1/mx[N]$, $ex[N] = 1/mx[N]$ and $Lx[N] = lx[N]/mx[N]$, which keeps the closed table consistent with $qx[N] = 1$. A user-supplied $ax[N]$ is therefore replaced, with a warning.

That rule assumes the force of mortality is roughly constant above the open age, which holds when the open interval is old (85+ or 90+) but not when the input stops early (70+ or 75+). The `close` argument addresses that directly: given a mortality-law code, the observed $mx[N]$ is replaced by the model-implied average over the open interval, $1/ex[N]$, without changing the age grid. The `omega` argument is a different response to the same problem: it extends the table to `omega` by extrapolating the rates and closes there. Either argument alone leaves the other's default in place, and the default (`close = NULL`, `omega = NULL`) is the standard reciprocal close, so results are unchanged.

Missing values in mx or qx are never masked. They localise to the interval quantities of the affected row (which are returned as NA) and to the cumulative Tx and ex columns at and before that row, while the survivorship chain lx is bridged across the gap, so the ages above it remain computable. A warning names the affected ages.

See Also

[LawTable](#) for generating life tables from a fitted parametric mortality law; [convertFx](#) for converting between mortality measures.

Examples

```
# Example 1 --- Full life tables with different inputs -----

y <- 1900
x <- as.numeric(rownames(ahmd$mx))
Dx <- ahmd$Dx[, paste(y)]
Ex <- ahmd$Ex[, paste(y)]

LT1 <- LifeTable(x, Dx = Dx, Ex = Ex)
LT2 <- LifeTable(x, mx = LT1$lt$mx)
LT3 <- LifeTable(x, qx = LT1$lt$qx)
LT4 <- LifeTable(x, lx = LT1$lt$lx)
LT5 <- LifeTable(x, dx = LT1$lt$dx)
LT6 <- LifeTable(x, ex = LT1$lt$ex)

LT1
LT5
LT6
ls(LT5)
```

```

# Example 2 --- Compute multiple life tables at once -----

LTs <- LifeTable(x, mx = ahmd$mx)
LTs
# A warning is printed if the input contains missing values.
# Some of the missing values can be handled automatically.

# Example 3 --- Abridged life table -----

x <- c(0, 1, seq(5, 110, by = 5))
mx <- c(.053, .005, .001, .0012, .0018, .002, .003, .004,
        .004, .005, .006, .0093, .0129, .019, .031, .049,
        .084, .129, .180, .2354, .3085, .390, .478, .551)
LT7 <- LifeTable(x, mx = mx, sex = "female")
LT7

# Example 4 --- Abridged life table using a custom 'ax' -----
# This example reuses the ages (x) and death rates (mx) from Example 3.
# Note that 'ax' must have the same length as 'x', otherwise an error
# will be returned.

my_ax <- c(0.1, 1.5, rep(2, 19), 1, 1, 1)

LT8 <- LifeTable(x = x, mx = mx, ax = my_ax)

# Example 5 --- The ax methods -----
# The default 'andreev_kingkade' follows the HMD Methods Protocol v6
# (Andreev-Kingkade a0, half-interval elsewhere). 'cfm' is the plain
# constant-force identity; 'preston' and 'coale_demeny' are the two
# Coale-Demeny conventions for the first two intervals (identical above
# m0 = 0.107).

LT9 <- LifeTable(x, mx = mx, sex = "female", ax = "andreev_kingkade")
LT10 <- LifeTable(x, mx = mx, sex = "female", ax = "cfm")
LT11 <- LifeTable(x, mx = mx, sex = "female", ax = "preston")
LT12 <- LifeTable(x, mx = mx, sex = "female", ax = "coale_demeny")

rbind(
  andreev_kingkade = LT9$lt$ax[1:2],
  cfm = LT10$lt$ax[1:2],
  preston = LT11$lt$ax[1:2],
  coale_demeny = LT12$lt$ax[1:2]
)

# Example 6 --- Closing the open interval accurately -----
# The data stop at 75+; closing there assumes a constant hazard above 75.
# 'close' argument corrects the open-interval rate with a fitted law, on the same
# age grid;
# 'omega' argument offers the option to extend the table to 110 before closing it.

x5 <- c(0, 1, seq(5, 75, by = 5))
mx5 <- c(.053, .005, .001, .0012, .0018, .002, .003, .004,
        .004, .005, .006, .0093, .0129, .019, .031, .049, .084)

```

```

LT13 <- LifeTable(x5, mx = mx5)
LT14 <- LifeTable(x5, mx = mx5, close = "kannisto")
LT15 <- LifeTable(x5, mx = mx5, close = "kannisto", omega = 110)

c(default = LT13$lt$ex[1], close = LT14$lt$ex[1], omega = LT15$lt$ex[1])

```

logLik.MortalityLaw *Extract the Log-Likelihood of a Fitted Mortality Law*

Description

Returns the maximised log-likelihood of a "MortalityLaw" fit. It is only defined when the objective was a likelihood, that is when the model was fitted with `opt.method = "poissonL"` or `"binomialL"`; for the loss-function objectives the value is `NaN`. For a multiple fit the log-likelihoods are returned as a named vector.

Usage

```

## S3 method for class 'MortalityLaw'
logLik(object, ...)

```

Arguments

<code>object</code>	An object of class "MortalityLaw".
<code>...</code>	Further arguments passed to or from other methods.

Value

An object of class "logLik" for a single fit, or a named numeric vector of log-likelihoods for a multiple fit.

See Also

[MortalityLaw](#); [AIC.MortalityLaw](#).

Examples

```

x <- 45:75
M1 <- MortalityLaw(x = x, Dx = ahmd$Dx[as.character(x), "1950"],
                  Ex = ahmd$Ex[as.character(x), "1950"],
                  law = "makeham", opt.method = "poissonL")

logLik(M1)

```

MortalityLaw

*Fit Mortality Laws***Description**

Fit parametric mortality models given a set of input data. The data can be supplied as death counts and mid-interval population estimates (D_x , E_x), age-specific death rates (m_x), or death probabilities (q_x). Use the `law` argument to specify the model to be fitted. Over 30 parametric models are currently implemented; run [availableLaws](#) to see the full list. Models can be fitted using maximum likelihood or by optimising a loss function. See the [availableLF](#) function for the implemented options.

Usage

```
MortalityLaw(x, Dx = NULL, Ex = NULL, mx = NULL, qx = NULL,
             law = NULL,
             opt.method = "LF2",
             parS = NULL,
             fit.this.x = x,
             custom.law = NULL,
             show = FALSE, ...)
```

Arguments

<code>x</code>	Numeric vector of ages at the beginning of each age interval. For a full life table, use single-year ages (e.g., 0:110). For an abridged life table, use the lower bound of each interval (e.g., c(0, 1, 5, 10, ..., 110)).
<code>Dx</code>	Death counts. Each element represents the total number of deaths during the calendar year to persons aged x to $x + n$ (where n is the length of the age interval). Must be provided together with <code>Ex</code> .
<code>Ex</code>	Exposure-to-risk in the period. This is usually approximated by the mid-year population aged x to $x + n$. Must be provided together with <code>Dx</code> .
<code>mx</code>	Age-specific death rate in the age interval $[x, x+n)$. Defined as D_x / E_x .
<code>qx</code>	Probability of dying within the age interval $[x, x+n)$.
<code>law</code>	The name of the mortality law to be used (e.g., "gompertz", "makeham"). Run availableLaws to see all options.
<code>opt.method</code>	The function to optimise. Available options: • "poissonL": Poisson log-likelihood. • "binomialL": Binomial log-likelihood. • "LF1": Squared relative error $(1 - \mu/\nu)^2$. • "LF2": Squared log-ratio $\log(\mu/\nu)^2$. • "LF3": Chi-squared-type $((\nu - \mu)^2)/\nu$. • "LF4": Squared error $(\nu - \mu)^2$. • "LF5": Deviance-type $(\nu - \mu) * \log(\nu/\mu)$.

- "LF6": Absolute error $\text{abs}(\text{nu} - \text{mu})$.

See [availableLF](#) for details.

<code>parS</code>	Optional starting parameter values for the optimisation. If NULL, sensible defaults are automatically chosen via <code>bring_parameters</code> .
<code>fit.this.x</code>	A subset of <code>x</code> over which to fit the model. The default is the entire <code>x</code> vector. Use this to exclude, for example, advanced ages where data are sparse.
<code>custom.law</code>	A user-defined function for fitting a model not included in the package. The function must accept arguments <code>x</code> (age vector) and <code>par</code> (named parameter vector) and return a list containing at least an element named <code>hx</code> (the hazard or force of mortality). See the examples below.
<code>show</code>	Logical. If TRUE, a progress bar is displayed during fitting. Default: FALSE.
<code>...</code>	Additional arguments passed to or from other methods.

Details

Optimisation: The PORT routines (via [nlminb](#)) are used for unconstrained and box-constrained optimisation. Parameters are estimated on the log scale to ensure positivity, and the routine is set to allow up to 5000 iterations. When the optimisation method is "poissonL" or "binomialL", the AIC, BIC and log-likelihood are computed from the likelihood. Otherwise these are set to NaN.

Scaling of the age vector: For models that cover only a portion of the lifespan (e.g., adult or old-age mortality), the age vector `x` is automatically re-scaled as $x = x - \min(x) + 1$ before fitting. This transformation improves numerical stability and helps the optimisation algorithm converge, especially when the starting age is far from zero. Models that apply this scaling are flagged with `SCALE_X = TRUE` in the table returned by [availableLaws](#). When using [predict.MortalityLaw](#) or [LawTable](#) with such models, the same scaling is applied internally, so predictions remain consistent with the fitted coefficients.

Handling matrix input: If `Dx`, `Ex`, `mx` or `qx` are provided as matrices (with one column per population or time period), the function iterates over the columns and fits a separate model to each, returning a collection of results.

Value

An object of class "MortalityLaw", which is a list with the following components:

<code>input</code>	List of input arguments, stored for reproducibility.
<code>info</code>	Model information (name, formula, date of fitting).
<code>coefficients</code>	Estimated parameters of the mortality law. A named vector for a single fit, or a matrix for multiple fits.
<code>fitted.values</code>	Fitted hazard rates (or death probabilities) evaluated at the input ages <code>x</code> .
<code>residuals</code>	Raw residuals, observed minus fitted values.
<code>deviance.residuals</code>	Deviance residuals. For the count cases (<code>Dx/Ex</code>) they are the Poisson deviance residuals; for the rate cases (<code>mx, qx</code>) they are the log-residuals.
<code>pearson.residuals</code>	Pearson residuals. For the count cases they are the Poisson Pearson residuals; for the rate cases they are the log-residuals.

<code>goodness.of.fit</code>	Named numeric vector (single fit) or matrix (one row per fit) with log-likelihood, AIC and BIC (NaN for non-likelihood methods). For count fits the log-likelihood is the Poisson or binomial kernel: the data-only additive constants are dropped, which leaves model comparison (AIC/BIC) unaffected but makes the absolute value differ from <code>glm</code> .
<code>opt.diagnosis</code>	Object returned by the optimisation routine, useful for checking convergence.
<code>df</code>	Number of parameters, residual degrees of freedom and the dispersion.
<code>dispersion</code>	Dispersion of the fit. For the count cases it is the Pearson chi-square divided by the residual degrees of freedom (the GLM dispersion, 1 for a correctly specified Poisson model); for the rate cases it is the mean squared log-residual.
<code>deviance</code>	The deviance of the fit. For the count cases this is the Poisson deviance, the quantity minimised by "poissonL"; for the rate cases it is the sum of squared log-residuals.

Author(s)

Marius D. Pascariu

See Also

[availableLaws](#) for a list of all implemented models; [availableLF](#) for loss function details; [LifeTable](#) for life table construction; [ReadHMD](#) for downloading data from the Human Mortality Database.

Examples

```
# Example 1: Fitting the Makeham model -----
x <- 45:75
Dx <- ahmd$Dx[paste(x), "1950"]
Ex <- ahmd$Ex[paste(x), "1950"]

M1 <- MortalityLaw(x = x, Dx = Dx, Ex = Ex, law = 'makeham')

M1
ls(M1)
coef(M1)
summary(M1)
fitted(M1)
predict(M1, x = 45:95)
plot(M1, which = 'fit')
plot(M1, which = 'diagnostics')

# Example 2: -----
# We can fit the same model using a different data format
# and a different optimization method.
x <- 45:75
mx <- ahmd$mx[paste(x), ]
M2 <- MortalityLaw(x = x, mx = mx, law = 'makeham', opt.method = 'LF1')
M2
```

```

fitted(M2)
predict(M2, x = 55:90)

# Example 3: -----
# Now let's fit a mortality law that is not defined
# in the package, say a reparameterized Gompertz in
# terms of modal age at death
#  $h_x = b \exp(b(x-m))$  (here  $b$  and  $m$  are the parameters to be estimated)

# A function with 'x' and 'par' as input has to be defined, which returns
# at least an object called 'hx' (hazard rate).
missov <- function(x, par = c(b = 0.13, M = 45)){
  hx <- with(as.list(par), b*exp(b*(x - M)) )
  return(as.list(environment()))
}

M3 <- MortalityLaw(x = x, Dx = Dx, Ex = Ex, custom.law = missov)
summary(M3)
plot(M3)
# predict M3 for different ages
predict(M3, x = 85:130)

# Example 4: -----
# Fit Heligman-Pollard model for a single
# year in the dataset between age 0 and 100 and build a life table.

x <- 0:100
mx <- ahmd$mx[paste(x), "1950"] # select data
M4 <- MortalityLaw(x = x, mx = mx, law = 'HP', opt.method = 'LF2')
M4
plot(M4, which = 'fit')
plot(M4, which = 'diagnostics')
plot(M4, which = 'both')

LifeTable(x = x, qx = fitted(M4))

```

Description

Fit the most popular human mortality 'laws', and construct full and abridged life tables given various input indices. A mortality law is a parametric function that describes the dying-out process of individuals in a population during a significant portion of their life spans. For a comprehensive review of the most important mortality laws see Tabeau (2001) [doi:10.1007/0306475626_1](https://doi.org/10.1007/0306475626_1). Practical functions for downloading data from various human mortality databases are provided as well.

Details

To learn more about the package, start with the vignettes: `browseVignettes(package = "MortalityLaws")`

Author(s)

Maintainer: Marius D. Pascariu <mpascariu@outlook.com> ([ORCID](#)) [copyright holder]

Authors:

- Marius D. Pascariu <mpascariu@outlook.com> ([ORCID](#)) [copyright holder]

Other contributors:

- Vladimir Canudas-Romo [contributor]

See Also

Useful links:

- <https://mpascariu.github.io/MortalityLaws/>
- <https://github.com/mpascariu/MortalityLaws>
- Report bugs at <https://github.com/mpascariu/MortalityLaws/issues>

plot.convertFx

Plot a Life Table Indicator Conversion

Description

Draws the result of `convertFx` in the house style: the input indicator in one panel and the converted output indicator in the other, sharing the age axis when they are stacked. Rates and probabilities (mx, qx) go on a log scale. A matrix input draws one curve per column in both panels.

Usage

```
## S3 method for class 'convertFx'
plot(x, split = NULL, ...)
```

Arguments

x	An object of class "convertFx", as returned by <code>convertFx</code> .
split	How to arrange the two panels: NULL (the default) puts them side by side, c(1, 2); give a length-2 integer c(nrow, ncol) to split them yourself, such as c(2, 1) to stack the output under the input on a shared age axis.
...	Further arguments; currently ignored.

Value

The object x, invisibly. Called for the plot it draws.

Author(s)

Marius D. Pascariu

See Also[convertFx.](#)**Examples**

```
# convert death rates (mx) to life expectancies (ex)
x <- 0:105
mx <- ahmd$mx[paste(x), "2010"]
ex <- convertFx(x = x, data = mx, from = "mx", to = "ex")
plot(ex)
plot(ex, split = c(2, 1))

# convert life expectancy (ex) to 1-year probability of dying (qx)
qx <- convertFx(x = x, data = ex, from = "ex", to = "qx")
plot(qx)
```

plot.LifeTable	<i>Plot a Life Table</i>
----------------	--------------------------

Description

Draws the classic life table figures in the house style: the survivorship l_x , the hazard m_x on a log scale, the death distribution d_x and the life expectancy e_x . When the table holds several life tables, from a matrix input or from a [LawTable](#) built on several parameter sets, each table is drawn as one curve and the legend labels the tables.

Usage

```
## S3 method for class 'LifeTable'
plot(x, which = c("all", "lx", "hazard", "dx", "ex"), split = NULL, ...)
```

Arguments

<code>x</code>	An object of class "LifeTable".
<code>which</code>	Which panels to draw: "all" (the default; the four panels), or a single panel: "lx", "hazard", "dx" or "ex".
<code>split</code>	How to arrange the panels when more than one is drawn: NULL (the default) draws the four panels in a c(2, 2) grid; give a length-2 integer c(nrow, ncol) to split them yourself, such as c(1, 4) for one row or c(4, 1) for one column.
<code>...</code>	Further arguments; currently ignored.

Value

The object `x`, invisibly. Called for the plot it draws.

Author(s)

Marius D. Pascariu

See Also

[LifeTable](#), [LawTable](#).

Examples

```
x <- as.numeric(rownames(ahmd$mx))
LT <- LifeTable(x = x, mx = ahmd$mx[, c("1950", "2010")])
plot(LT)
plot(LT, which = "lx")
plot(LT, split = c(1, 4))
```

plot.MortalityLaw	<i>Plot a Fitted Mortality Law</i>
-------------------	------------------------------------

Description

Draws the figures of a "MortalityLaw" fit in the house style. The fit chart puts the observed and the fitted mortality on a log scale, with the fitted age range shaded and a goodness-of-fit subtitle. The residual diagnostics give the deviance residuals against age and against the fitted values, each with a band of two standard deviations and a lowess smooth, plus a normal Q-Q plot and the residual distribution with a normal density.

Usage

```
## S3 method for class 'MortalityLaw'
plot(x, which = c("both", "fit", "diagnostics"), split = NULL, ...)
```

Arguments

<code>x</code>	An object of class "MortalityLaw".
<code>which</code>	Which figure to draw: "both" (the default; the fit chart and the residual panels in one figure), "fit" (the fit chart alone) or "diagnostics" (the four residual panels alone).
<code>split</code>	How to arrange the four diagnostic panels: NULL (the default) draws them in a c(2, 2) grid; give a length-2 integer c(nrow, ncol) to split them yourself, such as c(1, 4) for one row or c(4, 1) for one column. Ignored when only the fit chart is drawn.
<code>...</code>	Further arguments; currently ignored.

Value

The object `x`, invisibly. Called for the figures it draws.

Author(s)

Marius D. Pascariu

See Also[MortalityLaw](#).**Examples**

```
x <- 45:75
M1 <- MortalityLaw(x = x, Dx = ahmd$Dx[as.character(x), "1950"],
                  Ex = ahmd$Ex[as.character(x), "1950"], law = "makeham")
plot(M1, which = "fit")
plot(M1, which = "diagnostics")
plot(M1, which = "diagnostics", split = c(1, 4))
plot(M1, which = "both")
```

predict.MortalityLaw *Predict from a Fitted Mortality Law*

Description

Evaluates a fitted mortality law at new ages. The coefficients are reused as they are, so the prediction is an extrapolation of the fitted curve: it is meaningful over the ages that the law describes and becomes unreliable far outside the fitted range. Models that scale the age vector during fitting (the SCALE_X column of [availableLaws](#)) are rescaled internally, so the prediction stays consistent with the coefficients.

Usage

```
## S3 method for class 'MortalityLaw'
predict(object, x, ...)
```

Arguments

object	An object of class "MortalityLaw".
x	Vector of ages at which to evaluate the fitted law.
...	Additional arguments affecting the predictions produced.

Value

A named vector of predicted mortality values for a single fit, or a matrix with one column per fit. The values are hazards ($\mu[x]$) or death probabilities ($q[x]$), depending on the law; see the FIT column of [availableLaws](#).

Author(s)

Marius D. Pascariu

See Also[MortalityLaw](#); [fitted](#).

Examples

```
# Extrapolate old-age mortality with the Kannisto model
# Fit ages 80-94 and extrapolate up to 120.

Mx <- ahmd$mx[paste(80:94), "1950"]
M1 <- MortalityLaw(x = 80:94, mx = Mx, law = 'kannisto')
fitted(M1)
predict(M1, x = 80:120)

# See more examples in MortalityLaw function help page.
```

ReadAHMD

Download the Australian Human Mortality Database (AHMD)

Description

Download detailed mortality and population data for different provinces and territories in Australia, in a single object from the [Australian Human Mortality Database](#).

Usage

```
ReadAHMD(what, regions = NULL, interval = "1x1", save = FALSE, show = TRUE)
```

Arguments

what	What type of data are you looking for? The following options might be available for some or all the countries and regions: • "births" – birth records (the 1-year product only; HMD does not publish births by 5-year age group or by multi-year periods); • "Dx_lexis" – deaths by Lexis triangles; • "Ex_lexis" – exposure-to-risk by Lexis triangles; • "population" – population size; • "Dx" – death counts; • "Ex" – exposure-to-risk; • "mx" – central death-rates; • "LT_f" – period life tables for females; • "LT_m" – period life tables for males; • "LT_t" – period life tables both sexes combined; • "e0" – period life expectancy at birth; • "Exc" – cohort exposures; • "mxc" – cohort death-rates; • "LT_fc" – cohort life tables for females; • "LT_mc" – cohort life tables for males; • "LT_tc" – cohort life tables both sexes combined; • "e0c" – cohort life expectancy at birth;
------	---

regions	Specify the region specific data you want to download by adding the AHMD region code/s. Options: • "ACT" – Australian Capital Territory; • "NSW" – New South Wales; • "NT" – Northern Territory; • "QLD" – Queensland; • "SA" – South Australia; • "TAS" – Tasmania; • "VIC" – Victoria; • "WA" – Western Australia; • NULL – if NULL data for all the regions are downloaded.
interval	Datasets are given in various age and time formats based on which the records are aggregated. Interval options: • "1x1" – by age and year; • "1x5" – by age and 5-year time interval; • "1x10" – by age and 10-year time interval; • "5x1" – by 5-year age group and year; • "5x5" – by 5-year age group and 5-year time interval; • "5x10" – by 5-year age group and 10-year time interval.
save	Do you want to save a copy of the dataset on your local machine? Logical. Default: FALSE.
show	Choose whether to display a progress bar. Logical. Default: TRUE.

Details

The Australian Human Mortality Database is a "satellite" of the Human Mortality Database, built with the same methodology, so the two are directly comparable. It covers Australia, its states and its territories. See the AHMD website for its history and research team. The database is open, so no login is needed.

Value

A ReadAHMD object that contains:

input	List with the input values;
data	Data downloaded from AHMD;
download.date	Time stamp;
years	Numerical vector with the years covered in the data;
ages	Numerical vector with ages covered in the data.

Author(s)

Marius D. Pascariu

See Also[ReadHMD ReadCHMD](#)**Examples**

```
## Not run:
# Download demographic data for Australian Capital Territory and
# Tasmania regions in 5x1 format

# Death counts. We don't want to export data outside R.
AHMD_Dx <- ReadAHMD(what = "Dx",
                    regions = c('ACT', 'TAS'),
                    interval = "5x1",
                    save = FALSE)

AHMD_Dx

# Download life tables for female population in all the states and export data.
LTF <- ReadAHMD(what = "LT_f", interval = "5x1", save = FALSE)
LTF

## End(Not run)
```

ReadCHMD

*Download the Canadian Human Mortality Database (CHMD)***Description**

Download detailed mortality and population data for different provinces and territories in Canada, in a single object from the Canadian Human Mortality Database.

Usage

```
ReadCHMD(what, regions = NULL, interval = "1x1", save = FALSE, show = TRUE)
```

Arguments

what	What type of data are you looking for? The following options are available: • "births" – birth records; • "Dx_lexis" – deaths by Lexis triangles; • "population" – population size; • "Dx" – death counts; • "Ex" – exposure-to-risk; • "mx" – central death-rates; • "LT_f" – period life tables for females; • "LT_m" – period life tables for males; • "LT_t" – period life tables both sexes combined;
------	---

	• "e0" – period life expectancy at birth;
regions	Specify the region specific data you want to download by adding the CHMD region code/s. Options: • "CAN" – Canada - Sum of Canadian provinces and territories; • "NFL" – Newfoundland & Labrador; • "PEI" – Prince Edward Island; • "NSC" – Nova Scotia; • "NBR" – New Brunswick; • "QUE" – Quebec; • "ONT" – Ontario; • "MAN" – Manitoba; • "SAS" – Saskatchewan; • "ALB" – Alberta; • "BCO" – British Columbia; • "NWT" – Northwest Territories & Nunavut; • "YUK" – Yukon; • NULL – if NULL data for all the regions are downloaded.
interval	Datasets are given in various age and time formats based on which the records are aggregated. Interval options: • "1x1" – by age and year; • "1x5" – by age and 5-year time interval; • "1x10" – by age and 10-year time interval; • "5x1" – by 5-year age group and year; • "5x5" – by 5-year age group and 5-year time interval; • "5x10" – by 5-year age group and 10-year time interval.
save	Do you want to save a copy of the dataset on your local machine? Logical. Default: FALSE.
show	Choose whether to display a progress bar. Logical. Default: TRUE.

Details

The Canadian Human Mortality Database is a "satellite" of the Human Mortality Database, built with the same methodology, so the two are directly comparable. It covers Canada, its provinces and its territories. See the CHMD website for its history and research team; the data are validated and corrected for the period it covers.

Value

A ReadCHMD object that contains:

input	List with the input values;
data	Data downloaded from CHMD;
download.date	Time stamp;
years	Numerical vector with the years covered in the data;
ages	Numerical vector with ages covered in the data.

Author(s)

Marius D. Pascariu

See Also

[ReadHMD](#) [ReadAHMD](#)

Examples

```
## Not run:
# Download demographic data for Quebec and Saskatchewan regions in 1x1 format

# Death counts. We don't want to export data outside R.
CHMD_Dx <- ReadCHMD(what = "Dx",
                    regions = c('QUE', 'SAS'),
                    interval = "1x1",
                    save = FALSE)

# Download life tables for female population. To export data use save = TRUE.
LTF <- ReadCHMD(what = "LT_f",
               regions = c('QUE', 'SAS'),
               interval = "1x1",
               save = FALSE)

## End(Not run)
```

ReadHMD

Download The Human Mortality Database (HMD)

Description

Download detailed mortality and population data for different countries and regions in a single object from the [Human Mortality Database](#).

Usage

```
ReadHMD(
  what,
  countries = NULL,
  interval = "1x1",
  username,
  password,
  save = FALSE,
  show = TRUE
)
```

Arguments

what	What type of data are you looking for? The following options might be available for some or all the countries and regions: • "births" – birth records (the 1-year product only; HMD does not publish births by 5-year age group or by multi-year periods); • "Dx_lexis" – deaths by Lexis triangles; • "Ex_lexis" – exposure-to-risk by Lexis triangles; • "population" – population size; • "Dx" – death counts; • "Ex" – exposure-to-risk; • "mx" – central death-rates; • "LT_f" – period life tables for females; • "LT_m" – period life tables for males; • "LT_t" – period life tables both sexes combined; • "e0" – period life expectancy at birth; • "Exc" – cohort exposures; • "mxc" – cohort death-rates; • "LT_fc" – cohort life tables for females; • "LT_mc" – cohort life tables for males; • "LT_tc" – cohort life tables both sexes combined; • "e0c" – cohort life expectancy at birth;
countries	Specify the country data you want to download by adding the HMD country code/s. Options: "AUS", "AUT", "BEL", "BGR", "BLR", "CAN", "CHL", "HRV", "HKG", "CHE", "CZE", "DEUTNP", "DEUTE", "DEUTW", "DNK", "ESP", "EST", "FIN", "FRATNP", "FRACNP", "GRC", "HUN", "IRL", "ISL", "ISR", "ITA", "JPN", "KOR", "LTU", "LUX", "LVA", "NLD", "NOR", "NZL_NP", "NZL_MA", "NZL_NM", "POL", "PRT", "RUS", "SVK", "SVN", "SWE", "TWN", "UKR", "GBR_NP", "GBRTENW", "GBRCENW", "GBR_SCO", If NULL data for all the countries are downloaded at once;
interval	Datasets are given in various age and time formats based on which the records are aggregated. Interval options: • "1x1" – by age and year; • "1x5" – by age and 5-year time interval; • "1x10" – by age and 10-year time interval; • "5x1" – by 5-year age group and year; • "5x5" – by 5-year age group and 5-year time interval; • "5x10" – by 5-year age group and 10-year time interval.
username	Your HMD username. If you don't have one you can sign up for free on the Human Mortality Database website.
password	Your HMD password.
save	Do you want to save a copy of the dataset on your local machine? Logical. Default: FALSE.
show	Choose whether to display a progress bar. Logical. Default: TRUE.

Details

The Human Mortality Database is the reference source of detailed national mortality and population data; see the project's own pages for its history and research teams. A free account (and acceptance of the user agreement) is required, and a dataset is only as detailed as the country publishes: not every what exists for every country.

The login is performed once per call and reused for every country, so a long `countries` vector costs one authentication. The password is never stored in the returned object; `input` carries everything else, for reproducibility.

Value

A ReadHMD object that contains:

<code>input</code>	List with the input values (except the password).
<code>data</code>	Data downloaded from HMD.
<code>download.date</code>	Time stamp.
<code>years</code>	Numerical vector with the years covered in the data.
<code>ages</code>	Numerical vector with ages covered in the data.

Author(s)

Marius D. Pascariu

Examples

Not run:

```
# Download demographic data for 3 countries in 1x1 format
age_int <- 1 # age interval: 1,5
year_int <- 1 # year interval: 1,5,10
interval <- paste0(age_int, "x", year_int) # --> 1x1
# And the 3 countries: Sweden Denmark and USA. We have to use the HMD codes
cntr <- c('SWE', 'DNK', 'USA')

# Download death counts. We don't want to export data outside R.
HMD_Dx <- ReadHMD(what = "Dx",
                  countries = cntr,
                  interval = interval,
                  username = "user@email.com",
                  password = "password",
                  save = FALSE)

HMD_Dx

# Download life tables for female population and export data.
LTF <- ReadHMD(what = "LT_f",
              countries = cntr,
              interval = interval,
              username = "user@email.com",
```

```

password = "password",
save = TRUE)

LTF

## End(Not run)

```

ReadJMD

Download the Japanese Mortality Database (JMD)

Description

Download detailed mortality and population data of the 47 prefectures in Japan, in a single object. The source of data is the [Japanese Mortality Database](#).

Usage

```
ReadJMD(what, regions = NULL, interval = "1x1", save = FALSE, show = TRUE)
```

Arguments

what	What type of data are you looking for? The following options are available for JMD: • "births" – birth records; • "population" – population size; • "Dx" – death counts; • "Ex" – exposure-to-risk; • "mx" – central death-rates; • "LT_f" – period life tables for females; • "LT_m" – period life tables for males; • "LT_t" – period life tables both sexes combined; • "e0" – period life expectancy at birth;
regions	Specify the region specific data you want to download by adding the JMD region code/s. Options: "Japan", "Hokkaido", "Aomori", "Iwate", "Miyagi", "Akita", "Yamagata", "Fukushima", "Ibaraki", "Tochigi", "Gunma", "Saitama", "Chiba", "Tokyo", "Kanagawa", "Niigata", "Toyama", "Ishikawa", "Fukui", "Yamanashi", "Nagano", "Gifu", "Shizuoka", "Aichi", "Mie", "Shiga", "Kyoto", "Osaka", "Hyogo", "Nara", "Wakayama", "Tottori", "Shimane", "Okayama", "Hiroshima", "Yamaguchi", "Tokushima", "Kagawa", "Ehime", "Kochi", "Fukuoka", "Saga", "Nagasaki", "Kumamoto", "Oita", "Miyazaki", "Kagoshima", "Okinawa". If NULL data for all the regions are downloaded at once.
interval	Datasets are given in various age and time formats based on which the records are aggregated. Interval options: • "1x1" – by age and year; • "1x5" – by age and 5-year time interval; • "1x10" – by age and 10-year time interval;

	• "5x1" – by 5-year age group and year; • "5x5" – by 5-year age group and 5-year time interval; • "5x10" –by 5-year age group and 10-year time interval.
save	Do you want to save a copy of the dataset on your local machine? Logical. Default: FALSE.
show	Choose whether to display a progress bar. Logical. Default: TRUE.

Details

The Japanese Mortality Database is a mortality database reorganised to be consistent with the Human Mortality Database, for all Japan and by prefecture; see the JMD website for its research project and methods. The database is open, so no login is needed. Its life tables are built to be internationally comparable, so they need not match the official Japanese life tables, which use a different base population and estimation method.

The region codes are prefecture names, not the numeric JIS codes used in the server's folder names; the reader maps one to the other internally.

Value

A ReadJMD object that contains:

input	List with the input values;
data	Data downloaded from JMD;
download.date	Time stamp;
years	Numerical vector with the years covered in the data;
ages	Numerical vector with ages covered in the data.

Author(s)

Marius D. Pascariu

See Also

[ReadHMD](#) [ReadCHMD](#)

Examples

```
## Not run:
# Download demographic data for Fukushima and Tokyo regions in 1x1 format

# Death counts. We don't want to export data outside R.
JMD_Dx <- ReadJMD(what = "Dx",
                  regions = c('Fukushima', 'Tokyo'),
                  interval = "1x1",
                  save = FALSE)

JMD_Dx

# Download life tables for female population in all the states and export data.
```

```
LTF <- ReadJMD(what = "LT_f", interval = "5x5", save = FALSE)
LTF

## End(Not run)
```

summary.MortalityLaw *Summarise a Fitted Mortality Law*

Description

Collects the fitted coefficients, the fit measures and a summary of the residuals into a compact object for printing, and rounds them to digits. The fit measures cover the fit window (the ages fitted and the ages reported), the optimisation (method used, optimiser outcome), the deviance with its degrees of freedom and dispersion, and the R-squared and RMSE of the fit. For a likelihood-based fit the maximised log-likelihood with its information criteria is included. When more than four curves were fitted only the first and the last two are kept in the printed coefficients and fit measures, so the output fits on one screen.

Usage

```
## S3 method for class 'MortalityLaw'
summary(object, ..., digits = max(3L, getOption("digits") - 3L))
```

Arguments

object	An object of class "MortalityLaw".
...	Additional arguments affecting the summary produced.
digits	Number of significant digits to display.

Value

An object of class "summary.MortalityLaw", a list holding the model information, the matched call, the rounded coefficients, the goodness-of-fit measures, the deviance and the degrees of freedom, the R-squared and RMSE of the fit, the optimisation outcome, the fit window and the residual summaries on the raw and the deviance scale.

See Also

[MortalityLaw](#) to fit a law; [coef](#) and [fitted](#) for the extracted values.

Examples

```
x <- 45:75
M1 <- MortalityLaw(x = x, Dx = ahmd$Dx[as.character(x), "1950"],
                  Ex = ahmd$Ex[as.character(x), "1950"], law = "makeham")
summary(M1)
```

Index

* datasets

ahmd, [2](#)

ahmd, [2](#)

AIC.MortalityLaw, [3](#), [23](#)

availableHMD, [4](#)

availableLaws, [5](#), [13](#), [15](#), [19](#), [24–26](#), [31](#)

availableLF, [8](#), [8](#), [24–26](#)

coef, [41](#)

convertFx, [9](#), [21](#), [28](#), [29](#)

deviance.MortalityLaw, [11](#), [13](#)

df.residual.MortalityLaw, [11](#)

dispersion, [11](#), [12](#)

dispersion(dispersion.MortalityLaw), [12](#)

dispersion.MortalityLaw, [12](#)

fitted, [31](#), [41](#)

LawTable, [5](#), [8](#), [10](#), [13](#), [21](#), [25](#), [29](#), [30](#)

LifeTable, [9](#), [10](#), [13](#), [15](#), [17](#), [26](#), [30](#)

logLik.MortalityLaw, [3](#), [23](#)

MortalityLaw, [3](#), [5](#), [6](#), [8](#), [11–15](#), [23](#), [24](#), [31](#), [41](#)

MortalityLaws, [27](#)

MortalityLaws-package (MortalityLaws),
[27](#)

nlminb, [25](#)

plot.convertFx, [10](#), [28](#)

plot.LifeTable, [29](#)

plot.MortalityLaw, [30](#)

predict.MortalityLaw, [25](#), [31](#)

ReadAHMD, [32](#), [36](#)

ReadCHMD, [34](#), [34](#), [40](#)

ReadHMD, [2–4](#), [26](#), [34](#), [36](#), [36](#), [40](#)

ReadJMD, [39](#)

summary.MortalityLaw, [12](#), [41](#)