Package 'MortalityLaws'

November 6, 2018

Type Package

Title Parametric Mortality Models, Life Tables and HMD

Version 1.7.0

Date 2018-11-06

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

Description Fit the most popular human mortality 'laws', and construct full and abridge 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>. An elegant function for downloading data from Human Mortality Database <https://www.mortality.org> is provided as well.

License GPL-3

LazyData TRUE

Depends R (>= 3.0.0)

Imports minpack.lm (>= 1.2), RCurl (>= 1.95), pbapply (>= 1.3-4), tidyr (>= 0.8.1)

Suggests testthat, knitr, rmarkdown

URL https://github.com/mpascariu/MortalityLaws

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

RoxygenNote 6.1.0

VignetteBuilder knitr

NeedsCompilation no

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

Repository CRAN

Date/Publication 2018-11-06 10:20:08 UTC

R topics documented:

`ahmd` *MortalityLaws Test Data*

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. The data-set 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

An object of class `list` of length 3.

Source

Human Mortality Database

See Also

`ReadHMD`

Examples

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

`availableHMD` *Check Data Availability in HMD*

Description

The function returns information about available data in HMD (period life tables etc.), with the range of years covered by the life tables.

Usage

```
availableHMD(username, password, ...)
```

Arguments

`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.
`...`	Other parameters to be passed in `ReadHMD` function.

Value

An `availableHMD` object.

Author(s)

Marius D. Pascariu

See Also

`ReadHMD`

Examples

```
## Not run:
# This will take few seconds...
datainfo <- availableHMD(username = "your_username",
                         password = "your_password")
datainfo

## End(Not run)
```

`availableLaws` *Check Available Mortality Laws*

Description

The function returns information about the parametric models that can be called and fitted in the `MortalityLaw` function. For a comprehensive review of the most important mortality laws, 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.

Value

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

`table`	Table with mortality models and codes to be used in `MortalityLaw`
`legend`	Table with details about the section of the mortality curve

Author(s)

Marius D. Pascariu

References

1. 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.
2. 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.
3. 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.
4. 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.
5. Wittstein, T. and D. Bumsted. (1883). The Mathematical Law of Mortality. Journal of the Institute of Actuaries and Assurance Magazine, 24(3), 153-173.

6. Steffensen, J. (1930). Infantile mortality from an actuarial point of view. Skandinavisk Aktuarietidskrift 13, 272-286.
7. Perks, W. (1932). On Some Experiments in the Graduation of Mortality Statistics. Journal of the Institute of Actuaries, 63(1), 12-57.
8. Harper, F. S. (1936). An actuarial study of infant mortality. Scandinavian Actuarial Journal 1936 (3-4), 234-270.
9. Weibull, W. (1951). A statistical distribution function of wide applicability. Journal of applied mechanics 103, 293-297.
10. 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.
11. 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.
12. Siler, W. (1979), A Competing-Risk Model for Animal Mortality. Ecology, 60: 750-757.
13. Heligman, L., & Pollard, J. (1980). The age pattern of mortality. Journal of the Institute of Actuaries, 107(1), 49-80.
14. 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
15. 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)
16. Carriere J.F. (1992). Parametric models for life tables. Transactions of the Society of Actuaries. Vol.44
17. Kostaki A. (1992). A nine-parameter version of the Heligman-Pollard formula. Mathematical Population Studies. Vol. 3 277-288
18. 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
19. 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
20. Finkelstein M. (2012) Discussing the Strehler-Mildvan model of mortality Demographic Research, Vol. 26(9), 191-206.

See Also

`MortalityLaw`

Examples

```
availableLaws()
```

availableLF *Check Available Loss Function*

Description

The function returns information about the implemented loss function used by the optimization procedure in the `MortalityLaw` function.

Usage

```
availableLF()
```

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

Easy conversion between the life table indicators. This function is based on the `LifeTable` function and methods behind it.

Usage

```
convertFx(x, data, from, to, ...)
```

Arguments

x	Vector of ages at the beginning of the age interval.
data	Vector or data.frame/matrix containing the mortality indicators.
from	Specify the life table indicator in the input `data`. Character. Options: `mx, qx, dx, lx`.
to	What indicator would you like to obtain? Character. Options: `mx, qx, dx, lx, Lx, Tx, ex`.
...	Further arguments to be passed to the `LifeTable` function with impact on the results to be produced.

Author(s)

Marius D. Pascariu

See Also

`LifeTable`

Examples

```
# Data ---
x  <- 0:110
mx <- ahmd$mx

# mx to qx
qx <- convertFx(x, data = mx, from = "mx", to = "qx")
# mx to dx
dx <- convertFx(x, data = mx, from = "mx", to = "dx")
# mx to lx
lx <- convertFx(x, data = mx, from = "mx", to = "lx")


# There are 28 possible combinations --------------------------------
# Let generate all of them.
from <- c("mx", "qx", "dx", "lx")
to   <- c("mx", "qx", "dx", "lx", "Lx", "Tx", "ex")
K    <- expand.grid(from = from, to = to) # all possible cases/combinations

for (i in 1:nrow(K)) {
  In  <- as.character(K[i, "from"])
  Out <- as.character(K[i, "to"])
  N <- paste0(Out, "_from_", In)
  cat(i, " Create", N, "\n")
  # Create the 28 sets of results
  assign(N, convertFx(x = x, data = get(In), from = In, to = Out))
}
```

LawTable *Compute Life Tables from Parameters of a Mortality Law*

Description

Compute Life Tables from Parameters of a Mortality Law

Usage

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

Arguments

`x`	Vector of ages at the beginning of the age interval.
`par`	The parameters of the mortality model.
`law`	The name of the mortality law/model to be used. e.g. `gompertz`, `makeham`, ... To investigate all the possible options, see `availableLaws` function.
`sex`	Sex of the population considered here. Default: `NULL`. This argument affects the first two values in the life table ax column. If sex is specified the values are computed based on the Coale-Demeny method and are slightly different for males than for females. Options: `NULL, male, female, total`.
`lx0`	Radix. Default: 100 000.
`ax`	Numeric scalar. Subject-time alive in age-interval for those who die in the same interval. If `NULL` this will be estimated. A common assumption is `ax = 0.5`, i.e. the deaths occur in the middle of the interval. Default: `NULL`.

Details

The "life table" is also called "mortality table" or "actuarial table". This shows, for each age, what the probability is that a person of that age will die before his or her next birthday, the expectation of life across different age ranges or the survivorship of people from a certain population.

Value

The output is of the `"LifeTable"` class with the components:

`lt`	Computed life table;
`call`	`Call` in which all of the specified arguments are specified by their full names;
`process_date`	Time stamp.

Author(s)

Marius D. Pascariu

See Also

`LifeTable` `MortalityLaw`

Examples

```
# Example 1 --- Makeham --- 4 tables ----------
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 = 45:100, par = C1, law = L1)

# WARNING!!!

# It is important to know how the coefficients have been estimated. If the
# fitting of the model was done over the [x, x+) age-range, the LawTable
# function can be used to create a life table only for age x onward.

# What can go wrong?

# ** Example 1B - is OK.
LawTable(x = 45:100, par = c(0.00717, 0.07789, 0.00363), law = L1)

# ** Example 1C - Not OK, because the life expectancy at age 25 is
# equal with life expectancy at age 45 in the previous example.
LawTable(x = 25:100, par = c(0.00717, 0.07789, 0.00363), law = L1)

# Why is this happening?

# If we have a model that covers only a part of the human mortality curve
# (e.g. adult mortality), in fitting the x vector is scaled down, meaning age (x) becomes
# (x - min(x) + 1). And, the coefficients are estimated on a scaled x in ordered
# to obtain meaningful estimates. Otherwise the optimization process might
# not converge.

# What can we do about it?

# a). Know which mortality laws are rescaling the x vector in the fitting process.
# If these models are fitted with the MortalityLaw() function, you can find out
# like so:
A <- availableLaws()$table
A[, c("CODE", "SCALE_X")]

# b). If you are using one of the models that are applying scaling,
# be aware over what age-range the coefficients have been estimated. If they
# have been estimated using, say, ages 50 to 80, you can use the
# LawTable() to build a life tables from age 50 onwards.


# Example 2 --- Heligman-Pollard -- 1 table ----
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" is not scaling down the x vector, the output is not affected by
# the problem described above.

# Check
LawTable(x = 3:110, par = C2, law = L2)
# Note the e3 = 70.31 in both tables
```

LifeTable *Compute Life Tables from Mortality Data*

Description

Construct either a full or abridged life table with various input choices like: death counts and mid-interval population estimates (`Dx, Ex`) or age-specific death rates (`mx`) or death probabilities (`qx`) or survivorship curve (`lx`) or a distribution of deaths (`dx`). If one of these options is specified, the other can be ignored. The input data can be an object of class: numerical `vector`, `matrix` or `data.frame`.

Usage

```
LifeTable(x, Dx = NULL, Ex = NULL,
             mx = NULL,
             qx = NULL,
             lx = NULL,
             dx = NULL,
             sex = NULL,
             lx0 = 1e5,
             ax  = NULL)
```

Arguments

`x`	Vector of ages at the beginning of the age interval.
`Dx`	Object containing death counts. An element of the `Dx` object represents the number of deaths during the year to persons aged x to x+n.
`Ex`	Exposure in the period. `Ex` can be approximated by the mid-year population aged x to x+n.
`mx`	Death rate in age interval [x, x+n).
`qx`	Probability of dying in age interval [x, x+n).
`lx`	Probability of survival up until age x.
`dx`	Deaths by life-table population in the age interval [x, x+n).

sex	Sex of the population considered here. Default: NULL. This argument affects the first two values in the life table ax column. If sex is specified the values are computed based on the Coale-Demeny method and are slightly different for males than for females. Options: NULL, male, female, total.
lx0	Radix. Default: 100 000.
ax	Numeric scalar. Subject-time alive in age-interval for those who die in the same interval. If NULL this will be estimated. A common assumption is ax = 0.5, i.e. the deaths occur in the middle of the interval. Default: NULL.

Details

The "life table" is also called "mortality table" or "actuarial table". This shows, for each age, what the probability is that a person of that age will die before his or her next birthday, the expectation of life across different age ranges or the survivorship of people from a certain population.

Value

The output is of the "LifeTable" class with the components:

lt	Computed life table;
call	Call in which all of the specified arguments are specified by their full names;
process_date	Time stamp.

Author(s)

Marius D. Pascariu

See Also

LawTable convertFx

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)

LT1
LT5
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 by the function.

# 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)
lt = LifeTable(x, mx = mx, sex = "female")
lt
```

MortalityLaw *Fit Mortality Laws*

Description

Fit parametric mortality models given a set of input data which can be represented by death counts and mid-interval population estimates `(Dx, Ex)` or age-specific death rates `(mx)` or death probabilities `(qx)`. Using the argument `law` one can specify the model to be fitted. So far more than 27 parametric models have been implemented; check the `availableLaws` function to learn about the available options. The models can be fitted under the maximum likelihood methodology or by selecting a loss function to be optimised. See the implemented loss function by running the `availableLF` function.

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

`x`	Vector of ages at the beginning of the age interval.
`Dx`	Object containing death counts. An element of the `Dx` object represents the number of deaths during the year to persons aged x to x+n.
`Ex`	Exposure in the period. `Ex` can be approximated by the mid-year population aged x to x+n.
`mx`	Death rate in age interval [x, x+n).

qx	Probability of dying in age interval [x, x+n).
law	The name of the mortality law/model to be used. e.g. gompertz, makeham, ... To investigate all the possible options, see availableLaws function.
opt.method	How would you like to find the parameters? Specify the function to be optimize. Available options: the Poisson likelihood function poissonL; the Binomial likelihood function -binomialL; and 6 other loss functions. For more details, check the availableLF function.
parS	Starting parameters used in the optimization process (optional).
fit.this.x	Select the ages to be considered in model fitting. By default fit.this.x = x. One may want to exclude from the fitting procedure, say, the advanced ages where the data is sparse.
custom.law	Allows you to fit a model that is not defined in the package. Accepts as input a function.
show	Choose whether to display a progress bar during the fitting process. Logical. Default: FALSE.
...	Arguments to be passed to or from other methods.

Details

Depending on the complexity of the model, one of following optimization strategies is employed:

1. Nelder-Mead method: approximates a local optimum of a problem with n variables when the objective function varies smoothly and is unimodal. For details see optim
2. PORT routines: provides unconstrained optimization and optimization subject to box constraints for complicated functions. For details check nlminb
3. Levenberg-Marquardt algorithm: damped least-squares method. For details check nls.lm

Value

The output is of the "MortalityLaw" class with the components:

input	List with arguments provided in input. Saved for convenience.
info	Brief information about the model.
coefficients	Estimated coefficients.
fitted.values	Fitted values of the selected model.
residuals	Deviance residuals.
goodness.of.fit	List containing goodness of fit measures like AIC, BIC and log-Likelihood.
opt.diagnosis	Resultant optimization object useful for checking the convergence etc.
stats	List containing statistical measures like: parameter correlation, standard errors, degrees of freedom, deviance, gradient matrix, QR decomposition, covariance matrix etc.

Author(s)

Marius D. Pascariu

See Also

availableLaws availableLF LifeTable ReadHMD

Examples

```
# Example 1: --------------------------
# Fit Makeham Model for Year of 1950.

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)


# 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
# hx = 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).
my_gompertz <- 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 = my_gompertz)
summary(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)

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

`MortalityLaws` *MortalityLaws: Parametric Mortality Models, Life Tables and HMD*

Description

Fit the most popular human mortality 'laws', and construct full and abridge 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>. An elegant function for downloading data from Human Mortality Database <https://www.mortality.org> is 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> (0000-0002-2568-6489) [copyright holder]

Other contributors:

- Vladimir Canudas-Romo [contributor]

See Also

Useful links:

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

plot.MortalityLaw *Plot Function for MortalityLaw*

Description

Plot Function for MortalityLaw

Usage

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

Arguments

x	An object of class MortalityLaw
...	Arguments to be passed to methods, such as graphical parameters (see par).

Author(s)

Marius D. Pascariu

See Also

MortalityLaw

Examples

```
# See complete example in MortalityLaw help page
```

predict.MortalityLaw *Predict function for MortalityLaw*

Description

Predict function for MortalityLaw

Usage

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

Arguments

object	An object of class "MortalityLaw"
x	Vector of ages to be considered in prediction
...	Additional arguments affecting the predictions produced.

Author(s)

Marius D. Pascariu

See Also

MortalityLaw

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.
```

ReadHMD — *Download Mortality and Population Data (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 are available:

- `"births"` – birth records;
- `"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","CHE","CZE", "DEUTE","DEUTNP
interval	HMD data format: (age interval x year interval). Interval options: 1x1, 1x5, 1x10, 5x1, 5x5, 5x10.
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.

Value

A ReadHMD object that contains:

input	List with the input values (except the password).
data	Data downloaded from HMD.
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

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)
ls(HMD_Dx)
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)
```

Index