Introduction to `MortalityLaws`

Marius D. Pascariu

August 16, 2018

1 Package structure

MortalityLaws in an **R** package which exploits the available optimization methods to provide tools for fitting and analyzing a wide range of complex mortality models. The package can be used to construct full and abridged life tables given various input indices and to download data from Human Mortality Database as well. The main functions in the package are: `MortalityLaw`, `LifeTable`, `LawTable`, `convertFx` and `ReadHMD`. The package also provides support functions like `availableLaws`, `availableLF` and `availableHMD` that return information about the mortality laws implemented in the package, the loss functions used in optimization processes, and HMD data availability. Generic functions like `predict`, `plot`, `summary`, `fitted`, `residuals` are created for MortalityLaws objects. Small data set for testing purposes `ahmd` is saved in the package.

All functions are documented in the standard way, which means that once you load the package using `library(MortalityLaws)` you can just type, for example, `?MortalityLaw` to see the help file.

2 Downloading HMD Data

Download data form Human Mortality Database (2016) using the `ReadHMD` function:

```r
library(MortalityLaws)
```

```r
# Download HMD data - death counts
HMD_Dx <- ReadHMD(what = "Dx",
                  countries = "SWE",            # HMD country code for Sweden
                  interval  = "1x1",            # specify data format
                  username  = "user@email.com", # here add your HMD username
                  password  = "password",       # here add your HMD password
                  save = FALSE)                 # save data outside R
```

Here we download all the registered death counts `Dx` in Sweden from 1751 until 2014. In the same way one can download the following records: birth records `births`, exposure-to-risk `Ex`, deaths by Lexis triangles `lexis`, population size `population`, death-rates `mx`, life tables for females `LT_f`, life tables for males `LT_m`, life tables both sexes combined `LT_t`, life expectancy at birth `e0`, cohort death-rates `mxc` and cohort exposures `Exc` for over 40 countries and regions in different formats.

3 Model fitting and diagnosis

Once we have data from HMD or other sources we can start analyzing it. For example, let's fit a Heligman-Pollard (1980) model under a Poisson setting which is already implemented as one of the standard models in the package. We have to use the `MortalityLaw` function in this regard.

```
year     <- 1950
ages     <- 0:100
deaths   <- ahmd$Dx[paste(ages), paste(year)]
exposure <- ahmd$Ex[paste(ages), paste(year)]

fit <- MortalityLaw(x   = ages,
                    Dx  = deaths,   # vector with death counts
                    Ex  = exposure, # vector containing exposures
                    law = "HP",
                    opt.method = "LF2")
```

```
# inspect the output object
ls(fit)
## [1] "coefficients"    "deviance"      "df"            "fitted.values"
## [5] "goodness.of.fit" "info"          "input"         "opt.diagnosis"
## [9] "residuals"
```

A summary can be obtained using the `summary` function:

```
summary(fit)
## Heligman-Pollard model: q[x]/p[x] = A^[(x + B)^C] + D exp[-E log(x/F)^2] + G H^x
## Fitted values: mx
##
## Call: MortalityLaw(x = ages, Dx = deaths, Ex = exposure, law = "HP",
##     opt.method = "LF2")
##
## Deviance Residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -0.0312 -0.0002  0.0000  0.0009  0.0003  0.0462
##
## Parameters:
##      A       B       C       D       E      F_       G       H
##  0.0022  0.0146  0.1229  0.0009  2.7565 29.0080  0.0000  1.1141
##
## Residual standard error: 0.07118 on 93 degrees of freedom
```

The standard plot helps us to investigate visually the goodness of fit.

```
plot(fit)
```
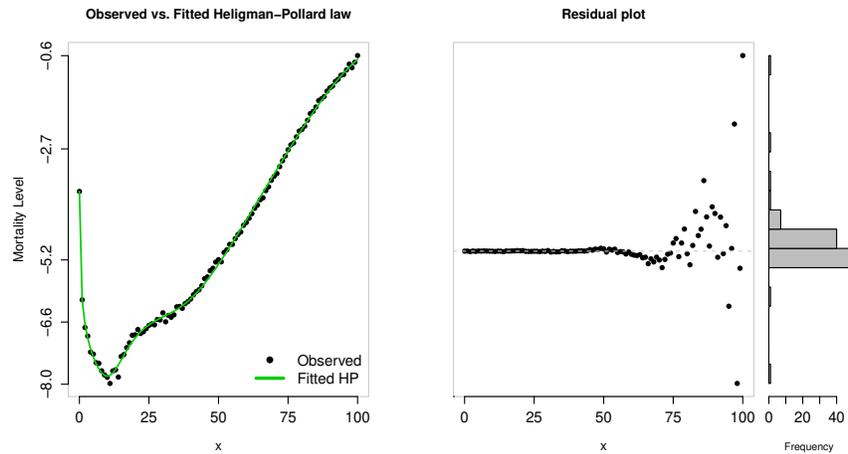
A model can be fitted using a subset of the data only by specifying in `fit.this.x` age range to be covered:

```r
fit.subset <- MortalityLaw(x   = ages,
                           Dx  = deaths,
                           Ex  = exposure,
                           law = "HP",
                           opt.method = "LF2",
                           fit.this.x = 0:65)
plot(fit.subset)
```
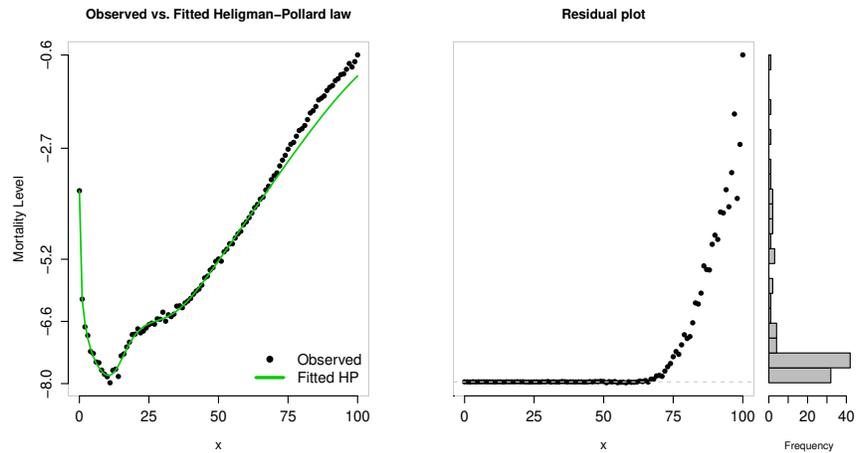

The gray area on the plot showing the fitted value indicates the age range used in fitting the model.

4 Mortality laws in the package

In **R** one can check the availability of the implemented models using `availableLaws`:

```r
availableLaws()
```

See table 1.

5 Loss function in the package

A parametric model is fitted by optimizing a loss function e.g. a likelihood function or a function that minimizes errors. In `MortalityLaws` 8 such functions are implemented and can be used to better capture different portions of a mortality curve. Check `availableLF` for more details.

```
availableLF()
##
## Loss functions available in the package:
##
##  LOSS FUNCTION                             CODE
##  L = -[Dx * log(mu) - mu*Ex]               poissonL
##  L = -[Dx * log(1 - exp(-mu)) - (Ex - Dx)*mu]   binomialL
##  L =  [1 - mu/ov]^2                        LF1
##  L =  log[mu/ov]^2                         LF2
##  L =  [(ov - mu)^2]/ov                     LF3
##  L =  [ov - mu]^2                          LF4
##  L =  [ov - mu] * log[ov/mu]               LF5
##  L =  abs(ov - mu)                         LF6
##
## LEGEND:
## Dx: Death counts
## Ex: Population exposed to risk
## mu: Estimated value
## ov: Observed value
```

6 Custom mortality laws

Now let's fit a mortality law that is not defined in the package, say a reparametrize version of Gompertz in terms of modal age at death (Missov et al. 2015),

$$\mu_x = \beta e^{\beta(x-M)}. \tag{1}$$

We have to define a function containing the desired hazard function and then using the `custom.law` argument it can be used in the `MortalityLaw` function.

```r
# Here we define a function for our new model and provide start parameters
my_gompertz <- function(x, par = c(b = 0.13, M = 45)){
  hx  <- with(as.list(par), b*exp(b*(x - M)) )
```

```r
  # return everything inside this function
  return(as.list(environment()))
}
```

```r
# Select data
year     <- 1950
ages     <- 45:85
deaths   <- ahmd$Dx[paste(ages), paste(year)]
exposure <- ahmd$Ex[paste(ages), paste(year)]
```

```r
# Use 'custom.law' argument to instruct the MortalityLaw function how to behave
my_model <- MortalityLaw(x = ages,
                         Dx = deaths,
                         Ex = exposure,
                         custom.law = my_gompertz)
```

```r
summary(my_model)
## Custom Mortality Law
## Fitted values: mx
##
## Call: MortalityLaw(x = ages, Dx = deaths, Ex = exposure, custom.law = my_gompertz)
##
## Deviance Residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -0.0029 -0.0011  0.0002  0.0009  0.0015  0.0122
##
## Parameters:
##       b       M
##  0.0993 35.8021
##
## Residual standard error: 0.02097 on 39 degrees of freedom
```

```r
plot(my_model)
```

Observed vs. Fitted Custom Mortality law   **Residual plot**
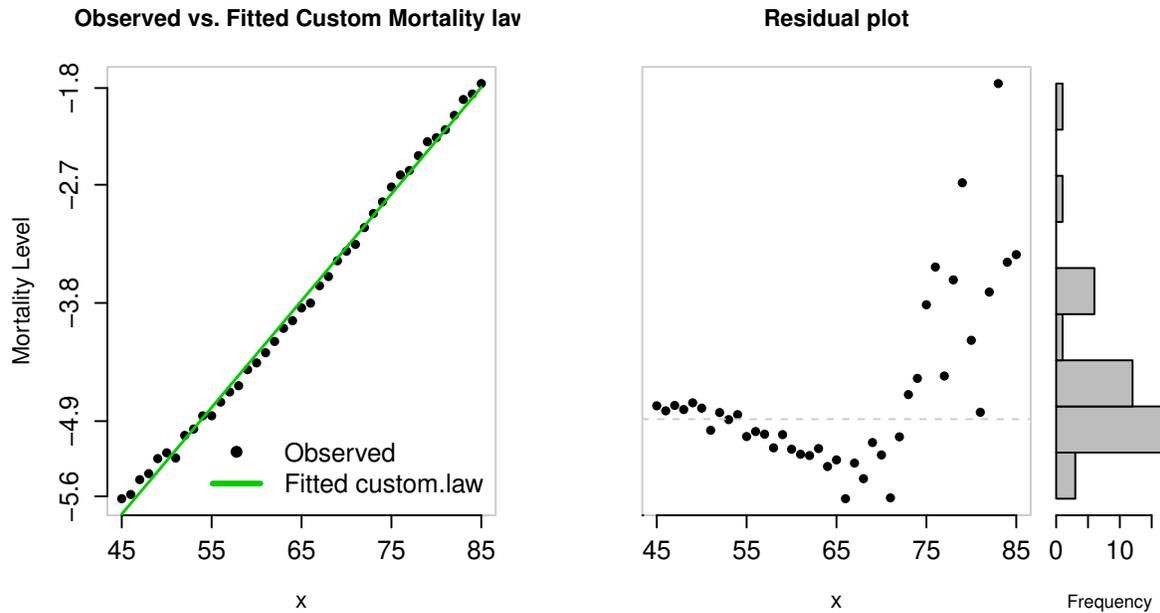
7 Full life tables

Using `LifeTable` function one can build 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 are specified, the other can be ignored.

```r
# Life table for year of 1900
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
##
## Full Life Table
```

```
## 
## Number of life tables: 1
## Dimension: 111 x 10
## Age intervals: [0,1) [1,2) [2,3) ...   ... [108,109) [109,110) [110,+)
## 
##       x.int    x     mx      qx     ax     lx     dx     Lx        Tx      ex
##       [0,1)    0 0.1513 0.1404  0.49  1e+05  14042  92802  4808384  48.08
##       [1,2)    1 0.0479 0.0468   0.5  85958   4023  83931  4715582  54.86
##       [2,3)    2 0.0191 0.0189   0.5  81935   1552  81157  4631651  56.53
##       [3,4)    3  0.013 0.0129   0.5  80383   1041  79861  4550494  56.61
##       [4,5)    4 0.0094 0.0093   0.5  79342    740  78971  4470633  56.35
##       [5,6)    5 0.0067 0.0067   0.5  78602    523  78340  4391661  55.87
##        <NA> ...    ...    ...   ...    ...    ...    ...      ...    ...
##   [108,109) 108   6.75 0.9988  0.15      0      0      0        0   0.15
##   [109,110) 109   6.75 0.9988  0.15      0      0      0        0   0.15
##     [110,+) 110   6.75      1  0.15      0      0      0        0   0.15
```

```
ls(LT1)
## [1] "call"           "lt"              "process_date"
```

8   Abridged life tables

```
# Example
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
## 
## Abridged Life Table
## 
## Number of life tables: 1
## Dimension: 24 x 10
## Age intervals: [0,1) [1,5) [5,10) ...   ... [100,105) [105,110) [110,+)
## 
##       x.int    x      mx     qx    ax     lx    dx     Lx       Tx     ex
##       [0,1)    0   0.053 0.0516   0.2  1e+05  5162   95878 6485467  64.85
##       [1,5)    1   0.005 0.0198  1.44  94838  1878  374547 6389590  67.37
##       [5,10)   5   0.001  0.005   2.5  92960   464  463640 6015042  64.71
##      [10,15)  10  0.0012  0.006   2.5  92496   553  461098 5551402  60.02
##      [15,20)  15  0.0018  0.009   2.5  91943   824  457653 5090304  55.36
##      [20,25)  20   0.002   0.01   2.5  91119   907  453326 4632651  50.84
##        <NA> ...     ...    ...   ...    ...    ...    ...     ...    ...
```

```
## [100,105) 100   0.39 0.8577 1.73   407  349     896    1014  2.49
## [105,110) 105  0.478 0.9084 1.59    58   53     110     119  2.05
##   [110,+) 110  0.551      1 1.59     5    5       8       8  1.59
```

9 Citation

```
citation(package = "MortalityLaws")
## Warning in citation(package = "MortalityLaws"): no date field in
## DESCRIPTION file of package 'MortalityLaws'
## Warning in citation(package = "MortalityLaws"): could not determine year
## for 'MortalityLaws' from package DESCRIPTION file
##
## To cite package 'MortalityLaws' in publications use:
##
##   Marius D. Pascariu (NA). MortalityLaws: Parametric Mortality
##   Models, Life Tables and HMD. R package version 1.8.1.
##   https://github.com/mpascariu/MortalityLaws
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {MortalityLaws: Parametric Mortality Models, Life Tables and HMD},
##     author = {Marius D. Pascariu},
##     note = {R package version 1.8.1},
##     url = {https://github.com/mpascariu/MortalityLaws},
##   }
```

References

Heligman, Larry, and John H Pollard. 1980. "The Age Pattern of Mortality." *Journal of the Institute of Actuaries* 107 (01): 49–80.

Missov, Trifon I, Adam Lenart, Laszlo Nemeth, Vladimir Canudas-Romo, and James W Vaupel. 2015. "The Gompertz Force of Mortality in Terms of the Modal Age at Death." *Demographic Research* 32: 1031–48.

University of California Berkeley, USA, and Max Planck Institute for Demographic Research, Germany. 2016. "Human Mortality Database." http://www.mortality.org/.

Table 1: Parametric functions build in the MortalityLaws package

Mortality Laws	Predictor	Code
Gompertz	$\mu(x) = Ae^{Bx}$	gompertz
Gompertz	$\mu(x) = \frac{1}{\sigma}exp\left\{\frac{x-M}{\sigma}\right\}$	gompertz0
Inverse-Gompertz	$\mu(x) = \frac{1}{\sigma}exp\left\{\frac{x-M}{\sigma}\right\}/\left(exp\left\{e^{\frac{-(x-M)}{\sigma}}\right\} - 1\right)$	invgompertz
Makeham	$\mu(x) = Ae^{Bx} + C$	`makeham`
Makeham	$\mu(x) = \frac{1}{\sigma}exp\left\{\frac{x-M}{\sigma}\right\} + C$	makeham0
Opperman	$\mu(x) = \frac{A}{\sqrt{x}} + B + C\sqrt[3]{x}$	opperman
Thiele	$\mu(x) = A_1 e^{-B_1 x} + A_2 e^{-\frac{1}{2}B_2(x-C)^2} + A_3 e^{B_3 x}$	thiele
Wittstein & Bumstead	$q(x) = \frac{1}{B}A^{-(Bx)^N} + A^{-(M-x)^N}$	wittstein
Perks	$\mu(x) = (A + BC^x)/(BC^{-x} + 1 + DC^x)$	perks
Weibull	$\mu(x) = \frac{1}{\sigma}\left(\frac{x}{M}\right)^{\frac{M}{\sigma}-1}$	weibull
Inverse-Weibull	$\mu(x) = \frac{1}{\sigma}\left(\frac{x}{M}\right)^{-\frac{M}{\sigma}-1}/\left(exp\left\{\left(\frac{x}{M}\right)^{-\frac{M}{\sigma}}\right\} - 1\right)$	invweibull
Van der Maen	$\mu(x) = A + Bx + Cx^2 + I/(N-x)$	vandermaen
Van der Maen	$\mu(x) = A + Bx + I/(N-x)$	vandermaen2
Quadratic	$\mu(x) = A + Bx + Cx^2$	quadratic
Beard	$\mu(x) = Ae^{Bx}/(1 + KAe^{Bx})$	beard
Makeham-Beard	$\mu(x) = Ae^{Bx}/(1 + KAe^{Bx}) + C$	makehambeard
Gamma-Gompertz	$\mu(x) = Ae^{Bx}/[1 + \frac{AG}{B}(e^{Bx} - 1)]$	ggompertz
Siler	$\mu(x) = A_1 e^{-B_1 x} + A_2 + A_3 e^{B_3 x}$	siler
Heligman - Pollard	$q(x)/p(x) = A^{(x+B)^C} + De^{-E(\ln x - \ln F)^2} + GH^x$	HP
Heligman - Pollard	$q(x) = A^{(x+B)^C} + De^{-E(\ln x - \ln F)^2} + \frac{GH^x}{1+GH^x}$	HP2
Heligman - Pollard	$q(x) = A^{(x+B)^C} + De^{-E(\ln x - \ln F)^2} + \frac{GH^x}{1+KGH^x}$	HP3
Heligman - Pollard	$q(x) = A^{(x+B)^C} + De^{-E(\ln x - \ln F)^2} + \frac{GH^{x^K}}{1+GH^{x^K}}$	HP3
Rogers-Planck	$q(x) = A_0 + A_1 e^{-Ax} + A_2 e^{\left\{B(x-U)-e^{-C(x-U)}\right\}} + A_3 e^{Dx}$	rogersplanck
Martinelle	$\mu(x) = (Ae^{Bx} + C)/(1 + De^{Bx}) + Ke^{Bx}$	martinelle
Carriere	$S(x) = \psi_1 S_1(x) + \psi_2 S_2(x) + \psi_3 S_3(x)$	carriere1
Carriere	$S(x) = \psi_1 S_1(x) + \psi_4 S_4(x) + \psi_3 S_3(x)$	carriere2
Kostaki	$q(x)/p(x) = A^{(x+B)^C} + De^{-E_i(\ln x - \ln F)^2} + GH^x$	kostaki
Kannisto	$\mu(x) = Ae^{Bx}/(1 + Ae^{Bx}) + C$	kannisto