Exercises in Pension Systems

University of Copenhagen, Autumn 2021

6 September – 12 September (Week 1)

Purpose

The purpose of this week's exercises is to get you acquainted with the data structure and the R6-classes we will be using throughout the course. You will need to understand how to load and extract mortality data with the `HMDdatClass` to support different kinds of calculations. Moreover, we will develop an understanding of the historical evolution of mortality, how to simulate from mortality models, and how to calculate life expectancies. Finally, there is an exercise on recreating the death counts available in HMD from raw data.

Remark 1: It is not required that you use ggplot for visualization.
Remark 2: To improve your coding skills and sharpen your understanding, have a look at the bonus questions.

Exercise 1: R warm-ups

In this exercise, we will make sure that our code setup works on your computer, and familiarize ourselves with basic data structures and plotting.

Question 1.1: Add two new folders to the course folder on your computer, one titled 'Code' and the other titled 'Data'. Download the files from the data folder on Absalon and place them in your 'Data' folder. Download the files from the R-code folder on Absalon and place them in your 'Code'-folder. Open structure.r, set the working directory at the top of the file, and execute the program line by line. Make sure there are no errors.

The R6-class `HMDdatClass` will be our workhorse for getting HMD-data into R and extracting the mortality data we need.

Question 1.2: Inspect `HMDobj` by calling `str(HMDobj)`. Inspect the loaded the data with `str(HMDobj$data)`. Try to overwrite `HMDobj$data` or anything within it. Hopefully, you won't have any luck.

Instead of working with the data in `HMDobj` directly, we will extract the parts we need using the `$getOEdata` and the `$getSmoothMu` methods. The first method extracts the empiric occurrence and exposure counts, while the second method returns a

smoothed empiric mortality surface. Both methods take three inputs: `groups` (character vector of group names), an `agelim` (numeric vector of length 1 or 2 specifying the age-span), and a `timelim` (numeric vector of length 1 or 2 specifying the time-span). The methods contain a series of argument checks ensuring internal consistency, i.e. you can only extract data that is actually in the object and it will warn you if you try to do something unreasonable. You can write `print(HMDobj)` to make the object tell you what it contains.

Question 1.3: Extract some data using `$getOEdata` and `$getSmoothMu`. If you do not specify any inputs, the method returns all available data. Inspect the outputs using the `str`-command. You should notice that both outputs follow a similar structure, preserving the inputs you specified.

The outputs from Question 1.3 are S3-classes. For our needs, you can think of this simply as a list with a class name, stored as an attribute, attached to it. You can ask for the class name using the `class`-command. The class name is useful as it can inform a function about the structure of the input data. You can construct your own S3-class using the command `structure(someList, class = "someClassName")`.

Question 1.4: Write a function that takes an `OEdata`-class as input and returns an object of class `surf` containing the group names, an agelim, a timelim, and the empiric mortality surface (O/E-rates). That is, the output of your function should be of the same form as the output of `HMDobj$getSmoothMu`. You can use the following skeleton

```
OEdata <- HMDobj$getOEdata()
calc.OErate <- function(OEdata) {
    # Verify that the input is of class OEdata. Stop if not.

    # Calculate OE-rates
    # for all groups, ages, and years within input object

    # Output an object of class 'surf'
}
OEsurf <- calc.OErate(OEdata)
str(OEsurf) # Compare to str(HMDobj$getSmoothMu())
```

Careful: What should your function do if there are no exposures in a given age-period cell?

We can exploit these standardized data structures to ease computations, for instance to make plotting more efficient.

`ggplot2` is a plotting package that makes it simple to create complex plots from data in a data frame. `ggplot2` plots and commands work best with data in the 'long' format, i.e., a column for every dimension, and a row for every observation. You will notice that the mortality surface in `surf`-objects are in a 'wide' format.

Question 1.5: Write a function `surf2plotdat` that takes a `surf`-object as input and returns a data frame based on `surf$mu` but transformed from wide to long format

with columns 'Group', 'Age', 'Year', and 'Mu'. (*Hint:* You may find the commands `lim2length` and `lim2vec` useful; type the commands in your console to see what they do. Alternatively, have a look at the `melt`-function from the `reshape2`-package.)

Question 1.6: Plot the O/E-rates for Danish females and males, ages 0–110 in the year 2020 on a suitable scale. Superimpose the smoothed rates. Describe (in words) the general pattern of mortality over the age-span and between genders.

Question 1.7: Plot the smoothed O/E-rates for Danish females and males, ages $0, 10, \ldots, 100$ over the years 1950–2020 on a suitable scale. Describe (in words) the general pattern of mortality over the time-span.

Exercise 2: Simulating mortality data

Let T be strictly positive stochastic variable, representing the life time of an individual, with density function f, distribution function F, and survival function $S = 1 - F$. Define the hazard function $\mu(t) := f(t)/S(t)$ for $t \in [0, \infty)$, and denote by $I(t) := \int_0^t \mu(s)\,\mathrm{d}s$ the cumulated hazard and assume that its inverse $I^{-1}(t)$ exists.

Question 2.1: Show that if E is exponential with rate 1, i.e. $E \sim \mathrm{Exp}(1)$, then the random variable $I^{-1}(E)$ has the same survival function as T.

Question 2.2: Suppose that death occurs according to Gompertz' law of mortality, that is

$$\mu(t) = \alpha \exp(\beta t) \tag{1}$$

for $t \in \mathbb{R}_+$ and parameters $\alpha, \beta \in \mathbb{R}_+$. What is the interpretation of the parameters α and β? Using the result from Question 2.1, simulate $100,000$ life times with $\alpha = \exp(-11)$ and $\beta = 0.1$. Make a plot of the resulting distribution, e.g. a density plot. Do you think this distribution could describe human mortality? Explain your reasoning.

Question 2.3: Sometimes, it is convenient to parametrize (1) in terms of the modal age at death M, i.e. the age at which most adult deaths occur. Find M. Insert the parameter values from Question 2.2 into your expression for M and verify that it corresponds to the mode of the distribution you simulated in the previous question. Give an alternative parametrization of (1) in terms of β and M.

Question 2.4: Find the survival function of $I^{-1}(I(x) + E)$ for $x \in (0, \infty)$ where $E \sim \mathrm{Exp}(1)$. How does this result relate to the result from Question 2.1?

Question 2.5: Consider two life time distributions T_1 and T_2 satisfying $T_1 \perp\!\!\!\perp T_2$. Show that if T_1 has death rate $\mu_1(t)$ and T_2 has death rate $\mu_2(t)$ then $T := \min(T_1, T_2)$ has death rate $\mu(t) = \mu_1(t) + \mu_2(t)$.

Question 2.6: Suppose that we are given covariates $Z \in \mathbb{R}^p$ and corresponding coefficients $b \in \mathbb{R}^p$. How can we interpret the model

$$\mu(t) = \mu_0(t) \exp(b^\top Z) \tag{2}$$

where μ_0 is some hazard function. Write down a way to simulate from (2).

Question 2.7: Suppose that mortality is piecewise constant so that $\mu(t) = \mu(\lfloor t \rfloor)$. Describe how we could simulate from this model using a Bernoulli distribution. (Here, $\lfloor t \rfloor$ denotes the integer part of t).

Exercise 3: Life expectancy

In this exercise, we will prove some useful formulas and expressions for (remaining) life expectancy. Assume that we have n iid observations on a continuous life time T with hazard $\mu(t)$, density $f(t)$, and survival function $S(t)$. Assume further that $\mathbb{E}[T] < \infty$.

Question 3.1: Show that the expected life time is $\mathbb{E}[T] = \int_0^\infty S(t)\,\mathrm{d}t$.

Let $\omega \in (0, \infty)$ be given. The number ω is frequently referred to as the *maximum attainable age*.

Question 3.2: Find a similar expression for $\mathbb{E}[T \wedge \omega]$[1]. Why might this quantity be more useful in practice compared to that of Question 3.1?

Question 3.3: Find a similar expression for $\mathbb{E}[(T \wedge \omega) - x \mid T > x]$, $x > 0$.

Question 3.4: Find $f(t)$, $S(t)$, $\mu(t)$ and compute $\mathbb{E}[T]$ when

- **Q.3.4.1:** $F(t) = \frac{t \wedge \omega}{\omega}$,
- **Q.3.4.2:** $F(t) = 1 - \exp(-\alpha t)$, $\alpha > 0$.

Suppose that we partition the compact interval $[0, \omega]$ into subintervals such that $0 = t_0 < t_1 < \cdots < t_J = \omega$. In the following, we assume that μ is piecewise constant on this grid, that is

$$\mu(t) = \mu(t_j), \quad t \in [t_j, t_{j+1}), \quad j = 0, \ldots, J-1.$$

Question 3.5: Let $x \in \{0, \ldots, J\}$. Show that

$$\mathbb{E}[(T \wedge \omega) - t_x \mid T > t_x] = \sum_{j=x}^{J-1} \frac{1 - e^{-\mu(t_j)(t_{j+1} - t_j)}}{\mu(t_j)} e^{-\sum_{i=x}^{j-1} \mu(t_i)(t_{i+1} - t_i)}. \tag{3}$$

The expressions for life expectancy in the above all follow a single cohort through time (so age and time advances synchronously). If we consider multiple cohorts, we add an additional index so that $\mu(x, t)$ describes the hazard for age x at time t. Let $\mu(x, t)$ be constant over the square $[x, x+1) \times [t, t+1)$ in the following.

[1] $x \wedge y := \min(x, y)$

Question 3.6: Suppose we want to find the remaining period life expectancy for a person born on January 1st. In that case we can consider integer ages x and we fix the period t to be a single calendar year. Equation (3) then reads (why?)

$$e_p(x,t) = \sum_{j=x}^{J-1} \frac{1-e^{-\mu(j,t)}}{\mu(j,t)} \exp\left(-\sum_{i=x}^{j-1} \mu(i,t)\right). \tag{4}$$

Write a function in R that computes (4). You can use the following skeleton

```
calc.period.life <- function(surf, group, age, year) {
    # Your implementation
}
```

taking a `surf`-object along with the `group`, `year` and `age` for which life expectancy should be computed as input. Extract the historic (smoothed) mortality surface using the command `HMDobj$getSmoothMu()` and verify that $e_p(60, 2015) = 24.84$ for a Danish female. What would you have to change to accommodate an individual born on another date than January 1st? (*Hint:* Before you code, make a sketch of the mortality pieces you need using a Lexis-diagram. How do they enter each of the two sums?)

(BONUS CODING 1) *Cohort life expectancy*: As in Question 3.6, implement a function computing the cohort life expectancy. Be careful: What should your function do, when the cohort has not yet reached age ω but you have run out of years for which you have data? We could choose to let (5) be undefined for any cohort that has not reached age ω before the last year of data (if so, implement a stopping criterion breaking the computation), but could we do something else?

- *Easy:* Suppose we want to compute the cohort life expectancy of a person born January 1st (so $x \in \mathbb{N}$) calculated in some year on January 1st (so $t \in \mathbb{Z}$). We can use the formula

$$e_c(x,t) = \sum_{j=x}^{J-1} \frac{1-e^{-\mu(j,t+j-x)}}{\mu(j,t+j-x)} \exp\left(-\sum_{i=x}^{j-1} \mu(i,t+i-x)\right). \tag{5}$$

  Verify that $e_c(20, 1920) = 54.925$ for a Danish female.

- *Intermediate:* Usually, when reporting life expectancies, one considers a person born July 1st ($x + 0.5$) calculated at July 1st ($t + 0.5$). With an outset in (3), implement a function that calculates the cohort life expectancy in this case. Verify that $e_c(20.5, 1925.5) = 56.164$ for a Danish female.

- *Hard:* Suppose we want to compute the cohort life expectancy of a person born on any date ($x \in [0, \omega]$) at any point in time ($t \in \mathbb{R}$). With an outset in (3), implement a function that calculates the cohort life expectancy in this case. Verify that $e_c(20, 1920) = 54.925$, $e_c(20.5, 1925.5) = 56.164$, and $e_c(20.75, 1930.2) = 56.345$ for a Danish female.

No matter the difficulty you choose, it is a good idea to visualize the calculation using a Lexis-diagram.

(BONUS CODING 2) *Implementing new R6-methods*: Add your implementation of `calc.period.life` as a public method to the `HMDdatClass`. The new method should take a group, an age and a year as input and return the period life expectancy (utilizing the historic data contained within the object). You may also expand on the method so that the input is vector of groups, an `agelim` and a `timelim` rather than single numbers, making the output an array of life expectancies. You may find it useful to look at `https://adv-r.hadley.nz/r6.html`.

Exercise 4: Recreating HMD death counts

The Human Mortality Database (HMD) is a collaborative project sponsored by the University of California at Berkeley (United States) and the Max Planck Institute for Demographic Research (Rostock, Germany). The purpose of the database is to provide researchers around the world with easy access to detailed and comparable national mortality data.

Most raw data requires various adjustments before being used. In this exercise, we will look at one of the most common problems where death counts are not available in the desired 1×1 format. HMD splits death counts in an $n \times 1$ configuration into 1×1 data using cubic splines fitted to the cumulative distribution of deaths within each calendar year. In the following we will recreate the 1×1 death counts for Danish females in 1835.

Question 4.1: The dataset *DNK_Deaths_raw.txt* contains the raw input death data for Denmark. Load the data using your favourite csv reader and create a subset of the data containing only data for females in 1835.

We will need three variables from the dataset: `Age`, `AgeInterval` and `Deaths`. The `AgeInterval` variable describes the length of the age interval. For example, the 3.315 deaths for age zero is the total number of deaths between age 0 and age 1 (ageInterval = 1), while the 1.448 deaths at age 1 is the total number of deaths between age 1 and age 3 (ageInterval = 2).

Let $Y(x)$ be the cumulative number of deaths up to age x. We assume that $Y(x)$ is known for limited age ranges, including always $x = 1, x = 5$, and $x = 105$. Our plan is to fit Y with cubic splines following the equation

$$Y(x) = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \alpha_3 x^3 + \beta_1 (x - k_1)^3 \mathsf{I}(x > k_1) + \cdots + \beta_n (x - k_n)^3 \mathsf{I}(x > k_n), \quad (6)$$

where I denotes the indicator function and $\gamma = (\alpha_0, \ldots, \alpha_3, \beta_1, \ldots, \beta_n)^\top$ are parameters to be estimated. As a matter of convention, let $k_1 = 1$ and k_n be the last age with data before $\omega = 105$. Thus, we know $Y(x)$ for $n + 2$ ages, namely $\{0, k_1, \ldots, k_n, \omega\}$, but the regression (6) has $n + 4$ parameters. Therefore, we propose the following two additional constraints

1. $Y'(\omega) = 0$,
2. $Y'(1) = \frac{Y(5)-Y(1)}{2}$.

Question 4.2: Find $Y'(x)$ and motivate the two constraints above.

To estimate the parameters of (6), we can write the system on matrix form $A\gamma = b$, where $A \in \mathbb{N}_0^{(n+4)\times(n+4)}$ is the design matrix and

$$b = (0, Y(k_1), \ldots, Y(k_n), Y(\omega), (Y(5) - Y(1))/2, 0)^\top \tag{7}$$

Question 4.3: Specify the design matrix A.

Question 4.4: Estimate the parameters by computing $\gamma = A^{-1}b$. Use the estimated coefficients to find fitted values $\hat{Y}(x)$ for $x = 0, 1, 2, \ldots, k_n$. Estimate the 1×1 death counts by differencing

$$\hat{O}_x = \hat{Y}(x+1) - \hat{Y}(x),$$

and verify that $\hat{O}_x$ corresponds to the HMD-data in *DNK_Deaths.txt* (that you load with the `HMDdatClass`).

Question 4.5: The spline method described above does not guarantee that $\hat{Y}$ is monotonically increasing over all ages. Why is this problematic? When would this problem typically occur in practice?