Journal of Statistical Software

December 2016, Volume 75, Issue 5. *doi: 10.18637/jss.v075.i05*

bayesPop: Probabilistic Population Projections

Hana Ševčíková
University of Washington

Adrian E. Raftery
University of Washington

Abstract

We describe **bayesPop**, an R package for producing probabilistic population projections for all countries. This uses probabilistic projections of total fertility and life expectancy generated by Bayesian hierarchical models. It produces a sample from the joint posterior predictive distribution of future age- and sex-specific population counts, fertility rates and mortality rates, as well as future numbers of births and deaths. It provides graphical ways of summarizing this information, including trajectory plots and various kinds of probabilistic population pyramids. An expression language is introduced which allows the user to produce the predictive distribution of a wide variety of derived population quantities, such as the median age or the old age dependency ratio. The package produces aggregated projections for sets of countries, such as UN regions or trading blocs. The methodology has been used by the United Nations to produce their most recent official population projections for all countries, published in the *World Population Prospects*.

Keywords: Bayesian hierarchical model, population projections, expression language, population pyramid, United Nations, World Population Prospects.

1. Introduction

Projections of countries' future populations, broken down by age and sex, are widely used by governments at all levels for planning purposes, by international organizations for monitoring development and other goals, such as the Millennium Development Goals, by social and health researchers, and the private sector for strategic and marketing decisions.

Most population projections are currently done deterministically using the cohort component method. This is an age- and sex-structured version of the basic demographic identity that the population of a country at the next time point is equal to the current population, plus the number of births, minus the deaths, plus the immigrants, minus the emigrants (Leslie 1945; Preston, Heuveline, and Guillot 2001).

Standard projections are deterministic, meaning that they yield a single value for each pro-

jected future population quantity of interest. However, probabilistic projections are widely desired because they are useful for decision-making when one wants to be reasonably sure of not under- or overpredicting a number, for assessing changes and deviations of population quantities from expectations, and for providing a general assessment of uncertainty.

A systematic framework for producing probabilistic population projections for all countries, both developed and developing, has recently been proposed by Raftery, Li, Ševčíková, Gerland, and Heilig (2012). It consists of probabilistically projecting total fertility rate and life expectancy using Bayesian hierarchical models (Alkema *et al.* 2011; Raftery, Chunn, Gerland, and Ševčíková 2013), converting the results to age-specific rates, and projecting the population forward using the cohort component method applied to each trajectory simulated from their predictive distributions (Ševčíková, Li, Kantorová, Gerland, and Raftery 2016a). The median projection from the method has been used as the official medium projection of the United Nations (UN) since the 2012 revision of the *World Population Prospects* (United Nations 2013). In July 2014 for the first time, the United Nations released official probabilistic population projections for all countries (Gerland *et al.* 2014; United Nations 2015).

Here we describe a package called **bayesPop** (Ševčíková, Raftery, and Buettner 2016b) that produces probabilistic population projections using this method. It is implemented in R (Ihaka and Gentleman 1996; R Core Team 2016), and it was developed to allow users beyond the UN to implement the methodology. The package allows an analyst to reproduce the UN projections, to generate variations on them corresponding to different inputs or modeling assumptions, or to use their own data. We also introduce a flexible expression language which allows probabilistic results to be summarized and visualized in graphs, maps or population pyramids. The software can be conveniently controlled from a graphical user interface.

The paper is organized as follows. In Section 2 we review the basic probabilistic population projection methodology underlying the package. In Section 3 we describe the **bayesPop** package and how to generate and view the probabilistic population projections using it. In Section 4, we show how to display probabilistic population pyramids for visualizing the age-specific results. In Section 5 we describe our expression language for generating probabilistic projections of user-defined derived population quantities. Examples include the median age of the population or the ratio of the population of one country to that of another. In Section 6 we indicate how to produce probabilistic projections of aggregated population quantities for groups of countries, such as UN regions or trading blocs. We conclude in Section 7 with a brief discussion of some related R packages.

2. Methodology

Most methods for predicting population P in country c at time period t are based on the *demographic balancing equation*, namely

$$P_{c,t} = P_{c,t-1} + B_{c,t} - D_{c,t} + M_{c,t}, \tag{1}$$

where B denotes the number of births, D denotes the number of deaths and M denotes net international migration. In most applications this equation is solved deterministically using the cohort component method (Whelpton 1928, 1936), which decomposes it into age- and sex-specific components.

As it has traditionally been implemented by the United Nations Population Division (United

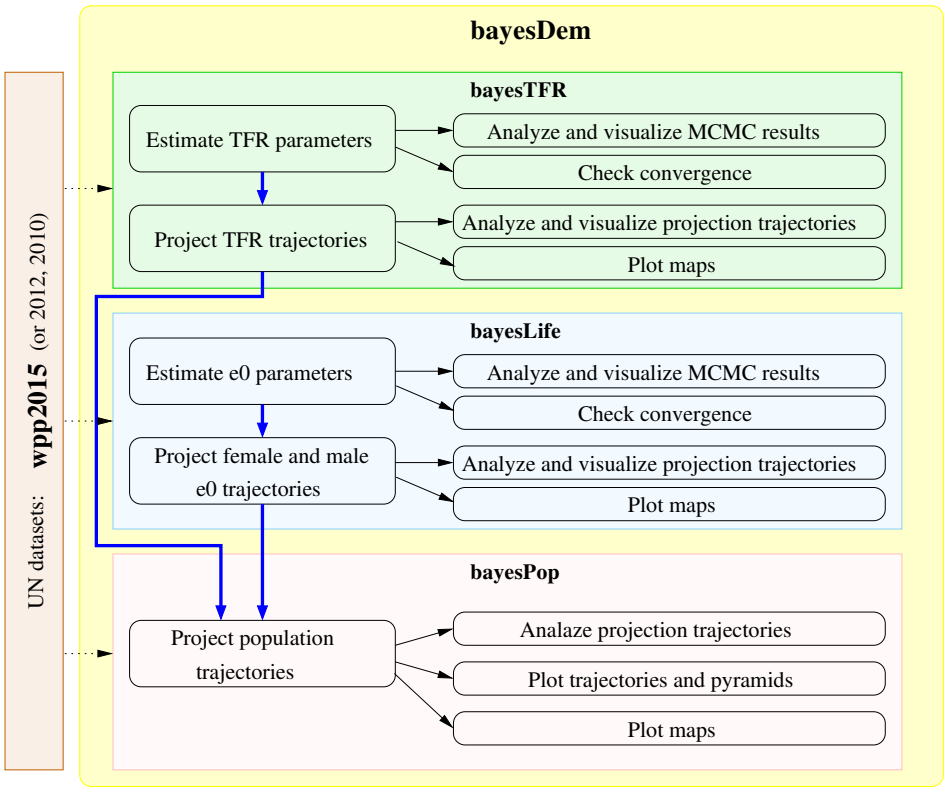

Figure 1: Structure of packages supported by **bayesDem**. Boxes shown on the left-hand side (connected by blue arrows) depict the main steps needed for generating probabilistic population projections. Boxes on the right-hand side show supporting functionality of the packages. The packages operate on UN datasets included in one of the **wpp** packages which can be explored and visualized using the **wppExplorer** package.

Nations 1956, 1989), the cohort component method for projecting a country's population by age and sex in future time periods $t > 0$ is deterministic, and requires the following inputs:

- sex- and age-specific population estimates at the initial time $t = 0$,
- projections of future total fertility rates (TFR),
- projections of sex ratio at birth,
- projections of female and male life expectancies (e_0),
- historical data on sex- and age-specific death rates (for $t \leq 0$),
- historical data on fertility distribution by age (for $t \leq 0$), and
- projections of future sex- and age-specific net international migration.

In each time period t, a projection of the fertility distribution by age is obtained using historical data (Ševčíková *et al.* 2016a). Then this distribution is used to convert the TFR to age-specific fertility rates in time period t. Using the historical data on death rates,

life expectancy is converted to age-specific mortality rates using a variant of the Lee-Carter method (Lee and Carter 1992). See Ševčíková *et al.* (2016a) for more detail on these conversion steps. Finally the cohort component method is applied.

To communicate uncertainty in the context of this deterministic approach, until recently the UN used three scenarios, the high, medium and low variants. The medium projection is the main one. The high and low variants are generated by adding half a child to or subtracting half a child from the TFR, respectively, and then applying the method above. Such an approach suffers from not having a probabilistic basis and can lead to inconsistencies (Lee and Tuljapurkar 1994; National Research Council 2000).

Methods for probabilistic projection of the two most important inputs have recently been developed, namely TFR (Alkema *et al.* 2011) and life expectancy (Raftery *et al.* 2013). Raftery *et al.* (2012) describes a way to combine these components into overall probabilistic population projections. The idea is to simulate a large set of trajectories of future values of TFR (as implemented in **bayesTFR**, see Ševčíková, Alkema, and Raftery 2011), then to simulate an equal number of trajectories of life expectancy (as implemented in **bayesLife**, see Ševčíková, Raftery, and Chunn 2016c), and finally to convert each of the trajectories into a future trajectory of all sex- and age-specific populations, using the current UN methodology as described above. The resulting set of values is viewed as a sample from the predictive distribution of population numbers. This approach is implemented in the R package **bayesPop** (Ševčíková *et al.* 2016b). A graphical user interface (GUI) for the three packages, **bayesTFR**, **bayesLife** and **bayesPop**, is provided by the R package **bayesDem** (Ševčíková 2016a). Together, these packages allow one to generate probabilistic projections of TFR and life expectancy, and combine those results into probabilistic population projections from a single interface; see Figure 1. In addition to the Comprehensive R Archive Network (`https://CRAN.R-project.org/`), all of these packages are hosted on `https://github.com/PPgp`. Furthermore, a mailing list accessible from `http://bayespop.csss.washington.edu/` is available to interested users.

3. Using bayesPop

3.1. Generating population projections

The main function for generating probabilistic population projection is called `pop.predict`. It can be run for a single country or for a given set of countries. By default, projections are generated for all countries for which inputs are available. The data packages **wpp2010** (Ševčíková and Gerland 2013), **wpp2012** (Ševčíková, Gerland, Andreev, Li, Gu, and Spoorenberg 2014), and **wpp2015** (United Nations 2016) provide such data for most countries. An argument `wpp.year` (which can be one of 2010, 2012, or 2015) determines the default dataset used in the projections. We will refer to the corresponding data package simply as a **wpp** package.

Inputs

The projection inputs listed in Section 2 are given in the argument `inputs` which is a list containing the various input components. The *deterministic* components include:

`popM`, `popF`: Initial male, female age-specific population counts.

`mxM`, `mxF`: Estimates of historical male and female age-specific death rates.

pasfr: Estimates of historical age-specific fertility rates as percentages of TFR.

srb: Projection of future sex ratio at birth.

mig.type: Migration base year, and an argument determining whether migration is assumed to occur at the end of the five-year interval or to be evenly distributed over the interval.

migM, migF: Projection of future male and female age-specific net international migration.

If any of these inputs is not specified, the default dataset from the given **wpp** package is used. If the user wishes to overwrite a default dataset, the corresponding component can be given as a tab-delimited text file (the manual for `pop.predict` describes the structure of these files). Note that the *pop* and *mig* components must be on the same scale which becomes the final scale of the projections. The **wpp** packages maintain their datasets on the scale of thousands.

Each of the *probabilistic* components of the `inputs` argument, namely TFR and e_0, can be specified in several ways, either as a directory, or as a file. In both cases, a set of trajectories is passed to the prediction function. The `pop.predict` function is designed to operate on the results of `tfr.predict` and `e0.predict` from **bayesTFR** and **bayesLife**, respectively. Thus one would specify here the directories in which the resulting TFR and e_0 trajectories are stored:

tfr.sim.dir: Simulation directory used to store results of `tfr.predict`.

e0F.sim.dir: Simulation directory used to store results of `e0.predict` for projections of female life expectancy.

e0M.sim.dir: This can be a directory with a simulation of male life expectancy that is independent of the simulation in `e0F.sim.dir`. Preferably, however, it can be the keyword `"joint_"` in which case it is assumed that male and female e_0 were generated jointly using the gap model (Raftery, Lalic, and Gerland 2014b), and thus, they are both extracted from `e0F.sim.dir`.

For convenience, these probabilistic inputs can also be specified as text files, which can be useful, for example, if generated with software other than **bayesTFR** and **bayesLife**.

If neither of the probabilistic components of the `inputs` argument is given, the function creates three trajectories which are extracted from the **wpp** package, namely the projection median, and the low and high variants.

For experimental purposes, the function allows the user to enter multiple trajectories of net migration which can be used for example to test different migration scenarios. These components are called `migMtraj` and `migFtraj` and if used they replace the deterministic components `migM` and `migF`.

Other arguments

Besides specifying countries (argument `countries`), one can define how many trajectories to generate (`nr.traj`), the end year of the projection (`end.year`), the initial year (`start.year`), or the present year (`present.year`). The initial year defines the first year of the historical death rates to be used to estimate the parameters of the Lee-Carter method. The present

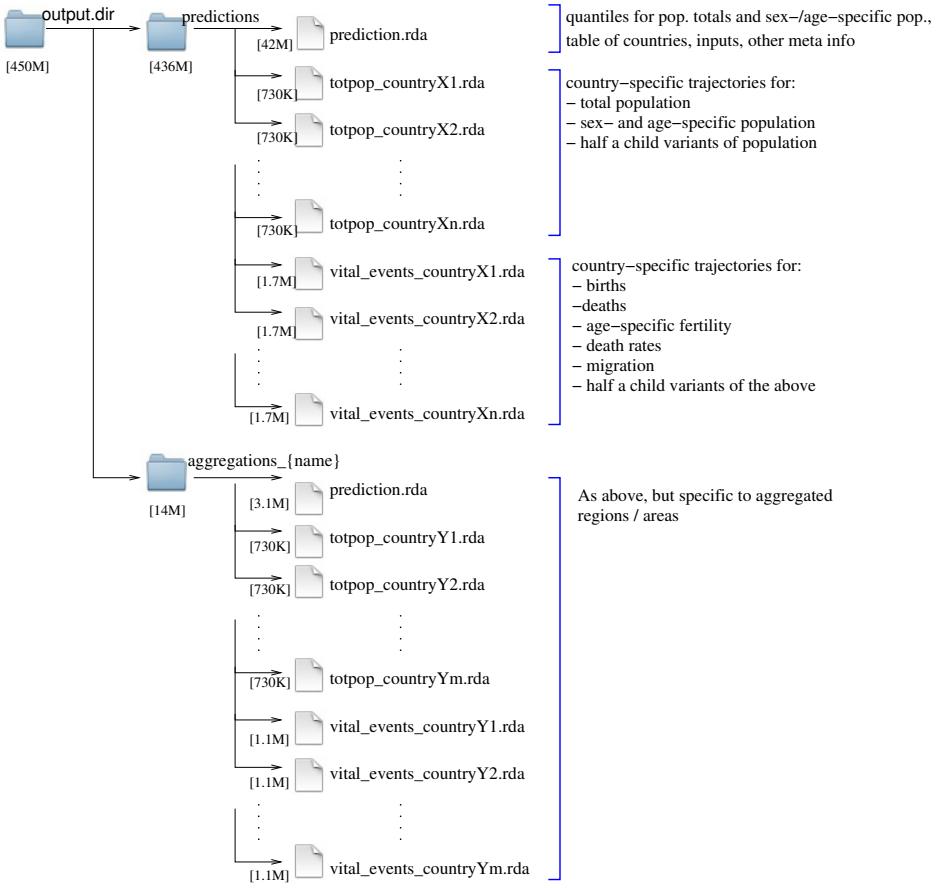

Figure 2: Structure of folders and files created when generating projections with **bayesPop**. The approximate sizes correspond to a simulation with 100 trajectories, 162 countries and 6 aggregated areas.

year defines the last observed time period, i.e., the initial time $t = 0$ from which projections start.

By default, vital events, such as births and deaths, used during the computation in the cohort component method are discarded. However, if the argument `keep.vital.events` is set to `TRUE`, they are stored together with the projection trajectories and can be used later for an analysis. It should be noted that storing vital events more than doubles the amount of data stored per country.

The argument `output.dir` specifies the location on disk where the results are to be stored.

Outputs

The `pop.predict` function applies the cohort component method to each set of trajectories (a set meaning one trajectory for each of TFR, female e_0 and male e_0), using the deterministic components for the remaining input data. As a result, we have a set of sex- and age-specific population trajectories which can be used to construct posterior distributions of various population quantities of interest.

The trajectories are stored in the `output.dir/predictions` directory, one file per country,

called `totpop_country`x`.rda`, where x is the numerical country code. If storing vital events is requested, trajectories for number of births by age of mother and sex of child, trajectories for sex- and age-specific numbers of deaths, net migration, and trajectories for sex- and age-specific fertility and mortality rates are stored in files called `vital_events_country`x`.rda`. Thus in such a case, there are two files per country. The resulting file structure is depicted in Figure 2. The approximate file sizes in this figure correspond to a simulation with 100 trajectories, such as in the example below. The optional aggregations sub-directory will be described in Section 6.

An object returned by the `pop.predict` function is of class `bayesPop.prediction`. It contains pre-computed quantiles for the main population quantities, including total population, total population by sex and total population by sex and age. It also contains a pointer to the disk location where the country-specific trajectories are stored. The package contains various functions that help the user to view and analyze those results; these will be described in the following sections.

3.2. Example

We first show an example of how to generate probabilistic population projection from scratch, including generating all the probabilistic components. The blue arrows in Figure 1 show the workflow in this process. Note that estimating and projecting TFR and e_0 (step 1 and 2 below) use packages **bayesTFR** and **bayesLife**, respectively, and can be therefore considered as prerequisites for generating population projection with **bayesPop**. The data used in this example are taken from the **wpp2015** package. At the time of writing, 2015 is the default value for the `wpp.year` argument in all three packages involved.

A word of caution about this example is in order. We will show an example that can be reproduced in a time-efficient manner and thus the Markov chain Monte Carlo chains (MCMC) of the TFR and e_0 models might not converge. Thus results that will be shown throughout this paper are not UN official projections and may not even be realistic. For a real simulation, increase the number of iterations (argument `iter`), set it to `"auto"`, or use default values. Using multiple chains and setting an argument `parallel` to `TRUE` in steps 1 and 2 below could improve estimation results in less run time. Finally, steps 1 and 2 can be carried out independently of one another. If all the steps are processed sequentially, allow about an half an hour processing time on a current standard laptop.

1. Estimate TFR parameters for the phase II and phase III models (Raftery, Alkema, and Gerland 2014a) and generate TFR projections (about 15 minutes):

```
R> sim.dir.tfr <- file.path(getwd(), "TFRprojections")
R> run.tfr.mcmc(iter = 1000, nr.chains = 1, thin = 1,
+    output.dir = sim.dir.tfr, seed = 1)
R> run.tfr3.mcmc(sim.dir = sim.dir.tfr, iter = 1000,
+    nr.chains = 1, thin = 1, seed = 1)
R> tfr.predict(sim.dir = sim.dir.tfr, nr.traj = 100,
+    burnin = 500, burnin3 = 500, seed = 1)
```

   For details on the estimation, MCMC diagnostics, TFR projections etc, see Ševčíková *et al.* (2011).

2. Estimate e_0 parameters using female data and generate joint female and male projections of life expectancy (about 10 minutes):

```
R> sim.dir.e0 <- file.path(getwd(), "e0_projections")
R> run.e0.mcmc(sex = "Female", iter = 1000, nr.chains = 1, thin = 1,
+    output.dir = sim.dir.e0, seed = 1)
R> e0.predict(sim.dir = sim.dir.e0, nr.traj = 100, burnin = 500,
+    seed = 1)
```

3. Generate probabilistic population projections (about 10 minutes):

```
R> sim.dir.pop <- file.path(getwd(), "pop_projections")
R> pop.pred <- pop.predict(output.dir = sim.dir.pop,
+    inputs = list(tfr.sim.dir = sim.dir.tfr,
+      e0F.sim.dir = sim.dir.e0, e0M.sim.dir = "joint_"),
+    keep.vital.events = TRUE, verbose = TRUE)
```

   The last call generates 100 trajectories, one for each trajectory of TFR and e_0. (In practice far more trajectories would be needed, but 100 can be run relatively quickly for illustrative purposes.)

At the end of this command sequence the user has three new directories in the working directory, for TFR, e_0 and population, respectively. One can now use functions from the **bayesTFR**, **bayesLife**, and **bayesPop** packages, respectively, to analyze and visualize results.

To access population projections in later sessions, issue the command:

```
R> pop.pred <- get.pop.prediction(sim.dir.pop)
```

3.3. Population trajectories

Population trajectories can be viewed on a country-specific basis. A simple `summary` function gives one a quick look at quantiles of a country's projections:

```
R> country <- "Netherlands"
R> summary(pop.pred, country)

Projections: 17 ( 2018 - 2098 )
Initial time point: 2013
Observed time points: 13 ( 1953 - 2013 )
Trajectories: 100
Number of countries: 178

Country: Netherlands

Projected Population:
      mean       SD  2.5%    5%   10%   25%   50%   75%   90%   95% 97.5%
2013 16925    0.000 16925 16925 16925 16925 16925 16925 16925 16925 16925
2018 17177   50.551 17081 17100 17117 17139 17176 17209 17247 17253 17285
```

```
2023 17388  119.019 17196 17210 17232 17307 17384 17462 17554 17572 17610
2028 17556  195.290 17235 17262 17305 17411 17553 17676 17792 17881 17940
2033 17647  267.376 17141 17191 17266 17486 17638 17805 18003 18050 18198
2038 17648  344.829 17049 17090 17171 17469 17635 17838 18067 18299 18408
2043 17582  429.780 16844 16962 17038 17321 17559 17781 18130 18522 18546
2048 17484  529.638 16602 16697 16882 17087 17455 17737 18217 18553 18712
2053 17386  633.788 16331 16434 16624 16897 17352 17706 18210 18568 18870
2058 17296  731.828 16075 16086 16463 16738 17265 17638 18301 18722 19029
2063 17239  828.020 15804 15932 16276 16692 17191 17625 18362 18812 19137
2068 17218  926.868 15655 15774 16163 16612 17124 17666 18480 18899 19354
2073 17216 1048.189 15471 15640 16011 16498 17100 17694 18677 19132 19626
2078 17193 1177.040 15218 15427 15790 16405 17067 17787 18840 19303 19851
2083 17155 1320.375 14911 15256 15520 16249 16999 17801 18974 19519 20199
2088 17103 1472.944 14623 14994 15219 16084 16926 17922 19121 19776 20524
2093 17045 1623.739 14398 14682 15030 15939 16827 18029 19496 19918 20844
2098 16990 1783.298 14106 14433 14771 15730 16785 18080 19601 20136 21120
```

The unit of the projections corresponds to the unit of the initial population, which is in this case a thousand.

Trajectories can be plotted using:

```
R> pop.trajectories.plot(pop.pred, country = country, sum.over.ages = TRUE)
```

The resulting plot is shown in the left panel of Figure 3. The `pop.trajectories.plot` accepts arguments for specifying sex and age. For example, the right panel of Figure 3 shows the projection for male population up to age 14:

```
R> pop.trajectories.plot(pop.pred, country = country, sex = "male",
+    age = 1:3, sum.over.ages = TRUE)
```

If `sum.over.ages` is `FALSE`, separate plots for each age group are generated. Regarding the `age` argument, see below about defining ages in **bayesPop**. An optional argument `nr.traj` can be used to control how many trajectories are plotted. It defaults to the total number of available trajectories, which is 100 in our example.

In addition to plotting trajectories by time, one can view them by age, see the left panel of Figure 4:

```
R> pop.byage.plot(pop.pred, country = country, year = 2100)
```

The argument `year` can be either a projected year or a past time point, i.e., any year from the x-axis of Figure 3. To compare the age structure from multiple years, the right panel of Figure 4 shows an analogous plot for 2060 (with 50 trajectories) and 1960 in the same graph:

```
R> pop.byage.plot(pop.pred, country = country, year = 2060,
+    pi = 80, nr.traj = 50)
R> pop.byage.plot(pop.pred, country = country, year = 1960, add = TRUE,
+    col = "blue", show.legend = FALSE)
```

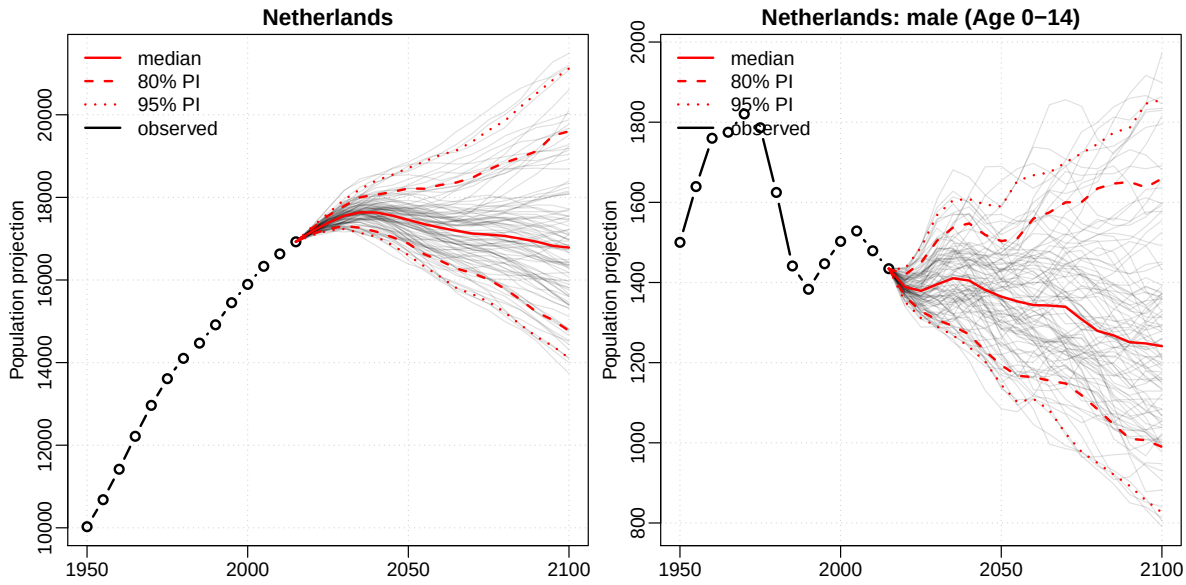

Figure 3: Projected trajectories by time for the Netherlands for total population (left panel) and for males aged 0–14 (right panel).

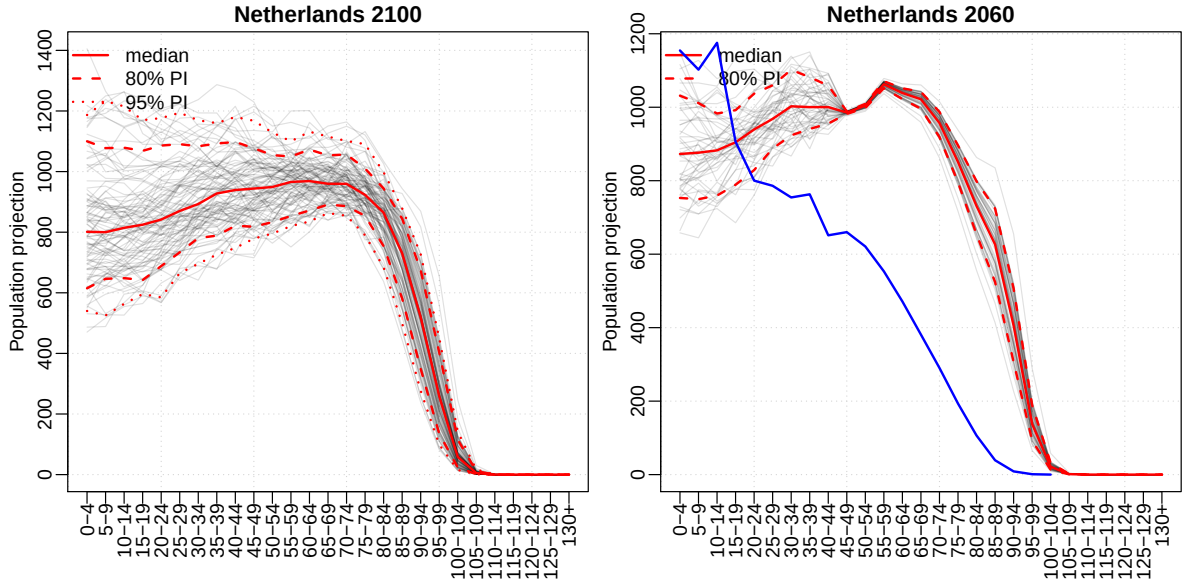

Figure 4: Projected trajectories by age for the Netherlands: in 2100 (left panel) and in 2060 (right panel). In the right panel, the main projections (red lines) are compared to 1960 (blue line).

Tabular analogues to the trajectory plots are implemented in the functions `pop.trajectories.table` and `pop.byage.table`, respectively. All four functions understand an argument `pi` specifying the probability intervals to be viewed. The functions `pop.trajectories.plotAll` and `pop.byage.plotAll` can be used to plot population trajectories for all countries at once.

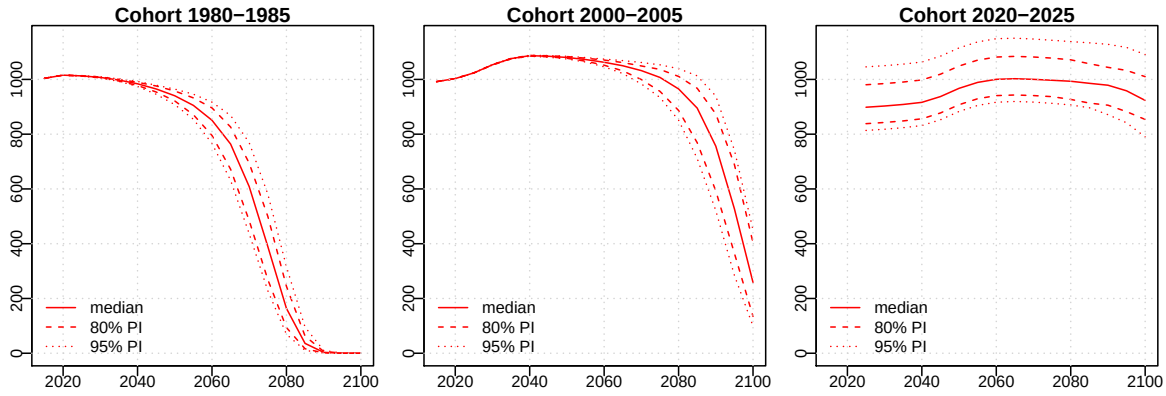

Figure 5: Projections of three different cohorts for the Netherlands.

Cohort projections

It is often of interest to obtain projections for specific cohorts. Simply running

```
R> pop.cohorts.plot(pop.pred, country = country)
```

will show projections of population born in ten different cohorts, starting with the present cohort. To extract the underlying data containing all cohorts (including the ones born in the past), one can use

```
R> cohort.data <- cohorts(pop.pred, country = country)
```

The result is a list of matrices where each list item corresponds to one cohort:

```
R> head(names(cohort.data))
```

```
[1] "1880-1885" "1885-1890" "1890-1895" "1895-1900" "1900-1905" "1905-1910"
```

Each of the elements is a matrix where the first row corresponds to the median and the following rows correspond to quantiles which are configurable via the optional argument `pi`. The `cohort.data` object can be also passed to the plotting function above. For example, the following code produces a plot of projections for two already born cohorts and one future cohort, as shown in Figure 5:

```
R> pop.cohorts.plot(pop.pred, cohort.data = cohort.data,
+    cohorts = c(1980, 2000, 2020))
```

Finally, all functions described in this section accept an argument called `expression` for exploring trajectories of other population quantities which will be described in Section 5.

Defining age

Many functions in the package accept an argument called `age`. It refers to an index of an ordered array of five-year age groups, as shown in Table 1. Functions that handle observed

Age group	0–4	5–9	10–14	15–19	20–24	25–29	30–34	35–39	40–44
`age`	1	2	3	4	5	6	7	8	9
Age group	45-49	50–54	55–59	60–64	65–69	70–74	75–79	80–84	85–89
`age`	10	11	12	13	14	15	16	17	18
Age group	90–94	95–99	100–104	105–109	110–114	115–119	120–124	125–129	130+
`age`	19	20	21	22	23	24	25	26	27

Table 1: Definition of the argument `age` in various functions of the package. It is an index of an ordered array of age groups.

data accept values of the age index up to 21 (in which case 21 corresponds to the age group 100+), whereas functions that deal with projections accept values of the age index up to 27. This distinction is due to the low availability of observed data for high ages. However, considering increasing longevity, it is useful to generate projections for those high ages. The package extrapolates mortality to higher ages using the Kannisto method. Other measures can be derived by using life tables and assuming that the initial population at those ages is zero. Ševčíková *et al.* (2016a) describes this in more detail.

A `bayesPop.prediction` object contains a component called `ages` which is an array of the starting ages of each age group. Thus it can be used to determine the correspondence between index and age, for example the one used in Figure 3 for ages 0–14:

```
R> which(pop.pred$ages < 15)

[1] 1 2 3

R> pop.pred$ages[1:3]

[1]  0  5 10
```

4. Probabilistic population pyramids

The **bayesPop** package supports plotting probabilistic population pyramids for any given country and year. In addition, multiple years can be plotted in one pyramid. There are two different kinds of pyramids – a *classic pyramid* consisting of boxes, and a so-called *trajectory pyramid* which is created using age trajectories. The classic pyramid can display projections for up to two years in one pyramid with one set of probability intervals; the trajectory pyramid can include any number of years and any number of probability intervals.

A classic pyramid can be created using the function `pop.pyramid`:

```
R> pop.pyramid(pop.pred, country, year = c(2100, 2015), age = 1:23)
```

Here we are comparing the end year of the projections with the current year (see Figure 6 on the left). An optional argument `pi` for defining the probability intervals shown can be given. In addition, the function accepts various arguments for controlling the appearance of

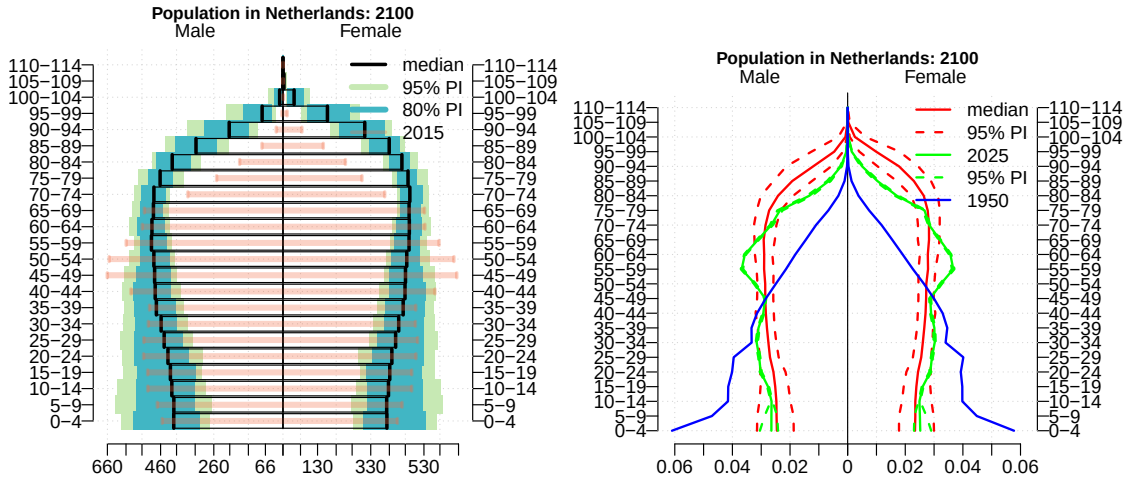

Figure 6: Probabilistic population pyramid for the Netherlands. The classic type on the left compares the end year of the projections with the current year. The trajectory type on the right compares three different years on the scale of proportions.

the pyramid, such as colors, height and thickness of the boxes etc.; see below for an example. Using the optional argument `indicator`, births and deaths can be also shown in a pyramid, if vital events were stored during the prediction.

The following code creates a trajectory pyramid with three years (the end year, the second prediction year, and the first observed year) with 95% probability intervals around the two prediction years:

```
R> pop.trajectories.pyramid(pop.pred, country, year = c(2100, 2025, 1950),
+     age = 1:23, pi = 95, nr.traj = 0, proportion = TRUE)
```

It results in the right graph of Figure 6. Here the argument `proportion` is used, which switches the x-axis to a proportional scale. Note that the order of the values in the `year` argument matters, especially in the classic pyramid case. The first value is used to create the main pyramid (i.e., with boxes of probability intervals in classic pyramid and using red color in case of trajectory pyramid), whereas the remaining ones are used for the additional pyramids in the graph.

The functions `pop.pyramidAll` and `pop.trajectories.pyramidAll` can be used to produce pyramids for all countries and for a set of years at once. The `year` argument is then expected to be a list where each element is a vector to be passed to the underlying function, i.e., `pop.pyramid` or `pop.trajectories.pyramid`. For example, to create the pyramid on the left-hand side of Figure 6 and a similar pyramid comparing 2050 to the current year for all countries, one can do:

```
R> pop.pyramidAll(pop.pred, year = list(c(2100, 2015), c(2050, 2015)),
+     age = 1:23, output.dir = "mypyramids")
```

This will create a PNG file for all combinations of countries and year elements, in this case two files per country, and place it into the directory "`mypyramids`". Alternatively, one can

set the optional argument `one.file` to `TRUE` in which case all pyramids are put into a single PDF file.

Pyramids for user-defined data

So far, we have shown how to create probabilistic pyramids for an object of class `bayesPop.prediction`. However, both pyramid functions are S3 methods that can be also applied to an object of class `bayesPop.pyramid`. This is a structure containing all the data necessary for a pyramid graph; they do not need to be created using **bayesPop**. Thus any data fitting into a pyramid structure can be used. An S3 method called `get.bPop.pyramid` can convert a matrix, a data frame, or a list of matrices or data frames into the `bayesPop.pyramid` structure. In addition, it can be also applied to a `bayesPop.prediction` object. One advantage of the latter conversion is that it gives the user a finer control over the plot.

The main element of the `bayesPop.pyramid` list is called `pyramid`. It is a list of data frames, each having two columns containing data for the left and right side of one pyramid and row names determining the age labels. Consider an example dataset containing population estimates for Washington State and King County in 2011:

```
R> data <- read.table(file.path(
+    find.package("bayesPop"), "ex-data", "popestimates_WAKing.txt"),
+    header = TRUE, row.names = 1)
R> head(data)
```

```
      WA.male WA.female King.male King.female
0-4    224883    214629     61537       58672
5-9    219450    209870     57814       55298
10-14  224828    213562     56848       53958
15-19  234336    221463     59294       56604
20-24  238228    223712     65651       64287
25-29  244381    233876     81481       78426
```

In order to show the two pyramids in one graph, we create two data frames with the same column names:

```
R> WA <- data[, c("WA.male", "WA.female")]
R> colnames(WA) <- c("M", "F")
R> Ki <- data[, c("King.male", "King.female")]
R> colnames(Ki) <- c("M", "F")
```

Now, one can create a `bayesPop.pyramid` object, specifying which columns contain data for the left and right part of the pyramid, respectively:

```
R> pyr <- get.bPop.pyramid(list(WA, Ki), LRcolnames = c("M", "F"),
+    legend = c("Washington", "King County"))
```

Simply using `plot` and optionally specifying some aesthetics will create the pyramid shown on the left-hand side of Figure 7:

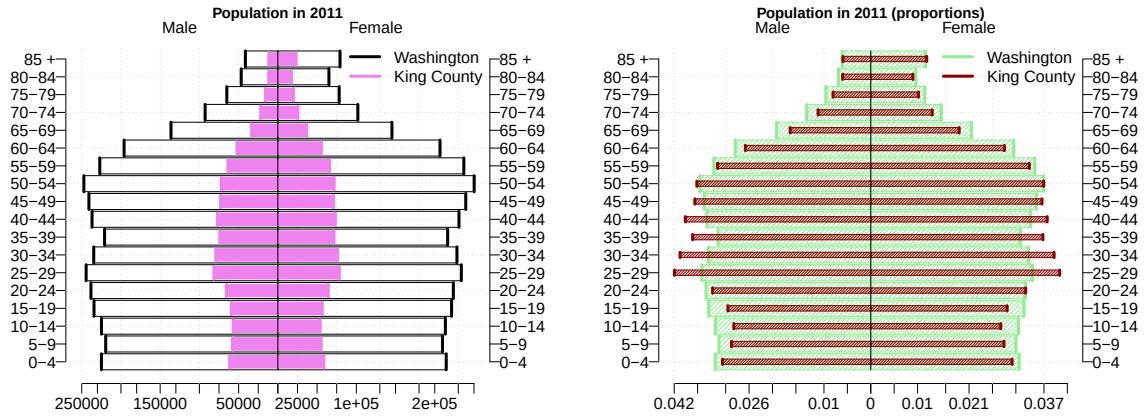

Figure 7: Population pyramids for user-defined data. The pyramid on the right is on the scale of proportions.

```
R> plot(pyr, main = "Population in 2011",
+    pyr2.par = list(height = 0.7, col = "violet", border = "violet"))
```

It can also be useful to compare such data on the scale of proportions, where what is plotted is not the actual numbers of people in each age group, but the numbers as a proportion of the total population. The following code creates such an object and its graph (shown in Figure 7 on the right):

```
R> pyr.prop <- get.bPop.pyramid(list(WA, Ki), is.proportion = NA,
+    LRcolnames = c("M", "F"), legend = c("Washington", "King County"))
R> pop.pyramid(pyr.prop, main = "Population in 2011 (proportions)",
+    pyr1.par = list(col = "lightgreen", border = "lightgreen",
+        density = 30),
+    pyr2.par = list(col = "darkred", border = "darkred",
+        density = 50, height = 0.3))
```

If the argument `is.proportion` has a logical value, it determines whether the data are on a proportional scale. An `NA` value means that the data frames, here `WA` and `Ki`, are not on a proportional scale but that such a scale is desired and thus, should be computed on the fly. Using the aesthetic arguments `pyr1.par` (for the main pyramid) and `pyr2.par` (for the secondary pyramid) allows the user to create a wide variety of different pyramid graphs.

The `plot` function is an alias for `pop.pyramid`. If the `pop.trajectories.pyramid` function is to be used, it should be called explicitly.

Apart from the `pyramid` element, the `bayesPop.pyramid` object also contains an element for storing the probability intervals, called `CI`, which can be passed directly to the function `get.bPop.pyramid`. Thus uncertainty can be included in the visualization of user-defined pyramid data. The `pop.trajectories.pyramid` can display any number of pyramids and probability interval sets into one graph, whereas `pop.pyramid` uses only the first two elements of `pyramid` and only one element of `CI`. See the manual for more details on the structure of `bayesPop.pyramid`.

5. bayesPop expressions

As mentioned in Section 3.1, a `bayesPop.prediction` object contains information about sex- and age-specific population projections. It is often of interest, however, to analyze quantities derived from the basic population counts and vital rates, such as potential support ratio, mean age at child-bearing, median age, and so on. For this purpose, the package implements a simple expression language that allows one to compute such quantities on the fly.

A **bayesPop** expression is a collection of *basic components* connected via usual arithmetic operators and combined using parentheses. Standard R functions and pre-defined functions can be also used within expressions.

Basic component

A basic component of an expression is a character string that consists of four sub-strings, the first two of which are mandatory. They must be in the following order:

1. Measure identification. The following upper-case characters are currently allowed and one of them must be provided:

   `P` Population

   `D` Deaths

   `B` Births

   `S` Survival ratio

   `F` Fertility rate

   `R` Percent age-specific fertility

   `M` Mortality rate

   `Q` Probability of dying

   `G` Net migration

   Note that all but the `P` and `G` indicators can be used only if `keep.vital.events` was switched on during the prediction. `P` and `G` are always available. Since net migration, `G`, is currently a deterministic input to the projections, it results in only one trajectory, namely the input.

2. Country part. This mandatory part can be a numerical country code, or a two- or three-character ISO 3166 code (International Organization for Standardization 2016), or characters `"XXX"` which serve as a wildcard for a country code. For example, `"P528"`, `"PNL"`, and `"PNLD"` are all expressions for the total population of the Netherlands. The use of `"XXX"` is limited to specific functions and will be discussed later in this section.

3. Sex sub-string. The country part can be optionally followed by either `"_F"` or `"_M"`, specifying female or male indicator, respectively. An expression consisting of two basic components `"P528_F / P528"` gives the ratio of female to total population in the Netherlands.

4. Age sub-string. If the age sub-string is used, the basic component is concluded by an array of age indices (as defined in Table 1). Such an array is delimited by either brackets

("[ ]") or curly braces ("{ }"). The former invokes a summation of counts over the given ages, while the latter is used when no summation is desired. Note that if the age sub-string is missing, the counts are automatically summed over all ages. To use all ages without summing, empty curly braces can be used. For example, the number of females in childbearing age in France can be calculated as "PFR_F[4:10]". As another example, the potential support ratio can be defined as "PFR[5:13] / PFR[14:27]".

In addition to the age index in Table 1, the indicators S, M and Q also allow an index -1 which corresponds to the age group 0–1, and an index 0 which corresponds to the age group 1–4.

Not all combinations of the four parts above make sense. For example, fertility rate can be combined only with female sex and a subset of the age groups, namely the childbearing ages (indices 4 to 10). Births are also restricted to those age groups. The rate-like indicators S, M, and Q should include all four components, since summing over sexes or age groups is meaningless for this type of measure.

Connecting components

When an expression is evaluated, each basic component is replaced by the corresponding data in the form of a four-dimensional array with the following dimensions:

1. Country dimension: It is equal to one if a specific country code is given. If "XXX" is used in the country sub-string, this dimension equals the number of countries in the prediction object.

2. Age dimension: It is equal to one if the age sub-string is missing or is defined within square brackets. If the age is defined within curly braces, this dimension corresponds to the length of the age array.

3. Time dimension: Depending on the context in which the expression is evaluated, this dimension corresponds to either the number of projection periods or the number of observation periods.

4. Trajectory dimension: It corresponds to the number of trajectories in the prediction object, or if the expression is evaluated on observed data, it is equal to one.

This array is returned by the function get.pop, which is evaluated either on projections or observed data, controlled by the logical argument observed.

Various arithmetic operations, such as +, -, *, /, ^, %%, %/%, and R functions can be performed on these four-dimensional arrays. An expression should be constructed in such a way that the age dimension is eliminated, for example by using the apply function or one of the pre-defined functions described below. An exception to this rule is when an expression is used in the context of functions pop.byage.plot, pop.byage.table, or the cohort functions, as illustrated below.

There are a few aspects to keep in mind when combining basic components. They are rooted in the fact that the combined arrays must have the same dimensions. For example, the deterministic indicator G cannot simply be combined with the probabilistic components, unless it is on observed data. In such cases, a special function, pop.combine (see below), needs to

be used. Furthermore, if using curly braces, the age index of the basic components must have the same length. For `B`, `F` and `R`, only age indices between 4 and 10 are allowed, so that `"BNL{}"` has length 7 on its age axis whereas `"PNL{}"` has length 27 for predictions and 21 for observed data. Therefore, if B, F and R are combined with other indicators, the age index specified must be of the correct length, e.g., `"BNL{} / PNL{4:10}"`. For debugging purposes, one can use the `get.pop` function to check dimensions of basic components:

```
R> B <- get.pop("BNL{}", pop.pred, observed = FALSE)
R> P <- get.pop("PNL{4:10}", pop.pred, observed = FALSE)
R> all(dim(B) == dim(P))

[1] TRUE
```

Pre-defined functions

There are a few pre-defined functions implemented in the package for convenience. The most commonly used are:

- `gmedian` and `gmean` for computing the median and mean of grouped data;
- `pop.apply` is an `apply` wrapper for applying a function along the age dimension;
- `pop.combine` for combining basic components of different shapes;
- `age.index01` for an index to age groups 0, 1–4, 5–9, ... (allowed for `S`, `M`, and `Q`);
- `age.index05` for an index to age groups 0–4, 5–9, ...

Furthermore, to generate an expression for the mean age of childbearing, one can use the function `mac.expression(...)`, see below for an example.

Using expressions in ***bayesPop*** *functions*

All functions that accept expressions have an argument called `expression`.

Expressions can be used to view projection trajectories by time using functions `pop.trajectories.plot` and `pop.trajectories.table`, as well as trajectories by age using functions `plot.byage.plot` and `pop.byage.table`. The former two evaluate expressions on both, observed and projected data. Each of the latter two accepts an argument `year` that specifies on which data it is evaluated. In the case of projected data, the functions use the trajectory dimension of the resulting array to compute desired quantiles, and possibly to show trajectories in the plot. For example, using the `pop.pred` object created in Section 3.2, we can plot the median age of women in childbearing ages in Germany by

```
R> expr <- "pop.apply(PDE_F{4:10}, gmedian,
+    cats = seq(15, by = 5, length = 8))"
R> pop.trajectories.plot(pop.pred, nr.traj = 20, expression = expr,
+    main = "Median age of women in childbearing ages")
```

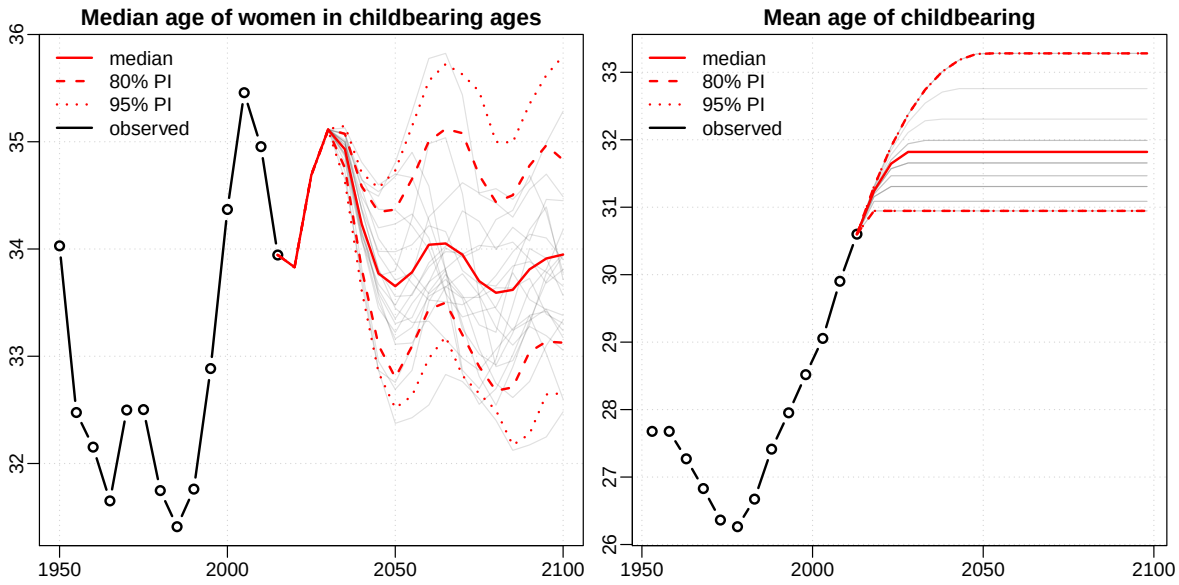

Figure 8: Results of evaluating expressions. Left panel: Median age of women in childbearing ages by time in Germany. Right panel: The mean age of childbearing in Germany.

which results in the left graph of Figure 8. The `cats` argument in the expression is passed to `gmedian` and it is a categories definition of the grouped data.

An expression for the mean age of childbearing can be obtained and plotted as follows:

```
R> expr <- mac.expression(country = "DE")
R> expr

[1] "(17.5*RDE[4] + 22.5*RDE[5] + 27.5*RDE[6] + 32.5*RDE[7] + 37.5*RDE[8] +
      42.5*RDE[9] + 47.5*RDE[10])/100"

R> pop.trajectories.plot(pop.pred, nr.traj = 20, expression = expr,
+    main = "Mean age of childbearing")
```

This results in the right graph of Figure 8.

For an age-specific plot, the number of births by mother's age per women of the same age can be viewed using

```
R> pop.byage.plot(pop.pred, expression = "BDE{} / PDE_F{4:10}", year = 2030,
+    nr.traj = 20, main = "Births by mother's age per woman - 2030")
```

which is shown on the left-hand side of Figure 9. This output can be obtained in tabular rather than graphical format using the function `pop.byage.table`. Both of these age-specific functions require the use of curly braces in the expressions, as the age axis of the resulting array must not be eliminated.

Expressions can be also used in the cohort functions described in Section 3.3. For example, the total number of children born to women of the next future cohort in Germany can be obtained by

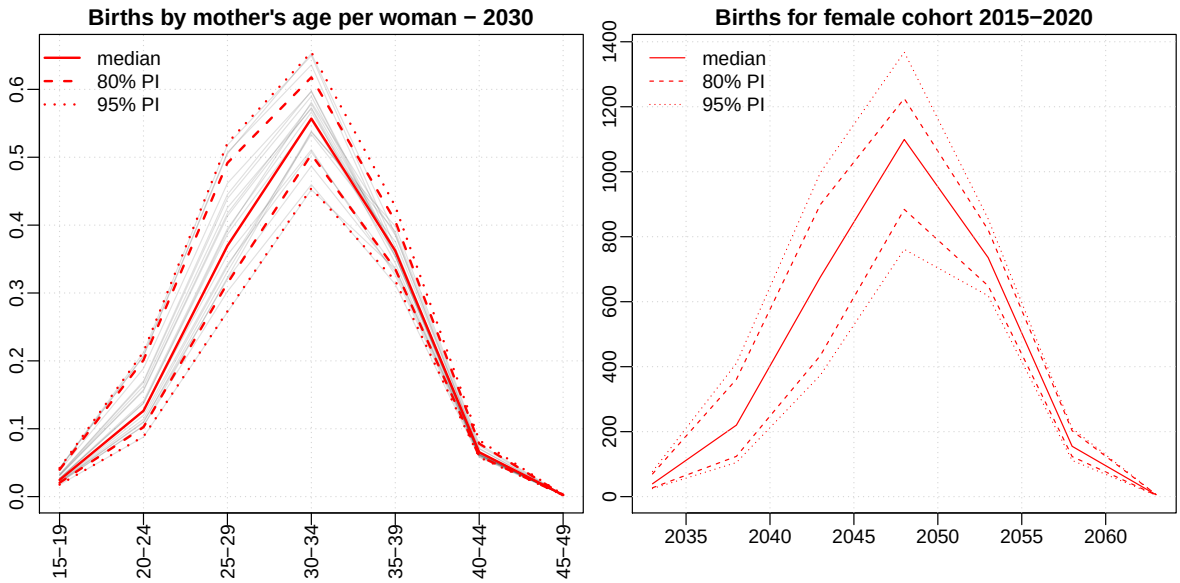

Figure 9: Results of evaluating expressions for Germany. Left panel: Births by mother's age per women of the same age in an age-specific plot. Right panel: Total number of births for women of the cohort 2015-2020 as a function of time.

```
R> pop.cohorts.plot(pop.pred, expression = "BDE{}", cohorts = 2015,
+    legend.pos = "topleft", main = "Births for female cohort 2015-2020")
```

which can be seen in the right panel of Figure 9. Note that expressions used in the cohort functions must contain curly braces.

In all functions mentioned so far in this section, the expressions must be country specific. However, basic components from different countries can be combined. For example, one could use `"PDE / PFR"` to view projection of the ratio of German to French population.

Expressions can also be used in maps. For that purpose, the `"XXX"` wildcard should be used. To generate a map with infant mortality in 2050, do

```
R> pop.map(pop.pred, expression = "MXXX[-1]", year = 2050)
```

The `pop.map` function is built on top of the **rworldmap** package (South 2016). Alternatively, one can use the `pop.map.gvis` function which builds on the **googleVis** package (Gesmann and de Castillo 2011). Creating a map via an expression involves, for each country, evaluating an expression in which the wildcard is replaced by its country code. This means loading trajectories of all countries from the disk one by one, evaluating the expression and obtaining results, which can be time-expensive. For that reason, the package has a simple caching mechanism, in which results of an expression evaluation for all time periods are stored on disk (in a file called `cache.rda` located in the prediction directory). Next time the same expression is used, for example with a different time point, the cached data are re-used. Even though creating a map with a new expression is processed in parallel if possible (depending on the number of cores in the user's computer) it still can take substantially more time than using a previously used expression. The cache is removed every time a new projection is

generated. Alternatively, the function `pop.cleanup.cache` can be used for a manual removal of the data.

In addition to maps, expressions with the `"XXX"` wildcard can be passed to other functions that involve all countries, such as `pop.trajectories.plotAll`, `plot.byage.plotAll`, or `write.pop.projection.summary`.

More examples of **bayesPop** expressions can be found by typing `?pop.expressions`.

6. Aggregations

In addition to producing population estimates at the country level, the UN also provides projections for population quantities aggregated over many different sets of countries, such as geographic regions and trading blocs. **bayesPop** offers two methods for producing probabilistic projections of aggregated quantities:

Country-based method: This combines the posterior samples from the different countries trajectory by trajectory: aggregation is done by simply summing the population counts in each trajectory across the countries of the regions in question.

If the input TFR and life expectancy trajectories came from the original Bayesian hierarchical models (BHM), this corresponds to the conditional independence assumptions in the BHM. If there is between-country correlation beyond that, the resulting posterior distributions may underestimate uncertainty.

However, this method can be made to take account of correlations between the forecast errors for different countries. Fosdick and Raftery (2014) proposed a method for taking the between-country correlation in TFR forecast errors into account. This is implemented in **bayesTFR** (controlled by the logical argument `use.correlation` in the `tfr.predict` function). If their method is used, the output from **bayesTFR** that is fed into **bayesPop** will take account of between-country correlations in TFR forecast errors.

Region-based method: Here aggregations are generated using a cohort component method, similarly to `pop.predict`, but the function operates on aggregated input components. While deterministic input components are aggregated on the fly, the method requires that aggregations of all probabilistic input components described in Section 3.1 exist. This can be achieved using the functions `run.tfr.mcmc.extra` and `tfr.predict.extra` from **bayesTFR** for TFR, and functions `run.e0.mcmc.extra` and `e0.predict.extra` from **bayesLife** for life expectancy.

In practice we have found that, when projecting aggregates of countries whose demographic histories are not well aligned, the regional method tends to overestimate uncertainty, often giving predictive intervals that are too wide.

Here is an example of aggregating over continents and over the whole world using the country-based method:

```
R> pop.aggr <- pop.aggregate(pop.pred, input.type = "country",
+    regions = c(900, 903, 908, 905, 935, 904))
```

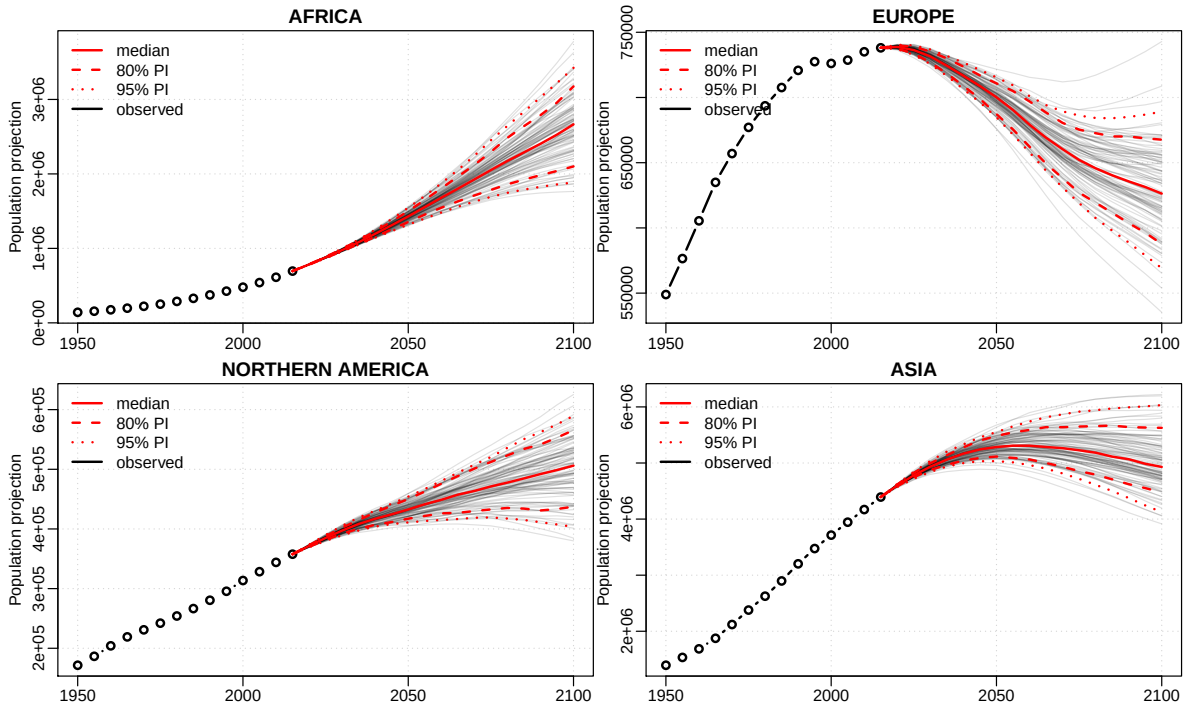

Figure 10: Population trajectories for aggregated regions obtained via the country-based method.

The region codes must correspond to the column "area_code" of the `UNlocations` dataset in the **wpp** package from which the corresponding names are extracted and kept in the `countries` table of the resulting object:

```
R> pop.aggr$countries
```

```
  code                              name
1  900                             WORLD
2  903                            AFRICA
3  908                            EUROPE
4  905                  NORTHERN AMERICA
5  935                              ASIA
6  904 LATIN AMERICA AND THE CARIBBEAN
```

Alternatively, user-defined aggregations are also supported (see the function help file for more information).

The function `pop.aggregate` accepts an optional argument `input.type` which defaults to the method name and is used for labeling the aggregation. Thus one can create several aggregations for the same prediction object. They are stored in the main simulation directory, here `sim.dir.pop` from Section 3.2, under the given name. In later R sessions, the object can be retrieved using

```
R> pop.aggr <- get.pop.aggregation(sim.dir.pop, name = "country")
```

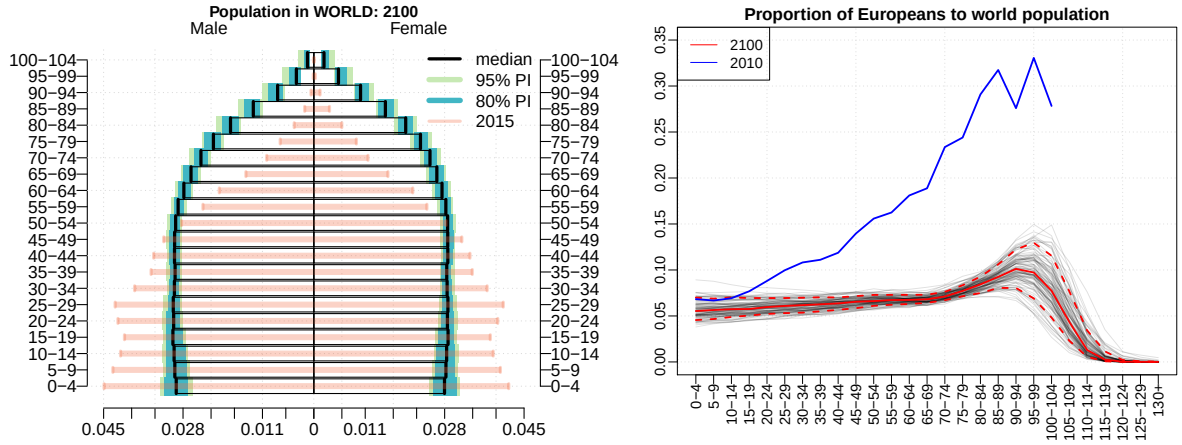

Figure 11: Visualizing results of aggregation. Left panel: Population pyramid for the world in 2010 and 2100. Right panel: Number of Europeans as a proportion of world population for each age group in 2100 and the same indicator in 2010, expressed as `"P908{} / P900{}"`.

The stored data have the same structure as in the case of the prediction object, as described in Section 3.1. Indeed, the object that results from the two calls above is again of class `bayesPop.prediction` and thus can be used in any of the summarizing and plotting function described in the previous sections, including in combination with expressions:

```
R> par(mfrow = c(2, 2))
R> for(country in c(903, 908, 905, 935))
+    pop.trajectories.plot(pop.aggr, country, sum.over.ages = TRUE)

R> pop.pyramid(pop.aggr, 900, year = c(2100, 2015), proportion = TRUE)

R> pop.byage.plot(pop.aggr, expression = "P908{} / P900{}", year = 2100,
+    main = "Proportion of Europeans to world population",
+    pi = 80, ylim = c(0,0.35), show.legend = FALSE)
R> pop.byage.plot(pop.aggr, expression = "P908{} / P900{}", year = 2015,
+    add = TRUE, show.legend = FALSE, col = "blue")
R> legend("topleft", legend = c(2100, 2015), col = c("red", "blue"), lty = 1)
```

The resulting graphs are shown in Figures 10 and 11. Note that the regions are aggregated only from countries that are available in the underlying `pop.pred` object. These do not yet include a large part of Africa, because we do not yet have probabilistic projections of life expectancy for the countries with generalized HIV/AIDS epidemics, many of which are in Africa. Thus the African projections are only illustrative.

The expression used in the last call of `pop.byage.plot` (right graph of Figure 11) combines indicators from two regions. It is also possible to combine non-aggregated indicators with aggregated ones:

```
R> pop.trajectories.plot(pop.pred, expression = "PIND / P900",
+    main = "Proportion of population of India to the world population")
```

In such a call, the original (non-aggregated) prediction object, here `pop.pred`, should be passed as the first argument. The function then tries to find the aggregated object automatically by iterating over the available aggregation objects until the region's code is found. In this process, an aggregation called "country" has priority above objects with other names.

7. Discussion

We have described an R package called **bayesPop** to produce and display probabilistic population projections, using a methodology which is being used by the United Nations Population Division as part of the process for producing its official population projections for all countries. The package produces a sample from a joint posterior predictive distribution of population quantities.

It allows the user to visualize the probabilistic projections in various ways, including different kinds of probabilistic population pyramids. It also includes an expression language that yields probabilistic projections of arbitrary user-defined derived future population quantities, such as the median age of the population, the potential support ratio or the ratio of the population of one country to that of another. Finally, it gives probabilistic projections of population quantities that are aggregated over an arbitrary set of countries, such as UN regions or trading blocs.

bayesPop is a command line package. However, there is also a graphical user interface, implemented in the **bayesDem** R package, for controlling all three packages **bayesPop**, **bayesTFR** and **bayesLife**, and visualizing their results. The UN's most recent official historical population estimates and population projections at the time of writing are contained in the data package **wpp2015**. The previous revisions of the UN's official *World Population Prospects* are available in the data packages **wpp2012** and **wpp2010**. Data in all three **wpp** packages can be visualized in a browser using the R package **wppExplorer** (Ševčíková 2016b), or viewed online at `https://rstudio.stat.washington.edu/shiny/wppExplorer/`.

There is now a wealth of R packages that do demographic analysis in some form, but relatively few oriented to human populations. Apart from **bayesPop**, the only one that we know of that does probabilistic projections of human populations is **demography** (Hyndman 2014), which does stochastic population forecasting using the functional data approach of Hyndman and Ullah (2007).

Several packages use statistical models to estimate and forecast age-specific mortality rates. The **YourCast** package is based on the methods of Girosi and King (2008). **MortalitySmooth** (Camarda 2012) uses P-splines to smooth and forecast age-specific mortality rates. The **HPbayes** package (Sharrow 2012; Sharrow, Clark, Collison, Kahn, and Tollman 2013) estimates the Heligman-Pollard model for age-specific mortality from mortality data using Bayesian Melding with incremental mixture importance sampling (IMIS; Raftery and Bao (2010)). The **LifeTables** package (Sharrow 2015; Clark and Sharrow 2011) produces model life tables by applying model-based clustering (Fraley and Raftery 2002) to the Human Mortality Database.

The **popReconstruct** package (Wheldon 2014) does probabilistic reconstruction of *past* population quantities rather than forecasting of the future; it is based on the methods of Wheldon, Raftery, Clark, and Gerland (2013). **Giza** (Striessnig 2012) is a graphics package that constructs panels of population pyramid plots.

There are several packages that provide tools for the construction and analysis of deterministic matrix population models, often oriented more to animal than to human populations. These include **popbio** (Stubben, Milligan, and Nantel 2016), based on the work of Caswell (2001), **popdemo** (Stott, Hodgson, and Townley 2016), and **primer** (Stevens 2012). The **IPMpack** package (Metcalf, McMahon, Salguero-Gomez, Jongelans, and Merow 2014) builds and analyzes integral projection models; these are also deterministic and take demographic rates as fixed inputs.

There are also several packages that provide tools for analyzing the interaction between demography and population genetics, again typically in the context of animal populations. These also usually treat demographic rates as fixed inputs. These include **AlleleRetain** (Welser, Grueber, and Jamieson 2012), which analyzes the effect of demography on allele retention, and **Biodem** (Boattini and Calboli 2015), which provides biodemographic functions, with an emphasis on kinship and inbreeding. Another such package is **lmf** (Kvaines 2013), which provides methods for inference about genetic selection in age-structured populations; it is based on the methods of Engen, Saether, Kvaines, and Jensen (2012).

Acknowledgments

This work was supported by NIH grants R01 HD054511 and R01 HD070936, and by Science Foundation Ireland under E.T.S. Walton visitor award 11/W.1/I2079. The authors thank the editor and two reviewers, Leontine Alkema, Samuel Clark and Patrick Gerland for very helpful comments and discussion. The authors thank the School of Mathematical Sciences and the Complex and Adaptive Systems Laboratory (CASL) at University College Dublin, Ireland, for their warm hospitality during the writing of this paper.

References

Alkema L, Raftery AE, Gerland P, Clark SJ, Pelletier F, Buettner T, Heilig GK (2011). "Probabilistic Projections of the Total Fertility Rate for All Countries." *Demography*, **48**, 815–839. doi:10.1007/s13524-011-0040-5.

Boattini A, Calboli FCF (2015). ***Biodem**: Biodemography Functions*. R package version 0.4, URL https://CRAN.R-project.org/package=Biodem.

Camarda CG (2012). "**MortalitySmooth**: An R Package for Smoothing Poisson Counts with P-Splines." *Journal of Statistical Software*, **50**(1), 1–24. doi:10.18637/jss.v050.i01.

Caswell H (2001). *Matrix Population Models: Construction, Analysis and Interpretation*. 2nd edition. Sinauer Associates, Sunderland.

Clark SJ, Sharrow DJ (2011). "Contemporary Model Life Tables for Developed Countries: An Application of Model-Based Clustering." *Working Paper 107*, Center for Statistics and the Social Sciences, University of Washington. URL http://www.csss.washington.edu/Papers/wp107.pdf.

Engen S, Saether BE, Kvaines T, Jensen H (2012). "Estimating Fluctuating Selection in Age-Structured Populations." *Journal of Evolutionary Biology*, **25**, 1487–1499. doi:10.1111/j.1420-9101.2012.02530.x.

Fosdick BK, Raftery AE (2014). "Regional Probabilistic Fertility Forecasting by Modeling Between-Country Correlations." *Demographic Research*, **30**, 1011–1034. doi:10.4054/demres.2014.30.35.

Fraley C, Raftery AE (2002). "Model-Based Clustering, Discriminant Analysis, and Density Estimation." *Journal of the American Statistical Association*, **97**, 611–631. doi:10.1198/016214502760047131.

Gerland P, Raftery AE, Ševčíková H, Li N, Gu D, Spoorenberg T, Alkema L, Fosdick BK, Chunn JL, Lalic N, Bay G, Buettner T, Heilig GK, Wilmoth J (2014). "World Population Stabilization Unlikely This Century." *Science*, **346**, 234–237. doi:10.1126/science.1257469.

Gesmann M, de Castillo D (2011). "**googleVis**: Interface between R and the Google Visualisation API." *The R Journal*, **3**(2), 40–44. URL https://journal.R-project.org/archive/2011-2/RJournal_2011-2_Gesmann+de~Castillo.pdf.

Girosi F, King G (2008). *Demographic Forecasting*. Princeton University Press.

Hyndman RJ (2014). ***demography***: *Forecasting Mortality, Fertility, Migration and Population Data*. R package version 1.18, URL https://CRAN.R-project.org/package=demography.

Hyndman RJ, Ullah S (2007). "Robust Forecasting of Mortality and Fertility Rates: A Functional Data Approach." *Computational Statistics & Data Analysis*, **51**, 4942–4956. doi:10.1016/j.csda.2006.07.028.

Ihaka R, Gentleman R (1996). "R: A Language for Data Analysis and Graphics." *Journal of Computational and Graphical Statistics*, **5**, 299–314. doi:10.1080/10618600.1996.10474713.

International Organization for Standardization (2016). "ISO 3166." http://en.wikipedia.org/wiki/ISO_3166-1_numeric, http://en.wikipedia.org/wiki/ISO_3166-1_alpha-2, http://en.wikipedia.org/wiki/ISO_3166-1_alpha-3.

Kvaines T (2013). ***lmf***: *Functions for Estimation and Inference of Selection in Age-Structured Populations*. R package version 1.2, URL https://CRAN.R-project.org/package=lmf.

Lee RD, Carter L (1992). "Modeling and Forecasting the Time Series of US Mortality." *Journal of the American Statistical Association*, **87**, 659–671. doi:10.1080/01621459.1992.10475265.

Lee RD, Tuljapurkar S (1994). "Stochastic Population Forecasts for the United States: Beyond High, Medium, and Low." *Journal of the American Statistical Association*, **89**, 1175–1189. doi:10.1080/01621459.1994.10476857.

Leslie PH (1945). "On the Use of Matrices in Certain Population Dynamics." *Biometrika*, **33**, 183–212. doi:10.1093/biomet/33.3.183.

Metcalf CJE, McMahon SM, Salguero-Gomez R, Jongelans E, Merow C (2014). *pkgIPMpack: Builds and Analyses Integral Projection Models (IPMs)*. R package version 2.1, URL https://CRAN.R-project.org/package=IPMpack.

National Research Council (2000). *Beyond Six Billion: Forecasting the World's Population*. National Academy Press, Washington, DC. doi:10.17226/9828.

Preston SH, Heuveline P, Guillot M (2001). *Demography: Measuring and Modeling Population Processes*. Blackwell, Malden.

Raftery AE, Alkema L, Gerland P (2014a). "Bayesian Population Projections for the United Nations." *Statistical Science*, **29**, 58–68. doi:10.1214/13-sts419.

Raftery AE, Bao L (2010). "Estimating and Projecting Trends in HIV/AIDS Generalized Epidemics Using Incremental Mixture Importance Sampling." *Biometrics*, **66**, 1162–1173. doi:10.1111/j.1541-0420.2010.01399.x.

Raftery AE, Chunn JL, Gerland P, Ševčíková H (2013). "Bayesian Probabilistic Projections of Life Expectancy for All Countries." *Demography*, **50**, 777–801. doi:10.1007/s13524-012-0193-x.

Raftery AE, Lalic N, Gerland P (2014b). "Joint Probabilistic Projection of Female and Male Life Expectancy." *Demographic Research*, **30**, 795–822. doi:10.4054/demres.2014.30.27.

Raftery AE, Li N, Ševčíková H, Gerland P, Heilig GK (2012). "Bayesian Probabilistic Population Projections for All Countries." *Proceedings of the National Academy of Sciences*, **109**, 13915–13921. doi:10.1073/pnas.1211452109.

R Core Team (2016). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Sharrow DJ (2012). ***HPbayes**: Heligman Pollard Mortality Model Parameter Estimation Using Bayesian Melding with Incremental Mixture Importance Sampling*. R package version 0.1, URL https://CRAN.R-project.org/package=HPbayes.

Sharrow DJ (2015). ***LifeTables**: A Package to Implement HMD Model Life Table System*. R package version 1.0, URL https://CRAN.R-project.org/package=LifeTables.

Sharrow DJ, Clark SJ, Collison MA, Kahn K, Tollman SM (2013). "The Age-Pattern of Increases in Mortality Affected by HIV: Bayesian Fit of the Heligman-Pollard Model to Data from the Agincourt HDSS Field Site in Rural Northeast South Africa." *Demographic Research*, **29**, 1039–1096. doi:10.4054/demres.2013.29.39.

South A (2016). ***rworldmap**: Mapping Global Data*. R package version 1.3-6, URL https://CRAN.R-project.org/package=rworldmap.

Stevens MHH (2012). ***primer**: Functions and Data for a Primer of Ecology with R*. R package version 1.0, URL https://CRAN.R-project.org/package=primer.

Stott I, Hodgson D, Townley S (2016). ***popdemo**: Demographic Modelling Using Projection Matrices*. R package version 0.2-3, URL https://CRAN.R-project.org/package=popdemo.

Striessnig E (2012). ***Giza***: *Constructing Panels of Population Pyramid Plots Based on Lattice*. R package version 1.0, URL https://CRAN.R-project.org/package=Giza.

Stubben C, Milligan B, Nantel P (2016). ***popbio***: *Construction and Analysis of Matrix Population Models*. R package version 2.4.3, URL https://CRAN.R-project.org/package=popbio.

United Nations (1956). *Manual III: Methods for Population Projections by Sex and Age*. New York, NY: Department of Economic and Social Affairs, Population Division. Volume 25 – Population Studies.

United Nations (1989). *The United Nations Population Projection Computer Program: A User's Manual*. New York, NY: Department of Economic and Social Affairs, Population Division.

United Nations (2013). *World Population Prospects: The 2012 Revision*. United Nations, New York.

United Nations (2015). *World Population Prospects: The 2015 Revision, Probabilistic Population Projections*. Population Division, Department of Economic and Social Affairs, United Nations, New York. URL http://esa.un.org/unpd/ppp/.

United Nations (2016). ***wpp2015***: *World Population Prospects 2015*. R package version 1.1-0, URL https://CRAN.R-project.org/package=wpp2015.

Ševčíková H (2016a). ***bayesDem***: *Graphical User Interface for* ***bayesTFR***, ***bayesLife*** *and* ***bayesPop***. R package version 2.5-1, URL https://CRAN.R-project.org/package=bayesDem.

Ševčíková H (2016b). ***wppExplorer***: *Explorer of World Population Prospects*. R package version 2.0-0, URL https://CRAN.R-project.org/package=wppExplorer.

Ševčíková H, Alkema L, Raftery AE (2011). "**bayesTFR**: An R Package for Probabilistic Projections of the Total Fertility Rate." *Journal of Statistical Software*, **43**(1), 1–29. doi:10.18637/jss.v043.i01.

Ševčíková H, Gerland P (2013). ***wpp2010***: *World Population Prospects 2010*. R package version 1.2-0, URL https://CRAN.R-project.org/package=wpp2010.

Ševčíková H, Gerland P, Andreev K, Li N, Gu D, Spoorenberg T (2014). ***wpp2012***: *World Population Prospects 2012*. R package version 2.2-1, URL https://CRAN.R-project.org/package=wpp2012.

Ševčíková H, Li N, Kantorová V, Gerland P, Raftery AE (2016a). "Age-Specific Mortality and Fertility Rates for Probabilistic Population Projections." In R Schoen (ed.), *Dynamic Demographic Analysis*, pp. 285–310. Springer. doi:10.1007/978-3-319-26603-9_15.

Ševčíková H, Raftery AE, Buettner T (2016b). ***bayesPop***: *Probabilistic Population Projection*. R package version 6.0-3, URL https://CRAN.R-project.org/package=bayesPop.

Ševčíková H, Raftery AE, Chunn J (2016c). ***bayesLife***: *Bayesian Projection of Life Expectancy*. R package version 3.0-1, URL https://CRAN.R-project.org/package=bayesLife.

Welser EL, Grueber CE, Jamieson IG (2012). "**AlleleRetain**: The Effect of Demography on Allele Retention in Populations of Animals." *Molecular Ecology Resources*, **12**, 1161–1167. doi:10.1111/j.1755-0998.2012.03176.x.

Wheldon MC (2014). ***popReconstruct**: Reconstruct Population Counts, Fertility, Mortality and Migration Rates of Human Populations of the Recent Past.* R package version 1.0-4, URL https://CRAN.R-project.org/package=popReconstruct.

Wheldon MC, Raftery AE, Clark SJ, Gerland P (2013). "Estimating Demographic Parameters with Uncertainty from Fragmentary Data." *Journal of the American Statistical Association*, **108**, 96–110. doi:10.1080/01621459.2012.737729.

Whelpton PK (1928). "Population of the United States, 1925–1975." *American Journal of Sociology*, **31**, 253–270. doi:10.1086/214667.

Whelpton PK (1936). "An Empirical Method for Calculating Future Population." *Journal of the American Statistical Association*, **31**, 457–473. doi:10.2307/2278370.

Affiliation:

Hana Ševčíková
Center for Statistics and the Social Sciences
University of Washington
Box 354322
Seattle, WA 98195-4322, United States of America
E-mail: hanas@uw.edu

Adrian E. Raftery
Departments of Statistics and Sociology
University of Washington
Box 354320
Seattle, WA 98195-4320, United States of America
Email: raftery@uw.edu

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
December 2016, Volume 75, Issue 5 *Submitted:* 2013-10-16
doi:10.18637/jss.v075.i05 *Accepted:* 2015-09-30