

# bayesPop: Probabilistic Population Projections

Hana Ševčíková and Adrian E. Raftery  
University of Washington, Seattle

Working Paper no. 135  
Center for Statistics and the Social Sciences  
University of Washington

October 16, 2013

## 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*.

KEY WORDS: Bayesian hierarchical model, 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 Millenium 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 *et al.* 2001).

Standard projections are deterministic, meaning that they yield a single value for each projected 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 *et al.* (2012). It consists of probabilistically projecting total fertility rate and life expectancy using Bayesian hierarchical models (Alkema *et al.* 2011; Raftery *et al.* 2013b), 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. The median projection from the method has been used as the official medium projection of the United Nations for all countries in the 2012 revision of the *World Population Prospects* (United Nations 2013).

Here we describe an R package called **bayesPop**, 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

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

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 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 fertility distribution by age,
- 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  $t \leq 0$ ), and
- projections of future sex- and age-specific net international migration.

In each time period  $t$ , the TFR is converted to age-specific fertility rates using the fertility distribution by age at  $t$ . 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). Then 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 plus and minus 0.5 child to the TFR, respectively, and 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.* 2013b). 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 *et al.* (2011)), then to simulate an equal number of trajectories of life expectancy (as implemented in **bayesLife**, see Ševčíková and Raftery (2013a)), 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 and Raftery 2013b). A graphical user interface (GUI) for the three packages, **bayesTFR**, **bayesLife** and **bayesPop**, is provided by the R package **bayesDem** (Ševčíková 2013a). 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.

## 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 **wpp2008**, **wpp2010**, and **wpp2012** (Ševčíková *et al.* 2013) provide such data for most countries. An argument `wpp.year` (which can be one of 2008, 2010, or 2012) determines the default dataset used in the projections. We will refer to the corresponding data package simply as a **wpp** package.

#### 3.1.1 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:

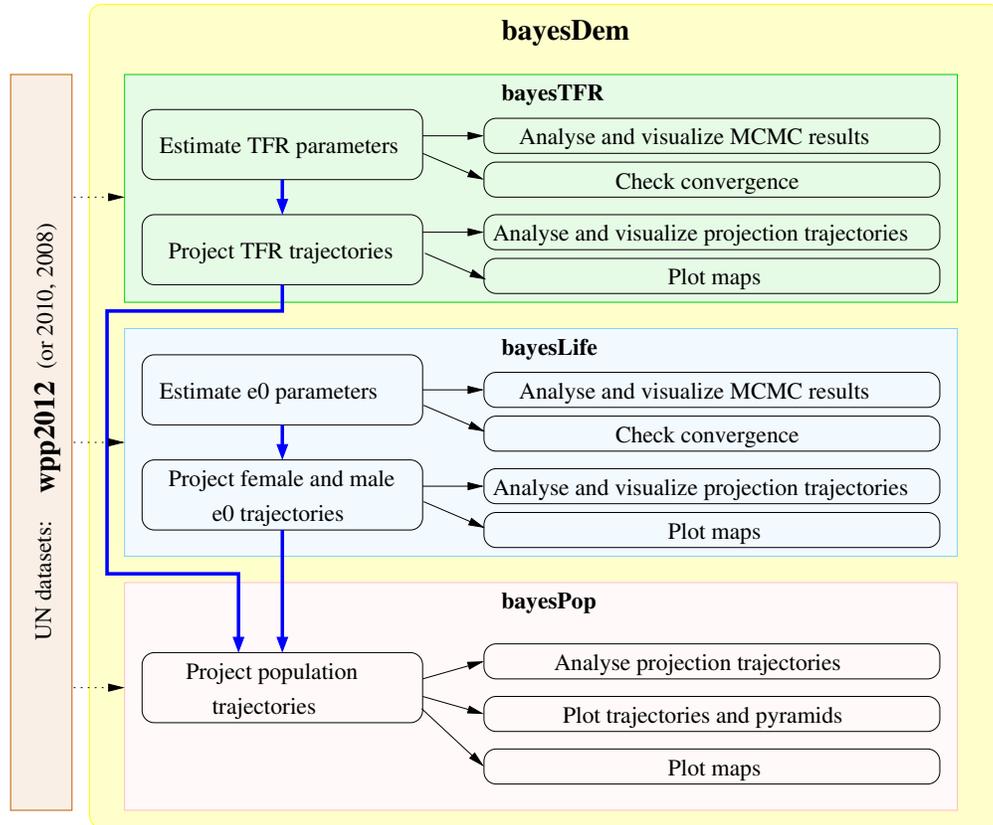


Figure 1: Structure of packages supported by **bayesDem**. Boxes shown on the left-hand side (connected by blue arrows) depict the main steps needed for generating probabilistic population projections. Boxes on the right-hand side show supporting functionality of the packages. The packages operate on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

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

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

**srb**: Projection of future sex ratio at birth.

**pasfr**: Projection of future age-specific fertility rates as percentages of TFR.

**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).

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, it is a set of trajectories that 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 (Lalic and Raftery 2012), and thus, they are 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 at most which are extracted from the **wpp** package, namely the projection median, and the low and high variants if available.

### 3.1.2 Other arguments

Besides specifying countries (argument `countries`), one can define the end year of the projection (`end.year`), the initial year (`start.year`), and how many trajectories to generate (`nr.traj`).

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.

### 3.1.3 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_countryx.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, and trajectories for sex- and age-specific fertility and mortality rates are stored in files called “`vital_events_countryx.rda`”. Thus, in such a case, there are two files per country.

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 projecting TFR and  $e_0$  can be done independently of one another. Each simulation runs in a different directory. The data used in this example are taken from the **wpp2012** package, as 2012 is currently 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 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.

1. Estimate TFR parameters for the phase II and phase III models (Raftery *et al.* 2013a) 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)
```

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**, 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: 18 ( 2013 - 2098 )
Initial time point: 2008
Observed time points: 12 ( 1953 - 2008 )
Trajectories: 100
Number of countries: 162
```

Country: Netherlands

Projected Population for both:

	mean	SD	2.5%	5%	10%	25%	50%	75%	90%	95%	97.5%
2008	16615	0.00	16615	16615	16615	16615	16615	16615	16615	16615	16615
2013	16844	56.16	16722	16747	16777	16810	16844	16882	16913	16939	16948
2018	17024	115.87	16826	16863	16886	16931	17024	17101	17158	17238	17260
2023	17155	195.24	16795	16847	16910	17007	17161	17277	17383	17492	17588
2028	17226	276.06	16669	16745	16867	17063	17230	17374	17601	17709	17735
2033	17231	365.87	16501	16572	16761	17010	17225	17428	17759	17799	17972
2038	17165	459.42	16279	16337	16536	16911	17157	17478	17768	17955	18068
2043	17041	555.71	15965	16026	16309	16710	17035	17348	17692	18028	18094
2048	16887	663.28	15658	15714	15978	16469	16926	17263	17649	17952	18212
2053	16738	778.18	15305	15414	15597	16227	16724	17249	17695	17972	18293
2058	16611	905.40	14853	15045	15374	16060	16586	17138	17752	18124	18328
2063	16512	1024.24	14464	14745	15163	15916	16506	17080	17730	18212	18414
2068	16441	1157.25	14191	14374	14863	15803	16484	17135	17769	18329	18595
2073	16378	1299.63	13896	14106	14625	15615	16426	17122	17781	18521	18882
2078	16324	1455.52	13553	13693	14499	15447	16368	17190	17807	18734	19218
2083	16270	1614.43	13239	13316	14283	15348	16307	17226	17899	19059	19521
2088	16210	1775.87	12867	12959	14126	15224	16160	17131	17956	19256	19762
2093	16154	1951.14	12503	12644	13962	15028	16073	17151	17982	19412	20033
2098	16093	2119.98	12134	12314	13814	14889	15930	17175	18086	19709	20228

Trajectories can be plotted using:

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

The resulting plot is shown in Figure 2. The `pop.trajectories.plot` accepts arguments for specifying sex and age. For example, Figure 3 shows the projection for male population

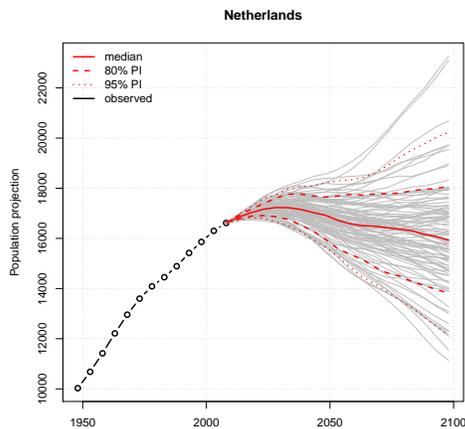


Figure 2: Projected trajectories for the Netherlands.

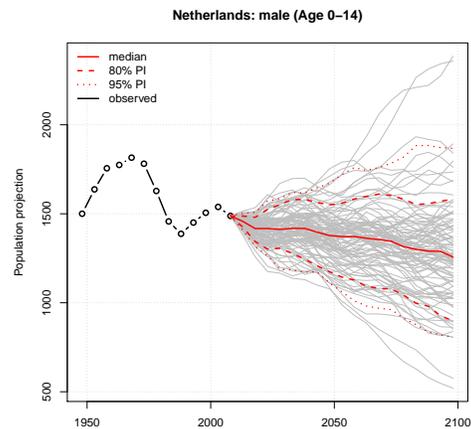


Figure 3: Projected trajectories for males aged 0–14 in the Netherlands.

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 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 2. To compare the age structure from multiple years, Figure 5 shows an analogous plot for 2060 and 1960 in the same graph:

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

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. Finally, all functions described in this section accept an

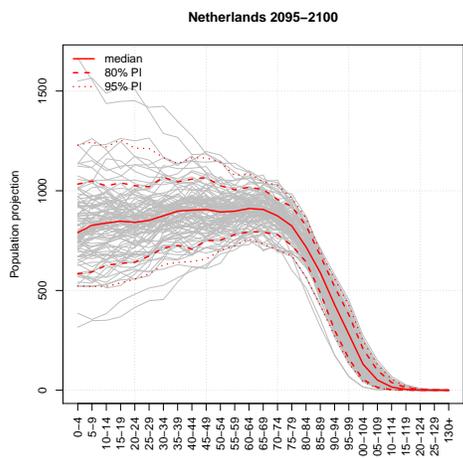


Figure 4: Projected trajectories by age for the Netherlands in 2100.

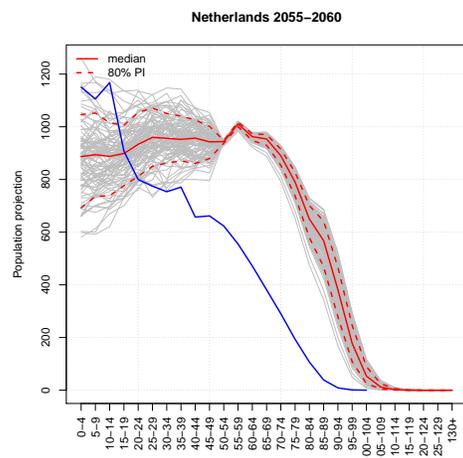


Figure 5: Projected population by age in the Netherlands in 2060 (red lines) compared to 1960 (blue line).

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

Age group	0–4	5–9	10–14	15–19	20–24	25–29	30–34	35–39	40–44
<code>age</code>	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
<code>age</code>	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+
<code>age</code>	19	20	21	22	23	24	25	26	27

argument called `expression` for exploring trajectories of other population quantities which will be described in Section 5.

### 3.3.1 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 data accept an age index up to 21, whereas functions that deal with projections accept an age index up to 27.

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 above for ages 0–14:

```
R> which(pop.pred$ages >= 0 & pop.pred$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(2096, 2010), age=1:25)
```

Here we are comparing the end year of the projections with the current year (see Figure 6). 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 the pyramid, such as colors, height and thickness of the boxes etc.; see below for an example.

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(2096, 2018, 1949),  
                           age=1:25, pi=95, nr.traj=0, proportion=TRUE)
```

It results in Figure 7. 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, e.g. `pop.pyramid` or `pop.trajectories.pyramid`. For example, to create a pyramid from Figure 6 and a similar pyramid comparing 2050 to the current year for all countries, one can do:

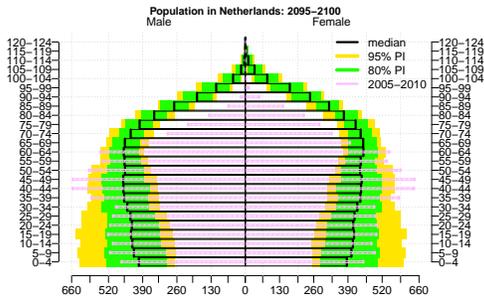


Figure 6: Probabilistic population pyramid (classic type) for the Netherlands, comparing the current year with the end year of the projections.

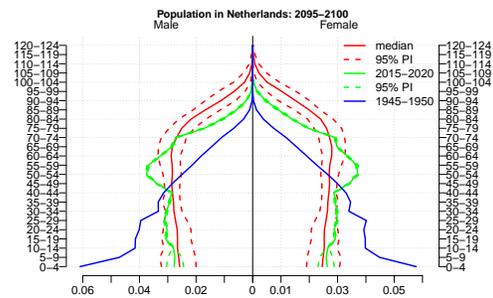


Figure 7: Probabilistic population pyramid (trajectory type) for the Netherlands, comparing three different years on the scale of proportions.

```
R> pop.pyramidAll(pop.pred, year=list(c(2096, 2010), c(2050, 2010)), age=1:25,
                  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 in the directory “mypyramids”.

#### 4.0.2 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")]; colnames(WA) <- c("M", "F")
R> Ki <- data[,c("King.male", "King.female")]; 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), legend=c("Washington", "King County"),
                          LRcolnames=c("M", "F"))
```

Simply using `plot` and optionally specifying some aesthetics will create the pyramid shown in Figure 8:

```
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:

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

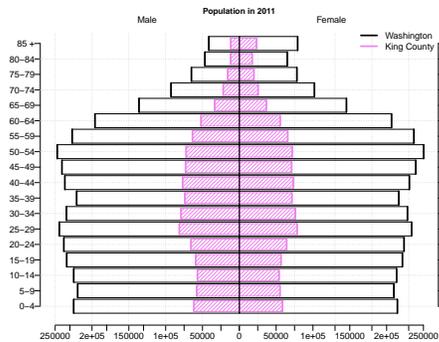


Figure 8: Population pyramid for user-defined data.

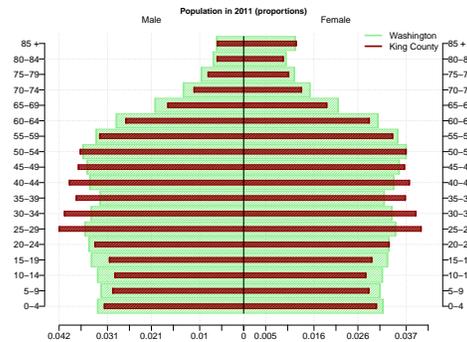


Figure 9: Population pyramid for user-defined data on a proportion scale.

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 `get.bPop.pyramid` function. Thus, uncertainty can be included in the visualization of user-defined pyramid data. As mentioned above, 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.

### 5.0.3 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

**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 2013), 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 a 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 of child-bearing 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 child-bearing ages (indices 4 to 10). Births are also restricted to those age groups. All the rate-like indicators (S, F, M, Q) should include all four components, since summing over sexes or age groups is meaningless for this type of measure.

#### 5.0.4 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 internal function `get.pop`, which is evaluated either on projections or observed data, controlled by the logical argument `observed`.

Some arithmetic operations, such as `+`, `-`, `*`, `/`, `^`, `%%`, `%/%`, and some 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 `plot.byage.plot` and `pop.byage.table`, 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 and F, 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 and F is 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 <- bayesPop:::get.pop("BNL{ }", pop.pred, observed=FALSE)
R> P <- bayesPop:::get.pop("PNL{4:10}", pop.pred, observed=FALSE)

R> all(dim(B) == dim(P))
TRUE
```

### 5.0.5 Pre-defined functions

There are a few pre-defined functions implemented in the package for convenience. These are:

- `gmedian` and `gmean` for computing the median and mean of grouped data;
- `age.func` for applying an arithmetic binary function to the data and corresponding age;
- `drop.age` for explicitly dropping the age dimension;
- `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, ...

### 5.0.6 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 accept an argument `year` that

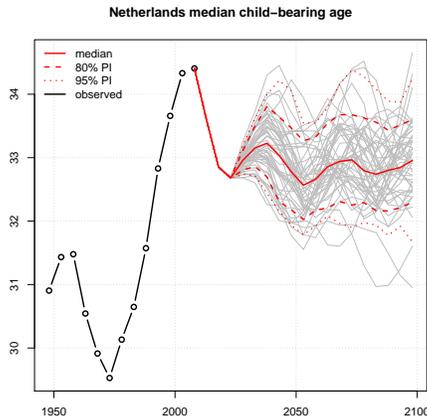


Figure 10: Result of the expression `"pop.apply(P528_F{4:10}, gmedian, cats=seq(15, by=5, length=8))"` (median child-bearing age) as a trajectories plot by time.

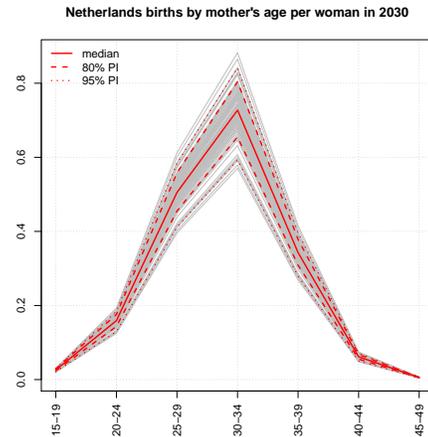


Figure 11: Result of the expression `"BNL{ } / PNL_F{4:10}"` (births by mother's age per women of the same age) as a trajectories plot by age.

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 a median child-bearing age in the Netherlands by

```
R> expr <- "pop.apply(P528_F{4:10}, gmedian, cats=seq(15, by=5, length=8))"
R> pop.trajectories.plot(pop.pred, nr.traj=50, expression=expr,
                        main="Netherlands median child-bearing age")
```

which results in Figure 10. The `cats` argument is passed to `gmedian` and it is a categories definition of the grouped data.

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="BNL{ } / PNL_F{4:10}", year=2030,
                 main="Netherlands births by mother's age per woman in 2030")
```

which is shown in Figure 11. 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.

In all four functions, 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 *et al.* 2012). 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:

**Independence 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 assumption

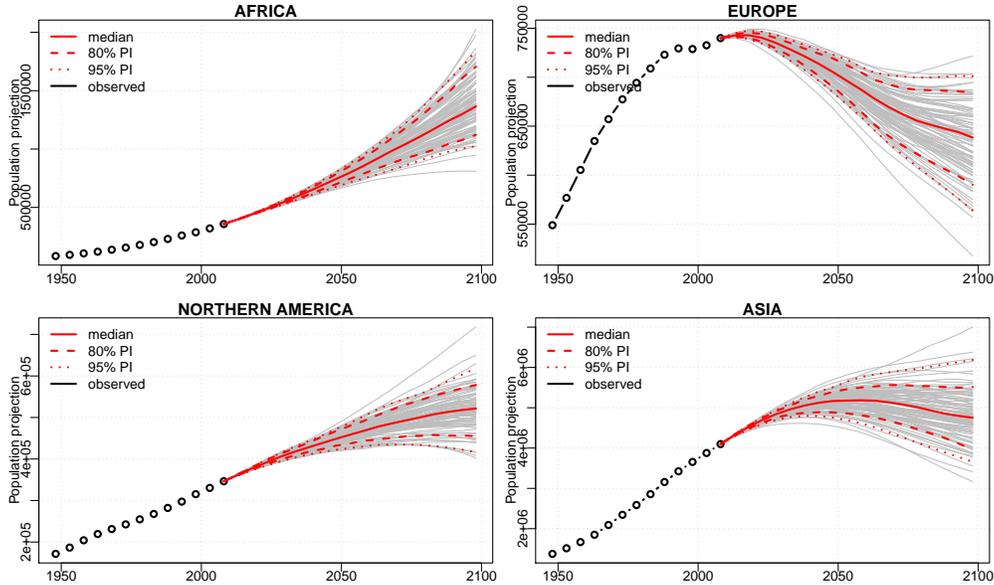


Figure 12: Population trajectories for aggregated regions obtained via the independence method.

tions 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 (2013) proposed a method for taking the between-country correlation in TFR forecast errors into account which is now implemented in newer versions of **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.

**Regional method:** Here aggregations are generated using a cohort component method, similarly to `pop.predict`, but where 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 `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 independence method:

```
R> pop.aggr <- pop.aggregate(pop.pred, method="independence",  
                             # World, Africa, Europe, Northern America, Asia, Latin Am.  
                             regions=c(900, 903, 908, 905, 935, 904))
```

The region codes must correspond to the column “area\_code” of the UNlocations dataset in the **wpp** package. Alternatively, user-defined aggregations are also supported (see the function help file for more information).

Function `pop.aggregate` accepts an optional argument `name` which defaults to the method name and is used for labelling 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)
```

and optionally passing its name in the `name` argument. 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 summarising 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, 2010), proportion=TRUE)  
  
R> pop.byage.plot(pop.aggr, expression="P908{} / P900{}", year=2100, pi=80,  
                 main = "Proportion of European population to world population",  
                 show.legend=FALSE)  
R> pop.byage.plot(pop.aggr, expression="P908{} / P900{}", year=2010,  
                 add=TRUE, show.legend=FALSE, col="blue")  
R> legend("topleft", legend=c(2100, 2010), col=c("red", "blue"), lty=1)
```

The resulting graphs are shown in Figures 12–14. Note that the regions are aggregated only from countries that are available in the underlying `pop.pred` object. These do not yet

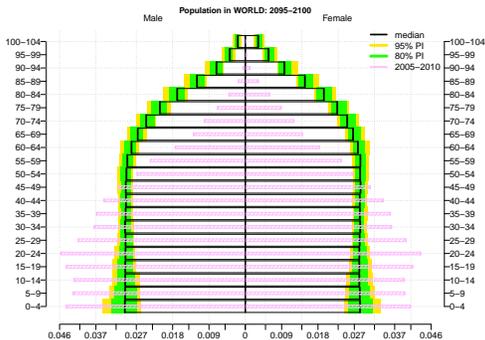


Figure 13: Population pyramid for the world in 2010 and 2100.

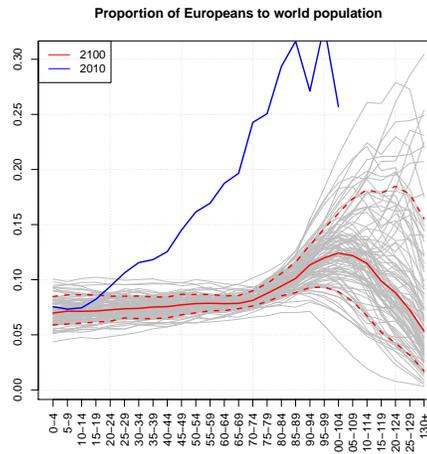


Figure 14: Number of Europeans as a proportion of world population for each age group in 2100 and the same indicator in 2010, expressed as “ $P_{908} / P_{900}$ ”.

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` (Figure 14) combines indicators from two regions. It is also possible to combine non-aggregated indicators with aggregated ones:

```
pop.trajectories.plot(pop.pred, expression="PIND / P900", sum.over.ages=TRUE,
  main="Proportion of population of India to the world population")
```

In such a call, the original (non-aggregated) prediction object 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 “independence” 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 (Ševčíková 2013a), 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 are contained in the data package **wpp2012** (Ševčíková *et al.* 2013). The previous revisions of the UN's official *World Population Prospects* are available in the data packages **wpp2010** and **wpp2008**. Data in all three **wpp** packages can be visualized in a browser using the R package **wppExplorer** (Ševčíková 2013b).

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 2013), which does stochastic population forecasting using the functional data approach of Hyndman and Ullah (2007). Two packages, **YourCast** (Giroso *et al.* 2013) and **MortalitySmooth** (Camarda 2012), use statistical models to forecast mortality rates. **YourCast** is based on the methods of Giroso and King (2008). The **popReconstruct** package (Wheldon 2013) does probabilistic reconstruction of *past* population quantities rather than forecasting of the future; it is based on the methods of Wheldon *et al.* (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 *et al.* 2012), based on the work of Caswell (2001), **popdemo** (Stott *et al.* 2012), and **primer** (Stevens 2012). The **IPMpack** package (Metcalf *et al.* 2013) 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 *et al.* 2012), which analyzes the effect of demography on allele retention, and **Biodem** (Boattini and Calboli 2012), 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 *et al.* (2012).

**Acknowledgements:** 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 Leontine Alkema, Samuel Clark and Patrick Gerland for helpful comments and discussion. Both 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.
- Boattini A, Calboli FCF (2012). *Biodem: Biodemography Functions*. R package version 0.3, URL <http://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. URL <http://www.jstatsoft.org/v50/i01/>.
- Caswell H (2001). *Matrix Population Models: Construction, Analysis and Interpretation*. 2nd edition. Sinauer Associates, Sunderland, Mass.
- Engen S, Saether BE, Kvaines T, Jensen H (2012). “Estimating Fluctuating Selection in Age-Structured Populations.” *Journal of Evolutionary Biology*, **25**, 1487–1499.
- Fosdick BK, Raftery AE (2013). “Regional Probabilistic Fertility Forecasting by Modeling between-Country Correlations.” Presented at the annual meeting of Population Association of America. <http://paa2013.princeton.edu/abstracts/130127>.
- Gesmann M, de Castillo D (2011). “**googleVis**: Interface between R and the Google Visualisation API.” *The R Journal*, **3**(2), 40–44. URL [http://journal.r-project.org/archive/2011-2/RJournal\\_2011-2\\_Gesmann+de~Castillo.pdf](http://journal.r-project.org/archive/2011-2/RJournal_2011-2_Gesmann+de~Castillo.pdf).
- Giroi F, King G (2008). *Demographic Forecasting*. Princeton University Press.

- Giroso F, King G, Villalon E, Bischof J, Kashin K (2013). *YourCast: Forecasting Age-Sex-Country-Cause Mortality Rates*. R package version 1.6, URL <http://CRAN.R-project.org/package=YourCast>.
- Hyndman RJ (2013). *demography: Forecasting Mortality, Fertility, Migration and Population Data*. R package version 1.15, URL <http://CRAN.R-project.org/package=demography>.
- Hyndman RJ, Ullah S (2007). “Robust Forecasting of Mortality and Fertility Rates: A Functional Data Approach.” *Computational Statistics and Data Analysis*, **51**, 4942–4956.
- International Organization for Standardization (2013). “ISO 3166.” [http://en.wikipedia.org/wiki/ISO\\_3166-1\\_numeric](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-2), [http://en.wikipedia.org/wiki/ISO\\_3166-1\\_alpha-3](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 1.1, URL <http://CRAN.R-project.org/package=lmf>.
- Lalic N, Raftery AE (2012). “Joint Probabilistic Projection of Female and Male Life Expectancy.” Presented at the annual meeting of Population Association of America. <http://paa2012.princeton.edu/abstracts/120140>.
- Lee RD, Carter L (1992). “Modeling and Forecasting the Time Series of US Mortality.” *Journal of the American Statistical Association*, **87**, 659–671.
- 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.
- Leslie PH (1945). “On the Use of Matrices in Certain Population Dynamics.” *Biometrika*, **33**, 183–212.
- Metcalf CJE, McMahon SM, Salguero-Gomez R, Jongelans E, Merow C (2013). *pkgIPMpack: Builds and Analyses Integral Projection Models (IPMs)*. R package version 2.0, URL <http://CRAN.R-project.org/package=IPMpack>.
- National Research Council (2000). *Beyond Six Billion: Forecasting the World’s Population*. National Academy Press, Washington, D.C.
- Preston SH, Heuveline P, Guillot M (2001). *Demography: Measuring and Modeling Population Processes*. Blackwell, Malden, Mass.

- Raftery AE, Alkema L, Gerland P (2013a). “Bayesian Population Projections for the United Nations.” *Statistical Science*, *in press*.
- Raftery AE, Chunn JL, Gerland P, Ševčíková H (2013b). “Bayesian Probabilistic Projections of Life Expectancy for all Countries.” *Demography*, **50**, 777–801.
- 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.
- South A, with contributions from Joe Scutt-Phillips, Rowlingson B, Foster RBP (2012). *rworldmap: Mapping Global Data, Vector and Raster*. R package version 1.02, URL <http://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 <http://CRAN.R-project.org/package=primer>.
- Stott I, Hodgson D, Townley S (2012). *popdemo: Provides Tools for Demographic Modelling Using Projection Matrices*. R package version 0.1-3, URL <http://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 <http://CRAN.R-project.org/package=Giza>.
- Stubben C, Milligan B, Nantel P (2012). *popbio: Construction and Analysis of Matrix Population Models*. R package version 2.4, URL <http://CRAN.R-project.org/package=popbio>.
- United Nations (1956). *Manual III: Methods for Population Projections by Sex and Age*. New York, NY: Dept. of Economic and Social Affairs, Population Division. Vol. 25 - Population Studies.
- United Nations (1989). *The United Nations Population Projection Computer Program: A User's Manual*. New York, NY: Dept. of Economic and Social Affairs, Population Division.
- United Nations (2013). *World Population Prospects: The 2012 Revision*. United Nations, New York, NY.
- Ševčíková H (2013a). *bayesDem: Graphical User Interface for bayesTFR, bayesLife and bayesPop*. R package version 2.3-2, URL <http://CRAN.R-project.org/package=bayesDem>.

- Ševčíková H (2013b). **wppExplorer**: *Explorer of World Population Prospects*. R package version 1.1-1, URL <http://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**, 1–29. URL <http://www.jstatsoft.org/v43/i01>.
- Ševčíková H, Gerland P, Andreev K, Li N, Gu D, Spoorenberg T (2013). **wpp2012**: *World Population Prospects 2012*. R package version 2.0-0, URL <http://CRAN.R-project.org/package=wpp2012>.
- Ševčíková H, Raftery AE (2013a). **bayesLife**: *Bayesian Projection of Life Expectancy*. R package version 2.0-2. Original WinBugs code written by Jennifer Chunn, URL <http://CRAN.R-project.org/package=bayesLife>.
- Ševčíková H, Raftery AE (2013b). **bayesPop**: *Probabilistic Population Projection*. R package version 4.2-1, URL <http://CRAN.R-project.org/package=bayesPop>.
- 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.
- Wheldon MC (2013). **popReconstruct**: *Reconstruct Population Counts, Fertility, Mortality and Migration Rates of Human Populations of the Recent Past*. R package version 1.0-3, URL <http://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.
- Whelpton PK (1928). “Population of the United States, 1925–1975.” *American Journal of Sociology*, **31**, 253–270.
- Whelpton PK (1936). “An Empirical Method for Calculating Future Population.” *Journal of the American Statistical Association*, **31**, 457–473.