Package 'bayesPop'

February 20, 2026

Type Package

Title Probabilistic Population Projection

Version 12.0-1

Date 2026-02-17

Maintainer Hana Sevcikova <hanas@uw.edu>

Depends R (>= 3.5.0), bayesTFR (>= 7.1-0), bayesLife (>= 5.0-0),
MortCast (>= 2.6-1)

Suggests wpp2017, wpp2015, wpp2010, knitr, R.rsp

Imports parallel, abind, data.table, wpp2019, wpp2012, graphics,
grDevices, stats, utils, rworldmap, fields, googleVis,
reshape2, plyr

Description Generating population projections for all countries of the world using several probabilistic components,
such as total fertility rate, life expectancy at birth and net migration (Raftery et al., 2012 <doi:10.1073/pnas.1211452109>).
The package can be also used for subnational population projections.

License GPL-3 | file LICENSE

LazyData true

URL https://bayespop.csss.washington.edu,

https://bayespop.csss.washington.edu/bayesPop/vignettes/population-projections.html

VignetteBuilder R.rsp

NeedsCompilation yes

Author Hana Sevcikova [aut, cre],
Adrian Raftery [aut],
Thomas Buettner [aut]

Repository CRAN

Date/Publication 2026-02-19 23:20:02 UTC

Contents

---

bayesPop–package   *Probabilistic Population Projection*

Description

The package allows to generate population projections for all countries of the world using several probabilistic components, such as total fertility rate (TFR) and life expectancy. Generating subnational projections is also supported.

Details

The main function is called `pop.predict`. It uses trajectories of TFR from the **bayesTFR** package and life expectancy from the **bayesLife** package and for each trajectory it computes a population projection using the cohort component method. It results in probabilistic age and sex specific projections. Various plotting functions are available for results visualization (`pop.trajectories.plot`, `pop.pyramid`, `pop.trajectories.pyramid`, `pop.map`), as well as a summary function. Aggregations can be derived using `pop.aggregate`. An expression language is available to obtain the distribution of various population quantities.

Subnational projections can be generated using `pop.predict.subnat`. Function `pop.aggregate.subnat` aggregates such projections.

Author(s)

Hana Sevcikova, Adrian Raftery, Thomas Buettner

Maintainer: Hana Sevcikova <hanas@uw.edu>

References

H. Sevcikova, A. E. Raftery (2016). bayesPop: Probabilistic Population Projections. Journal of Statistical Software, 75(5), 1-29. doi:10.18637/jss.v075.i05

A. E. Raftery, N. Li, H. Sevcikova, P. Gerland, G. K. Heilig (2012). Bayesian probabilistic population projections for all countries. Proceedings of the National Academy of Sciences 109:13915-13921. doi:10.1073/pnas.1211452109

P. Gerland, A. E. Raftery, H. Sevcikova, N. Li, D. Gu, T. Spoorenberg, L. Alkema, B. K. Fosdick, J. L. Chunn, N. Lalic, G. Bay, T. Buettner, G. K. Heilig, J. Wilmoth (2014). World Population Stabilization Unlikely This Century. Science 346:234-237.

H. Sevcikova, N. Li, V. Kantorova, P. Gerland and A. E. Raftery (2016). Age-Specific Mortality and Fertility Rates for Probabilistic Population Projections. In: Dynamic Demographic Analysis, ed. Schoen R. (Springer), pp. 285-310. Earlier version in arXiv:1503.05215.

H. Sevcikova, J. Raymer J., A. E. Raftery (2024). Forecasting Net Migration By Age: The Flow-Difference Approach. arXiv:2411.09878.

See Also

bayesTFR, bayesLife

Examples

```
## Not run:
sim.dir <- tempfile()
# Generates population projection for one country
country <- "Netherlands"
pred <- pop.predict(countries=country, output.dir=sim.dir)
summary(pred, country)
pop.trajectories.plot(pred, country)
dev.off()
pop.trajectories.plot(pred, country, sum.over.ages=TRUE)
pop.pyramid(pred, country)
pop.pyramid(pred, country, year=2100, age=1:26)
unlink(sim.dir, recursive=TRUE)

## End(Not run)

# Here are commands needed to run probabilistic projections
# from scratch, i.e. including TFR and life expectancy.
# Note that running the first four commands
# (i.e. predicting TFR and life expectancy) can take
# LONG time (up to several days; see below for possible speed-up).
# For a toy simulation, set the number of iterations (iter)
# to a small number.
## Not run:
```

```
sim.dir.tfr <- "directory/for/TFR"
sim.dir.e0 <-  "directory/for/e0"
sim.dir.pop <- "directory/for/pop"

# Estimate TFR parameters (speed-up by including parallel=TRUE)
run.tfr.mcmc(iter="auto", output.dir=sim.dir.tfr, seed=1)

# Predict TFR (if iter above < 4000, reduce burnin and nr.traj accordingly)
tfr.predict(sim.dir=sim.dir.tfr, nr.traj=2000, burnin=2000)

# Estimate e0 parameters (females) (speed-up by including parallel=TRUE)
# Can be run independently of the two commands above
run.e0.mcmc(sex="F", iter="auto", output.dir=sim.dir.e0, seed=1)

# Predict female and male e0
# (if iter above < 22000, reduce burnin and nr.traj accordingly)
e0.predict(sim.dir=sim.dir.e0, nr.traj=2000, burnin=20000)

# Population prediction
pred <- pop.predict(output.dir=sim.dir.pop, verbose=TRUE,
    inputs = list(tfr.sim.dir=sim.dir.tfr,
                  e0F.sim.dir=sim.dir.e0, e0M.sim.dir="joint_"))
pop.trajectories.plot(pred, "Madagascar", nr.traj=50, sum.over.ages=TRUE)
pop.trajectories.table(pred, "Madagascar")

## End(Not run)
```

---

age.specific.migration

Generate Sex- and Age-specific Migration

Description

Creates sex- and age-specific net migration datasets out of the total net migration using different methods. The age.specific.migration is a legacy function that distributes UN 5-year totals into ages using a residual method. The migration.totals2age distribute given totals using Rogers-Castro and the Flow Difference Method (FDM).

Usage

```
age.specific.migration(wpp.year = 2019, years = seq(1955, 2100, by = 5),
    countries = NULL, smooth = TRUE, rescale = TRUE, ages.to.zero = 18:21,
    write.to.disk = FALSE, directory = getwd(), file.prefix = "migration",
    depratio = wpp.year == 2015, verbose = TRUE)

migration.totals2age(df, ages = NULL, annual = FALSE, time.periods = NULL,
    scale = 1, method = "rc", sex = "M",
    id.col = "country_code", mig.is.rate = FALSE,
    rc.data = NULL, pop = NULL, pop.glob = NULL, ...)
```

```
rcastro.schedule(annual = FALSE)
```

Arguments

wpp.year	Integer determining which **wpp** package should be used to get the necessary data from. That package is required to have a dataset on total net migration (called [migration](#)).
years	Array of years that the reconstruction should be made for. This should be a subset of years for which the total net migration is available.
countries	Numerical country codes to do the reconstruction for. By default it is performed on all countries included in the [migration](#) dataset where aggregations are excluded.
smooth	Logical controlling if smoothing of the reconstructed curves is required. Due to rounding issues the residual method often yields unrealistic zig-zags on migration curves by age. Smoothing usually improves their look.
rescale	Logical controlling if the resulting migration should be rescaled to match the total migration.
ages.to.zero	Indices of age groups where migration should be set to zero. Default is 85 and older.
write.to.disk	If TRUE results are written to disk.
directory	Directory where to write the results if write.to.disk is TRUE.
file.prefix	If write.to.disk is TRUE results are written into two text files with this prefix, a letter "M" and "F" determining the sex, and concluded by the ".txt" suffix. By default "migrationM.txt" and "migrationF.txt".
depratio	If it is TRUE it will use an internal dataset on migration dependency ratios to adjust the first three age groups. It can also be a name of a binary file containing such dataset. The default dataset is only available for 2015.
verbose	Logical controlling the amount of output messages.
df	data.frame, marix or data.table containing total migration counts or rates. Columns correspond to time, rows correspond to locations. Column "country_code" (or column identified by id.col) contains identifiers of the locations. Names of the time columns should be either single years if annual is TRUE, e.g. "2018", "2019" etc., or five year time periods if annual is FALSE, e.g. "2010-2015", "2015-2020" etc.
ages	Labels of age groups into which the total migration is to be disaggregated. If it is missing, default age groups are determined depending on the argument annual.
annual	Logical determining if the age groups are 5-year age groups (FALSE) or 1-year ages (TRUE) on which the choice of the default schedule is dependent, if schedule is missing. It also determines the expected syntax of the names of time columns in df.
time.periods	Character vector determining which columns should be considered in the df dataset. It should be a subset of column names in df. By default, all time columns in df are considered.

scale	The migration schedule is multiplied by this number. It can be used for example, if total migration needs to be distributed between sexes.
method	Method to use for the distribution of totals into age groups. The "rc" method uses either a basic Rogers-Castro disaggregation via the function rcastro.schedule, or a schedule given in the rc.data argument. The "fdmp" and "fdmnop" methods use the Flow Difference Method, where "fdmp" weights the flows by population.
sex	"M" or "F" determining the sex of this schedule. It only impacts the FDM methods.
id.col	Name of the unique identifier of the locations.
mig.is.rate	Logical indicating if the data in df should be interpreted as rates. If FALSE, df represent counts.
rc.data	data.table containing either a family of Rogers-Castro proportions if method = "rc", or various inputs for the FDM methods if method is either "fdmp" or "fdmnop".
	For the "rc" method, mandatory columns are "age" and "prop". Optionally, it can have a column "mig_sign" with values "Inmigration" and "Emigration" (distinguishing schedules to be applied for positive and negative migration, respectively) and a column "sex" with values "Female" and "Male". The format corresponds to the dataset DemoTools::mig_un_families, subset to a single family.
	For the FDM methods, it has columns contained in the rcFDM dataset, as well as columns "beta0" (intercept), "beta1" (slope), "min" (minimum rate), "in_sex_factor" (inflow female proportion), and "out_sex_factor" (outflow female proportion), used in the FDM methods. These columns correspond to columns "MigFDMb0", "MigFDMb1", "MigFDMmin", "MigFDMsrin" and "MigFDMsrout", respectively, in the vwBaseYear dataset.
pop	data.table with population counts needed for the FDM methods. It should have a location identifier column of the same name as id.col, further columns "year", "age", and "pop".
pop.glob	data.table with global population needed for the weighted FDM method ("fdmp"). It should have columns "year", "age", and "pop".
...	Further arguments passed to the underlying functions.

Details

Function age.specific.migration: Unlike in wpp2012, for the four releases of the WPP between 2015 and 2022, the **wpp2015**, **wpp2017**, **wpp2019**, and **wpp2022**, the UN Population Division did not publish the sex- and age-specific net migration counts, only the totals. However, since the sex- and age-schedules are needed for population projections, the age.specific.migration function attempts to reconstruct those missing datasets. It uses the published population projections by age and sex, fertility and mortality projections from the **wpp** package. It computes the population projection without migration and sets the residual to the published population projection as the net migration. By default such numbers are then scaled so that the sum over sexes and ages corresponds to the total migration count.

If smooth is TRUE a smoothing procedure is performed over ages where necessary. Also, for simplicity, we set migration of old ages to zero (default is 85+). Both is done before the scaling. If it is desired to obtain raw residuals without any additional processing, set smooth=FALSE, rescale=FALSE, ages.to.zero=c().

This function works only for 5-year data.

Function migration.totals2age: This function should be used when working with annual data or data from **wpp2022** and **wpp2024**. It allows users to disaggregate total migration counts or rates (for multiple time periods and multiple locations) into age-specific ones by either a schedule similar to the one used by the UN in WPP2024 (method = "fdmnop"), a Rogers-Castro (method = "rc"), or by FDM weighted by population (method = "fdmp") as described in Sevcikova et al (2024). The FDM method needs additional info passed via the arguments rc.data, pop and pop.glob. The default Rogers-Castro schedule can be accessed via the function rcastro.schedule where the annual argument specifies if it is for 1-year or 5-year age groups. Alternatively, an external schedule can be given via the rc.data argument, where one can distinguish between schedules for each sex, as well as for positive and negative net migration. It has the same structure as the dataset DemoTools::mig_un_families, but it should be a subset for a single family and converted to data.table.

Value

Function age.specific.migration returns a list of two data frames (male and female), each having the same structure as [migrationM](#).

Function migration.totals2age returns a data.table with the disaggregated counts.

Function rcastro.schedule returns a vector of proportions for each age group.

Warning

Due to rounding issues and slight differences in the methodology, the functions do not reproduce the unpublished UN datasets exactly. It is only an approximation! Especially, the first age groups might be more off than other ages.

Note

These functions are called automatically from [pop.predict](#) if needed, depending on the inputs. Thus, only users that need sex- and age-specific migration for other purposes, or modify the defaults, will need to call these functions explicitly.

Further note that the **wpp2024** package does contain the age-specific net migration for projected years (datasets migprojAge1dt, migprojAge5dt). Thus, if running [pop.predict](#) with wpp.year = 2024 and the default migration totals, no disaggregation is necessary for the projected time periods. The disaggregation is only triggerered for the past time periods, or in a case when user-specific net migration totals are used.

Author(s)

Hana Sevcikova

References

H. Sevcikova, J. Raymer J., A. E. Raftery (2024). Forecasting Net Migration By Age: The Flow-Difference Approach. arXiv:2411.09878.

See Also

pop.predict, migration migrationM, rcFDM, vwBaseYear

Examples

```
## Not run:
asmig <- age.specific.migration()
head(asmig$male)
head(asmig$female)
## End(Not run)

# simple disaggregation for one location
totmig <- c(30, -50, -100)
names(totmig) <- 2018:2020
asmig.simple <- migration.totals2age(totmig, annual = TRUE, method = "rc")
head(asmig.simple)

## Not run:
# disaggregate WPP 2019 migration for all countries, one sex
data(migration, package = "wpp2019")
# assuming equal sex migration ratio
asmig.all <- migration.totals2age(migration, scale = 0.5, method = "rc")
# plot result for the US in 2095-2100
mig1sex.us <- subset(asmig.all, country_code == 840)[["2095-2100"]]
plot(ts(mig1sex.us))
# check that the sum is half of the original total
sum(mig1sex.us) == subset(migration, country_code == 840)[["2095-2100"]]/2
## End(Not run)
```

---

get.countries.table   *Accessing Country Information*

Description

The function returns a data frame containing codes and names of all countries used in the prediction.

Usage

```
## S3 method for class 'bayesPop.prediction'
get.countries.table(object, ...)
```

Arguments

object	Object of class bayesPop.prediction.
...	Not used.

Value

Data frame with columns `code` and `name`.

Author(s)

Hana Sevcikova

---

get.pop.prediction *Accessing Prediction Object*

Description

Function `get.pop.prediction` retrieves results of a prediction from disk and creates an object of class [bayesPop.prediction](#). Function `has.pop.prediction` checks an existence of such results.

Usage

```
get.pop.prediction(sim.dir, aggregation = NULL, write.to.cache = TRUE)

has.pop.prediction(sim.dir)

pop.cleanup.cache(pop.pred)
```

Arguments

sim.dir	Directory where the prediction is stored. It should correspond to the value of the `output.dir` argument used in the [pop.predict](#) function.
aggregation	If given, the prediction object is considered to be an aggregation and both arguments are passed to [get.pop.aggregation](#).
write.to.cache	Logical controlling if other functions are allowed to write the cache of this prediction object (see Details).
pop.pred	Object of class [bayesPop.prediction](#).

Details

The [pop.predict](#) function stores resulting trajectories into a directory called `output.dir/prediction`. Here the argument `sim.dir` should correspond to `output.dir` (i.e. without the "prediction" part).

In addition to retrieving prediction results, the `get.pop.prediction` function also looks for a file called 'cache.rda' and loads it into an environment called cache. If it does not exist, it creates an empty cache environment. See [pop.map](#) - Section Performance and Caching. The environment can be cleaned up using the `pop.cleanup.cache` function which also deletes the 'cache.rda' file on disk. If `write.to.cache` is `FALSE`, other functions are not allowed to manipulate the 'cache.rda' file.

Value

Function `has.pop.prediction` returns a logical indicating if a prediction exists.

Function `get.pop.prediction` returns an object of class `bayesPop.prediction`.

Author(s)

Hana Sevcikova

See Also

`bayesPop.prediction`, `get.pop.aggregation`

Examples

```
sim.dir <- file.path(find.package("bayesPop"), "ex-data", "Pop")
pred <- get.pop.prediction(sim.dir)
summary(pred)
```

LifeTableMx   *Life Table Functions*

Description

Functions for obtaining life table quantities.

Usage

```
LifeTableMx(mx, sex = c("Male", "Female", "Total"), include01 = TRUE,
abridged = TRUE, radix = 1, open.age = 130)

LifeTableMxCol(mx, colname = c("Lx", "lx", "qx", "mx", "dx", "Tx", "sx", "ex", "ax"), ...)
```

Arguments

mx	Vector of age-specific mortality rates nmx. If `abridged` is TRUE, the elements correspond to 1m0, 4m1, 5m5, 5m10, ..., otherwise they corresppond single year age groups. In the abridged case teh vector can have no more than 28 elements which corresponds to age up to 130. In the `LifeTableMxCol` function, this argument can be a two-dimensional matrix with first dimension being the age.
sex	For which sex is the life table.
include01	Logical. If it is FALSE the first two age groups (0-1 and 1-4) are collapsed to one age group (0-4). Only considered if `abridged` is TRUE.
abridged	Logical. If TRUE (default) the life table and the `mx` argument is assumed for 5-year age groups. Otherwise 1-year age groups are assumed.
radix	Base of the life table.

open.age	Open age group. If smaller than the last age group of mxm, the life table is truncated.
colname	Name of the column of the life table that should be returned.
...	Arguments passed to underlying functions, e.g. abridged. In addition for abridged life table only, argument age05 is a logical vector of size three, specifying if the age groups 0-1, 1-4 and 0-5 should be included. Default value of c(FALSE, FALSE, TRUE) includes the 0-5 age group only.

Details

Function LifeTableMx returns a life table for one set of mortality rates. Function LifeTableMxCol returns one column of the life table for (possibly) multiple sets of mortality rates. The underlying workhorse here is the life.table function from the MortCast package. These functions only collapse the first age groups if needed for an abridged life table (LifeTableMx) or/and combine results for multiple time periods into one object (LifeTableMxCol).

Value

Function LifeTableMx returns a data frame with the following elements:

age	Age groups
mx	mx, the input vector of mortality rates.
qx	nqx, probability of dying between ages x ad x+n.
lx	lx, number left alive at age x.
dx	ndx, cohort deaths between ages x ad x+n.
Lx	nLx, person-years lived between ages x and x+n.
sx	sx, survival rate at age x.
Tx	Tx, person-years lived above age x.
ex	e0x, expectation of life at age x.
ax	nax, average person-years lived in the interval by those dying in the interval.

Function LifeTableMxCol returns one given column of the life table, possibly as a matrix (if mx is a matrix).

Author(s)

Hana Sevcikova, Thomas Buettner, Nan Li, Patrick Gerland

References

Preston, P., Heuveline, P., Guillot, M. (2001): Demography. Blackwell Publishing Ltd.

See Also

life.table, pop.expressions for examples on retrieving some life table quantities.

Examples

```
## Not run:
sim.dir <- tempfile()
pred <- pop.predict(countries="Ecuador", output.dir=sim.dir, wpp.year=2015,
    present.year=2015, keep.vital.events=TRUE, fixed.mx=TRUE, fixed.pasfr=TRUE)
# get male mortality rates from 2020 for age groups 0-1, 1-4, 5-9, ...
mxm <- pop.byage.table(pred, expression="MEC_M{age.index01(27)}", year=2020)[,1]
print(LifeTableMx(mxm), digits=3)
# female LT with first two age categories collapsed
mxf <- pop.byage.table(pred, expression="MEC_F{age.index01(27)}", year=2020)[,1]
print(LifeTableMx(mxf, sex="Female", include01=FALSE), digits=3)
unlink(sim.dir, recursive=TRUE)
## End(Not run)
```

---

mac.expression   *Expression Generator*

Description

Help functions to easily generate commonly used expressions.

Usage

```
mac.expression(country)
mac.expression1(country)
mac.expression5(country)
```

Arguments

country   Country code as defined for expressions.

Details

mac.expression and mac.expression1 generate expressions for the mean age of childbearing of the given country, for 5-year age groups and 1-year age groups, respectively. mac.expression5 is a synonym for mac.expression. Note that pop.predict has to be run with keep.vital.events=TRUE for this to work.

Value

mac.expression returns a character string corresponding to the formula $(17.5 * R_c(15 - 19) + 22.5 * R_c(20 - 24) + ... + 47.5 * R_c(45 - 49))/100$ where $R_c(x)$ denotes the country-specific percent age-specific fertility for the age group x.

mac.expression1 returns a character string corresponding to the formula $(10.5 * R_c(10 - 11) + 11.5 * R_c(11 - 12) + ... + 54.5 * R_c(54 - 55))/100$

See Also

pop.expressions

Examples

```
## Not run:
sim.dir <- tempfile()
# Run pop.predict with storing vital events
pred <- pop.predict(countries=c("Germany", "France"), nr.traj=3,
        keep.vital.events=TRUE, output.dir=sim.dir)
# plot the mean age of childbearing
pop.trajectories.plot(pred, expression=mac.expression("FR"), cex.main = 0.7)
unlink(sim.dir, recursive=TRUE)
## End(Not run)
```

MLTbx   *Dataset on Lee-Carter bx for Modeled Countries*

Description

Dataset with values of the Lee-Carter bx parameter for countries where mortality was obtained using model life tables.

Usage

```
data(MLTbx)
```

Format

A data frame with nine rows and 28 columns. Each row corresponds to one mortality age pattern as defined in the [vwBaseYear](#) dataset. Each column corresponds to an age group, starting with 0-1, 1-4, 5-9, 10-14, ... up to 125-129, 130+.

Details

These values are used for countries for which the column AgeMortalityType in [vwBaseYear](#) is equal to "Model life tables". In such a case a row is selected that corresponds to the corresponding value of the column AgeMortalityPattern (also in [vwBaseYear](#)). These values are then used instead of estimating the Lee-Carter b_x from the country's historical data.

Source

Data provided by the United Nations Population Division.

See Also

[vwBaseYear](#)

Examples

```
data(MLTbx)
str(MLTbx)
```

---

peak.probability   *Probability of Peaks in Population Indicators*

Description

For a given indicator and a country, the function computes the probability of a peak happening before a given year, as well as a range of years between which a peak happens with given probability.

Usage

```
peak.probability(pop.pred, country = NULL, expression = NULL, year = NULL,
    pi = 95, verbose = TRUE, ...)
```

Arguments

pop.pred	Object of class bayesPop.prediction.
country	Name or numerical code or ISO-2 or ISO-3 character code of a country. If given, population is used as an indicator and the expression argument is ignored.
expression	Expression defining an indicator. For syntax see pop.expressions. It must be defined by time (i.e. either without or with square brackets, and no curly braces). Only used if country is not speicified.
year	Used for computing the probability of a peak happenning before year.
pi	Probability between 0 and 100. Used for selecting a range of years between which a peak happens with probability given by this argument.
verbose	Logical. If TRUE, results are printed.
...	Additional arguments passed to the underlying functions. If country is given, these are arguments passed to pop.trajectories, e.g. sex, age or adjust. If the indicator is given via expression, it can be e.g. adj.to.file.

Details

Given an indicator, the function computes two quantities:

- probability that the indicator reaches its peak before given year;
- range of years between which a peak happens with the given probability pi.

The indicator can be either population (if country is given), or it can be any expression defined as a function of time (see pop.expressions).

Value

List with elements:

prob.peak.less.given.year
                Probability that the indicator reaches its peak before year.

given.year   The value of year.

peak.quantiles The lower bound, the upper bound and the median of years defining a time interval in which a peak happens with the given probability pi

.

all.prob.peak.by.time

Data frame containing the probability of peak happening in each projected year, as well as the corresponding cummulative probability. Years in which no peak is projected are not included.

Author(s)

Hana Sevcikova

See Also

pop.expressions

Examples

```
sim.dir <- file.path(find.package("bayesPop"), "ex-data", "Pop")
pred <- get.pop.prediction(sim.dir, write.to.cache=FALSE)

# probability that population of Netherlands peaks before 2040
# and between which years it will peak with probablity 80%
peak.probability(pred, "NL", year = 2040, pi = 80)

# check visually with
# pop.trajectories.plot(pred, "NL")

# the same for female of age 45-49
peak.probability(pred, "NL", year = 2040, pi = 80, sex = "female", age = 10)

# probability of a peak for the potential support ratio in Ecuador
peak.probability(pred, expression = "PEC[5:13]/PEC[14:27]")

# check visually that it already peaked
# pop.trajectories.plot(pred, expression = "PEC[5:13]/PEC[14:27]")
```

---

pop.aggregate   *Aggregation of Population Projections*

Description

Aggregation of existing countries' population projections into projections of given regions, and accessing such aggregations.

Usage

```
pop.aggregate(pop.pred, regions,
    input.type = c("country", "region"), name = input.type,
   inputs = list(e0F.sim.dir = NULL, e0M.sim.dir = "joint_", tfr.sim.dir = NULL),
    my.location.file = NULL, verbose = FALSE, ...)

get.pop.aggregation(sim.dir = NULL, pop.pred = NULL, name = NULL,
    write.to.cache = TRUE)

pop.aggregate.subnat(pop.pred, regions, locations, ..., verbose = FALSE)
```

Arguments

pop.pred
: Object of class `bayesPop.prediction` containing country-specific population projections.

regions
: Vector of numerical codes of regions. It should correspond to values in the column "country_code" in the `UNlocations` dataset or in `my.location.file` (see below). For `pop.aggregate.subnat` it is a numerical code of a country over which subregions are aggregated.

input.type
: There are two methods for aggregating projections depending on the type of inputs, "country"- and "region"-based, see Details.

name
: Name of the aggregation. It becomes a part of a directory name where aggregation results are stored.

inputs
: This argument is only used when the "region"-based method is selected. It is a list of inputs of probabilistic components of the projection:

  e0F.sim.dir Simulation directory with projections of female life expectancy (generated using **bayesLife**). It must contain projections for the given regions (see functions `run.e0.mcmc.extra`, `e0.predict.extra`). If it is not given, the same e0 directory is taken which was used for generating the `pop.pred` object, in which case the e0 projections are re-loaded from disk.

  e0M.sim.dir Simulation directory with projections of male life expectancy. By default (value NULL or "joint_") the function assumes a joint female-male projections of life expectancy and thus tries to load the male projections from the female projection object created using the `e0F.sim.dir` argument.

  tfr.sim.dir Simulation directory with projections of total fertility rate (generated using **bayesTFR**). It must contain projections for the given regions (see functions `run.tfr.mcmc.extra`, `tfr.predict.extra`). If it is not given, the same TFR directory is taken which was used for generating the `pop.pred` object, in which case the TFR projections are re-loaded from disk.

my.location.file
: User-defined location file that can contain other aggreggation groups than the default UN location file. It should have the same structure as the `UNlocations` dataset, see below.

verbose
: Logical switching log messages on and off.

sim.dir	Simulation directory where aggregation is stored. It is the same directory used for creating the pop.pred object. Alternatively, pop.pred can be used. Either sim.dir or pop.pred must be given.
write.to.cache	Logical controlling if functions operating on this object are allowed to write into its cache (see Details of get.pop.prediction).
locations	Name of a tab-delimited file that contains definitions of the sub-regions. It should be the same file as used for the locations argument in pop.predict.subnat.
...	Additional arguments. For a country-type aggregation, it can be logical use.kannisto which determines if the Kannisto method should be used for old ages when aggregating mortality rates. A logical argument keep.vital.events determines if vital events should be computed for aggregations. Argument adjust determines if country-level population numbers should be adjusted to the WPP values.

Details

Function pop.aggregate triggers an aggregations over countries while function pop.aggregate.subnat is used for aggregation over sub-regions to a country. The following details refer to the use of pop.aggregate. For sub-national aggregation see Example in pop.predict.subnat.

The dataset UNlocations or my.location.file is used to determine countries to be aggregated, in particular the field "location_type" of the entries with "country_code" given in the regions argument. One can aggregate over the following location types: Type 0 means aggregating all countries of the world (or in the file), type 2 is aggregating over continents, type 3 is aggregating over regions within continents, and any other integer (except 4) correponds to user-defined aggregations. Note that type 4 is reserved as a location type of countries and thus, all aggregations are performed over entries of this type. For type 2, countries are matched using the "area_code" column; for type 3 the matching is done using the "reg_code" column of the UNlocations dataset. E.g., if regions=908 (Europe) which has location type 2 in the default UNlocations dataset, all countries are aggregated for which values of 908 are found in the "area_code" column. If the location type is other than 0, 2, 3 and 4, there must be a column in the file called "agcode_x" with x being the location type. This column is then used to match the countries to be aggregated.

Consider the following example. Say we want to pair four countries (Germany [DE], France [FR], Netherlands [NL], Italy [IT]) in two different ways, so we have two overlapping groupings, each of which has two groups (A,B):

1. group A = (DE, FR), group B = (NL, IT)
2. group A = (DE, NL), group B = (FR, IT)

Then, my.location.file should have the following entries:

country_code	name	location_type	agcode_98	agcode_99
1001	grouping1_groupA	98	-1	-1
1002	grouping1_groupB	98	-1	-1
1003	grouping2_groupA	99	-1	-1
1004	grouping2_groupB	99	-1	-1
276	Germany	4	1001	1003
250	France	4	1001	1004

258	Netherlands	4	1002	1003
380	Italy	4	1002	1004
1005	all	0	-1	-1

The "country_code" of the groups is user-specific, but it must be unique within the file. Values of "country_code" for countries must match those in the prediction object. To run the aggregation for the four groups above we set regions=1001:1004. Having "location_type" being 98 and 99, it is expected the file to have columns "agcode_98" and "agcode_99" containing assignements to each of the two groupings. Values in this columns corresponding to groups are not used and thus can have any value. For aggregating over all four countries, set regions=1005 which has "location_type" equal 0 and thus, it is aggregated over all entries with "location_type" equals 4.

There are two methods available for generating aggregations of population projection:

Country-based Method Aggregations are created by summing trajectories over countries of the given region.

Region-based Method The aggregation is generated using the same algorithm as population projections for single countries (function pop.predict), but it operates on aggregated input components. These are created as follows. Here c denotes countries over which we aggregate a region R, $s \in \{m, f\}$, a, and t denote sex, age category and time, respectively. $t = P$ denotes the present year of the prediction. $N_{s,a,t}^c$ and $M_{s,a,t}^c$, respectively, denotes the historical population count and the Bayesian predictive median of population, respectively, of sex s, in age category a at time t for country c (refer to the links in parentheses for description of the data):

Initial sex and age-specific population (popM, popF): $N_{s,a,t=P}^R = \sum_c N_{s,a,t=P}^c$

Sex and age-specific death rates (mxM, mxF): $mx_{s,a,t}^R = \frac{\sum_c (mx_{s,a,t}^c \cdot N_{s,a,t})}{\sum_c N_{s,a,t}}$

Sex ratio at birth (srb): $SRB_t^R = \frac{\sum_c M_{s=m,a=1,t}^c}{\sum_c M_{s=f,a=1,t}^c}$

Percentage age-specific fertility rate (pasfr): $PASFR_{a,t}^R = \frac{\sum_c (PASFR_{a,t}^c \cdot M_{s=f,a,t})}{\sum_c M_{s=f,a,t}}$

Migration code and start year (mig.type): Aggregated migration code is the code of maximum counts over aggregated countries weighted by $N_{t=P}^c$. Migration start year is the maximum of start years over aggregated countries.

Sex and age-specific migration (migM, migF): $mig_{s,a,t}^R = \sum_c mig_{s,a,t}^c$

Probabilistic projection of life expectancy: We assume an aggregation of life expectancy for the given regions was generated prior to this call, using the run.e0.mcmc.extra and e0.predict.extra functions of the **bayesLife** package.

Probabilistic projection of total fertility rate: We assume an aggregation of total fertility for the given regions was generated prior to this call, using the run.tfr.mcmc.extra and tfr.predict.extra functions of the **bayesTFR** package.

Results of the aggregations are stored in the same top directory as the pop.pred object, in a sudirectory called 'aggregations_*name*'. They can be accessed using the function get.pop.aggregation. Note that multiple runs of this function with the same name will overwrite previous aggregations results of the same name.

Value

Object of class bayesPop.prediction containing the aggregated results. In addition it contains elements aggregation.method giving the input.type used, and aggregated.countries which is a list of countries aggregated for each region.

Author(s)

Hana Sevcikova, Adrian Raftery

References

H. Sevcikova, A. E. Raftery (2016). bayesPop: Probabilistic Population Projections. Journal of Statistical Software, 75(5), 1-29. doi:10.18637/jss.v075.i05

See Also

[pop.predict](), [tfr.predict.extra](), [e0.predict.extra]()

Examples

```
## Not run:
sim.dir <- tempfile()
pred <- pop.predict(countries=c(528,218,450), output.dir=sim.dir)
aggr <- pop.aggregate(pred, 900) # aggregating World (i.e. all countries available in pred)
pop.trajectories.plot(aggr, 900, sum.over.ages=TRUE)
# countries over which we aggregated:
subset(UNlocations, country_code %in% aggr$aggregated.countries[["900"]])
unlink(sim.dir, recursive=TRUE)
## End(Not run)
```

---

pop.cohorts.plot   *Extracting and Plotting Cohort Data*

Description

Extracts and plots population counts or results of expressions by cohorts.

Usage

```
cohorts(pop.pred, country = NULL, expression = NULL, pi = c(80, 95))

pop.cohorts.plot(pop.pred, country = NULL, expression = NULL, cohorts = NULL,
    cohort.data = NULL, pi = c(80, 95), dev.ncol = 5, show.legend = TRUE,
  legend.pos = "bottomleft", ann = par("ann"), add = FALSE, xlab = "", ylab = "",
    main = NULL, xlim = NULL, ylim = NULL, col = "red", ...)
```

Arguments

pop.pred	Object of class [bayesPop.prediction]().
country	Name or numerical code of a country. If it is not given, expression must be specified.
expression	Expression defining the population measure to be plotted. For syntax see [pop.expressions](). It must be country-specific, i.e. "XXX" is not allowed, and it must contain curly braces, i.e. be age specific.

pi	Probability interval. It can be a single number or an array.
cohorts	Years of the cohorts to be plotted. By default, 10 future cohorts (starting from the last observed one) are used. It can be a single number or an array.
cohort.data	List with the cohort data obtained via the cohorts function. If it is not given, function cohorts is called internally, but by passing this argument the processing is faster.
dev.ncol	Number of column for the graphics device.
show.legend	Logical controlling whether the legend should be drawn.
legend.pos	Position of the legend passed to the [legend](#) function.
ann, xlab, ylab, main, xlim, ylim, col, ...	
	Graphical parameters passed to the plot function.
add	Logical specifying if the plot should be added to an existing graphics.

Details

pop.cohorts.plot plots all cohorts passed in the cohorts argument on the same scale of the y-axis.

Value

Function cohorts returns a list where each element corresponds to one cohort. Each cohort element is a matrix with columns corresponding to years and rows corresponding to the median (first row) and quantiles of the given probability intervals.

Author(s)

Hana Sevcikova

See Also

[pop.trajectories.plot](#), [pop.byage.plot](#), [pop.expressions](#)

Examples

```
sim.dir <- file.path(find.package("bayesPop"), "ex-data", "Pop")
pred <- get.pop.prediction(sim.dir)
# Population cohorts
pop.cohorts.plot(pred, "Netherlands")
# plot specific cohorts using expression (must contain {})
pop.cohorts.plot(pred, expression="P528{}", cohorts=c(1960, 1980, 2000, 2020))
# the same as
cohort.data <- cohorts(pred, expression="P528{}")
pop.cohorts.plot(pred, cohort.data=cohort.data, cohorts=c(1960, 1980, 2000, 2020))
```

---

pop.expressions   *Expressions as used in Population Output Functions*

Description

Documentation of expressions supported by functions `pop.trajectories.plot`, `pop.trajectories.plotAll`, `pop.trajectories.table`, `pop.byage.plot`, `pop.byage.table`, `cohorts`, `pop.cohorts.plot`, `pop.map`, `pop.map.gvis`, `write.pop.projection.summary`, `get.pop.ex`, `get.pop.exba`.

Details

The functions above accept an argument `expression` which should define a population measure, i.e. a quantity that can be computed from population projections, observed population data or vital events. Such an expression is a collection of *basic components* connected via usual arithmetic operators, such as `+`, `-`, `*`, `/`, `^`, `%%`, `%/%`, and combined using parentheses. In addition, standard R functions or predefined functions (see below) can be used within expressions.

A **basic component** is a character string constituted of four parts, two of which are optional. They must be in the following order:

1. Measure identification. One of the folowing upper-case characters:

   - 'P' - population,
   - 'D' - deaths,
   - 'B' - births,
   - 'S' - survival ratio,
   - 'F' - fertility rate,
   - 'R' - percent age-specific fertility,
   - 'M' - mortality rate,
   - 'Q' - probability of dying,
   - 'E' - life expectancy,
   - 'G' - net migration,
   - 'A' - a_x column of the life table.

   All but the 'P' and 'G' indicators are available only if the `pop.predict` function was run with `keep.vital.events=TRUE`.

2. Country part. One of the following:

   - Numerical country code (as used in `UNlocations`, see https://en.wikipedia.org/wiki/ISO_3166-1_numeric),
   - two- or three-character ISO 3166 code, see https://en.wikipedia.org/wiki/ISO_3166-1_alpha-2, https://en.wikipedia.org/wiki/ISO_3166-1_alpha-3,
   - characters "XXX" which serves as a wildcard for a country code.

3. Sex part (optional): The country part can be followed by either "_F" (for female) or "_M" (for male).

4. Age part (optional): If used, the basic component is concluded by an age index given as an array. Such array is embraced by either brackets ("[" and "]") or curly braces ("{" and "}"). The former invokes a summation of counts over given ages, the latter is used when no summation is desired. Note that if this part is missing, counts are automatically summed over all ages. To use all ages without summing, empty curly braces can be used.

   - For 5x5 predictions, the age index 1 corresponds to age 0-4, index 2 corresponds to age 5-9 etc. Indicators 'S', 'M', 'Q' and 'E' allow an index -1 which corresponds to age 0-1 and an index 0 which corresponds to age 1-4. Use the pre-defined functions age.index01(...) and age.index05(...) (see below) to define the right indices.
   - For 1x1 predictions, the age index starts with 0 for all indicators and matches exactly the age. I.e., indices 0,1,2,... correspond to ages 0,1,2,....

Not all combinations of the four parts above make sense. For example, 'F' and 'R' can be only combined with female sex, 'B', 'F' and 'R' can be only combined with a subset of the age groups, namely child-bearing ages (indices 4 to 10 in 5x5, or 11 to 55 in 1x1). Or, there is no point in summing the life table based indicators (M, Q, E, S, A) over multiple age groups, i.e. using brackets, or over sexes. Thus, if the sex part is omitted for the life table indicators, the life table is correctly aggregated over sexes, instead of a simple summation.

Examples of basic components are "P276", "D50_F[4:10]", "PXXX{14:27}", "SCZE_M{}", "QIE_M[-1]".

When the expression is evaluated on a prediction object, each basic component is substituted by an array of four dimensions (using the get.pop function):

1. Country dimension: Equals to one if a specific country code is given, or it equals the number of countries in the prediction object if a wildcard is used.

2. Age dimension: Equals to one if the third component above is missing or the age 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 time context of the expression, this dimension corresponds to either the number of projection periods or the number of observation periods.

4. Trajectory dimension: Corresponds to the number of trajectories in the prediction object, or one if the component is evaluated on observed data.

Depending on the context from which the expression is called, the trajectory dimension of the result of the expression can be reduced by computing given quantiles, and if only one country is evaluated, the first dimension is removed. In addition, with an exception of functions pop.byage.plot, pop.byage.table, cohorts, and pop.cohorts.plot, the expression should be constructed in a way that the age dimension is eliminated. This can be done for example by using brackets to define age, by using the apply function or one of the pre-defined functions described below. When using within pop.byage.plot, pop.byage.table, cohorts, or pop.cohorts.plot, the expression MUST include curly braces.

While get.pop can be used to obtain results of a basic component, functions get.pop.ex and get.pop.exba evaluate whole expressions.

Pre-defined functions

The following functions can be used within an expression:

- gmedian(f, cat)
  It gives a median for grouped data with frequencies f and categories cat. This function is to be used in combination with apply or pop.apply (see below) along the age dimension. For example,
  "apply(P380{}, c(1,3,4), gmedian, cats=seq(0, by=5, length=28))"
  is an expression for median age in Italy. (See pop.apply below for a simplified version.)

- gmean(f, cat)
  Works like gmedian but gives the grouped mean.

- age.func(data, fun="*")
  This function applies fun to data and the corresponding age (the middle point of each age category). The default case would multiply data by the corresponding age. As gmedian, it is to be used in combination with apply or pop.apply.

- drop.age(data)
  Drops the age dimension of the data. For example, if two basic components are combined where one is used within the apply function, the other will need to change its dimension in order to have conformable arrays. For example,
  "apply(age.func(P752{}), c(1,3,4), sum) / drop.age(P752)"
  is an expression for the average age in Sweden. (See pop.apply below for a simplified version.)

- pop.apply(data, fun, ..., split.along=c("None", "age", "traj", "country"))
  By default applies function fun to the age dimension of data and converts the result into the same format as returned by a basic component. This allows combining the apply function with other basic components without having to modify their dimensions. For example,
  "pop.apply(age.func(P752{}), fun=sum) / P752" gives the average age in Sweden, or
  "pop.apply(P380{}, gmedian, cats=seq(0, by=5, length=28))" gives the median age of Italy.
  If slice.along is not 'None', it can be used as an apply function where the data is sliced along one axis.

- pop.combine(data1, data2, fun, ..., split.along=c("age", "traj", "country"))
  Can be used if two basic components should be combined that result in different shapes. It tries to put data into the right format and calls pop.apply. For example,
  "pop.combine(PIND{}, PIND, '/')" give population by age per total population in India, or
  "pop.combine(BFR - DFR, GFR, '+', split.along='traj')" gives births minus deaths plus net migration in France. Here, pop.combine is necessary, because 'GFR' is a deterministic component and thus, has only one trajectory, whereas births and deaths are probabilistic.

- age.index01(end)
  Can be used with indicators 'S', 'M', 'Q' and 'E' only. It returns an array of age group indices that include ages 0-1 and 1-4 and exclude 0-4. The last age index is end.

- age.index05(end)
  Returns an array of age group indices starting with group 0-4, 5-9 until the age group corresponding to index end.

There is also a help function available that generates an expression for the mean age of childbearing, see mac.expression.

Note

The expression parser is simple and far from being perfect. We recommend to leave spaces around the basic components.

Author(s)

Hana Sevcikova, Adrian Raftery

References

H. Sevcikova, A. E. Raftery (2016). bayesPop: Probabilistic Population Projections. Journal of Statistical Software, 75(5), 1-29. doi:10.18637/jss.v075.i05

See Also

mac.expression, get.pop, pop.trajectories.plot, pop.map, write.pop.projection.summary.

Examples

```
sim.dir <- file.path(find.package("bayesPop"), "ex-data", "Pop")
pred <- get.pop.prediction(sim.dir, write.to.cache=FALSE)

# median age of women in child-bearing ages in Netherlands and all countries - trajectories
pop.trajectories.plot(pred, nr.traj=0,
    expression="pop.apply(P528_F{4:10}, gmedian, cats= seq(15, by=5, length=8))")
## Not run:
pop.trajectories.plotAll(pred, nr.traj=0,
    expression="pop.apply(PXXX_F{4:10}, gmedian, cats= seq(15, by=5, length=8))")

## End(Not run)
# mean age of women in child-bearing ages in Netherlands - table
pop.trajectories.table(pred,
    expression="pop.apply(age.func(P528_F{4:10}), fun=sum) / P528_F[4:10]")
# - gives the same results as with "pop.apply(P528_F{4:10}, gmean, cats=seq(15, by=5, length=8))"
# - for the mean age of childbearing, see ?mac.expression

# migration per capita by age
pop.byage.plot(pred, expression="GNL{} / PNL{}", year=2000)

## Not run:
# potential support ratio - map (with the two countries
#        contained in pred object)
pop.map(pred, expression="PXXX[5:13] / PXXX[14:27]")
## End(Not run)

# proportion of 0-4 years old to whole population - export to an ASCII file
dir <- tempfile()
write.pop.projection.summary(pred, expression="PXXX[1] / PXXX", output.dir=dir)
unlink(dir)

## Not run:
# These are vital events only available if keep.vital.events=TRUE in pop.predict, e.g.
# sim.dir.tmp <- tempfile()
# pred <- pop.predict(countries="Netherlands", nr.traj=3,
#               keep.vital.events=TRUE, output.dir=sim.dir.tmp)
# log female mortality rate by age for Netherlands in 2050, including 0-1 and 1-4 age groups
pop.byage.plot(pred, expression="log(MNL_F{age.index01(27)})", year=2050)
```

```
# trajectories of male 1q0 and table of 5q0 for Netherlands
pop.trajectories.plot(pred, expression="QNLD_M[-1]")
pop.trajectories.table(pred, expression="QNLD_M[1]")
# unlink(sim.dir.tmp)
## End(Not run)
```

---

pop.map   *World Map of Population Measures*

Description

Generates a world map of various population measures for a given quantile and a projection or observed period, using different techniques: pop.map use **rworldmap**, pop.ggmap uses **ggplot2**, and pop.map.gvis creates an interactive map via **GoogleVis**.

Usage

```
pop.map(pred, sex = c("both", "male", "female"), age = "all", expression = NULL, ...)

pop.ggmap(pred, sex=c('both', 'male', 'female'), age='all', expression=NULL, ...)

get.pop.map.parameters(pred, expression = NULL, sex = c("both", "male", "female"),
    age = "all", range = NULL, nr.cats = 50, same.scale = TRUE, quantile = 0.5, ...)

pop.map.gvis(pred, ...)
```

Arguments

pred	Object of class [bayesPop.prediction]().
sex	One of "both" (default), "male" or "female". By default the male and female counts are summed up. This argument is only used if expression is NULL.
age	Either a character string "all" (default) or an integer vector of age indices. Value 1 corresponds to age 0-4, value 2 corresponds to age 5-9 etc. Last age goup 130+ corresponds to index 27. This argument is only used if expression is NULL.
expression	Expression defining the population measure to be plotted. For syntax see [pop.expressions](). The country components of the expression should be given as "XXX".
range	Range of the population measure to be displayed. It is of the form c(*min*, *max*).
nr.cats	Number of color categories.
same.scale	Logical controlling if maps for all years of this prediction object should be on the same color scale.
quantile	Quantile for which the map should be generated. It must be equal to one of the values in dimnames(pred$quantiles[[2]]), i.e. 0, 0.025, 0.05, 0.1, 0.2, 0.25, 0.3, 0.4, 0.5, 0.6, 0.7, 0.75, 0.8, 0.9, 0.95, 0.975, 1. Value 0.5 corresponds to the median.

	Additional arguments passed to the underlying functions. In `pop.map`, these are `quantile`, `year`, `projection.index`, `device`, `main`, and `device.args` (see `tfr.map`). For `pop.ggmap`, these are arguments that can be passed to `tfr.ggmap`. For `pop.map.gvis`, these are all arguments that can be passed to `tfr.map.gvis`. In addition, `pop.map` and `get.pop.map.parameters` accept arguments passed to the `mapCountryData` function of the **rworldmap** package.

Details

`pop.map` creates a single map for the given time period and quantile. If the package **fields** is installed, a color bar legend at the botom of the map is created.

Function `get.pop.map.parameters` can be used in combination with `pop.map`. It sets breakpoints for the color scheme.

Function `pop.ggmap` is similar to `pop.map`, but uses the **ggplot2** package in combination with the `geom_sf` function.

Function `pop.map.gvis` creates an interactive map using the **googleVis** package and opens it in an internet browser. It also generates a table of the mapped values that can be sorted by columns interactively in the browser.

Value

`get.pop.map.parameters` returns a list with elements:

pred	The object of class `bayesPop.prediction` used in the function.
quantile	Value of the argument `quantile`.
catMethod	If the argument `same.scale` is TRUE, this element contains breakpoints for categorization. Otherwise, it is NULL.
numCats	Number of categories.
coulourPalette	Subset of the rainbow palette, starting from dark blue and ending at red.
...	Additional arguments passed to the function.

Performance and Caching

If the `expression` argument or a non-standard combination of sex and age is used, quantiles are computed on the fly. In such a case, trajectory files for all countries have to be loaded from disk, which can be quite time expensive. Therefore a simple caching mechanism was added to the prediction object which allows re-using data from previously used expressions. The prediction object points to an environment called `cache` which is a collection of data arrays that are results of evaluating expressions. The space-trimmed expressions are the names of the `cache` entries. Every time a map function is called, it is checked if the corresponding expression is contained in the cache. If it is not the case, the quantiles are computed on the fly, otherwise the existing values are taken.

When computing on the fly, the function tries to process it in parallel if possible, using the package **parallel**. In such a case, the computation is split into n nodes where n is either the number of cores detected automatically (default), or the value of `getOption("cl.cores")`. Use `options(cl.cores=n)` to modify the default. If a sequential processing is desired, set `cl.cores` to 1.

The cache data are also stored on disk, namely in the simulation directory of the prediction object. By default, every update of the cache in memory is also updated on the disk. Thus, data expression

results can be re-used in multiple R sessions. Function `pop.cleanup.cache` deletes the content of the cache. This behaviour can be turned off by setting the argument `write.to.cache=FALSE` in the `get.pop.prediction` function. We use this settings in the examples throughout this manual whenever the example data from the installation directory is used, in order to prevent writing into the installation directory.

Author(s)

Hana Sevcikova

See Also

`tfr.map`

Examples

```
## Not run:
###########################
# This example only makes sense if there is a simulation
# for all countries. Below, only two countries are included,
# so the map is useless.
###########################
sim.dir <- file.path(find.package("bayesPop"), "ex-data", "Pop")
pred <- get.pop.prediction(sim.dir=sim.dir, write.to.cache=FALSE)

# Using ggplot2
pop.ggmap(pred)
pop.ggmap(pred, year = 2100)

# Using rworldmap
# Uses heat colors with seven categories by default
pop.map(pred, sex="female", age=4:10)
# Female population in child-bearing age as a proportion of totals
pop.map(pred, expression="PXXX_F[4:10] / PXXX")
# The same with more colors
params <- get.pop.map.parameters(pred, expression="PXXX_F[4:10] / PXXX")
do.call("pop.map", params)
# Another projection year on the same color scale
do.call("pop.map", c(list(year=2043), params))

# Interactive map of potential support ratio (requires Flash)
pop.map.gvis(pred, expression="PXXX[5:13] / PXXX[14:27]")
## End(Not run)
```

---

pop.predict   *Probabilistic Population Projection*

Description

The function generates trajectories of probabilistic population projection for all countries for which input data is available, or any subset of them.

Usage

```
pop.predict(end.year = 2100, start.year = 1950, present.year = 2020,
    wpp.year = 2019, countries = NULL,
    output.dir = file.path(getwd(), "bayesPop.output"),
    annual = FALSE,
    inputs = list(popM=NULL, popF=NULL, mxM=NULL, mxF=NULL, srb=NULL,
        pasfr=NULL, patterns=NULL,
        migM=NULL, migF=NULL, migMt=NULL, migFt=NULL, mig=NULL,
        mig.fdm = NULL, e0F.file=NULL, e0M.file=NULL, tfr.file=NULL,
        e0F.sim.dir=NULL, e0M.sim.dir=NULL, tfr.sim.dir=NULL,
        migMtraj = NULL, migFtraj = NULL, migtraj = NULL,
        migFDMtraj = NULL, GQpopM = NULL, GQpopF = NULL,
        average.annual = NULL),
    nr.traj = 1000, keep.vital.events = FALSE,
    fixed.mx = FALSE, fixed.pasfr = FALSE,
    lc.for.hiv = TRUE, lc.for.all = TRUE, mig.is.rate = FALSE,
    mig.age.method  = c("auto", "fdmp", "fdmnop", "rc"), mig.rc.fam = NULL,
    my.locations.file = NULL, replace.output = FALSE, verbose = TRUE, ...)
```

Arguments

end.year	End year of the projection.
start.year	First year of the historical data.
present.year	Year for which initial population data is to be used.
wpp.year	Year for which WPP data is used. The functions loads a package called **wpp**x where x is the wpp.year and uses the various datasets as default if the corresponding inputs element is missing (see below).
countries	Array of country codes or country names for which a projection is generated. If it is NULL, all available countries are used. If it is NA and there is an existing projection in output.dir and replace.output=FALSE, then a projection is performed for all countries that are not included in the existing projection. Names of countries are matched to those in the [UNlocations](#) dataset (or in the dataset loaded from my.locations.file if used).
output.dir	Output directory of the projection. If there is an existing projection in output.dir and replace.output=TRUE, everything in the directory will be deleted.
annual	Logical. If TRUE it is assumed that this is 1x1 simulation, i.e. one year age groups and one year time periods. Note that this is still an experimental feature!
inputs	A list of file names where input data is stored. It contains the following elements (Unless otherwise noted, these are tab delimited ASCII files; Names of default datasets from the corresponding **wpp** package which are used if the corresponding element is NULL are shown in brackets):

    popM, popF Initial male/female age-specific population (at time present.year) [[popM](#), [popF](#)].

    mxM, mxF Historical data and (optionally) projections of male/female age-specific death rates [[mxM](#), [mxF](#)] (see also argument fixed.mx).

    srb Projection of sex ratio at birth. [[sexRatio](#)]

pasfr Historical data and (optionally) projections of percentage age-specific fertility rate [`percentASFR`] (see also argument `fixed.pasfr`).

patterns, mig.type Migration type and base year of the migration. In addition, this dataset gives information on country's specifics regarding mortality and fertility age patterns as defined in [`vwBaseYear`]. `patterns` and `mig.type` have the same meaning and can be used interchangeably.

migM, migF, migMt, migFt, mig Projection and (optionally) historical data of net migration on the same scale as the initital population. There are three ways of defining this quantity, here in order of priority: 1. via `migM` and `migF` which should give male and female age-specific migration [`migrationM`, `migrationF`]; 2. via `migMt` and `migFt` which should give male and female total net migration; 3. via `mig` which should give the total net migration. For 2. and 3., the totals are disagregated into age-specific migration by applying a schedule defined by the `mig.age.method` argument. If all of these input items are missing, for `wpp.year` = 2024 or 2012, the UN age schedules are used. For other WPP revisions, the migration schedules are reconstructed from total migration counts derived from `migration` using either the `age.specific.migration` or the `migration.totals2age` function.

mig.fdm If `mig.age.method` is "fdmp" or "fdmnop", this file is used to disaggregate total in- and out-migration into ages, giving proportions of the migration in-flow and out-flow for each age. It should have columns "country_code", "age", "in" and "out", where the latter two should each sum to 1 for each location. By default the function uses the `rc1FDM` (annual) or `rc5FDM` (5-year) datasets. For locations where the unique identifier does not match the country code in these default datasets, Rogers-Castro curves are used, obtained via the function `rcastro.schedule`.

e0F.file Comma-delimited CSV file with results of female life expectancy (generated using **bayesLife**, function `convert.e0.trajectories`, file "ascii_trajectories.csv"). Required columns are "LocID", "Year", "Trajectory", and "e0". If this element is not NULL, the argument `e0F.sim.dir` is ignored. If both `e0F.file` and `e0F.sim.dir` are NULL, data from the corresponding **wpp** package is taken, namely the median projections as one trajectory and the low and high variants (if available) as second and third trajectory. For 5-year simulations, column "Year" should be the middle year of the time period, e.g. 2023, 2028 etc.

e0M.file Comma-delimited CSV file containing results of male life expectancy (generated using **bayesLife**, function `convert.e0.trajectories`, file "ascii_trajectories.csv"). Required columns are "LocID", "Year", "Trajectory", and "e0". If this element is not NULL, the argument `e0M.sim.dir` is ignored. As in the female case, if both `e0M.file` and `e0M.sim.dir` are NULL, data from the corresponding **wpp** package is taken.

tfr.file Comma-delimited CSV file with results of total fertility rate (generated using **bayesTFR**, function `convert.tfr.trajectories`, file "ascii_trajectories.csv"). Required columns are "LocID", "Year", "Trajectory", and "TF". If this element is not NULL, the argument `tfr.sim.dir` is ignored. If both `tfr.file` and `tfr.sim.dir` are NULL, data from the corresponding **wpp** package is taken (median and the low and high variants as three trajectories). Alternatively, this argument can be the keyword "median_" in which case only the

wpp median is taken.

e0F.sim.dir Simulation directory with results of female life expectancy (generated using **bayesLife**). It is only used if e0F.file is NULL.

e0M.sim.dir Simulation directory with results of male life expectancy (generated using **bayesLife**). Alternatively, it can be the string "joint_", in which case it is assumed that the male life expectancy was projected jointly from the female life expectancy (see joint.male.predict) and thus contained in the e0F.sim.dir directory. The argument is only used if e0M.file is NULL.

tfr.sim.dir Simulation directory with results of total fertility rate (generated using **bayesTFR**). It is only used if tfr.file is NULL.

migMtraj, migFtraj, migtraj Comma-delimited CSV file with male/female age-specific migration trajectories, or total migration trajectories (migtraj). If present, it replaces deterministic projections given by the mig* items. It has a similar format as e.g. e0M.file with columns "LocID", "Year", "Trajectory", "Age" (except for migtraj) and "Migration". For a five-year simulation, the "Age" column must have values "0-4", "5-9", "10-14", …, "95-99", "100+", and the "Year" column should be the middle year of the time period, e.g. 2023, 2028 etc. In an annual simulation, age is given by a single number between 0 and 100, and "Year" contains all projected years.

migFDMtraj Comma-delimited CSV file with trajectories of in- and out-migration schedules used for the FDM migration method, i.e. if mig.age.method is "fdmp" or "fdmnop". The values have te same meaning as in the mig.fdm input item, except that here multiple trajectories of such schedules can be provided. It should have columns "LocID", "Age", "Trajectory", "Value", and "Parameter". For "Age", the same rules apply as for migMtraj above. The "Parameter" column should have values "in" for in-migration, "out" for out-migration and "v" for values of the variance denominator v used in Equation 22 of Sevcikova et al (2024). For the v parameter, the "Age" column should be left empty.

GQpopM, GQpopF Age-specific population counts (male and female) that should be excluded from application of the cohort component method (CCM). It can be used for defining group quarters. These counts are removed from population before the CCM projection and added back afterwards. It is not used when computing vital events on observed data. The datasets should have columns "country_code", "age" and "gq". In such a case the "gq" amount is applied to all years. If it is desired to destinguish the amount that is added back for individual years, the "gq" column should be replaced by columns indicating the individual years, i.e. single years for an annual simulation and time periods (e.g. "2020-2025", "2025-2030") for a 5-year simulation. For a five-year simulation, the "age" column should include values "0-4", "5-9", "10-14", …, "95-99", "100+". However, rows with zeros do not need to be included. In an annual simulation, age is given by a single number between 0 and 100.

average.annual Character string with values "TFR", "e0M", "e0F". If this is a 5-year simulation, but the inputs of TFR or/and e0 comes from an annual simulation, including the corresponding string here will cause that the TFR or/and e0 trajectories are converted into 5-year averages.

nr.traj	Number of trajectories to be generated. If this number is smaller than the number of available trajectories of the probabilistic components (TFR, life expectancy and migration), the trajectories are equidistantly thinned. If all of those components contain less trajectories than nr.traj, the value is adjusted to the maximum of available trajectories of the components. For those that have less trajectories than the adjusted number, the available trajectories are re-sampled, so that all components have the same number of trajectories.
keep.vital.events	
	Logical. If TRUE age- and sex-specific vital events of births and deaths as well as other objects are stored in the prediction object, see Details.
fixed.mx	Logical. If TRUE, it is assumed the dataset of death rates (mxM and mxF) include data for projection years and they are then used instead of the life expectancy.
fixed.pasfr	Logical. If TRUE, it is assumed the dataset on percent age-specific fertility rate (percentASFR) include data for projection years and they are then used instead of computing it on the fly.
lc.for.hiv	Logical controlling if the modified Lee-Carter method should be used for projection of mortality rates for countries with HIV epidemics. If FALSE, the function hiv.mortmod from the HIV.LifeTables package is used.
lc.for.all	Logical controlling if the modified Lee-Carter method should be used for projection of mortality rates for all countries. If FALSE, the corresponding method is determined by the columns "AgeMortProjMethod1" and "AgeMortProjMethod2" of the [vwBaseYear](#) dataset.
mig.is.rate	Logical determining if migration data are to be interpreted as net migration rates (TRUE) or counts (FALSE, default). It can also be a vector of two logicals, where the first element refers to observed data and the second element refers to predictions. A value of c(FALSE, TRUE) could for example be used if observed data in inputs$mig are counts, and migration trajectories in inputs$migtraj are rates.
mig.age.method	If migration is given as totals, this argument determines a method to disaggregate into age-specific migration.
	The "rc" method uses a simple Rogers-Castro disaggregation, via the function [rcastro.schedule](#). An alternative schedule can be passed via the mig.rc.fam argument.
	Values "fdmp" and "fdmnop" trigger the Flow Difference Method (Sevcikova et al, 2024), where "fdmp" weights the flows by population, while "fdmnop" is an unweighted version. They both split the total net migration into total in- and out-migration and then disaggregate these flows separately. These two FDM methods use additional inputs in the inputs$rc.fdm and/or inputs$migFDMtraj components.
	The "auto" method (default) uses "rc" if sex-specific migration totals are given, i.e. in inputs$migFt and inputs$migMt. If annual is FALSE and wpp.year is 2015, 2017 or 2019, then the residual method using the function [age.specific.migration](#) is used. Otherwise the "fdmp" method is applied.
mig.rc.fam	Data frame providing a single family of Rogers-Castro parameters to be used if mig.age.method is set to "rc". Mandatory columns are "age" and "prop". Optionally, it can have a column "mig_sign" with values "Inmigration" and "Emi-

gration" (distinguishing schedules to be applied for positive and negative migration, respectively) and a column "sex" with values "Female" and "Male". The format corresponds to the dataset DemoTools::mig_un_families, subset to a single family. If this argument is NULL and mig.age.method = "rc", the function rcastro.schedule with equal sex ratio is used to distribute total migration into ages.

my.locations.file

Name of a tab-delimited ascii file with a set of all locations for which a projection is generated. Use this argument if you are projecting for a country/region that is not included in the standard UNlocations dataset. It must have the same structure.

replace.output Logical. If TRUE, everything in the directory output.dir is deleted prior to the prediction.

verbose Logical controlling the amount of output messages.

... Additional arguments passed to the underlying function. These can be parallel and nr.nodes for parallel processing and the number of nodes, respectively, as well as further arguments passed for creating a parallel cluster.

Details

The population projection is computed using the cohort component method and is based on an algorithm used by the United Nation Population Division (see also Sevcikova et al (2016b) in the References below). For each country, one projection is calculated for each trajectory of male and female life expectancy, TFR and possibly migration. This results in a set of trajectories of population projection which forms its posterior distribution. The trajectories of life expectancy and TFR can be given either in its binary form generated by the packages **bayesLife** and **bayesTFR**, respectively (as directories e0M.sim.dir, e0F.sim.dir, tfr.sim.dir of the inputs argument), or they can be given as ASCII tables in csv format, see above. The number of trajectories for male and female life expectancy must match, as does for male and female migration.

The projection is generated sequentially location by location. Results are stored in a sub-directory of output.dir called 'prediction'. There is one binary file per location, called 'totpop_country$x$.rda', where x is the country code. It contains six objects: totp, totpf, totpm (trajectories of total population, age-specific female and age-specific male, respectively), totp.hch, totpf.hch, totpm.hch (the UN half-child variant for total population, age-specific female and age-specific male, respectively). Optionally, if keep.vital.events is TRUE, there is an additional file per country, called 'vital_events_country$x$.rda', containing the following objects: btm, btf (trajectories for births by age of mothers for male and female child, respectively), deathsm, deathsf (trajectories for age-specific male and female deaths, respectively), asfert (trajectories of age-specific fertility), mxm, mxf (trajectories of male and female age-specific mortality rates), migm, migf (if used, these are trajectories of male and female age-specific migration), btm.hch, btf.hch, deathsm.hch, deathsf.hch, asfert.hch, mxm.hch, mxf.hch (the UN half-child variant for age- and sex-specific births, deaths, fertility rates and mortality rates). An object of class bayesPop.prediction is stored in the same directory in a file 'prediction.rda'. It is updated every time a country projection is finished.

See pop.trajectories for extracting trajectories.

To access a previously stored prediction object, use get.pop.prediction.

Value

Object of class bayesPop.prediction with the following elements:

base.directory   Full path to the base directory output.dir.

output.directory

Sub-directory relative to base.directory with the projections.

nr.traj   The actual number of trajectories of the projections.

quantiles   Three-dimensional array of projection quantiles (countries x number of quantiles x projection periods). The second dimension corresponds to the following quantiles: $0.025, 0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 0.95, 0.975$.

traj.mean.sd   Three-dimensional array of projection mean and standard deviation (countries x 2 x projection periods). First and second matrix of the second dimension, respectively, is the mean and standard deviation, respectively.

quantilesM, quantilesF

Quantiles of male and female projection, respectively. Same structure as quantiles.

traj.mean.sdM, traj.mean.sdF

Same as traj.mean.sd corresponding to male and female projection, respectively.

quantilesMage, quantilesFage

Four-dimensional array of age-specific quantiles of male and female projection, respectively (countries x age groups x number of quantiles x projection periods). The same quantiles are used as in quantiles.

quantilesPropMage, quantilesPropFage

Array of age-specific quantiles of male and female projection, respectively, divided by the total population. The dimensions are the same as in quantilesMage.

estim.years   Vector of time for which historical data was used in the projections.

proj.years   Vector of projection time periods starting with the present period.

wpp.year   The wpp year used.

inputs   List of input data used for the projection.

function.inputs

Content of the inputs argument passed to the function.

countries   Matrix of countries for which projection exists. It contains two columns: code, name.

ages   Vector of age groups.

annual   If TRUE, this object corresponds to a 1x1 prediction, otherwise 5x5.

cache   This component is added by get.pop.prediction and modified and used by pop.map and write.pop.projection.summary. It is an environment for caching and re-using results of expressions.

write.to.cache   Logical determining if cache should be modified.

is.aggregation   Logical determining if this object is a result of pop.predict or pop.aggregate.

Author(s)

Hana Sevcikova, Thomas Buettner, based on code of Nan Li and helpful comments from Patrick Gerland

References

H. Sevcikova, A. E. Raftery (2016a). bayesPop: Probabilistic Population Projections. Journal of Statistical Software, 75(5), 1-29. doi:10.18637/jss.v075.i05

A. E. Raftery, N. Li, H. Sevcikova , P. Gerland, G. K. Heilig (2012). Bayesian probabilistic population projections for all countries. Proceedings of the National Academy of Sciences 109:13915-13921.

P. Gerland, A. E. Raftery, H. Sevcikova, N. Li, D. Gu, T. Spoorenberg, L. Alkema, B. K. Fosdick, J. L. Chunn, N. Lalic, G. Bay, T. Buettner, G. K. Heilig, J. Wilmoth (2014). World Population Stabilization Unlikely This Century. Science 346:234-237.

H. Sevcikova, N. Li, V. Kantorova, P. Gerland and A. E. Raftery (2016b). Age-Specific Mortality and Fertility Rates for Probabilistic Population Projections. In: Dynamic Demographic Analysis, ed. Schoen R. (Springer), pp. 285-310. Earlier version in arXiv:1503.05215.

H. Sevcikova, J. Raymer J., A. E. Raftery (2024). Forecasting Net Migration By Age: The Flow-Difference Approach. arXiv:2411.09878.

See Also

[pop.trajectories.plot](), [pop.pyramid](), [pop.trajectories](), [get.pop.prediction](), [age.specific.migration]()

Examples

```
## Not run:
sim.dir <- tempfile()
# Countries can be given as a combination of numerical codes and names
pred <- pop.predict(countries=c("Netherlands", 218, "Madagascar"), nr.traj=3,
         output.dir=sim.dir)
pop.trajectories.plot(pred, "Ecuador", sum.over.ages=TRUE)
unlink(sim.dir, recursive=TRUE)

## End(Not run)
```

---

pop.predict.subnat *Subnational Probabilistic Population Projection*

Description

Generates trajectories of probabilistic population projection for subregions of a given country.

Usage

```
pop.predict.subnat(end.year = 2060, start.year = 1950, present.year = 2020,
        wpp.year = 2019, output.dir = file.path(getwd(), "bayesPop.output"),
        locations = NULL, default.country = NULL, annual = FALSE,
        inputs = list(
            popM = NULL, popF = NULL,
            mxM = NULL, mxF = NULL, srb = NULL,
```

```
              pasfr = NULL, patterns = NULL,
              migM = NULL, migF = NULL,
              migMt = NULL, migFt = NULL, mig = NULL, mig.fdm = NULL,
              e0F.file = NULL, e0M.file = NULL, tfr.file = NULL,
              e0F.sim.dir = NULL, e0M.sim.dir = NULL, tfr.sim.dir = NULL,
              migMtraj = NULL, migFtraj = NULL, migtraj = NULL,
              migFDMtraj = NULL, GQpopM = NULL, GQpopF = NULL,
              average.annual = NULL
         ),
         nr.traj = 1000, keep.vital.events = FALSE,
         fixed.mx = FALSE, fixed.pasfr = FALSE, lc.for.all = TRUE,
          mig.is.rate = FALSE, mig.age.method = c("rc", "fdmp", "fdmnop"),
          mig.rc.fam = NULL, pasfr.ignore.phase2 = FALSE,
          replace.output = FALSE, verbose = TRUE)
```

Arguments

end.year	End year of the projection.
start.year	First year of the historical data on mortality rates. It determines the length of the historical time series used in the Lee-Carter estimation.
present.year	Year for which initial population data is to be used.
wpp.year	Year for which WPP data is used. The function loads a package called **wpp**x where x is the wpp.year and uses its data (corresponding to the default.country) as default datasets if region-specific alternatives are not given (see more details below).
output.dir	Output directory of the projection.
locations	Name of a tab-delimited file that contains definitions of the subregions. It has a similar structure as [UNlocations](), with mandatory columns reg_code (unique identifier of the subregions) and name (name of the subregions). Optionally, location_type should be set to 4 for subregions to be processed. Column country_code can be included with the numerical code of the corresponding country. A row with location_type of 0 determines the country that the subregions belong to and is used for extracting default "national" datasets if the argument default.country is missing. In such a case, the code of the default country is taken from its column country_code. This is a mandatory argument.
default.country	Numerical code of a country to which the subregions belong to. It is used for extracting default datasets from the **wpp** package if some region-specific input datasets are missing. Alternatively, it can be also included in the locations file, see above. In either case, the code must exists in the [UNlocations]() dataset.
annual	Logical. If TRUE it is assumed that this is 1x1 simulation, i.e. one year age groups and one year time periods.
inputs	A list of file names where input data is stored. Unless otherwise noted, these are tab delimited ASCII files with a mandatory column reg_code giving the numerical identifier of the subregions. If an element of this list is NULL, usually a default dataset corresponding to default.country is extracted from the

wpp package. Names of these default datasets are shown in brackets. This list contains the following elements:

popM, popF Initial male/female age-specific population (at time present.year). Mandatory items, no defaults. Must contain columns reg_code and age and be of the same structure as popM from **wpp**.

mxM, mxF Historical data and (optionally) projections of male/female age-specific death rates [[mxM](mxM), [mxF](mxF)] (see also argument fixed.mx).

srb Projection of sex ratio at birth. [sexRatio]

pasfr Historical data and (optionally) projections of percentage age-specific fertility rate [percentASFR] (see also argument fixed.pasfr).

patterns Information on region's specifics regarding migration type, base year of the migration, mortality and fertility age patterns as defined in [vwBaseYear]. In addition, it can contain columns defining migration shares between the subregions, see Details below.

migM, migF, migMt, migFt, mig Projection and (optionally) historical data of net migration on the same scale as the initital population. There are three ways of defining this quantity, here in order of priority: 1. via migM and migF which should give male and female age-specific migration [migrationM, migrationF]; 2. via migMt and migFt which should give male and female total net migration; 3. via mig which should give the total net migration. For 2. and 3., the totals are disagregated into age-specific migration by applying a Rogers-Castro schedule. For 3., the totals are equally split between sexes. If all of these input items are missing, the migration schedules are constructed from total migration counts of the default.country derived from migration using Rogers Castro for age distribution. Migration shares between subregions (including sex-specific shares) can be given in the patterns file, see above and Details below. If no shares are given, it is distributed by population shares.

mig.fdm If mig.age.method is "fdmp" or "fdmnop", this file is used to disaggregate total in- and out-migration into ages, giving proportions of the migration in-flow and out-flow for each age. It should have columns "reg_code", "age", "in" and "out", where the latter two should each sum to 1 for each location. By default Rogers-Castro curves are used, obtained via the function rcastro.schedule.

e0F.file Comma-delimited CSV file with projected female life expectancy. It has the same structure as the file "ascii_trajectories.csv" generated using bayesLife::convert.e0.trajectories (which currently works for country-level results only). Required columns are "LocID", "Year", "Trajectory", and "e0". If e0F.file is NULL, data from the corresponding **wpp** package (for default.country) is taken, namely the median projections as one trajectory and the low and high variants (if available) as second and third trajectory. Alternatively, this element can be the keyword "median_" in which case only the median is taken.

e0M.file Comma-delimited CSV file containing projections of male life expectancy of the same format as e0F.file. As in the female case, if e0M.file is NULL, data for default.country from the corresponding **wpp** package is taken.

tfr.file Comma-delimited CSV file with results of total fertility rate (generated using **bayesTFR**, function `convert.tfr.trajectories`, file "ascii_trajectories.csv"). Required columns are "LocID", "Year", "Trajectory", and "TF". If this element is not NULL, the argument `tfr.sim.dir` is ignored. If both `tfr.file` and `tfr.sim.dir` are NULL, data for `default.country` from the corresponding **wpp** package is taken (median and the low and high variants as three trajectories). Alternatively, this argument can be the keyword "median_" in which case only the wpp median is taken.

e0F.sim.dir Simulation directory with results of female life expectancy, generated using bayesLife::`e0.predict.subnat`. It is only used if `e0F.file` is NULL. Alternatively, it can be set to the keyword "median_" which has the same effect as when `e0F.file` is "median_".

e0M.sim.dir This is analogous to `e0F.sim.dir`, here for male life expectancy. Use `e0M.file` instead of this item.

tfr.sim.dir Simulation directory with projections of total fertility rate (generated using bayesTFR::`tfr.predict.subnat`). It is only used if `tfr.file` is NULL.

migMtraj, migFtraj, migtraj Comma-delimited CSV file with male/female age-specific migration trajectories, or total migration trajectories (`migtraj`). If present, it replaces deterministic projections given by the `mig*` items. It has a similar format as e.g. `e0M.file` with columns "LocID", "Year", "Trajectory", "Age" (except for `migtraj`) and "Migration". For a five-year simulation, the "Age" column must have values "0-4", "5-9", "10-14", ..., "95-99", "100+". In an annual simulation, age is given by a single number between 0 and 100.

migFDMtraj Comma-delimited CSV file with trajectories of in- and out-migration schedules used for the FDM migration method, i.e. if `mig.age.method` is "fdmp" or "fdmnop". The values have te same meaning as in the `mig.fdm` input item, except that here multiple trajectories of such schedules can be provided. It should have columns "LocID", "Age", "Trajectory", "Value", and "Parameter". For "Age", the same rules apply as for `migMtraj` above. The "Parameter" column should have values "in" for in-migration, "out" for out-migration and "v" for values of the variance denominator v used in Equation 22 of Sevcikova et al (2024). For the v parameter, the "Age" column should be left empty.

GQpopM, GQpopF Age-specific population counts (male and female) that should be excluded from application of the cohort-component method (CCM). It can be used for defining group quarters. These counts are removed from population before the CCM projection and added back afterwards. It is not used when computing vital events on observed data. The datasets should have columns "reg_code", "age" and "gq". In such a case the "gq" amount is applied to all years. If it is desired to destinguish the amount that is added back for individual years, the "gq" column should be replaced by columns indicating the individual years, i.e. single years for an annual simulation and time periods (e.g. "2020-2025", "2025-2030") for a 5-year simulation. For a five-year simulation, the "age" column should include values "0-4", "5-9", "10-14", ..., "95-99", "100+". However, rows with zeros do not need to be included. In an annual simulation, age is given by a single

number between 0 and 100.

average.annual Character string with values "TFR", "e0M", "e0F". If this is a 5-year simulation, but the inputs of TFR or/and e0 comes from an annual simulation, including the corresponding string here will cause that the TFR or/and e0 trajectories are converted into 5-year averages.

nr.traj, keep.vital.events, fixed.mx, fixed.pasfr, lc.for.all, mig.is.rate, mig.age.method, mig.rc.fam, replace.output, verbose

These arguments have the same meaning as in `pop.predict`.

pasfr.ignore.phase2

Logical. If TRUE the TFR for all locations is considered being in phase III when predicting PASFR.

Details

Population projection for subnational units (regions) is performed by applying the cohort component method to subnational datasets on projected fertility (TFR), mortality and net migration, starting from given sex- and age-specific population counts. The only required inputs are the initial sex- and age-specific population counts in each region (popM and popF elements of the inputs argument) and a file with a set of locations (argument locations). If no other input datasets are given, those datasets are replaced by the corresponding "national" values, taken from the corresponding **wpp** package. The argument default.country determines the country for those default "national" values. The default country can be also included in the locations file as a record with location.type being set to 0.

The TFR component can be given as a set of trajectories generated using the `tfr.predict.subnat` function of the **bayesTFR** package (tfr.sim.dir element). Alternatively, trajectories can be given in an ASCII file (tfr.file).

Similarly, the e_0 component can be given as a set of trajectories using the `e0.predict.subnat` function of the **bayesLife** package (e0F.sim.dir element). If male projections are generated jointly (i.e. predict.jmale = TRUE), set e0M.sim.dir = "joint_". Alternatively, trajectories can be given in an ASCII files (e0F.file, e0M.file).

Having a set of subnational TFR and e_0 trajectories, the cohort component method is applied to each of them to yield a distribution of future subnational population.

Projection of net migration can either be given as disaggregated sex- and age-specific datasets (migM and migF), or as sex totals (migMt and migFt), or as totals (mig), or as sex- and age-specific trajectories (migMtraj and migFtraj), or as total trajectories (migtraj). Alternatively, it can be given as shares between regions as columns in the patterns dataset. These are: inmigrationM_share, inmigrationF_share, outmigrationM_share, outmigrationF_share. The sex specification and/or direction specification (in/out) can be omitted, e.g. it can be simply migration_share. The function extracts the values of net migration projection on the national level and distributes it to regions according to the given shares. For positive (national) values, it uses the in-migration shares; for negative values it uses the out-migration shares. If the in/out prefix is omitted in the column names, the given migartion shares are used for both, positive and negative net migration projection. By default, if no migration datasets neither region-specific shares are given, the distribution between regions is proportional to the size of population. The age-specific schedules follow by default the Rogers-Castro age schedules. Note that when handling migration using shares as described here, it only affects the distribution of international migration into regions. It does not take into account between-region migration.

The package contains example datasets for Canada. Use these as templates for your own data. See Example below.

Value

Object of class `bayesPop.prediction` containing the subnational projections. Note that this object can be used in the various **bayesPop** functions exactly the same way as an object with national projections. However, the meaning of the argument `country` in many of these functions (e.g. in `pop.trajectories.plot`) changes to an identification of the region (either as a numerical code or name as defined in the `locations` file).

Acknowledgment

We are greatful to Patrice Dion from Statistics Canada for providing us with example data. Note that the example datasets included in the package are not official STATCAN data - they only serve the purpose of illustration and templates. Data for the time period 2015-2020 has been imputed by the author.

Author(s)

Hana Sevcikova

See Also

`pop.predict`, `tfr.predict.subnat`, `pop.aggregate.subnat`

Examples

```
## Not run:
# Subnational projections for Canada
#########
data.dir <- file.path(find.package("bayesPop"), "extdata")

# Use national data for tfr and e0
###
sim.dir <- tempfile()
pred <- pop.predict.subnat(output.dir = sim.dir,
          locations = file.path(data.dir, "CANlocations.txt"),
          inputs = list(popM = file.path(data.dir, "CANpopM.txt"),
                        popF = file.path(data.dir, "CANpopF.txt"),
                        tfr.file = "median_"
                      ),
          verbose = TRUE)
pop.trajectories.plot(pred, "Alberta", sum.over.ages = TRUE)
unlink(sim.dir, recursive=TRUE)

# Use subnational probabilistic TFR simulation
###
# Subnational TFR projections for Canada (from ?tfr.predict.subnat)
my.subtfr.file <- file.path(find.package("bayesTFR"), 'extdata', 'subnational_tfr_template.txt')
tfr.nat.dir <- file.path(find.package("bayesTFR"), "ex-data", "bayesTFR.output")
tfr.reg.dir <- tempfile()
```

```
tfr.preds <- tfr.predict.subnat(124, my.tfr.file = my.subtfr.file,
    sim.dir = tfr.nat.dir, output.dir = tfr.reg.dir, start.year = 2013)

# Use subnational probabilistic e0
###
# Subnational e0 projections for Canada (from ?e0.predict.subnat)
# (here using the same female and male data, just for illustration)
my.sube0.file <- file.path(find.package("bayesLife"), 'extdata', 'subnational_e0_template.txt')
e0.nat.dir <- file.path(find.package("bayesLife"), "ex-data", "bayesLife.output")
e0.reg.dir <- tempfile()
e0.preds <- e0.predict.subnat(124, my.e0.file = my.sube0.file,
    sim.dir = e0.nat.dir, output.dir = e0.reg.dir, start.year = 2018,
    predict.jmale = TRUE, my.e0M.file = my.sube0.file)

# Population projections
sim.dir <- tempfile()
pred <- pop.predict.subnat(output.dir = sim.dir,
            locations = file.path(data.dir, "CANlocations.txt"),
            inputs = list(popM = file.path(data.dir, "CANpopM.txt"),
                          popF = file.path(data.dir, "CANpopF.txt"),
                          patterns = file.path(data.dir, "CANpatterns.txt"),
                          tfr.sim.dir = file.path(tfr.reg.dir, "subnat", "c124"),
                          e0F.sim.dir = file.path(e0.reg.dir, "subnat_ar1", "c124"),
                          e0M.sim.dir = "joint_"
                        ),
            verbose = TRUE)
pop.trajectories.plot(pred, "Alberta", sum.over.ages = TRUE)
pop.pyramid(pred, "Manitoba", year = 2050)
get.countries.table(pred)

# Aggregate to country level
aggr <- pop.aggregate.subnat(pred, regions = 124,
            locations = file.path(data.dir, "CANlocations.txt"))
pop.trajectories.plot(aggr, "Canada", sum.over.ages = TRUE)

unlink(sim.dir, recursive = TRUE)
unlink(tfr.reg.dir, recursive = TRUE)
unlink(e0.reg.dir, recursive = TRUE)

## End(Not run)
```

---

pop.pyramid   *Probabilistic Population Pyramid*

Description

Functions for plotting probabilistic population pyramid. pop.pyramid creates a classic pyramid
using rectangles; pop.trajectories.pyramid creates one or more pyramids using vertical lines
(possibly derived from population trajectories). They can be used to view a prediction object cre-
ated with this package, or any user-defined sex- and age-specific dataset. For the latter, function
get.bPop.pyramid should be used to translate user-defined data into a bayesPop.pyramid object.

Usage

```
## S3 method for class 'bayesPop.prediction'
pop.pyramid(pop.object, country, year = NULL,
    indicator = c("P", "B", "D"), pi = c(80, 95),
    proportion = FALSE, age = NULL, plot = TRUE, pop.max = NULL, ...)

## S3 method for class 'bayesPop.pyramid'
pop.pyramid(pop.object, main = NULL, show.legend = TRUE,
    pyr1.par = list(border="black", col=NA, density=NULL, height=0.9),
    pyr2.par = list(density = -1, height = 0.3),
    show.birth.year = FALSE,
    col.pi = NULL, ann = par("ann"), axes = TRUE, grid = TRUE,
    cex.main = 0.9, cex.sub = 0.9, cex = 0.8, cex.axis = 0.8, ...)

pop.pyramidAll(pop.pred, year = NULL,
    output.dir = file.path(getwd(), "pop.pyramid"),
    output.type = "png", one.file = FALSE, verbose = FALSE, ...)

## S3 method for class 'bayesPop.prediction'
pop.trajectories.pyramid(pop.object, country, year = NULL,
    indicator = c("P", "B", "D"), pi = c(80, 95), nr.traj = NULL,
    proportion = FALSE, age = NULL, plot = TRUE, pop.max = NULL, ...)

## S3 method for class 'bayesPop.pyramid'
pop.trajectories.pyramid(pop.object, main = NULL, show.legend = TRUE,
    show.birth.year = FALSE, col = rainbow, col.traj = "#00000020",
    omit.page.pars = FALSE, lwd = 2, ann = par("ann"), axes = TRUE, grid = TRUE,
    cex.main = 0.9, cex.sub = 0.9, cex = 0.8, cex.axis = 0.8, ...)

pop.trajectories.pyramidAll(pop.pred, year = NULL,
    output.dir = file.path(getwd(), "pop.traj.pyramid"),
    output.type = "png", one.file = FALSE, verbose = FALSE, ...)

## S3 method for class 'bayesPop.pyramid'
plot(x, ...)

## S3 method for class 'bayesPop.prediction'
get.bPop.pyramid(data, country, year = NULL,
    indicator = c("P", "B", "D"), pi = c(80, 95),
    proportion = FALSE, age = NULL, nr.traj = 0, sort.pi=TRUE, pop.max = NULL, ...)

## S3 method for class 'data.frame'
get.bPop.pyramid(data, main.label = NULL, legend = "observed",
    is.proportion = FALSE, ages = NULL, pop.max = NULL,
    LRmain = c("Male", "Female"), LRcolnames = c("male", "female"), CI = NULL, ...)

## S3 method for class 'matrix'
get.bPop.pyramid(data, ...)
```

```
## S3 method for class 'list'
get.bPop.pyramid(data, main.label = NULL, legend = NULL, CI = NULL, ...)
```

Arguments

pop.object	Object of class [bayesPop.prediction](#) or bayesPop.pyramid (see Value section).
pop.pred	Object of class [bayesPop.prediction](#).
x	Object of class bayesPop.pyramid.
data	Data frame, matrix, list or object of class [bayesPop.prediction](#). For data frame and matrix, it must have columns defined by LRcolnames ("male" and "female" by default). The row names will determine the age labels. For lists, it can be a collection of such data frames. The names of the list elements are used for legend, unless legend is given.
country	Name or numerical code of a country. It can also be given as ISO-2 or ISO-3 characters.
year	Year within the projection or estimation period to be plotted. Default is the start year of the prediction. It can also be a vector of years. pop.pyramid draws the first two, pop.trajectories.pyramid draws all of them. In the functions pop.pyramidAll and pop.trajectories.pyramidAll, the year argument can be a list of years, in which case the pyramids are created for all elements in the list.
indicator	One of the characters "P" (population), "B" (births), "D" (deaths) determining the pyramid indicator.
pi	Probability interval. It can be a single number or an array.
proportion	Logical. If TRUE the pyramid contains the distribution of rates of age-specific counts and population totals.
age	Integer vector of age indices. In a 5-year simulation, value 1 corresponds to age 0-4, value 2 corresponds to age 5-9 etc. In a 1x1 simulation, values 1, 2, 3 correpond to ages 0, 1, 2. Last available age goup is 130+ which corresponds to index 27 in a 5-year simulation and index 131 in an annual simulation. The purpose of this argument here is mainly to control the height of the pyramid.
plot	If FALSE, nothing is plotted. It can be used to retrieve the pyramid object without drawing it.
main	Titel of the plot. By default it is the country name and projection year if known.
show.legend	Logical controlling if the plot legend is drawn.
pyr1.par, pyr2.par	
	List of graphical parameters (color, border, density and height) for drawing the pyramid rectangles, for the first and second pyramid, respectively (see Details). The height component should be a number between 0 (corresponds to a line) and 1 (for non-overelapping rectangles). If density is NULL, the rectangles are transparent, see the argument density in [rect](#).
show.birth.year	
	Logical. If TRUE the corresponding birth years are shown on the right vertical axis.

col.pi	Vector of colors for drawing the probability boxes. If it is given, it must be of the same length as pi.
ann	Logical controlling if any annotation (main and legend) is plotted.
axes	Logical controlling if axes are plotted.
grid	Logical controlling if grid lines are plotted.
cex.main, cex.sub, cex, cex.axis	
	Magnification to be used for the title, secondary titles on the right and left panels, legend and axes, respectively.
output.dir	Directory into which resulting graphs are stored.
output.type	Type of the resulting files. It can be "png", "pdf", "jpeg", "bmp", "tiff", or "postscript".
one.file	Logical. If TRUE the output is put into one single file, by default a PDF.
verbose	Logical switching log messages on and off.
nr.traj	Number of trajectories to be plotted. If NULL, all trajectories are plotted, otherwise they are thinned evenly.
col	Colors generating function. It is called with an argument giving the number of pyramids to be plotted. Each color is then used for one pyramid, including its confidence intervals.
col.traj	Color used for trajectories. If more than one pyramid is drawn with its trajectories, this can be a vector of the size of number of pyramids.
omit.page.pars	Logical. If TRUE, no page parameters are set. Can be used if multiple pyramids are to be put on one page.
lwd	Line width for the pyramids.
sort.pi	Logical controlling if the probability intervals are sorted in decreasing order. This has an effect on the order in which they are plotted and thus on overlapping of pyramid boxes. By default the largest intervals are plotted first.
main.label	Optional argument for the main title.
legend	Legend to be used. In case of multiple pyramids, this can be a vector for each of them. If not given and data is a list, names of the list elements are taken as legend.
is.proportion	Either logical, indicating if the values in data are proportions, or NA in which case the proportions are computed.
ages	Vector of age labels. It must be of the same length as the number of rows of data. If it is not given, the age labels are considered to be the row names of data.
pop.max	Maximum value to be drawn in the pyramid. If it is not given, max(data) is taken.
LRmain	Vector of character strings giving the secondary titles for the left and right panel, respectively.
LRcolnames	Vector of character strings giving the column names of data to be used for the left and right panel of the pyramid, respectively.

CI	Confidence intervals. It should be of the same format as the `bayesPop.pyramid$CI` object, see below.
...	Arguments passed to the underlying functions. For `get.bPop.pyramid`, these can be additional items to be added to the resulting object, e.g. `pyr.year` and `is.annual`.

Details

The `pop.pyramid` function generates one or two population pyramids in one plot. The first (main) one is usually the median of a future year prediction, but it can also be the current year or any population estimates. The second one serves the purpose of comparing two pyramids with one another and is drawn on top of the main pyramid. For example, one can use it to compare a future prediction with the present, or two different time points in the past, or two different geographies. The main pyramid can have confidence intervals associated with it, which are also plotted. If `pop.pyramid` is called on a `bayesPop.prediction` object, the main and secondary pyramid, respectively, is generated from data of a time period given by the first and second element, respectively, of the `year` argument. In such a case, confidence intervals only of the first year are shown. Thus, it makes sense to set the first year to be a prediction year and the second year to an observed time period. If `pop.pyramid` is called on a `bayesPop.pyramid` object, data in the first and second element, respectively, of the `bayesPop.pyramid$pyramid` list are used, and only the first element of `bayesPop.pyramid$CI` is used.

Pyramids generated via the `pop.trajectories.pyramid` function have different appearance and therefore more than two pyramids can be put into one figure. Furthermore, confidence intervals of more than one pyramid can be shown. Thus, all elements of `bayesPop.pyramid$pyramid` and `bayesPop.pyramid$CI` are plotted. In addition, single trajectories given in `bayesPop.pyramid$trajectories` can be shown by setting the argument `nr.traj` larger than 0.

Both, `pop.pyramid` and `pop.trajectories.pyramid` (if called with a `bayesPop.prediction` object) use data from one country. Functions `pop.pyramidAll` and `pop.trajectories.pyramidAll` create such pyramids for all countries for which a projection is available and for all years given by the `year` argument which should be a list. In this case, one pyramid figure (possibly containing multiple pyramids) is created for each country and each element of the `year` list.

The core of these functions operates on a `bayesPop.pyramid` object which is automatically created when called with a `bayesPop.prediction` object. If used with a user-defined data set, one has to convert such data into `bayesPop.pyramid` using the function `get.bPop.pyramid` (see an example below). In such a case, one can simply use the `plot` function which then calls `pop.pyramid`.

Value

`pop.pyramid`, `pop.trajectories.pyramid` and `get.bPop.pyramid` return an object of class `bayesPop.pyramid` which is a list with the following components:

label	Label used for the main titel.
pyramid	List of pyramid data, one element per pyramid. Each component is a data frame with at least two columns, containing data for the left and right panels of the pyramid. Their names must correspond to `LRcolnames` (see below). There is one row per age group and the row names are used for labeling the y-axis. Names of the list elements are used in the legend.

CI   List of lists of confidence intervals with one element per pyramid. The order corresponds to the order in the `pyramid` component and it is NULL if the corresponding pyramid does not have confidence intervals. Each element is a list with one element per probability interval whose names are the values of the intervals. Each element is again a list with components `low` and `high` which have the same structure as `pyramid` and contain the lower and upper bounds of the corresponding interval.

trajectories   List of lists of trajectories with one element per pyramid. As in the case of CI, it is ordered the same way as the `pyramid` component and is NULL if the corresponding pyramid does not have any trajectories to be shown. Each element is again a list with two components, one for the left part and one for the right part of the pyramid. Their names correspond to `LRcolnames` and each of them is a matrix of size number of age categories x number of trajectories. This is only used by the `pop.trajectories.pyramid` function.

is.proportion   Logical indicating if values in the various data frames in this object are proportions or raw values.

is.annual   Logical indicating if the data correspond to 1-year age groups. If FALSE, the ages are considered to be 5-year age groups.

pyr.year   Year of the main pyramid. It is used as the base year when `show.birth.year` is TRUE.

pop.max   Maximum value for the x-axis.

LRmain   Vector of character strings determining the titles for the left and right panels, respectively.

LRcolnames   Vector of character strings determining the column names in `pyramid`, CI and `trajectories` used to plot data into the left and right panel, respectively.

Author(s)

Hana Sevcikova, Adrian Raftery, using feedback from Sam Clark and the bayesPop group at the University of Washington.

See Also

pop.trajectories.plot, bayesPop.prediction, summary.bayesPop.prediction

Examples

```
# pyramids for bayesPop prediction objects
###########################################
sim.dir <- file.path(find.package("bayesPop"), "ex-data", "Pop")
pred <- get.pop.prediction(sim.dir)
pop.pyramid(pred, "Netherlands", c(2045, 2010))
dev.new()
pop.trajectories.pyramid(pred, "NL", c(2045, 2010, 1960), age=1:25, proportion=TRUE)
# using manual manipulation of the data: e.g. show only the prob. intervals
pred.pyr <- get.bPop.pyramid(pred, country="Ecuador", year=2090, age=1:27)
pred.pyr$pyramid <- NULL
plot(pred.pyr, show.birth.year = TRUE)
```

```
# pyramids for user-defined data
#################################
# this example dataset contains population estimates for the Washington state and King county
# (Seattle area) in 2011
data <- read.table(file.path(find.package("bayesPop"), "ex-data", "popestimates_WAKing.txt"),
    header=TRUE, row.names=1)
# extract data for two pyramids and put it into the right format
head(data)
WA <- data[,c("WA.male", "WA.female")]; colnames(WA) <- c("male", "female")
King <- data[,c("King.male", "King.female")]; colnames(King) <- c("male", "female")
# create and plot a bayesPop.pyramid object
pyramid <- get.bPop.pyramid(list(WA, King), legend=c("Washington", "King"))
plot(pyramid, main="Population in 2011", pyr2.par=list(height=0.7, col="violet", border="violet"))
# show data as proportions and include birth year
pyramid.prop <- get.bPop.pyramid(list(WA, King), is.proportion=NA,
    legend=c("Washington", "King"), pyr.year = 2011)
pop.pyramid(pyramid.prop, main="Population in 2011 (proportions)",
    pyr1.par=list(col="lightgreen", border="lightgreen", density=2),
    pyr2.par=list(col="darkred", border="darkred"),
    show.birth.year = TRUE)
```

---

pop.scale.prediction   *Population Scaling To Match Aggregated Targets*

Description

Functions scale given sex- and age-specific population counts for each location and year, so that the sum of the scaled counts over all locations matches the given target for each year, sex and age. Function scale.pop.prediction creates a new prediction object that contain the scaled trajectories. Function write.scaled.pop writes a file containing the scaled statistics for each location, year, sex and age.

Usage

```
pop.scale.prediction(pop.pred, target.file, output.dir,
    target.code = NULL, variant.name = "mean",
    target.id.column = "country_code", stat = "mean",
    exclude.codes = NULL, verbose = TRUE)


write.scaled.pop(pop.pred, target.file,
    output.file = "adjusted_population.txt", target.code = NULL,
    variant.name = "mean", target.id.column = "country_code",
    output.id.column = "reg_code", stat = "mean",
    exclude.codes = NULL, verbose = TRUE)
```

Arguments

pop.pred	Object of class [bayesPop.prediction](#) containing population projections to be scaled.
target.file	Comma-delimited file containing the summary statistics of the aggregated targets. For example, if locations in pop.pred are states, this file would contain the summary statistics for the corresponding country. It is expected to have the same format as a file created with [write.pop.projection.summary](#)(what = 'popsexage'). That is, it should have the following columns: country_code (unique identifier of the aggregated locations), sex (either "male" or "female"), age (values from 0 to 130), columns for each year or time period. Optionally, column variant can be present.
output.dir	Directory to write the scaled projection into.
target.code	If the target.file contains more than one aggregated location, this argument gives the code of the location to use.
variant.name	If the target.file contains the column variant, this argument gives the value of that column for selecting the summary statistics. If the target.file is created via the [write.pop.projection.summary](#) function, run it with the argument include.means = TRUE to generate a file with the "mean" variant, which is the default value here.
target.id.column	Name of the column in target.file that determines the unique identifiers of the aggregated locations.
stat	Summary statistics to use for the scaling. By default, the scaling shifts are determined by comparing the means of the trajectories with the targets.
exclude.codes	Vector of location codes that should be excluded from the scaling.
output.file	File name to write the scaled sumary statistics into.
output.id.column	Column name of the unique identifier of the locations in the output file.
verbose	Logical switching processing info on and off.

Details

These functions can be used if it is necessary to scale the population distribution so that a summary statistics of the sum of the locations matches some given targets. For example, if states should sum up to a given country projection. Thus, its use is meant to be mainly for subnational projections.

The scaling is done for each sex and age, and the scaling proportions are determined by the population sizes within each sex and age group.

The function scale.pop.prediction creates a new prediction object, saved into the directory given by output.dir. It contains adjusted population trajectories, as well as recomputed quantiles. Note however that only population quantities are affected by this scaling. There is no change to vital events or rates.

Function write.scaled.pop derives the necessary shifts for the scaling and stores the adjusted summary statistics into a file. It does not create any new projection object.

Value

Function `scale.pop.prediction` returns an object of class `bayesPop.prediction` containing the scaled predictions. Function `write.scaled.pop` returns the name of the file that it created.

Author(s)

Hana Sevcikova

---

pop.trajectories   *Accessing Trajectories*

Description

Obtain projection trajectories of population and vital events/rates. `get.pop` allows to access trajectories using a basic component of an expression. `get.pop.ex` and `get.pop.exba` returns results of an expression defined "by time" and "by age", respectively. `get.trajectory.indices` creates a link to the probabilistic components of the projection by providing indices to the trajectories of TFR, e0 and migration. `extract.trajectories.eq` returns trajectories (of population or expression) and their indices that are closest to given values or a quantile. Similarly, functions `extract.trajectories.ge` and `extract.trajectories.le` return trajectories and their indices that are greater equal and less equal, respectively, to the given values or a quantile.

Usage

```
pop.trajectories(pop.pred, country, sex = c("both", "male", "female"),
    age = "all", ...)

get.pop(object, pop.pred, aggregation = NULL, observed = FALSE, ...)

get.pop.ex(expression, pop.pred, observed = FALSE, as.dt = FALSE, ...)

get.pop.exba(expression, pop.pred, observed = FALSE, as.dt = FALSE, ...)

get.trajectory.indices(pop.pred, country,
    what = c("TFR", "e0M", "e0F", "migM", "migF"))

extract.trajectories.eq(pop.pred, country = NULL, expression = NULL,
    quant = 0.5, values = NULL, nr.traj = 1, ...)

extract.trajectories.ge(pop.pred, country = NULL, expression = NULL,
    quant = 0.5, values = NULL, all = TRUE, ...)

extract.trajectories.le(pop.pred, country = NULL, expression = NULL,
    quant = 0.5, values = NULL, all = TRUE, ...)
```

Arguments

pop.pred	Object of class [bayesPop.prediction](#).
country	Name or numerical code of a country.
sex	One of "both" (default), "male" or "female". By default the male and female projections are summed up.
age	Either a character string "all" (default) or an integer vector of age indices. In a 5x5 simulation, value 1 corresponds to age 0-4, value 2 corresponds to age 5-9 etc. Last age goup 130+ corresponds to index 27. In a 1x1 simulation, value 1 corresponds to age 0, value 2 to age 1 etc, up to 131 corresponding to the last age group. Results is summed over the given age categories.
object	Character string giving a basic component of an expression (see [pop.expressions](#)).
aggregation	If the basic component is to be evaluated on an aggregated prediction object, this argument gives the name of the aggregation (corresponds argument name in [pop.aggregate](#)). By default, the function searches for available aggregations and gives priority to the one called "country".
observed	Logical. Determines if the evaluation uses observed data (TRUE) or predictions (FALSE).
expression	Expression defining the trajectories measure. For syntax see [pop.expressions](#). It must be define by age (i.e. contain curly braces) if used in get.pop.exba, and the opposite applies to get.pop.ex.
as.dt	Logical indicating if the result should be returned as a data.table object in long format. This can be useful especially if results for all countries are requested.
what	A character string that defines to which component should the indices link to. Allowable options are "TFR", "e0M" (male life expectancy), "e0F" (female life expectancy), "migM" (male migration), "migF" (female migration).
quant	Quantile used to select the closest trajectories to.
values	Vector of values used to select the closest trajectories to. If it is not of length 1, it has to be of the same length as the number of projected time periods. If it is not given, quant is used.
nr.traj	Number of trajectories to return. This argument can be passed to any of the functions that contains . . . .
all	Logical indicating if the corresponding condition should apply to all time periods of a trajectory. If it is FALSE, a trajectory is extracted if the condition is fulfilled in at least one time period.
...	Additional argument passed to the underlying functions. In case of get.pop, get.pop.ex and get.pop.exba, this is only used for observed=FALSE. It can be either nr.traj giving the number of trajectories or logical typical.trajectory.

Details

Function pop.trajectories returns an array of population trajectories for given sex and age.

Function get.pop evaluates a basic component of an expression and results in a four-dimensional array. Internally, this function is used for evaluation after an expression is decomposed into basic

components. It can be useful for example for debugging purposes, to obtain results from parts of an expression. In addition, while pop.trajectories works only for population counts, get.pop can be used for obtaining trajectories of vital events and rates. Note that the wildcard "XXX" in the expression cannot be used in get.pop; use get.pop.ex or get.pop.exba instead.

Functions get.pop.ex and get.pop.exba evaluate a whole expression and the dimensions of the resulting array is collapsed depending on the specific expression. Use get.pop.ex if the expected result of the expression does not contain the age dimension, i.e. it uses no brackets or square brackets. If it is not the case, i.e. the expression is defined using curly braces in order to include the age dimension, the get.pop.exba function is to be used. Argument nr.traj can be used to restrict the number of trajectories returned. Use one of those functions if results for all countries (i.e. if using "XXX") is desired.

Function get.trajectory.indices returns an array of indices that link back to the given probabilistic component. It is of the same length as number of trajectories in the prediction object. For example, an array of c(10, 15, 20) (for a prediction with three trajectories) obtained with what="TFR" means that the 1st, 2nd and 3rd population trajectory, respectively, were generated with the 10th, 15th and 20th TFR trajectory, respectively. If the input TFR and e0 were generated using bayesTFR and bayesLife, functions get.tfr.trajectories and get.e0.trajectories can be used to extract the corresponding TFR and e0 trajectories.

Function extract.trajectories.eq can be used to select a given number of trajectories of any population quantity, including vital events, that are close to either specific values or to a given quantile. For example the default seting with quant=0.5 and nr.traj=1 returns the one trajectory that is "closest" to the median projection. As a measure of "closeness" the sum of absolute differences (across all time periods) is used.

Similarly, function extract.trajectories.ge (extract.trajectories.le) selects all trajectories that are greater (less) equal to the specific values or a given quantile. The argument all specifies, if the greater/less condition should be valid for all time periods of the selected trajectories or at least one time period.

Value

Function pop.trajectories returns a two-dimensional array (time x trajectory).

Function get.pop returns an array of four dimensions (country x age x time x trajectory). See pop.expressions for more details.

Functions get.pop.ex and get.pop.exba return an array of trajectories. Its dimensions depend on the expression and whether it is evaluated on observed data or projections. If as.dt is TRUE these functions return data.table objects in long format.

Function get.trajectory.indices returns a 1-d array of indices. If the given component is deterministic, it returns NULL.

Functions extract.trajectories.eq, extract.trajectories.ge, extract.trajectories.le return a list with two components. trajectories: 2-d array of trajectories; index: indices of the selected trajectories relative to the whole set of available trajectories.

Author(s)

Hana Sevcikova

See Also

pop.expressions

Examples

```
sim.dir <- file.path(find.package("bayesPop"), "ex-data", "Pop")
pred <- get.pop.prediction(sim.dir, write.to.cache=FALSE)

# observed female of Netherlands by age; 1x21x15x1 array
popFNL <- get.pop("PNL_F{}", pred, observed=TRUE)

# observed population for all countries in the prediction object,
# here 2 countries; 2x1x15x1 array
popAll <- get.pop("PXXX", pred, observed=TRUE)

# future migration for all countries in the prediction object,
# here 2 countries; 2x17 array
migAll <- get.pop.ex("GXXX", pred)

# projection population for Ecuador with 3 trajectories;
# 1x1x17x3 array
popEcu <- get.pop("P218", pred, observed=FALSE)

# the above is equivalent to
popEcu2 <- pop.trajectories(pred, "Ecuador")

# Expression "PNL_F{} / PNL_M{}" evaluated on projections
# is internally replaced by
FtoM <- get.pop("PNL_F{}", pred) / get.pop("PNL_M{}", pred)
# should return the same result as
FtoMa <- get.pop.exba("PNL_F{} / PNL_M{}", pred)

# the same expression by time (summed over ages)
FtoMt <- get.pop.ex("PNL_F / PNL_M", pred)

# the example simulation was generated with 3 TFR trajectories ...
get.trajectory.indices(pred, "Netherlands", what="TFR")
# ... and 1 e0 trajectory
get.trajectory.indices(pred, "Netherlands", what="e0M")

# The three trajectories of the population ratio of Ecuador to Netherlands
get.pop.ex("PEC/PNL", pred)
# Returns the trajectory closest to the upper 80% bound, including the corresponding index
extract.trajectories.eq(pred, expression="PEC/PNL", quant=0.9)
# Returns the median trajectory and the high variant, including the corresponding index
extract.trajectories.ge(pred, expression="PEC/PNL", quant=0.45)
```

---

pop.trajectories.plot   *Output of Probabilistic Population Projection*

Description

The functions plot and tabulate the distribution of population projection for a given country, or for all countries, including the median and given probability intervals.

Usage

```
pop.trajectories.plot(pop.pred, country = NULL, expression = NULL, pi = c(80, 95),
    sex = c("both", "male", "female"), age = "all", sum.over.ages = TRUE,
    half.child.variant = FALSE, nr.traj = NULL, typical.trajectory = FALSE,
    main = NULL, dev.ncol = 5, lwd = c(2, 2, 2, 2, 1),
    col = c("black", "red", "red", "blue", "#00000020"), show.legend = TRUE,
    ann = par("ann"), xshift = 0, ...)

pop.trajectories.plotAll(pop.pred,
    output.dir=file.path(getwd(), "pop.trajectories"),
    output.type="png", expression = NULL, verbose=FALSE, ...)

pop.trajectories.table(pop.pred, country = NULL, expression = NULL, pi = c(80, 95),
    sex = c("both", "male", "female"), age = "all", half.child.variant = FALSE,
    xshift = 0, ...)

pop.byage.plot(pop.pred, country = NULL, year = NULL, expression = NULL,
    pi = c(80, 95), sex = c("both", "male", "female"),
    half.child.variant = FALSE, nr.traj = NULL, typical.trajectory=FALSE,
    xlab = "", ylab = "Population projection",
    lwd = c(2,2,2,1), col = c("red", "red", "blue", "#00000020"),
    show.legend = TRUE, add = FALSE, type = "l", pch = NA, ...)

pop.byage.plotAll(pop.pred,
    output.dir=file.path(getwd(), "pop.byage"),
    output.type="png", expression = NULL, verbose=FALSE, ...)

pop.byage.table(pop.pred, country = NULL, year = NULL, expression = NULL,
    pi = c(80, 95), sex = c("both", "male", "female"),
    half.child.variant = FALSE, ...)
```

Arguments

pop.pred	Object of class [bayesPop.prediction].
country	Name or numerical code of a country. It can also be given as ISO-2 or ISO-3 characters.
expression	Expression defining the population measure to be plotted. For syntax see [pop.expressions]. For pop.trajectories.plot, pop.trajectories.table, pop.byage.plot and pop.byage.table the basic components of the expression must be country-specific. For pop.trajectories.plotAll and pop.byage.plotAll the country part should be given as "XXX". In addition, expressions passed into pop.byage.plot and pop.byage.table must contain curly braces (i.e. be age specific).
pi	Probability interval. It can be a single number or an array.

sex	One of "both" (default), "male" or "female". By default the male and female projections are summed up.
age	Either a character string "all" (default) or an integer vector of age indices. In a five year simulation, value 1 corresponds to age 0-4, value 2 corresponds to age 5-9 etc. Last age goup 130+ corresponds to index 27. In an annual simulation, the age indices 1, 2, 3, ..., 131 corrrespond to ages 0, 1, 2, ..., 130+.
sum.over.ages	Logical. If TRUE, the values are summed up over given age groups. Otherwise there is a separate plot for each age group.
half.child.variant	
	Logical. If TRUE the United Nations "+/-0.5 child" variant computed with fertility $+/-0.5*$ TFR median and the median of life expectancy is shown.
nr.traj	Number of trajectories to be plotted. If NULL, all trajectories are plotted, otherwise they are thinned evenly.
typical.trajectory	
	Logical. If TRUE one trajectory is shown that has the smallest distance to the median.
xlab, ylab, main, ann	
	Graphical parameters passed to the plot function.
xshift	Constant added to the x-axis (year).
dev.ncol	Number of column for the graphics device if sum.over.ages is FALSE. If the number of age groups is smaller than dev.ncol, the number of columns is automatically decreased.
lwd, col	For the first three functions it is a vector of five elements giving the line width and color for: 1. observed data, 2. median, 3. quantiles, 4. half-child variant, 5. trajectories. For functions that show results by age it is a vector of four elements - as above without the first item (observed data).
type, pch	Currently works for plotting by age only. It is a vector of four elements giving the plot type and point type for: 1. median, 2. quantiles, 3. half-child variant, 4. trajectories. The last element of the array is recycled.
show.legend	Logical controlling whether the legend should be drawn.
...	Additional graphical arguments. Functions pop.trajectories.plotAll and pop.byage.plotAll accept also any arguments of pop.trajectories.plot and pop.byage.plot, respectively, except country.
output.dir	Directory into which resulting graphs are stored.
output.type	Type of the resulting files. It can be "png", "pdf", "jpeg", "bmp", "tiff", or "postscript".
verbose	Logical switching log messages on and off.
year	Any year within the time period to be outputted.
add	Logical specifying if the plot should be added to an existing graphics.

Details

pop.trajectories.plot plots trajectories of population projection by time for a given country. pop.trajectories.table gives the same output as a table. pop.trajectories.plotAll creates

a set of graphs (one per country) that are stored in `output.dir`. The projections can be visualized separately for each sex and age groups, or summed up over both sexes and/or given age groups. This is controlled by the arguments `sex`, `age` and `sum.over.ages`.

`pop.byage.plot` and `pop.byage.table` plots/tabulate the posterior distribution by age for a given country and time period. `pop.byage.plotAll` creates such plots for all countries.

The median and given probability intervals are computed using all available trajectories. Thus, `nr.traj` does not influence those values - it is used only to control the number of trajectories plotted.

If plotting results of an expression and the function fails, to debug obtain values of that expression using the functions `get.pop.ex` (for pop.trajectories.plot) and `get.pop.exba` (for pop.byage.plot).

Author(s)

Hana Sevcikova

See Also

`bayesPop.prediction`, `summary.bayesPop.prediction`, `pop.pyramid`, `pop.expressions`, `get.pop`

Examples

```
sim.dir <- file.path(find.package("bayesPop"), "ex-data", "Pop")
pred <- get.pop.prediction(sim.dir)
pop.trajectories.plot(pred, country="Ecuador", pi=c(80, 95))
pop.trajectories.table(pred, country="ECU", pi=c(80, 95))
# female population of Ecuador in child bearing ages (by time)
pop.trajectories.plot(pred, expression="PEC_F[4:10]")
# Population by age in Netherands for two different years
pop.byage.plot(pred, country="Netherlands", year=2050)
pop.byage.plot(pred, expression="PNL{}", year=2000)
```

---

project.pasfr   *Projections of Percent Age-Specific Fertily Rate*

Description

The projections of percent age-specific fertility rate (PASFR) is normally computed within the `pop.predict` function for each trajectory. This function allows to project PASFR outside of population projections for the median total fertility rate (TFR) or user-provided TFR, and export it.

Usage

```
project.pasfr(inputs = NULL, present.year = 2020, end.year = 2100,
    wpp.year = 2019, annual = FALSE, nr.est.points = if(annual) 15 else 3,
    digits = 2, out.file.name = "percentASFR.txt", verbose = FALSE)

project.pasfr.traj(inputs = NULL, countries = NULL, nr.traj = NULL,
```

```
            present.year = 2020, end.year = 2100, wpp.year = 2019,
            annual = FALSE, nr.est.points = if(annual) 15 else 3,
            digits = 2, out.file.name = "percentASFRtraj.txt", verbose = FALSE)
```

Arguments

inputs	List of input data (file names) with the same meaning as in [pop.predict](). The relevant items here are: either tfr.file or tfr.sim.dir (TFR estimates and projections), pasfr (PASFR for observed time periods), and patterns (PASFR patterns). All entries are optional. By default the data is taken from the corresponding wpp package. See Details below.
present.year	Year of the last observed data point.
end.year	End year of the projection.
wpp.year	Year for which WPP data is used if one of the inputs components is left out.
annual	Logical that should be TRUE if the provided data on TFR and PASFR are annual-based data.
nr.est.points	Number of time points to be used for estimating the continuation of the observed PASFR trend. By default it is 15 years, corresponding to three time points for 5-year data.
digits	Number of decimal places in the results.
out.file.name	Name of the resulting file. If NULL nothing is written.
verbose	Logical switching verbose messages on and off.
countries	Vector of numerical country codes. By default the function is applied to all countries.
nr.traj	Number of trajectories on which the function should be applied. By default all trajectories are taken. Otherwise they are thinned appropriately.

Details

If the input TFR is given as an ASCII file (in inputs$tfr.file), it can be either a csv (comma-separated) file in long format, with columns "LocID", "Year", "Trajectory" and "TF". Or, it can be a tab-separated (wide format) file with column "country_code" and each year or time period as a separate column (see [tfr]()). In the latter case, an additional inputs entry tfr.file.type = "w" must be provided to specify the file is in the wide format, which is a case whe there is only one trajectory. Note that the TFR input should cover all projection time period as well as observed TFR as the function assesses the start of Phase III, which could be in the past.

If observed PASFR is given (in inputs$pasfr), it is a tab-separated file in wide format as in [percentASFR](). Fertility age patterns can be controlled by country via the inputs$patterns entry, which is a dataset in the same format and meaning as [vwBaseYear]().

In addition, if the present year differs by country, the inputs list accepts the entry last.observed, which is a tab-separated file with columns "country_code" and "last.observed". It can contain the year of the last observed time period for each country.

In the project.pasfr function, if the TFR input (given either as a long file or as a simulation directory), contains more than one trajectory, the median is derived over the trajectories for each time

period. Then, PASFR corresponding to this median is projected using the method from Sevcikova et al (2016).

For `project.pasfr.traj`, the PASFR is projected for single trajectories of TFR.

Value

Returns invisible data frame with the projected PASFR.

Author(s)

Hana Sevcikova, Igor Ribeiro

References

H. Sevcikova, N. Li, V. Kantorova, P. Gerland and A. E. Raftery (2016). Age-Specific Mortality and Fertility Rates for Probabilistic Population Projections. In: Dynamic Demographic Analysis, ed. Schoen R. (Springer), pp. 285-310. Earlier version in arXiv:1503.05215.

See Also

pop.predict

Examples

```
# using TFR in simulation directory
inputs <- list(tfr.sim.dir=file.path(find.package("bayesTFR"), "ex-data", "bayesTFR.output"))
pasfr <- project.pasfr(inputs, out.file.name = NULL)
head(pasfr)

## Not run:
pasfr.traj <- project.pasfr.traj(inputs, out.file.name = NULL)
head(pasfr.traj)
## End(Not run)

# using TFR in wide-format file
inputs2 <- list(tfr.file = file.path(find.package("wpp2019"), "data", "tfrprojMed.txt"),
    tfr.file.type = "w")
pasfr2 <- project.pasfr(inputs2, out.file.name = NULL)
head(pasfr2)
```

---

rcFDM   *Datasets on Inflow and Outflow Migration Schedules for FDM Method*

Description

Age-specific schedules of the inflow and outflow migration distribution used as input for the FDM method. `rc1FDM` corresponds to 1-year ages, while `rc5FDM` corresponds to 5-year age groups.

Usage

```
data(rc1FDM)
data(rc5FDM)
```

Format

A data frame where countries and ages are rows. It has four columns:

country_code Numerical Location Code (3-digit codes following ISO 3166-1 numeric standard) - see https://en.wikipedia.org/wiki/ISO_3166-1_numeric.

age Either single ages from 0 to 100 (rc1FDM) or 5-year age groups, such as "0-4", "5-9", ..., "100+" (rc5FDM).

Details

These datasets are used as the default datasets in pop.predict if mig.age.method is either "fdmp" or "fdmnop" and the inputs item "mig.fdm" is not given. Other default parameters of the FDM method are read from the vwBaseYear dataset.

Source

Most of the values were provided by the United Nations Population Division.

References

H. Sevcikova, J. Raymer J., A. E. Raftery (2024). Forecasting Net Migration By Age: The Flow-Difference Approach. arXiv:2411.09878.

See Also

vwBaseYear

Examples

```
data(rc1FDM)
head(rc1FDM)
```

---

summary.bayesPop.prediction

Summary of Probabilistic Population Projection

Description

Summary of an object bayesPop.prediction created using the pop.predict function. The summary contains the mean, standard deviation and several commonly used quantiles of the simulated trajectories.

Usage

```
## S3 method for class 'bayesPop.prediction'
summary(object, country = NULL,
    sex = c("both", "male", "female"), compact = TRUE, ...)
```

Arguments

object	Object of class `bayesPop.prediction`.
country	Country name or code. It can also be given as ISO-2 or ISO-3 characters. If it is NULL, only meta information included.
sex	One of "both" (default), "male", or "female". If it is not "both", the summary is given for sex-specific trajectories.
compact	Logical switching between a smaller and larger number of displayed quantiles.
...	A list of further arguments.

Author(s)

Hana Sevcikova

See Also

`bayesPop.prediction`

Examples

```
sim.dir <- file.path(find.package("bayesPop"), "ex-data", "Pop")
pred <- get.pop.prediction(sim.dir)
summary(pred, "Netherlands")
```

---

vwBaseYear	*Datasets on Migration Base Year and Type, and Mortality and Fertility Age Patterns*

Description

Datasets giving information on the baseyear and type of migration for each country. The 2012, 2015, 2017, 2019, 2022 and 2024 datasets also give information on country's specifics regarding mortality, fertility and migration age patterns.

Usage

```
data(vwBaseYear2024)
data(vwBaseYear2022)
data(vwBaseYear2019)
data(vwBaseYear2017)
data(vwBaseYear2015)
data(vwBaseYear2012)
data(vwBaseYear2010)
```

Format

A data frame containing the following variables:

country_code Numerical Location Code (3-digit codes following ISO 3166-1 numeric standard) - see https://en.wikipedia.org/wiki/ISO_3166-1_numeric.

country Country name. Not used by the package.

isSmall UN internal code. Not used by the package.

ProjFirstYear The base year of migration.

MigCode Type of migration. Zero means migration is evenly distributed over each time interval. Code 9 means migration is captured at the end of each interval.

WPPAIDS Dummy indicating if the country has generalized HIV/AIDS epidemics.

AgeMortalityType Type of mortality age pattern. Only relevant for countries with the entry "Model life tables". In such a case, the b_x Lee-Carter parameter is not estimated from historical data. Instead is taken from the dataset MLTbx using a pattern given in the AgeMortalityPattern column.

AgeMortalityPattern If AgeMortalityType is equal to "Model life tables", this value determines which b_x is selected from the MLTbx dataset. It must sorrespond to one of the rownames of MLTbx, e.g. "CD East", "CD West", "UN Latin American".

AgeMortProjMethod1 Method for projecting age-specific mortality rates. It is one of "LC" (modified Lee-Carter, uses function mortcast), "PMD" (pattern mortality decline, uses function copmd), "modPMD" (modified pattern mortality decline, uses function copmd(... use.modpmd = TRUE)), "MLT" (model life tables, uses function mlt), "LogQuad" (log quadratic method, uses function logquad), or "HIVmortmod" (HIV model life tables as implemented in the **HIV.LifeTables** package which can be installed from the PPgP/HIV.LifeTables GitHub repo).

AgeMortProjMethod2 If the mortality rates are to be projected via a blend of two methods (see mortcast.blend), this column determines the second method. The options are the same as in the column AgeMortProjMethod1.

AgeMortProjPattern If one of the AgeMortProjMethodX colums contains the "MLT" method, this column determines the type of the life table (see the argument type in the mlt function).

AgeMortProjMethodWeights If the mortality rates are to be projected via a blend of two methods, this column determines the weights in the first and the last year of the projection, respectively. It should be given as an R vector, e.g. "c(1, 0.5)" (see the argument weights in mortcast.blend).

AgeMortProjAdjSR Code determining how the "PMD" method should be adjusted if it's used. 0 means no adjustment, 1 means the argument sexratio.adjust in copmd is set to TRUE, and code 3 means that the argument adjust.sr.if.needed in copmd is set to TRUE.

LatestAgeMortalityPattern, LatestAgeMortalityPattern1 Indicator n for how many latest time periods of historical mortality rates should be averaged to compute the a_x Lee-Carter and modPMD parameter. If n is zero, all time periods are used. If n is one, only the latest time period is used. If n is negative, the latest n time periods are excluded. This can have also a form of a vector where the first element is either a negative or a zero. If it is negative, the vector must have only two elements. In such a case, the first element (must be negative) determines how many latest time periods should be excluded, while the second element (must be positive) determines how many latest time periods to include after the exclusion. If the vector starts with

a zero, the following numbers are interpreted as individual indices to the time periods starting from the latest time point. Here are a few examples, assuming the available mortality rates are on annual scale, from 1950 to 2023:

"0": using all years from 1950 to 2023

"3": using 2023, 2022, 2021

"-3": using 1950 - 2020

"c(-2, 3)": 2023 and 2022 are excluded; using 2021, 2020, 2019

"c(-2, 1, 3)": invalid specification - must have two elements if it starts with a negative

"c(0, 3)": interpreted as an individual index; thus, using 2021 only

"c(0, 1, 3, 4)": interpreted as individual indices; using 2023, 2021, 2020

If the LatestAgeMortalityPattern1 column is present, it should contain values related to an annual simulation (1x1) while the LatestAgeMortalityPattern column relates to a 5x5 simulation.

SmoothLatestAgeMortalityPattern If LatestAgeMortalityPattern is not zero, this column indicates if the a_x should be smoothed.

SmoothDFLatestAgeMortalityPattern, SmoothDFLatestAgeMortalityPattern1 Degree of free-dom for smoothing a_x. By default (value 0) a half of the number of age groups is taken. If the SmoothDFLatestAgeMortalityPattern1 column is present, it should contain values related to a 1x1 simulation while the SmoothDFLatestAgeMortalityPattern column relates to a 5x5 simulation.

PasfrNorm Type of norm for computing age-specific fertility pattern to which the country belongs to. Currently only "GlobalNorm" is used.

PasfrGlobalNorm, PasfrFarEastAsianNorm, PasfrSouthAsianNorm Dummies indicating which country to include to compute the specific norms.

MigFDMb0, MigFDMb1, MigFDMmin, MigFDMsrin, MigFDMsrout Available in the 2024 dataset. These are parameters of the Flow Difference Method to generate age-specific net migration patterns (Sevcikova et. al, 2024). They correspond to the intercept, slope, minimum flow rate, female sex ratio for the in-flow and out-flow, respectively.

Details

There is one record for each country. See Sevcikova et al (2016) on how information from the various columns is used for projections.

Source

Data provided by the United Nations Population Division.

References

H. Sevcikova, N. Li, V. Kantorova, P. Gerland and A. E. Raftery (2016). Age-Specific Mortality and Fertility Rates for Probabilistic Population Projections. In: Dynamic Demographic Analysis, ed. Schoen R. (Springer), pp. 285-310. Earlier version in arXiv:1503.05215.

H. Sevcikova, J. Raymer J., A. E. Raftery (2024). Forecasting Net Migration By Age: The Flow-Difference Approach. arXiv:2411.09878.

Examples

```
data(vwBaseYear2019)
str(vwBaseYear2019)
```

---

write.pop.projection.summary

Writing Projection Summary and Trajectory Files

Description

Functions for creating ASCII files containing projection summaries, such as the median, the lower and upper bound of the 80 and 95% probability intervals, respectively, as well as containing individual trajectories.

Usage

```
write.pop.projection.summary(pop.pred, what = NULL, expression = NULL,
    output.dir = NULL, ...)

write.pop.trajectories(pop.pred, expression = "PXXX",
    output.file = "pop_trajectories.csv", byage = FALSE,
    observed = FALSE,  wide = FALSE, digits = NULL,
    include.name = FALSE, sep = ",", na.rm = TRUE, ...)
```

Arguments

pop.pred	Object of class bayesPop.prediction.
what	A character vector specifying what kind of projection to write. Total population is specified by "pop". Vital events are specified by "births", "deaths", "sr" (survival rate), "fertility" and "pfertility" (percent fertility). Each of these strings can (some must) have a suffix "sex" and/or "age" if sex- and/or age-specific measure is desired. For example, "popage", "birthssexage", "deaths", "deathssex", are all valid values. Note that for survival, only "srsexage" is allowed. For percent fertility, only "pfertilityage" is allowed. Suffix "sex" cannot be used in combination with "fertility". Moreover, "fertility" (without age) corresponds to the total fertility rate. If the argument is NULL, all valid combinations are used. The argument is not used if expression is given. Note that vital events can be only used if the prediction object contains vital events, i.e. if it was generated with the keep.vital.events argument being TRUE (see pop.predict).
expression	Expression defining the measure to be written. If it is not NULL, argument what is ignored. For expression syntax see pop.expressions. The country components of the expression should be given as "XXX".
output.dir	Directory in which the resulting files will be stored. If NULL pop.pred$output.directory is used.
output.file	File name to write the trajectories into.

byage	Logical indicating if the expression is defined by age, i.e. if it includes curly braces (TRUE), of if it is defined by time (FALSE), see pop.expressions for more detail on the expression syntax.
observed	Logical indicating if observed data should be written (TRUE) or projected trajectories (FALSE).
wide	Logical indicating if the data format should be wide. By default, trajectories are written in long format.
digits	To how many decimal digits should the indicator be rounded. By default no rounding takes place.
include.name	Logical indicating if country names should be included in the dataset.
sep	The field separator string.
na.rm	Logical indicating if records with NA values should be included in the dataset.
...	For write.pop.projection.summary, these are:

- character file.suffix defines the file suffix;
- if expression is given, then one can use here expression.label which defaults to the actual expression and is put as the first line in the resulting file;
- logical include.observed determines if observed data should be included;
- integer digits defines the number of decimal places in the resulting file;
- for 5-year projections, logical end.time.only determines if the time columns should be in form of time periods (as XXXX-YYYY) or just the end years (YYYY);
- integer vector locations (location identifiers) limits the output to those locations only (used only if expression is not given);
- logical include.means determines if distribution means should be included in the output (defaults to FALSE; used only if expression is not given);
- logical adjust determines if the numbers should be adjusted; in such a case, adj.to.file and allow.negative.adj give the file name to which to adjust and a switch if negatives are allowed for the adjustments, respectively.

For write.pop.trajectories, these are arguments passed to [get.pop.ex](get.pop.ex) (if byage is FALSE) or get.pop.exba (if byage is TRUE).

Details

The write.pop.projection.summary function creates one file per value of what, or expression, called 'projection_summary_'*suffix*'.csv', where *suffix* is either what or, if an expression is given, the value of file.suffix. It is a comma-separated table with the following columns:

- "country_name": country name
- "country_code": country code
- "variant": name of the variant, such as "median", "lower 80", "upper 80", "lower 95", "upper 95", optionally "mean"
- period1: e.g. "2005-2010", or "2010": Given population measure for the first time period

- period2: e.g. "2010-2015", or "2015": Given population measure for the second time period

- . . . further time period columns

If `expression` is given, `expression.label` (by default the full expression) is written as the first line of the file starting with #. The file contains one line per country, and possibly sex and age.

Function `write.pop.trajectories` writes out all trajectories, either in long format (default) or, if `wide = TRUE` in wide format (years become columns).

Note

If the `expression` argument is used, the same applies as for [pop.map](#) in terms of Performance and Caching.

Author(s)

Hana Sevcikova

See Also

[pop.predict](#), [pop.map](#), [pop.expressions](#)

Examples

```
outdir <- tempfile()
dir.create(outdir)
sim.dir <- file.path(find.package("bayesPop"), "ex-data", "Pop")
pred <- get.pop.prediction(sim.dir=sim.dir, write.to.cache=FALSE)

# proportion of 65+ years old to the whole population
write.pop.projection.summary(pred, expression="PXXX[14:27] / PXXX", file.suffix="age65plus",
    output.dir=outdir, include.observed=TRUE, digits=2)

# various measures
write.pop.projection.summary(pred, what=c("pop", "popsexage", "popsex"),
    output.dir = outdir, include.means = TRUE)

unlink(outdir, recursive=TRUE)
```

Index