Package 'propop'

November 18, 2026

Title Project Population Growth in Switzerland Using the Cohort Component Method

Version 2.2.2

Description Project the development of the population at different spatial levels (e.g., cantons, districts, municipalities) using the cohort component method, broken down by 101 one-year age classes, gender, and nationality (see e.g. Preston et al. (2000, ISBN: 978-1-557-86451-2)).
Includes helper functions to download required population data and projection parameters as open data for Switzerland, plus tools for model evaluation against benchmarks such as population records or a base model.

URL https://statistik-aargau.github.io/propop/, https://github.com/statistik-aargau/propop

BugReports https://github.com/statistik-aargau/propop/issues

License GPL (>= 3)

Depends R (>= 4.1.0)

Imports assertthat, BFS, cli, curl, dplyr, stringi, tidyr, vctrs, progressr, purrr

Suggests devtools, DT, ggplot2, htmltools, tibble, knitr, rmarkdown, stringr, testthat, rlang, lifecycle, colorspace

VignetteBuilder knitr

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Config/testthat/edition 3

NeedsCompilation no

Author Norah Efosa [aut, cre],
Adrian Gadient [aut] (ORCID: <https://orcid.org/0000-0003-0026-3153>),
Tina Cornioley [aut],
Jan Wunder [aut],
Niklas Haffert [aut],

Andrea Plüss [ctb],
Nadine Herrmann [ctb],
Lutz Benson [ctb],
Eric Meyer [ctb],
Thomas Knecht [ctb],
Moana Stadelmann [ctb],
Statistik Aargau [fnd, cph]

Maintainer Norah Efosa <norah.efosa@ag.ch>

Repository CRAN

Date/Publication 2026-11-18 07:30:02 UTC

Contents

aggregate_measures 2
ag_migration_subregional 4
ag_population_subregional 4
calculate_rates 5
calculate_shares 6
check_balance 7
compute_measures 8
fso_parameters 10
fso_population 11
get_parameters 12
get_population 15
prepare_evaluation 16
propop 18

Index **22**

`aggregate_measures` *Aggregate evaluation measures*

Description

Returns descriptive summary statistics of model accuracy and bias measures across demographic groups and years. The returned statistics are particularly useful for comparing the model performance for different groups or different models.

Usage

```
aggregate_measures(data, weight_groups = NULL)
```

Arguments

`data`	data frame created with function `compute_measures`.
`weight_groups`	character, optional argument indicating one or more column names to obtain evaluation criteria weighted for specific groups (e.g., age groups, nationality).

Value

A data frame. The data frame includes the following summary measures:

- `mpe` is the mean percentage error (**mpe**; or mean algebraic percentage error **malpe**); it is a bias indicator as it takes the **direction** of the error into account. Positive values indicate that the projections were, overall, too high. Negative values indicate that the projections were, overall, too low. The closer the value is too zero, the lower the bias.
- `medpe` is the median (or middle value) of the percentage error (**medpe**). Particularly useful for small samples or skewed distributions. The closer the value is too zero, the lower the bias.
- `mape` is the mean **absolute** percentage / proportional error (**mape**). It considers variance (or amplitude) and can be seen as a measure of precision. The smaller the value, the lower is the average error.
- `medape` is the median (or middle value) of the **absolute** percentage error (**medape**). Particularly useful for small samples or skewed distributions. The smaller the value, the lower is the average error.
- `rmse` is the root mean square error; it is an indication of the robustness or quality of the projection. The smaller the value, the more robust the projection.
- `wmape` is the **weighted** mean **absolute** percentage error (**wmape**); in contrast to `mape`, this measure weights each absolute percentage error according to the population size of the "focal" group (e.g., nationality, age group) and thus considers domain size. Put differently, errors count more in populous groups than in less populous groups. This measure is particularly useful when population sizes vary strongly. The closer the value, the more precise is the projection.
- `n_measure` is the number of times a summary measure occurs (per weight group if requested).
- `ape_under_1` is a measure to gauge the error distribution; it indicates the proportion of observations that have absolute percentage errors smaller than 1%.
- `ape_under_5` is a measure to gauge the error distribution; it indicates the proportion of observations that have absolute percentage errors smaller than 5%.

References

Baker, J., et al. (2015). Sub-county population estimates using administrative records: A municipal-level case study in New Mexico. In M. N. Hoque & L. B. Potter (Eds.), Emerging techniques in applied demography (pp. 63-79). Springer, doi:10.1007/9789401789905_6

Bérard-Chagnon, J. (2015) Using tax data to estimate the number of families and households in Canada. In M. N. Hoque & L. B. Potter (Eds.), Emerging techniques in applied demography (pp. 137-153). Springer, doi:10.1007/9789401789905_10

Reinhold M. & Thomsen, S. L. (2015) Subnational population projections by age: An evaluation of combined forecast techniques, Population Research and Policy Review, 34, 593-613, doi:10.1007/s1111301593620

Wilson, T. (2012). Forecast accuracy and uncertainty of Australian Bureau of Statistics state and territory population projections, International Journal of Population Research, 1, 419824, doi:10.1155/2012/419824

Wilson, T. (2016). Evaluation of alternative cohort-component models for local area population forecasts, Population Research and Policy Review, 35, 241-261, doi:10.1007/s111130159380y

Examples

```
evaluation <- tibble::tibble(
  year = rep(2020L, 3L),
  n_benchmark = c(2605, 2679, 2650),
  n_projected = c(2526, 2558, 2594),
  error = c(-79, -121, -56),
  pe = c(-3.032629558541267, -4.516610675625234, -2.1132),
  ape = c(3.03, 4.516610675625234, 2.1132075471698113),
)

aggregate_measures(evaluation)
```

---

ag_migration_subregional

Subregional migration records for the canton of Aargau (five subregions)

Description

Raw migration data from the canton of Aargau divided into five subregions.

Usage

```
ag_migration_subregional
```

Format

The example data includes historic migration records for immigration and emigration. Dimensions correspond to the number of people of each demographic group (nationality (2) X sex (2) X age classes (101)) for five spatial units of the canton of Aargau in 2025.

---

ag_population_subregional

Population data for the canton of Aargau (five subregions)

Description

Population from the canton of Aargau divided into five subregions.

Usage

```
ag_population_subregional
```

Format

The example population records include the number of people of each demographic group (nationality (2) X sex (2) X age classes (101)) for five spatial units of the canton of Aargau in 2025.

`calculate_rates` *Calculate emigration rates to subregions*

Description

The function calculates the mean of past emigration to other municipalities within the canton for all municipalities and all possible combinations of age group, sex, and nationality.

Usage

```
calculate_rates(
  past_migration,
  n_jan,
  births,
  emi_n,
  spatial_unit,
  method,
  year_range = NULL,
  age_group = 1,
  binational = TRUE,
  two_sex = TRUE
)
```

Arguments

`past_migration`	data frame, containing aggregated migration records. The data frame must include one row for each combination of year, spatial unit, nationality, sex, and age.
`n_jan`	character, column containing the starting population (typically the number of people per group in January).
`births`	character, column containing the number of births in the group of 0-year olds.
`emi_n`	character, column containing the total number of people emigrating per spatial unit and demographic group.
`spatial_unit`	character, column containing the spatial units.
`method`	character, method to calculate average shares, i.e. `mean` or `median`.
`year_range`	**(optional)** vector, years taken into consideration to calculate historical shares. Default uses all years present in the data.
`age_group`	**(optional)** integer, divides continuous age values into intervals for calculating shares. If the argument is not specified, the default uses 1-year age groups.
`binational`	**(optional)** boolean, `TRUE` indicates that the calculation discriminates between two groups of nationalities. `FALSE` indicates that the calculation does not distinguish between nationalities.
`two_sex`	**(optional)** boolean, `TRUE` indicates that the calculation discriminates between two sexes. `FALSE` indicates that the calculation does not distinguish between sexes.

Value

A data frame that includes the average emigration rate.

Examples

```
# Compute mean emigration rate
calculate_rates(
  past_migration = ag_migration_subregional,
  n_jan = n_jan,
  births = births,
  emi_n = hist_emi_sub_n,
  spatial_unit = spatial_unit,
  method = "mean",
  year_range = c(2022:2024),
  age_group = 5,
  binational = TRUE,
  two_sex = TRUE
)
```

---

calculate_shares *Calculate shares for distributing immigration among subregions*

Description

Calculates historical immigration shares across spatial units within a region.
To compute the shares, the counts are summed within each spatial unit and across all spatial units combined, across all years. Dividing the spatial unit count by the total gives the group's share.
These shares are used to allocate emigrants moving from one subregion to another subregion (which can be done with `calculate_rates`).

Usage

```
calculate_shares(
  past_migration,
  imm_n,
  year_range = NULL,
  age_group = 1,
  binational = TRUE,
  two_sex = TRUE
)
```

Arguments

`past_migration`	data frame, historical records (e.g., immigration from other cantons, countries or subregions). Required columns are: `year`, `spatial_unit`, `age` and a column that contains aggregated historical migration records. The columns `nat` and `sex` are optional.

imm_n	character, name of the column which contains the data for aggregated historical migration records.
year_range	**(optional)** vector, years taken into consideration to calculate historical shares. Default uses all years present in the data.
age_group	**(optional)** integer, divides continuous age values into intervals for calculating shares. If the argument is not specified, the default uses 1-year age groups.
binational	**(optional)** boolean, TRUE indicates that the calculation discriminates between two groups of nationalities. FALSE indicates that the calculation does not distinguish between nationalities.
two_sex	**(optional)** boolean, TRUE indicates that the calculation discriminates between two sexes. FALSE indicates that the calculation does not distinguish between sexes.

Value

A data frame that includes the average share per demographic group and spatial unit. imm_share can be used as imm_sub parameter when propop::propop() uses subregional = "rate".

See Also

propop() for details on how to account for subregional migration using the rate method, calculate_rates() for calculating the associated emigration rate emi_sub.

Examples

```
# Calculate shares to distribute subregional immigration among spatial units
calculate_shares(
  past_migration = ag_migration_subregional,
  imm_n = hist_imm_sub_n,
  year_range = c(2022:2024),
  age_group = 10,
  binational = TRUE,
  two_sex = TRUE
)
```

---

check_balance *Check if population equation components add up within one year*

Description

Function to check if the components in the propop::propop() output add up. Takes the population at the beginning of the year (n_jan), adds all obligatory components (births - mor_n - emi_int_n - emi_nat_n + imm_int_n + imm_nat_n) and optional components (acq_n, mig_sub, emi_sub_n, imm_sub_n) and checks if the sum is equal to the population at the end of the year (n_dec).

Usage

```
check_balance(.data)
```

Arguments

`.data`	data frame containing population projections; can be created with `propop::propop()`.

Value

data frame containing summary statistics.

Examples

```
propop(
  parameters = fso_parameters,
  year_first = 2026,
  year_last = 2027,
  population = fso_population,
  subregional = FALSE,
  binational = TRUE
) |>
  check_balance()
```

`compute_measures` *Compute evaluation measures*

Description

Uses the differences between the results from a projection and a benchmark to compute performance measures.

Usage

```
compute_measures(combined, weight_groups = NULL)
```

Arguments

`combined`	data frame created with `propop::prepare_evaluation()`.
`weight_groups`	character, optional argument indicating one or more column names to obtain evaluation criteria weighted for specific groups (e.g., age groups, nationality).

Details

The input is a data frame created with `propop::prepare_evaluation()`. It includes a benchmark (typically the observed population records, i.e., the number of people per spatial unit, demographic group, and year) and the corresponding projected number of people. The input can range from low resolution (e.g., total number of people per municipality) to high resolution (e.g., 101 age classes, nationality, sex).

For more details on usage, see `vignette("evaluate", package = "propop")`.

Value

A data frame. The following evaluation criteria can directly be interpreted and used for descriptive comparisons:

- `error` is the forecast error; it quantifies the level of under-projection (negative values) and over-projection (positive values) relative to the benchmark `n_benchmark`.
- `pe` is the percentage error and expresses the under- / over-projection in percent of the benchmark `n_benchmark`.
- `ape` is the absolute percentage error; it is the absolute deviation in percent of the benchmark `n_benchmark`, thus only showing the extent of the error but not the direction.
- `w_ape` is the weighted absolute percentage error; it weighs each absolute percentage error according to the population size of the focal group (e.g., nationality, age group). The weighted version is useful as an aggregated measure when groups vary strongly in terms of population size. Only returned when the argument `weight_groups` contains at least one grouping variable.

The following helper variables are used to compute aggregate measures. They are only returned when weight groups are provided via the argument `weight_groups`.

- `n_tot` is the total number of people (i.e., sum of the number of people in all demographic groups); used to compute the weighted absolute percentage error.
- `group_tot` is the number of people in the focal group; used to compute the weighted absolute percentage error.
- `weight` is the share of the (optional) focal group (e.g., municipality type / size, nationality, age group) relative to all people; used to compute the weighted absolute percentage error.

References

Baker, J., et al. (2015). Sub-county population estimates using administrative records: A municipal-level case study in New Mexico. In M. N. Hoque & L. B. Potter (Eds.), Emerging techniques in applied demography (pp. 63-79). Springer, doi:10.1007/9789401789905_6

Wilson, T. (2012). Forecast accuracy and uncertainty of Australian Bureau of Statistics state and territory population projections, International Journal of Population Research, 1, 419824, doi:10.1155/2012/419824

Wilson, T. (2016). Evaluation of alternative cohort-component models for local area population forecasts, Population Research and Policy Review, 35, 241-261, doi:10.1007/s111130159380y

Examples

```
combined <- tibble::tibble(
  year = 2020,
  spatial_unit = "Aargau",
  sex = "f",
  age = c("age_00_19", "age_20_64", "age_65_"),
  n_benchmark = c(50020, 17917, 53005),
  n_projected = c(49691, 18519, 52395),
)
```

```
compute_measures(combined)
```

`fso_parameters` *Sample parameters to run population projection*

Description

Data frame containing the rates and number of people from the Federal Statistical Office (FSO) required to project the development of four demographic groups for a selected canton (Aargau). The sample data include three scenarios for the years 2025-2055.

Usage

```
fso_parameters
```

Format

The example data include the required parameters for each demographic group (nationality (2) X sex (2) X age classes (101)) for the years 2025-2055.

Demographic groups

The returned data frame includes parameters for each unique combination of the following demographic variables:

- `nat`: ch = Swiss; int = foreign / international.
- `sex`: f = female, m = male.
- `age`: 101 one-year age classes, ranging from 0 to 100 (including those older than 100).

Parameters

The following parameters are included in the returned data frame:

- `year`: numeric, year of projection.
- `scen`: character, projection scenario.
- `spatial_unit`: character, indicating the user requested spatial unit(s).
- `birthrate`: numeric, number of births per mother by age, nationality and year.
- `int_mothers`: numeric, proportion of children with Swiss nationality born to non-Swiss mothers.
- `mor`: numeric, prospective mortality (probability of death).
- `emi_int`: numeric, rate of people emigrating to other countries.
- `emi_nat`: numeric, rate of people emigrating to other cantons.
- `acq`: numeric, rate of acquisition of Swiss citizenship.
- `imm_int_n`: numeric, number of people immigrating from abroad.
- `imm_nat_n`: numeric, number of people immigrating from other cantons.
- `emi_nat_n`: numeric, number of people emigrating to other cantons.
- `mig_nat_n`: numeric, national / inter-cantonal net migration.

Projected population

`n_projected` is the number of people per demographic group and year on December 31 (as projected by the FSO in the 2025 model).

Details about calculated variables

`births_int_ch` is calculated by dividing the number of live newborns with Swiss citizenship born to non-Swiss mothers by the number of all live newborns born to non-Swiss mothers.

`mig_ch` is calculated as the difference between the immigration from other cantons and the emigration to other cantons.

Source

Data obtained from the Swiss Federal Statistical Office (FSO):

- https://www.pxweb.bfs.admin.ch/pxweb/en/px-x-0104020000_101/-/px-x-0104020000_101.px/
- https://www.pxweb.bfs.admin.ch/pxweb/en/px-x-0104020000_102/-/px-x-0104020000_102.px
- https://www.pxweb.bfs.admin.ch/pxweb/en/px-x-0104020000_103/-/px-x-0104020000_103.px/
- https://www.pxweb.bfs.admin.ch/pxweb/en/px-x-0104020000_106/-/px-x-0104020000_106.px/
- https://www.pxweb.bfs.admin.ch/pxweb/en/px-x-0104020000_109/-/px-x-0104020000_109.px/

`fso_population` *Sample population data from the Federal Statistical Office*

Description

Example data frame containing the starting population required to project the development of four demographic groups for a selected canton (Aargau). Population data can be downloaded from the FSO using the `get_population`.

Usage

```
fso_population
```

Format

The example population records include the number of people of each demographic group (nationality (2) X sex (2) X age classes (101)) for the canton of Aargau in 2024.

Value

A data frame. For each of the four demographic groups (female / male, Swiss / foreign nationals), there are 101 age classes, resulting in a total of 404 rows per requested year and spatial unit. Columns included in the returned data frame:

year numeric, year in which the population was recorded.

spatial_unit character, indicating the spatial entities (e.g., cantons, districts, municipalities).

nat character, `ch` = Swiss, `int` = foreign / international.

sex character `f` = female, `m` = male.

age numeric, 101 one-year age classes, ranging from `0` to `100` (including those older than 100).

n numeric, number of people per year, spatial entity, and demographic group.

Source

Federal Statistical Office: https://www.pxweb.bfs.admin.ch/pxweb/en/px-x-0102010000_101/-/px-x-0102010000_101.px/

`get_parameters` *Get projection parameters from FSO*

Description

Users who do not have the mandatory projection parameters for `propop::propop()` can use this convenience function to download them from the Federal Statistical Office (FSO). The parameters are only available on the level of cantons. For smaller-scale projections, the parameters must be scaled down. In addition to the parameters, the function also returns the projected population (i.e., number of people estimated in the FSO model released in 2025). All parameters and projections are from the FSO model published in 2025. The parameters are available for the years 2024-2055. The variables `int_mothers` and `mig_nat_n` are not directly available from the FSO. They are calculated within this function.

To get projection parameters, you must use the spelling defined in the corresponding FSO table. See `vignette("prepare_data", package = "propop")`. Inspecting the column 'valueTexts' in the following package data may also help:

- data('stattab_101_snap')
- data('stattab_102_snap')
- data('stattab_103_snap')
- data('stattab_106_snap')
- data('stattab_109_snap')

Changes to the API interface may break this function. If problems occur, we recommend following the step-by-step procedure described in `vignette("prepare_data", package = "propop")`.

Usage

```
get_parameters(
  number_fso_ref = "px-x-0104020000_101",
  number_fso_high = "px-x-0104020000_102",
  number_fso_low = "px-x-0104020000_103",
  number_fso_rates = "px-x-0104020000_109",
  number_fso_births = "px-x-0104020000_106",
  year_first,
  year_last,
  spatial_units
)
```

Arguments

number_fso_ref	character, px-x table ID for number parameters (reference scenario), defaults to "px-x-0104020000_101".
number_fso_high	character, px-x table ID for number parameters (high growth scenario), defaults to "px-x-0104020000_102".
number_fso_low	character, px-x table ID for for number parameters (low growth scenario, defaults to "px-x-0104020000_103".
number_fso_rates	character, px-x table ID for rate parameters, defaults to "px-x-0104020000_109".
number_fso_births	character, px-x table ID required to compute the share of Swiss newborns from non-Swiss mothers, defaults to "px-x-0104020000_106".
year_first	numeric, first year for which the parameters and projections are to be downloaded.
year_last	numeric, last year for which the parameters and projections are to be downloaded.
spatial_units	character vector, indicating at least one spatial entity for which the projection will be run. Typically a canton.

Value

A data frame with the rates and number of people from the Federal Statistical Office (FSO) required to project the population development of the requested spatial entities. For each of the four demographic groups (nationality x sex), there are 101 age classes, resulting in a total of 404 rows per requested year and spatial unit.

Demographic groups

The returned data frame includes parameters for each unique combination of the following demographic variables:

- nat: ch = Swiss; int = foreign / international.
- sex: f = female, m = male.
- age: 101 one-year age classes, ranging from 0 to 100 (including those older than 100).

Parameters

The following parameters are included in the returned data frame:

- `year`: numeric, year of projection.
- `scen`: character, projection scenario.
- `spatial_unit`: character, indicating the user requested spatial unit(s).
- `birthrate`: numeric, number of births per mother by age, nationality and year.
- `int_mothers`: numeric, proportion of children with Swiss nationality born to non-Swiss mothers.
- `mor`: numeric, prospective mortality (probability of death).
- `emi_int`: numeric, rate of people emigrating to other countries.
- `emi_nat`: numeric, rate of people emigrating to other cantons.
- `acq`: numeric, rate of acquisition of Swiss citizenship.
- `imm_int_n`: numeric, number of people immigrating from abroad.
- `imm_nat_n`: numeric, number of people immigrating from other cantons.
- `emi_nat_n`: numeric, number of people emigrating to other cantons.
- `mig_nat_n`: numeric, national / inter-cantonal net migration.

Projected population

`n_projected` is the number of people per demographic group and year on December 31 (as projected by the FSO in the 2025 model).

Details about calculated variables

`births_int_ch` is calculated by dividing the number of live newborns with Swiss citizenship born to non-Swiss mothers by the number of all live newborns born to non-Swiss mothers.

`mig_ch` is calculated as the difference between the immigration from other cantons and the emigration to other cantons.

Source

Data obtained from the Swiss Federal Statistical Office (FSO):

- https://www.pxweb.bfs.admin.ch/pxweb/en/px-x-0104020000_101/-/px-x-0104020000_101.px/
- https://www.pxweb.bfs.admin.ch/pxweb/en/px-x-0104020000_102/-/px-x-0104020000_102.px
- https://www.pxweb.bfs.admin.ch/pxweb/en/px-x-0104020000_103/-/px-x-0104020000_103.px/
- https://www.pxweb.bfs.admin.ch/pxweb/en/px-x-0104020000_106/-/px-x-0104020000_106.px/
- https://www.pxweb.bfs.admin.ch/pxweb/en/px-x-0104020000_109/-/px-x-0104020000_109.px/

Examples

```
two_cantons_4years <- get_parameters(
  year_first = 2025,
  year_last = 2028,
  spatial_units = c("Aargau", "Zug")
)
```

`get_population` *Get population data from FSO*

Description

Users who do not have the required population data can use this convenience function to get the mandatory starting population for `propop::propop()` from the Federal Statistical Office (FSO). The function can also be used to obtain the population records for several years (e.g., for model performance evaluations). This function can be used to obtain data at various spatial levels (e.g., cantons, municipalities). The most recent data are usually about 6 to 18 months old.

To get the population data, you must use the **spelling** defined in the corresponding FSO table (STATTAB cube px-x-0102010000_101). Inspecting the column 'valueTexts' in the following package data may also help: `data('stattab_pop_snap')`.

Changes to the API interface may break this function.

Usage

```
get_population(
  number_fso = "px-x-0102010000_101",
  year,
  year_last = NULL,
  spatial_units
)
```

Arguments

`number_fso`	character, px-x table ID for population records, defaults to `px-x-0102010000_101`.
`year`	numeric, year for which the population records are to be downloaded. This usually is the starting population. To download longer time periods, use `year` to indicate the first year of the period.
`year_last`	numeric **(optional)**; specifies the final year of the time period for which data will be downloaded.
`spatial_units`	character vector, indicating at least one spatial entity for which the projection will be run. Typically a canton, several districts or municipalities.

Value

A data frame. For each of the four demographic groups (female / male, Swiss / foreign nationals), there are 101 age classes, resulting in a total of 404 rows per requested year and spatial unit. Columns included in the returned data frame:

year numeric, year in which the population was recorded.

spatial_unit character, indicating the spatial entities (e.g., cantons, districts, municipalities).

nat character, `ch` = Swiss, `int` = foreign / international.

sex character `f` = female, `m` = male.

age numeric, 101 one-year age classes, ranging from `0` to `100` (including those older than 100).

n numeric, number of people per year, spatial entity, and demographic group.

Source

Federal Statistical Office: https://www.pxweb.bfs.admin.ch/pxweb/en/px-x-0102010000_101/-/px-x-0102010000_101.px/

Examples

```
get_population(
  year = 2025,
  spatial_units = c("- Aargau", "......0301 Aarberg")
)
```

`prepare_evaluation` *Prepare data for evaluation*

Description

Takes population projections and benchmark data (e.g., population records) and prepares a combined data frame to evaluate the performance of the projection. For more details on usage, see `vignette("evaluate", package = "propop")`.

Usage

```
prepare_evaluation(
  data_benchmark,
  n_benchmark,
  data_projected,
  n_projected,
  age_groups = NULL
)
```

Arguments

`data_benchmark`	data frame containing benchmark data (e.g., actual / official population records; can be obtained with `propop::get_population()`).
`n_benchmark`	numeric column containing the benchmark population of each demographic group.
`data_projected`	data frame containing population projections; can be created with `propop::propop()`.
`n_projected`	numeric column containing the projected size of each demographic group.
`age_groups`	character, optional argument with options `"age_groups_3"` or `"age_groups_5"`. The option `"age_groups_3"` groups the data into three age ranges (0-19, 20-64, 65 years and older). The option `"age_groups_5"` groups the data into five age ranges (0-19, 20-39, 40-59, 60-79, 80 years and older). Using aggregated groups will lead to smaller projection errors than using 101 age classes. Defaults to using 101 one-year age classes if no option is chosen.

Value

Returns a data frame with the number of people from the benchmark and from the projection. Each row contains a unique combination of year, spatial unit, and demographic group.

Input data and variables

Both input data frames must contain the following variables for the **same range of years**:

year character, year in which the population was recorded.

spatial_unit character, indicating the projected spatial entities (e.g., cantons, districts, municipalities).

nat character, `ch` = Swiss, `int` = foreign / international.

sex character, `f` = female, `m` = male.

age numeric, 101 one-year age classes, ranging from 0 to 100 (including those older than 100).

n numeric, number of people per year, spatial entity, and demographic group.

Examples

```
data_benchmark <- tibble::tibble(
  year = 2030,
  spatial_unit = "Aargau",
  nat = c("ch", "ch"),
  sex = c("m", "f"),
  age = c(42, 42),
  n = c(3551, 3655)
)

data_projected <- tibble::tibble(
  year = 2030,
  spatial_unit = "Aargau",
  nat = c("ch", "ch"),
  sex = c("m", "f"),
```

```
  age = c(42, 42),
  n = c(3474, 3536)
)

combined <- prepare_evaluation(
  data_benchmark = data_benchmark,
  n_benchmark = "n",
  data_projected = data_projected,
  n_projected = "n"
)
```

---

propop *Project population development*

Description

Wrapper function to project population development using the cohort component method. This function iterates across years and spatial units calling the function `project_population.R`, which performs the calculations.

You can either use your own parameters and starting population or download these data from the Swiss Federal Statistical Office (FSO). For instructions on how to download this information from STAT-TAB, see `vignette("prepare_data", package = "propop")`.

The projection parameters need to be passed to `propop::propop()` as a **single data frame** (with the parameters as columns). The column types, names, and factor levels need to match the specifications listed below under `parameters`.

If nothing else is indicated in argument `scenarios`, `propop()` runs and returns all **scenarios** provided via `parameters`.

For more details on how to use this function to project the population development on the level of a canton, see `vignette("project_single_region", package = "propop")`.

Usage

```
propop(
  parameters,
  population,
  year_first,
  year_last,
  scenarios = NULL,
  age_groups = 101,
  fert_first = 15,
  fert_last = 49,
  share_born_female = 100/205,
  subregional = FALSE,
  binational = TRUE,
  spatial_unit = "spatial_unit"
)
```

Arguments

`parameters` data frame containing the FSO rates and numbers to run the projection for a specific spatial level (e.g., canton, municipality).

- `year`, character, projection year.
- `spatial_unit`, character, ID of spatial entity (e.g., canton, municipality) for which to run the projections.
- `scen`, character, one or several projection scenario(s). The main scenarios are usually "reference", "low" growth, and "high" growth.
- `nat`, character, nationality (`ch` = Swiss; `int` = foreign / international). Required if binational = `TRUE`.
- `sex`, character (`f` = female, `m` = male).
- `age`, numeric, typically ranging from 0 to 100 (incl. >100).
- `birthrate`, numeric, number of births per mother by age, nationality and year.
- `int_mothers`, numeric, proportion of children with Swiss nationality born to non-Swiss mothers. Required if binational = `TRUE`.
- `mor`, numeric, prospective mortality rate (probability of death).
- `acq`, numeric, rate of acquisition of Swiss citizenship. Required if binational = `TRUE`.
- `emi_int`, numeric, rate of people emigrating abroad (number of immigrants - number of emigrants).
- `emi_nat`, rate of people emigrating to other cantons.
- `imm_int_n`, numeric, number of people immigrating from abroad.
- `imm_nat_n`, numeric, number of people immigrating from other cantons.
- `mig_sub` **(optional)**, numeric, net migration per subregion; this is the migration from / to other subregions (e.g., municipalities, districts) within the main superordinate projection unit (e.g., a canton). Accounts for movements between different subregions. Needs to be provided by the user.
- `emi_sub` **(optional)**, numeric, emigration rate per subregion; Accounts for movements between different subregions. Needs to be provided by the user. * `imm_sub` **(optional)**, numeric, immigration rate per subregion; Accounts for movements between different subregions. Needs to be provided by the user.

`population` data frame including the starting population of each demographic group and each spatial unit. Possible values are the same as in `parameters` (apart from year). The data frame only includes one year, usually the one preceding the first year of the projection.

- `year` character, should be `year_first` - 1.
- `spatial_unit` character.
- `nat` character.
- `sex` character.
- `age` numeric.
- `n` numeric, number of people per demographic group.

`year_first` numeric, first year to be projected.

`year_last` numeric, last year to be projected.

`scenarios` **(optional)**, character, indicating which projection scenario(s) shall be run; the corresponding information must be available in `parameters`. Defaults to the values in variable `scen` in `parameters`.

`age_groups` numeric, number of age classes. Creates a vector with 1-year age classes running from `0` to (`age_groups - 1`). Must currently be set to `= 101` (FSO standard number of age groups).

`fert_first` numeric, first year of female fertility. Defaults to 15 (FSO standard value).

`fert_last` numeric, last year of female fertility. Defaults to 49 (FSO standard value).

`share_born_female`
numeric, fraction of female babies. Defaults to 100 / 205 (FSO standard value).

`subregional` character or FALSE, indicates if subregional migration patterns (e.g., movement between municipalities within a canton) are part of the projection (default `subregional = FALSE`). Requires input on the level of subregions (in `parameters` and `population`). Two calculation methods are supported to distribute people between subregions: With `subregional = "net"`, the net migration between subregions is added to the population balance. Net migration numbers must be specified in a data column `mig_sub` in `parameters`. With `subregional = "rate"`, the numbers for subregional emigrants are subtracted from the population balance, then redistributed back to all subregional units as subregional immigration; `parameters` must contain the columns `emi_sub` and `imm_sub`.

`binational` boolean, `TRUE` indicates that projections discriminate between two groups of nationalities. `FALSE` indicates that the projection is run without distinguishing between nationalities.

`spatial_unit` character, name of variable containing the names of the region or subregions for which the projection shall be performed.

Value

Returns a **data frame** that includes the number of people for each demographic group per year (for projected years) and spatial unit. The number of rows is the product of all scenarios times all years times all demographic groups times all spatial units. The output includes several **identifiers** that indicate to which scenario, demographic group, year, and spatial unit the results in the rows refer to:

`year` integer, indicating the projected years.

`scen` character, indicating the projected scenario(s).

`spatial_unit` factor, spatial units for which the projection was run (e.g., canton, districts, municipalities).

`age` integer, ranging from `0`n to `100` years (including those older than 100).

`sex` factor, female (`f`) and male (`m`).

`nat` factor, indicates if the nationality is Swiss (`ch`) or international / foreign (`int`). This variable is only returned if `binational = TRUE`.

The output also includes columns related to the **size and change of the population:**

n_jan	numric, start-of-year population per demographic group.
n_dec	numeric, end-of-year population per demographic group.
delta_n	numeric, population change per demographic group from the start to the end of the year in *absolute numbers*.
delta_perc	numeric, population change per demographic group from the start to the end of the year in *percentages*.

The **components** that are used to project the development of the population are also included in the output:

births	numeric, number of births (non-zero values are only available for age = 0).
mor_n	numeric, number of deaths.
emi_int_n	numeric, number of people who emigrate to other countries.
emi_nat_n	numeric, number of people who emigrate to other cantons.
imm_int_n	numeric, number of people who immigrate from other countries.
imm_nat_n	numeric, number of people who immigrate from other cantons.
acq_n	numeric, number of people who acquire Swiss citizenship (only returned if binational = TRUE.)
mig_sub	numeric, migration between subregions (only returned if subregional = "net".)
emi_sub_n	numeric, emigration from subregions (only returned if subregional = "rate".)
imm_sub_n	numeric, immigration from subregions (only returned if subregional = "rate".)

Examples

```
# Run projection for the sample data (whole canton of Aargau)
propop(
  parameters = fso_parameters,
  year_first = 2026,
  year_last = 2027,
  population = fso_population,
  subregional = FALSE,
  binational = TRUE
)
propop(
  parameters = fso_parameters |>
    dplyr::filter(scen == "reference" | scen == "high"),
  year_first = 2026,
  year_last = 2027,
  scenarios = c("reference", "high"),
  population = fso_population,
  subregional = FALSE,
  binational = TRUE
)
```

Index

∗ **datasets**
ag_migration_subregional, 4
ag_population_subregional, 4
fso_parameters, 10
fso_population, 11

ag_migration_subregional, 4
ag_population_subregional, 4
aggregate_measures, 2

calculate_rates, 5
calculate_rates(), *7*
calculate_shares, 6
check_balance, 7
compute_measures, 8

fso_parameters, 10
fso_population, 11

get_parameters, 12
get_population, 15

prepare_evaluation, 16
propop, 18
propop(), *7*