inedemogR: Tidy Access to Spanish INE Demographic Data

A Comprehensive Tutorial

J. R. Caro-Barrera

October 20, 2026

Contents

1 Introduction **3**
1.1 Installation . . . 3
1.2 A note on live data . . . 3

2 Part I — Data Retrieval, Cleaning, and Storage **5**
2.1 System A: quick multi-geography totals . . . 5
2.1.1 `list_ine_indicators()` . . . 5
2.1.2 `get_ine_demog()` . . . 5
2.1.3 `get_ine_geo()` . . . 6
2.1.4 `update_ine_data()` . . . 6
2.2 System B: age/sex-disaggregated province data . . . 6
2.2.1 Caching layer (`helpers.R`) . . . 6
2.2.2 `get_ine_population()` . . . 7
2.2.3 `get_ine_births()` . . . 7
2.2.4 `get_ine_births_by_age()` . . . 7
2.2.5 `get_ine_deaths()` . . . 8
2.3 The mortality pipeline . . . 8
2.3.1 Exposure-to-risk: `compute_exposure()` . . . 8
2.3.2 Central death rates: `compute_death_rates()` . . . 9
2.3.3 Period life tables: `build_life_tables()` . . . 10
2.3.4 Abridged life tables: `build_abridged_life_tables()` . . . 11
2.4 Exporting to files: `download_ine_data()` . . . 12

3 Part II — Demographic Analysis and Visualization **13**
3.1 Summary demographic indicators . . . 13
3.1.1 Age dependency ratios: `age_dependency_ratio()` . . . 13
3.1.2 Aging index: `aging_index()` . . . 13
3.1.3 Sex ratio: `sex_ratio()` . . . 13
3.1.4 Crude birth and death rates . . . 13
3.1.5 General fertility rate: `general_fertility_rate()` . . . 14
3.1.6 Fertility schedule: ASFR, TFR, MAC, GRR, NRR . . . 14
3.1.7 Total fertility rate: `total_fertility_rate()` . . . 15
3.1.8 Mean age at childbearing: `mean_age_at_childbearing()` . . . 15

3.1.9 Gross and net reproduction rates . . . 15
3.1.10 Infant mortality rate: `infant_mortality_rate()` . . . 15
3.1.11 Rate of natural increase: `rate_of_natural_increase()` . . . 15
3.1.12 Birth-to-death ratio: `birth_death_ratio()` . . . 16
3.1.13 Life expectancy: `life_expectancy_summary()` and `life_expectancy()` . . . 16
3.2 Mortality decomposition: `decompose_life_expectancy()` . . . 16
3.3 Mapping functions . . . 18
3.3.1 `plot_ine_map()` . . . 18
3.3.2 `map_indicator()` . . . 18
3.3.3 `map_life_expectancy()` . . . 18
3.4 Population pyramids and trend charts . . . 18
3.4.1 `plot_population_pyramid()` . . . 18
3.4.2 `plot_demog_trend()` . . . 19
3.5 Lexis diagrams: `plot_lexis_diagram()` . . . 19
3.6 Worked examples . . . 19
3.6.1 Birth-to-death ratio map (`birth_death_ratio_map.R`) . . . 19
3.6.2 Andalusian population pyramids (`andalusia_pyramids.R`) . . . 20
3.6.3 Madrid/Barcelona Lexis diagrams (`madrid_barcelona_lexis.R`) . . . 20
3.6.4 Life expectancy comparison (`life_expectancy_comparison.R`) . . . 21
3.6.5 Natural increase analysis (`natural_increase_analysis.R`) . . . 21
3.6.6 Fertility and infant mortality analysis (`fertility_infant_mortality_analysis.R`) 21
3.6.7 Fertility indicators analysis (`fertility_indicators_analysis.R`) . . . 22
3.6.8 Mortality decomposition analysis (`mortality_decomposition_analysis.R`) . . . 23
3.6.9 Abridged life tables analysis (`abridged_life_tables_analysis.R`) . . . 23

Appendix **25**

1 Introduction

inedemogR provides tidy, harmonized, reproducible access to demographic data published by the Spanish National Statistics Institute (Instituto Nacional de Estadística, INE), retrieved live via the official **ineapir** API wrapper, with spatial integration via **mapSpain** and plotting built on **ggplot2**.

The package is organized around two complementary data systems that serve different purposes. They are deliberately kept parallel rather than merged into one interface, because they operate at genuinely different levels of demographic detail.

System A `get_ine_demog()`, `get_ine_geo()`, `list_ine_indicators()`, `plot_ine_map()`, `map_indicator()`, `update_ine_data()`. Quick, multi-geography (municipality or province) totals for population, births, and deaths, with no age or sex breakdown. Best for national/regional choropleths and quick comparisons.

System B `get_ine_population()`, `get_ine_births()`, `get_ine_births_by_age()`, `get_ine_deaths()`, `compute_exposure()`, `compute_death_rates()`, `build_life_tables()`, `build_abridged_life_tabl` `download_ine_data()`, and the summary indicators, decomposition, and charts built on top of them. Rigorous, province-level, age/sex-disaggregated analysis following the Human Mortality Database (HMD) Methods Protocol V6, as implemented for the Spanish subnational Human Mortality Database (SHMD) pipeline. This is the only system capable of producing dependency ratios, population pyramids, Lexis diagrams, period life tables, fertility schedules (ASFR/TFR/MAC/GRR/NRR), and mortality decomposition, since these all require an age breakdown that System A's data does not carry.

This tutorial is split into two parts, matching the two purposes a user of **inedemogR** typically has:

- **Part I (Section 2)** covers strictly data *retrieval, cleaning, validation, and storage* — the functions in `get_ine_demog.R`, `get_ine_geo.R`, `population.R`, `births.R`, `deaths.R`, `exposure.R`, `death_rates.R`, `life_tables.R`, `abridged_life_tables.R`, `download_ine_data.R`, and the caching layer in `helpers.R`.
- **Part II (Section 3)** covers *demographic analysis and visualization* — the indicator functions in `indicators.R`, the mortality-decomposition functions in `decomposition.R`, and the charting/mapping functions in `plot_demog.R`, `plot_ine_map.R`, and `lexis.R` — together with full worked examples.

Every mortality-pipeline calculation (exposure-to-risk, central death rates, life tables) is presented with the exact mathematical formulation implemented in the code, following the HMD Methods Protocol V6.

1.1 Installation

```
# From a local clone or source tarball:
# install.packages("devtools")
devtools::install()

library(inedemogR)
```

1.2 A note on live data

Every retrieval function in this package calls the real INE API (or, for `get_ine_geo()`, the real **mapSpain** boundary service) over the network. There is no bundled offline dataset. All code listings in this tutorial are real, runnable calls against live INE data as of the 2023–2024 reference years used in the examples;

results will naturally update as INE publishes new years. Section 2.2.1 explains how to avoid repeatedly re-downloading the same data during iterative analysis.

2 Part I — Data Retrieval, Cleaning, and Storage

2.1 System A: quick multi-geography totals

2.1.1 `list_ine_indicators()`

Returns the internal registry (`ine_variables`) mapping each indicator code to the real INE table it is retrieved from, the geographic dimension column INE's API returns for that table, and the geographic granularity the table is actually published at.

```
list_ine_indicators()
#> # A tibble: 3 x 6
#>   indicator      name            description  id_table geo_var geo_level
#>   population_total Resident pop. ...           29005   Municipios municipality
#>   births_total    Live births    ...           6506    Provincias province
#>   deaths_total    Deaths         ...           6545    Provincias province
```

Migration indicators are not included in this release: INE's migration tables are origin×destination×year flow matrices, a different shape from the simple geography×year stock tables backing the three indicators above, so wiring one in needs its own design pass rather than a registry one-liner. `get_ine_demog()` still raises an explicit `"Unknown indicator(s)"` error for any indicator code not in `list_ine_indicators()`, migration or otherwise.

2.1.2 `get_ine_demog()`

Retrieves one or more indicators via `ineapir::get_data_table()` and tidies the result into one row per geography × year, one column per indicator.

```
get_ine_demog(indicator, geo_level = NULL, year = NULL,
              region = NULL, sex = "Total", geometry = FALSE)
```

indicator Character vector of indicator codes (see `list_ine_indicators()`).

geo_level The geographic level to fetch; inferred from `indicator` if `NULL`. All requested indicators must share the same level (population is municipality-level; births/deaths are province-level only) — requesting a mix across levels is a hard error.

year Numeric/character year; `NULL` returns only the latest published period.

region Optional regex vector matched against the geography name.

sex One of `"Total"`, `"Hombres"`, `"Mujeres"`.

geometry If `TRUE`, joins `get_ine_geo()`'s geometry and returns an `sf` object.

```
# Population totals for every municipality, 2023
pop_totals <- get_ine_demog(indicator = "population_total", year = 2023)

# Births and deaths together, province level, one call
vitals <- get_ine_demog(
  indicator = c("births_total", "deaths_total"), year = 2023
)
```

2.1.3 `get_ine_geo()`

Retrieves province or municipality boundaries via **mapSpain** (`esp_get_prov()` / `esp_get_munic()`), reprojected to ETRS89 / UTM zone 30N (EPSG:25830), with a `GEOID` column that matches `get_ine_demog()`'s output exactly so the two can be joined directly.

```
get_ine_geo(geo_level = c("municipality", "province"), region = NULL,
            moveCAN = TRUE, can_gap_km = 60)
```

When `moveCAN = TRUE` (the default), the Canary Islands are translated so they sit `can_gap_km` kilometres from the nearest mainland coastline point (near Cádiz/Huelva) rather than left in their true, far-southwest position — the standard cartographic convention for compact national maps of Spain. The shifted islands' bounding box is attached as the `"can_box"` attribute, used by `plot_ine_map()` and `map_indicator()` to draw an inset separator frame.

```
prov_geo <- get_ine_geo(geo_level = "province")
mun_geo <- get_ine_geo(geo_level = "municipality", region = "Sevilla")
```

2.1.4 `update_ine_data()`

Checks every indicator wired in the registry for its latest published year and refreshes a local `.rds` cache (under `tools::R_user_dir("inedemogR", "cache")` by default) only if INE has published a newer year than what is cached, or if `force = TRUE`.

```
update_ine_data(force = FALSE, data_dir = tools::R_user_dir("inedemogR", "cache"))
```

2.2 System B: age/sex-disaggregated province data

2.2.1 Caching layer (`helpers.R`)

`get_ine_population()`, `get_ine_births()`, and `get_ine_deaths()` share a common disk-cache wrapper, `with_ine_cache()`. Fetching province-level age-specific deaths and population from INE's Tempus3 API (sometimes via a full CSV bulk-export fallback when the JSON API rejects an oversized query) is the slowest part of the package, often taking minutes; results are cached as `.rds` files keyed by the call's arguments so repeated calls during iterative analysis are near-instant.

Every System B retrieval function exposes the same three arguments:

use_cache (default `TRUE`) Return a previously cached result for the same arguments instead of re-fetching.

force (default `FALSE`) Ignore any existing cache and re-fetch from INE, refreshing the cache afterwards.

cache_dir Directory for the cache (default: `tools::R_user_dir("inedemogR", "cache")`, the same location `update_ine_data()` uses).

```
# First call: fetches live and caches the result.
pop <- get_ine_population(n_periods = 10)

# Second call, same arguments: returns instantly from cache.
pop <- get_ine_population(n_periods = 10)

# Force a fresh fetch, e.g. after INE has published a new year.
```

```
pop <- get_ine_population(n_periods = 10, force = TRUE)
```

2.2.2 get_ine_population()

Retrieves Spanish January-1st provincial population by single-year age (0–100+) and sex from INE's Padrón Municipal Continuo, per SHMD Protocol Section 12, Step 3.

```
get_ine_population(table_id = 56945, n_periods = 30,
                   use_cache = TRUE, force = FALSE,
                   cache_dir = tools::R_user_dir("inedemogR", "cache"))
```

Returns `list(data, qc)`: `data` has columns `ine_code`, `nuts3_code`, `nuts2_code`, `province_name`, `year`, `age`, `female`, `male`, `total`; `qc` is the output of `validate_population()`, which checks for negative/missing counts, incomplete age ranges, duplicate rows, year gaps, and implausible age-to-age jumps (>50%, informational only).

```
pop <- get_ine_population()
head(pop$data)
pop$qc$passed
```

2.2.3 get_ine_births()

Retrieves annual live births by province and sex from INE's MNPN operation.

```
get_ine_births(table_id = 6506, n_periods = 100,
               use_cache = TRUE, force = FALSE,
               cache_dir = tools::R_user_dir("inedemogR", "cache"))
```

Returns `list(data, qc)`; `data` has `ine_code`, `nuts3_code`, `nuts2_code`, `province_name`, `year`, `female`, `male`, `total`. `validate_births()` checks the same invariants as `validate_population()` (no age dimension here, so no age-range check).

2.2.4 get_ine_births_by_age()

`get_ine_births()` has no age-of-mother breakdown, so it cannot feed a true total fertility rate. `get_ine_births_by_age()` retrieves the same MNPN operation's age-of-mother table instead (table 6508: province, single year of age of the mother 15–49 plus the two open intervals `under15`/`50plus`, and sex of the newborn) — the input Section 3.1.6's age-specific fertility rate schedule needs.

```
get_ine_births_by_age(table_id = 6508, n_periods = 100,
                      use_cache = TRUE, force = FALSE,
                      cache_dir = tools::R_user_dir("inedemogR", "cache"))
```

Returns `list(data, qc)`; `data` has `ine_code`, `nuts3_code`, `nuts2_code`, `province_name`, `year`, `age` (integer, `NA` for the `under15` group), `age_group` (`"under15"`, `"15"` — `"49"`, `"50plus"`), `female`, `male`, `total`. INE's table omits a (province, age, sex-of-newborn, year) series entirely once its value is zero across the whole fetched window — the same behavior already seen in `compute_death_rates()` (Section 2.3.2) — so `clean_births_by_age()` builds the complete grid explicitly and fills absent combinations with zero rather than leaving them missing, before `validate_births_by_age()` checks for negative/missing counts, duplicate rows, and complete single-year age coverage (15–49) per province-year.

```
births_age <- get_ine_births_by_age()
births_age$data
```

2.2.5 `get_ine_deaths()`

Retrieves annual deaths by province and sex (INE's MNPD operation, table 6545) *and* an age-specific companion series (table 6547), since the mortality pipeline (`compute_exposure()`, `compute_death_rates()`) needs deaths by single-year age, $D(x,t)$, not just the yearly total $D(t)$.

```
get_ine_deaths(table_id = 6545, age_table_id = 6547, n_periods = 100,
               use_cache = TRUE, force = FALSE,
               cache_dir = tools::R_user_dir("inedemogR", "cache"))
```

Returns `list(data_provinces, data_national, qc, qc_age)`: `data_provinces` is the age-specific series (`nuts3_code`, `province_name`, `year`, `age`, `female`, `male`, `total`) that feeds the rest of the mortality pipeline; `data_national` is Spain's age-less annual total, used internally to cross-check the sum of provincial totals against the official national figure (`validate_deaths()`'s `national_vs_provincial_mismatch` check) and also validates the sex ratio at death lies in the plausible range $[0.90, 1.50]$.

```
deaths <- get_ine_deaths()
deaths$data_provinces # age-specific, feeds compute_exposure()/compute_death_rates()
deaths$data_national  # Spain total, used for QC cross-checks
```

2.3 The mortality pipeline

The three functions below implement, in order, Steps 4–6 of the SHMD Protocol (equivalently, HMD Methods Protocol V6): exposure-to-risk, central death rates, and period life tables. Each stage consumes the previous stage's output directly.

2.3.1 Exposure-to-risk: `compute_exposure()`

Exposure-to-risk $E(x,t)$ approximates the person-years lived at age x during calendar year t. HMD Methods Protocol V6 (Eq. 57, uniform distribution of births within cohorts) gives

$$E(x,t) \;=\; \tfrac{1}{2}\big[P(x,t)+P(x,t+1)\big] \;+\; \tfrac{1}{6}\big[D^{L}(x,t)-D^{U}(x,t)\big] \tag{1}$$

where $P(x,t)$ is the January 1 population aged x in year t; $D^{L}(x,t)$ counts deaths in the lower Lexis triangle (the cohort born in $t-x$, which reaches age x during year t) and $D^{U}(x,t)$ deaths in the upper triangle (the cohort born in $t-x-1$, already aged x on 1 January). If `deaths` carries a `cohort` column the triangles are taken from it. INE's death tables are not classified by cohort, so by default deaths are split evenly between the triangles, the correction vanishes, and exposure is the mean of the two January-1 stocks. This is an *adaptation* of the protocol, which splits square-cell deaths by regression (Appendix A) and corrects for the monthly distribution of births (Appendix E); the sensitivity of e_0 and e_{65} to the split is reported in the package's validation study (`reproducibility/`).

```
compute_exposure(population, deaths)
```

population `get_ine_population()$data`.

deaths `get_ine_deaths()$data_provinces`.

Because death registration lags population estimates by roughly a year, the most recent year in `population` commonly has no matching rows in `deaths` yet. That year is still used as the $P(x,t+1)$ boundary needed to compute the *previous* year's exposure, but is not itself returned. If $P(x,t+1)$ is missing for a year that is kept, exposure falls back to $P(x,t)$ and the row is flagged `is_boundary_year`.

```
pop <- get_ine_population()
deaths <- get_ine_deaths()
exposure <- compute_exposure(pop$data, deaths$data_provinces)
exposure$qc$passed
```

`validate_exposure()` flags cells with no usable source population, any remaining negative or missing exposure, and exposure exceeding twice the source population.

2.3.2 Central death rates: `compute_death_rates()`

Given deaths $D(x,t)$ and exposure $E(x,t)$, the central death rate at single-year age x in year t is

$$m(x,t) \;=\; \frac{D(x,t)}{E(x,t)} \qquad \text{(1x1: single-year age, one-year period)} \tag{2}$$

Missing data. Three situations are kept apart:

- *Structural zeros.* INE's age-specific deaths table omits cells with no deaths, so an age missing from a province-year that is otherwise present is a true zero.
- *Unavailable or suppressed counts.* An explicit `NA` death count (INE suppresses some small sex-specific cells; those recoverable as Total minus the other sex are recovered at retrieval) stays `NA` and propagates to the rate, the grouped rate, and the ASDR, and makes `validate_death_rates()` fail.
- *Missing province-years.* A province-year with exposure but no death records at all is excluded with a warning; it is never treated as zero mortality.

A pooled 5-year-age-group ($n = 5$) rate is also computed from the same counts:

$$m(x,n,t) \;=\; \frac{\sum_{k=0}^{4} D(x+k,t)}{\sum_{k=0}^{4} E(x+k,t)} \qquad \text{(5x1)} \tag{3}$$

and an age-standardised death rate per 100,000 using the European Standard Population 2013 (ESP2013):

$$\mathrm{ASDR}(t) \;=\; \sum_{x=0}^{100+} m(x,t)\,w(x), \qquad \sum_{x} w(x) = 100{,}000. \tag{4}$$

ESP2013 gives weights to five-year groups up to an open group 95+ (weight 200). Each group's weight is spread evenly over its single ages; the 95+ weight is spread over the six cells $95,\ldots,99,100+$. The ASDR is `NA` unless every age has a finite rate.

```
compute_death_rates(deaths, exposure)
```

Returns `list(mx_1x1, mx_5x1, asdr, qc)`; `mx_1x1` and `mx_5x1` carry the deaths (`d_*`) and exposures (`e_*`) beside the rates, as the old-age model below needs them. `validate_death_rates()` fails on negative, missing or infinite rates, on rates ≥ 1 below age 90, and on incomplete age coverage.

```
rates <- compute_death_rates(deaths$data_provinces, exposure$data)
rates$mx_1x1  # age x year x province mortality schedule
rates$asdr    # age-standardised rate per 100,000, one row per province-year
```

2.3.3 Period life tables: `build_life_tables()`

`build_life_tables()` constructs single-year period life tables (ages 0 to 110+) for every province and year in `mx_1x1`, following HMD Methods Protocol V6, Section 7.1.

Kannisto smoothing of old-age mortality. Observed rates are replaced from the age Y upward, where Y is the lowest age from 80 with at most 100 female or 100 male deaths, constrained to $80 \leq Y \leq 95$ (the same Y for both sexes). The fitted hazard is the Kannisto logistic model with asymptote one,

$$\mu_x(a,b) = \frac{a\,e^{b(x-80)}}{1+a\,e^{b(x-80)}}, \qquad a,b \geq 0, \tag{5}$$

estimated by Poisson maximum likelihood, $D_x \sim \text{Poisson}(E_x\,\mu_{x+1/2})$, maximising $\sum_x \left[D_x \log \mu_{x+1/2} - E_x\,\mu_{x+1/2}\right]$ over ages 80–99 (L-BFGS-B, analytic gradient, grid-search start). HMD fits ages 80–110+; INE's single-year detail ends at an open 100+ group, so the package fits 80–99. The constraint $b \geq 0$ means smoothed rates cannot fall with age. The smoothed rate at age x is $\hat{\mu}_{x+1/2}$. For the combined-sex table, rates from Y are the sex-specific smoothed rates weighted by a smoothed female share of exposure (weighted least squares of $\text{logit}\,\pi_x^F$ on a quadratic in age, HMD Eqs. 69–73).

Andreev–Kingkade a_0. The mean time lived in the first year by those who die before age 1 uses the period formulas of Andreev and Kingkade (2015) as tabulated in HMD Methods Protocol V6, Table 3:

$$a_0 = \begin{cases} 0.14929 - 1.99545\,m_0 & \text{male},\ m_0 < 0.02300 \\ 0.02832 + 3.26021\,m_0 & \text{male},\ 0.02300 \leq m_0 < 0.08307 \\ 0.29915 & \text{male},\ m_0 \geq 0.08307 \\ 0.14903 - 2.05527\,m_0 & \text{female},\ m_0 < 0.01724 \\ 0.04667 + 3.88089\,m_0 & \text{female},\ 0.01724 \leq m_0 < 0.06891 \\ 0.31411 & \text{female},\ m_0 \geq 0.06891 \end{cases} \tag{6}$$

For the combined-sex table, a_0 is the average of the female and male values weighted by their age-0 deaths (HMD Eq. 77). Every other closed age has $a_x = 0.5$.

Life-table recursion. Given m_x and a_x for $x = 0, 1, \ldots, \omega$ (open interval at $\omega = 110$):

$$q_x = \frac{m_x}{1 + (1 - a_x) m_x}, \qquad q_\omega = 1 \tag{7}$$

$$l_0 = 100{,}000, \qquad d_x = l_x q_x, \qquad l_{x+1} = l_x - d_x \tag{8}$$

$$L_x = \begin{cases} l_x - (1 - a_x)\, d_x & x < \omega \\ l_\omega / m_\omega & x = \omega \end{cases} \tag{9}$$

$$T_x = \sum_{y \geq x} L_y, \qquad e_x = T_x / l_x \tag{10}$$

so that $m_x = d_x / L_x$ holds at every age, including the open interval ($a_\omega = 1/m_\omega$).

Failure behaviour and coverage. A province-year is tabulated only if exposure is available at every single age 0–100 and rates are finite below Y; tables that do not qualify are withheld and listed, with the reason, in `$failed`. INE's provincial population by single year of age is top-coded at 85+ before 2002, so the supported period is 2002 onwards. Zero exposure at ages $\geq Y$ (for example, very old men in Ceuta and Melilla) is allowed, because those rates come from the fitted model. `validate_life_table()` fails on missing or non-contiguous ages, non-finite values, increasing l_x, non-positive L_x or e_x, $q_x \notin [0,1]$, a broken $m_x = d_x / L_x$ identity, or $e_0 \notin (15, 100]$. The arguments `kannisto_age` (a fixed Y) and `fit_min` (first fitting age) support sensitivity analysis.

```
build_life_table(mx_df, sex = c("female", "male")) # single year, single sex
build_life_tables(mx_1x1)                          # every province and year
```

`build_life_tables()` returns `list(fltper, mltper, bltper, qc, failed)`: female, male, and both-sex period life tables, each a tibble with `nuts3_code`, `province_name`, `year`, `age`, `mx`, `qx`, `ax`, `lx`, `dx`, `Lx`, `Tx`, `ex`; a `qc` tibble (one row per table, with the replacement age Y); and the withheld tables.

```
lt <- build_life_tables(rates$mx_1x1)
lt$fltper[lt$fltper$age == 0, ] # e0 by province and year, female
```

With the frozen inputs of the package's `reproducibility/` folder, this gives for A Coruña, 2024, $e_0 =$ 86.95 years (female) and 81.21 years (male).

2.3.4 Abridged life tables: `build_abridged_life_tables()`

`build_abridged_life_tables()` builds abridged period life tables (`"00-04"`, `"05-09"`, ..., `"95-99"`, `"100+"`) directly from the observed grouped rates `mx_5x1`. It is an independent estimate, not an abridgement of the single-year table: it uses no Kannisto smoothing and closes at 100+ rather than 110+.

The youngest group is the one exception to using `mx_5x1` alone: a single-year sub-table for ages 0–4 (Andreev–Kingkade a_0, $a_x = 0.5$ at ages 1–4) is built from `mx_1x1` and aggregated, which avoids an external ${}_5a_0$ approximation for the sharp infant gradient.

For every other closed interval $[x, x+n)$ ($n = 5$) the package uses Greville's ${}_na_x$ with the matching Chiang conversion (Preston, Heuveline & Guillot, 2001, Box 3.1):

$$ {}_na_x = \frac{n}{2} - \frac{n^2}{12}\left({}_nm_x - k_x\right), \qquad k_x = \frac{1}{2n}\log\frac{{}_nm_{x+n}}{{}_nm_{x-n}} \tag{11} $$

$$ {}_nq_x = \frac{n\,{}_nm_x}{1+(n-{}_na_x)\,{}_nm_x}, \qquad {}_nL_x = n\,l_{x+n} + {}_na_x\,{}_nd_x \tag{12} $$

This pair satisfies ${}_nm_x = {}_nd_x/{}_nL_x$ exactly. Where k_x is undefined (a zero rate in a neighbouring group) or ${}_nq_x$ would exceed 1, the group falls back to a constant hazard, ${}_nq_x = 1 - e^{-n\,{}_nm_x}$ and ${}_nL_x = {}_nd_x/{}_nm_x$, which also preserves the identity. The open interval has $q = 1$ and $L = l/m$.

```
build_abridged_life_table(mx_1x1, mx_5x1, sex = c("female", "male")) # one province/year
build_abridged_life_tables(mx_1x1, mx_5x1)                          # every province/year
```

Returns `list(fltper, mltper, bltper, qc)`, matching `build_life_tables()`'s structure with `age_group`/`age_start` in place of `age`. `validate_abridged_life_table()` also checks the identity ${}_nm_x = {}_nd_x/{}_nL_x$ in every closed group.

```
alt <- build_abridged_life_tables(rates$mx_1x1, rates$mx_5x1)
alt$fltper[alt$fltper$age_group == "00-04", ]
```

For A Coruña, 2024, the abridged female e_0 is 86.95 years, within 0.01 years of the single-year table. Because the two tables share inputs and part of their method, this agreement is a consistency check, not an external validation; the comparison with INE's published tables is reported in `reproducibility/`.

2.4 Exporting to files: `download_ine_data()`

For users who prefer to work outside R (a spreadsheet, another statistical package, or HMD-compatible mortality software), `download_ine_data()` runs any subset of the six pipeline stages and writes the results to a folder as CSV and/or HMD-format `.txt` files.

```
download_ine_data(out_dir,
                  stages = c("births", "deaths", "population",
                             "exposure", "mx", "life_tables"),
                  format = c("csv", "txt"),
                  n_periods = 30)
```

Later stages automatically pull in their dependencies (`exposure` needs `population` and `deaths`; `mx` needs `deaths` and `exposure`; `life_tables` needs `mx`), since they are fetched along the way regardless.

```
# Everything, both file formats, into ./ine_data
result <- download_ine_data("ine_data")
result$files            # every path written
result$life_tables$fltper # also available in memory, no re-read needed

# Just births and deaths, CSV only
download_ine_data("ine_data", stages = c("births", "deaths"), format = "csv")

# One HMD-format .txt file per province
download_ine_data("ine_data", stages = "life_tables", format = "txt")
```

`"csv"` writes one combined file per data table (e.g. `population.csv`, one row per province×age×year); `"txt"` writes HMD-format files, one per province, matching the layout HMD-compatible tooling expects.

3 Part II — Demographic Analysis and Visualization

3.1 Summary demographic indicators

The functions in this section all operate on System B's age/sex-disaggregated province data (or, for `birth_death_ratio()`, on System A's totals — noted explicitly below); `get_ine_demog()`'s total-only indicators cannot be used for any indicator that requires an age breakdown.

3.1.1 Age dependency ratios: `age_dependency_ratio()`

With P_{young}, P_{work}, P_{old} the population aged ≤ 14, 15–64, and ≥ 65 respectively (cut-offs configurable via `young_max`/`old_min`):

$$\text{youth dependency ratio} = \frac{P_{\text{young}}}{P_{\text{work}}} \times 100 \tag{13}$$

$$\text{old-age dependency ratio} = \frac{P_{\text{old}}}{P_{\text{work}}} \times 100 \tag{14}$$

$$\text{total dependency ratio} = \frac{P_{\text{young}} + P_{\text{old}}}{P_{\text{work}}} \times 100 \tag{15}$$

```
pop <- get_ine_population()
adr <- age_dependency_ratio(pop$data)
adr[adr$nuts3_code == "ES111", ]
```

3.1.2 Aging index: `aging_index()`

$$\text{aging index} = \frac{P_{\text{old}}}{P_{\text{young}}} \times 100 \tag{16}$$

```
ai <- aging_index(pop$data)
```

3.1.3 Sex ratio: `sex_ratio()`

$$\text{sex ratio} = \frac{P_{\text{male}}}{P_{\text{female}}} \times 100 \tag{17}$$

computed over all ages by default, or per single year of age with `by_age = TRUE`.

```
sex_ratio(pop$data)
sex_ratio(pop$data, by_age = TRUE)
```

3.1.4 Crude birth and death rates

$$\text{CBR} = \frac{B}{P} \times 1000, \qquad \text{CDR} = \frac{D}{P} \times 1000 \tag{18}$$

where B, D, P are total annual births, deaths, and population.

```
births <- get_ine_births()
deaths <- get_ine_deaths()
cbr <- crude_birth_rate(births$data, pop$data)
cdr <- crude_death_rate(deaths$data_provinces, pop$data)
```

These are crude rates, not a total fertility rate (TFR). TFR requires age-of-mother-specific birth counts; INE's underlying MNPN table as retrieved by `get_ine_births()` carries no age-of-mother breakdown, and this package does not currently compute TFR. Do not substitute `crude_birth_rate()` for TFR in fertility analysis.

3.1.5 General fertility rate: `general_fertility_rate()`

GFR improves on CBR by dividing births by the female population actually at risk of childbearing (ages 15–49 by default), removing the distortion CBR suffers when two provinces have different age/sex structures despite similar underlying fertility behavior:

$$\text{GFR} = \frac{B}{W_{15-49}} \times 1000 \tag{19}$$

```
gfr <- general_fertility_rate(births$data, pop$data, age_min = 15, age_max = 49)
```

GFR is still not a TFR: it is a single aggregate rate over all reproductive-age women, not an age-specific fertility schedule.

3.1.6 Fertility schedule: ASFR, TFR, MAC, GRR, NRR

Where CBR and GFR (above) are single aggregate rates, the functions below use `get_ine_births_by_age()`'s (Section 2.2.4) age-of-mother breakdown to build the full age-specific fertility schedule and the standard summary indicators derived from it.

Age-specific fertility rate: `age_specific_fertility_rate()`. The package follows INE's methodology for its Indicadores Demográficos Básicos:

$$\text{ASFR}(x) = \frac{B(x)}{\bar{W}(x)}, \qquad \text{ASFR}_{\text{female}}(x) = \frac{B_{\text{female}}(x)}{\bar{W}(x)}, \qquad \bar{W}(x) = \tfrac{1}{2}\left[W(x,t) + W(x,t+1)\right] \tag{20}$$

for single years of age $x = 15, \ldots, 49$, where $B(x)$ is births in year t to mothers of completed age x and $\bar{W}(x)$ is the mean annual female population (the mean of the January-1 stocks of t and $t+1$; if the $t+1$ stock is not available the t stock is used and the row is flagged `is_boundary_year`). Births to mothers younger than 15 are added to age 15 and births to mothers aged 50 or over to age 49, so no births are lost. Rates are births *per woman*; the column `asfr_per_1000` gives the same schedule per 1,000 women for display. $\text{ASFR}_{\text{female}}$ uses INE's sex-of-newborn breakdown rather than an assumed sex ratio at birth.

```
births_age <- get_ine_births_by_age()
pop <- get_ine_population()
asfr <- age_specific_fertility_rate(births_age$data, pop$data, age_min = 15, age_max = 49)
```

3.1.7 Total fertility rate: `total_fertility_rate()`

$$\text{TFR} = \sum_{x=15}^{49} \text{ASFR}(x) \quad \text{(children per woman)} \tag{21}$$

```
tfr <- total_fertility_rate(asfr)
```

3.1.8 Mean age at childbearing: `mean_age_at_childbearing()`

Completed age x covers the interval $[x, x+1)$, so its mid-point is used:

$$\text{MAC} = \frac{\sum_x (x+0.5)\,\text{ASFR}(x)}{\sum_x \text{ASFR}(x)} \tag{22}$$

```
mac <- mean_age_at_childbearing(asfr)
```

3.1.9 Gross and net reproduction rates

$$\text{GRR} = \sum_{x=15}^{49} \text{ASFR}_{\text{female}}(x), \qquad \text{NRR} = \sum_{x=15}^{49} \text{ASFR}_{\text{female}}(x)\,\frac{L_x}{l_0} \tag{23}$$

using the female period life table of the same province and year (Section 2.3.3), so NRR $\leq$ GRR.

```
grr <- gross_reproduction_rate(asfr)
nrr <- net_reproduction_rate(asfr, lt$fltper) # lt$fltper: female life table, build_life_tables()
```

With the frozen inputs, Madrid 2023 gives GRR $= 0.533$ and NRR $= 0.529$.

3.1.10 Infant mortality rate: `infant_mortality_rate()`

$$\text{IMR} = \frac{D_0}{B} \times 1000 \tag{24}$$

where D_0 is deaths under age 1 in year t and B is live births in that *same* year t (the birth cohort those deaths are drawn from) — a different denominator convention from CDR. IMR is closely related to, but not identical to, q_0 in the period life table: q_0 is derived from the exposure-based m_0 via the Andreev-Kingkade a_0, while IMR here uses the simpler, standard deaths/births ratio directly; the two will usually be close but need not match exactly.

```
imr <- infant_mortality_rate(deaths$data_provinces, births$data)
```

3.1.11 Rate of natural increase: `rate_of_natural_increase()`

$$\text{RNI} = \text{CBR} - \text{CDR} \tag{25}$$

combining already-computed `crude_birth_rate()` and `crude_death_rate()` output. A positive value means the population is growing from natural change alone (ignoring migration); negative means the reverse — the point in time and space where CDR overtakes CBR is often called the "demographic crossover."

```
rni <- rate_of_natural_increase(cbr, cdr)
```

3.1.12 Birth-to-death ratio: `birth_death_ratio()`

Unlike the indicators above, this needs no age breakdown, so it operates directly on `get_ine_demog()`'s totals (System A):

$$\text{birth-death ratio} = \frac{B}{D} \tag{26}$$

```
vitals <- get_ine_demog(indicator = c("births_total", "deaths_total"), year = 2023)
ratio <- birth_death_ratio(vitals)
```

A value of 2 means 2 live births per death; 0.5 means 1 birth per 2 deaths.

3.1.13 Life expectancy: `life_expectancy_summary()` and `life_expectancy()`

`life_expectancy_summary()` extracts e_0 (life expectancy at birth) and e_{65} (remaining life expectancy at 65) for every province and year from a `build_life_tables()` output table:

```
le <- life_expectancy_summary(lt$fltper, sex = "female")
```

`life_expectancy()` is a one-province convenience wrapper: it builds the life table for a single matched province directly from `compute_death_rates()`'s `mx_1x1` output, without requiring `build_life_tables()` to be run for every province first.

```
life_expectancy(rates$mx_1x1, province = "A Coruna", sex = "female")
```

3.2 Mortality decomposition: `decompose_life_expectancy()`

Life expectancy summaries answer *what* e_0 is; mortality decomposition answers *why* two life tables' e_0 values differ — attributing the gap between two provinces (same year) or one province (two years) to age-specific contributions. Two independently-derived methods are implemented so results can be cross-checked against each other (Ponnapalli, 2005, found the two are not sensitive to which one is used):

```
decompose_life_expectancy(lt1, lt2, method = c("arriaga", "pollard"))
```

lt1, lt2 Single life tables (one province-year-sex slice of `build_life_tables()`'s `fltper`/`mltper`/`bltper`, or `build_life_table()`'s direct output), over the *same* set of ages.

Arriaga's method (1984, the default). An exact discrete decomposition. For a non-terminal age interval $[x, x+n)$, splitting into a direct effect (extra/fewer years lived *within* the interval) and a combined indirect/interaction effect (extra/fewer survivors carrying forward into *later* ages):

$$\mathrm{DE}(x) = \frac{l_x^{(1)}}{l_0^{(1)}} \left(\frac{{}_nL_x^{(2)}}{l_x^{(2)}} - \frac{{}_nL_x^{(1)}}{l_x^{(1)}} \right) \tag{27}$$

$$\mathrm{IE}(x) = \frac{T_{x+n}^{(2)}}{l_0^{(1)}} \left(\frac{l_x^{(1)}}{l_x^{(2)}} - \frac{l_{x+n}^{(1)}}{l_{x+n}^{(2)}} \right) \tag{28}$$

$$\Delta(x) = \mathrm{DE}(x) + \mathrm{IE}(x) \tag{29}$$

with superscripts $(1)/(2)$ denoting the reference/comparison life table. For the terminal open interval ω, only a direct effect applies (since ${}_\infty L_\omega = T_\omega$):

$$\Delta(\omega) = \frac{l_\omega^{(1)}}{l_0^{(1)}} \left(\frac{T_\omega^{(2)}}{l_\omega^{(2)}} - \frac{T_\omega^{(1)}}{l_\omega^{(1)}} \right) \tag{30}$$

and $\sum_x \Delta(x) = e_0^{(2)} - e_0^{(1)}$ exactly.

Pollard's method (1988). An exact *continuous*-time decomposition. The package uses the symmetric average of Pollard's two exact forms, $e_0^{(2)} - e_0^{(1)} = \int \left(\mu^{(1)}(x) - \mu^{(2)}(x)\right) \frac{1}{2}\left[\frac{l^{(2)}(x)}{l_0^{(2)}} e^{(1)}(x) + \frac{l^{(1)}(x)}{l_0^{(1)}} e^{(2)}(x)\right] dx$, and integrates each closed single-year interval by the midpoint rule,

$$\Delta(x) = \tfrac{1}{2}\left(m_x^{(1)} - m_x^{(2)}\right) \left(\frac{L_x^{(2)}}{l_0^{(2)}} \bar{e}_x^{(1)} + \frac{L_x^{(1)}}{l_0^{(1)}} \bar{e}_x^{(2)} \right), \qquad \bar{e}_x = \tfrac{1}{2}\left(e_x + e_{x+1}\right), \tag{31}$$

and the open interval ω exactly under its constant hazard ($l(\omega+t) = l_\omega e^{-\mu t}$, $e(\omega+t) = 1/\mu$):

$$\Delta(\omega) = \left(m_\omega^{(1)} - m_\omega^{(2)}\right) e_\omega^{(1)} e_\omega^{(2)} \cdot \tfrac{1}{2} \left(\frac{l_\omega^{(1)}}{l_0^{(1)}} + \frac{l_\omega^{(2)}}{l_0^{(2)}} \right). \tag{32}$$

The sum differs from $e_0^{(2)} - e_0^{(1)}$ by a small discretisation residual, returned with the result:

```
lt <- build_life_tables(rates$mx_1x1)
female <- lt$fltper # female tables
lt_a <- female[female$province_name == "A Coruna" & female$year == 2023, ]
lt_b <- female[female$province_name == "Madrid" & female$year == 2023, ]

decomp <- decompose_life_expectancy(lt_a, lt_b, method = "arriaga")
sum(decomp$contribution)       # recovers the e0 gap exactly
attr(decomp, "e0_difference")  # e0(Madrid) - e0(A Coruna)
attr(decompose_life_expectancy(lt_a, lt_b, "pollard"), "residual")
```

With the frozen inputs, the female e_0 gap between A Coruña and Madrid in 2023 is 1.04 years, with ages 0, 58 and 90 the largest single-age contributors; Pollard's contributions sum to the gap within 0.0003 years. For a temporal comparison, A Coruña 2016 versus 2023, the gain is 0.76 years.

3.3 Mapping functions

3.3.1 `plot_ine_map()`

Plots the full map of Spain at province or municipality level, with one or more regions highlighted — primarily a diagnostic tool for confirming which regions a regex `region`/`highlight` pattern actually matches.

```
plot_ine_map(geo_level = "province", highlight = "^Cordoba$")
```

3.3.2 `map_indicator()`

General-purpose choropleth: joins any tidy indicator tibble keyed by `GEOID` onto province or municipality geometry, as either a binned discrete-legend map (default, matching the classic "N regions above/below a threshold" choropleth style) or a continuous gradient.

```
map_indicator(df, value_col, geo_level = c("province", "municipality"),
              region = NULL, binned = TRUE, breaks = NULL,
              palette = if (binned) "PiYG" else "D",
              moveCAN = TRUE, can_gap_km = 60,
              legend_title = NULL, title = NULL)
```

```
pop_data <- get_ine_demog(indicator = "population_total", year = 2023)
map_indicator(pop_data, "population_total", geo_level = "province")
```

3.3.3 `map_life_expectancy()`

A thin wrapper around `map_indicator()` that bridges System B's `life_expectancy_summary()` output with System A's `get_ine_geo()` geometry via the internal province lookup table.

```
map_life_expectancy(le_df, year, sex = "both",
                    moveCAN = TRUE, can_gap_km = 60, title = NULL)
```

```
map_life_expectancy(le, year = 2024, sex = "female")
```

3.4 Population pyramids and trend charts

3.4.1 `plot_population_pyramid()`

Draws a mirrored population pyramid (female right, male left as negative counts) for one province/region and year.

```
plot_population_pyramid(pop_df, year, region = NULL, title = NULL)
```

```
plot_population_pyramid(pop$data, year = 2024, region = "A Coruna")
```

Note `geom_col(width = 1)` is used internally rather than **ggplot2**'s default `width = 0.9`: at the default width, the systematic 10% gap between each of the 101 single-year-age bars renders as visible thin white stripes across the pyramid.

3.4.2 `plot_demog_trend()`

A generic time-series chart for any indicator tibble keyed by `nuts3_code`/`province_name`/`year` — works unmodified on the output of any indicator function above.

```
plot_demog_trend(df, value_col, region = NULL, title = NULL, ylab = NULL)
```

```
plot_demog_trend(ai, "aging_index", region = "A Coruna")
```

3.5 Lexis diagrams: `plot_lexis_diagram()`

A Lexis diagram plots age (vertical axis) against calendar year (horizontal axis), shaded by mortality rate, with birth-cohort diagonals overlaid. Every point along one diagonal line represents the same birth cohort, aging one year for every calendar year that passes:

$$\text{age} = \text{year} - \text{cohort} \tag{33}$$

```
plot_lexis_diagram(mx_df, province, sex = c("total", "female", "male"),
                   log_scale = TRUE, rate_per = 1000,
                   cohort_lines = TRUE, cohort_step = 10, title = NULL)
```

```
rates <- compute_death_rates(deaths$data_provinces, exposure$data)
plot_lexis_diagram(rates$mx_1x1, province = "A Coruna")
plot_lexis_diagram(rates$mx_1x1, province = "Madrid", cohort_lines = FALSE)
```

Mortality is color-scaled on a log10 axis by default (`log_scale = TRUE`), the standard convention for Lexis surfaces since m_x spans several orders of magnitude across ages. Grey cells mean no rate could be computed for that age×year cell — almost always because the source population data has no single-year age breakdown for the oldest ages in the earliest years of the series (so exposure, the rate's denominator, is missing there), not a defect in the calculation; this typically appears as a rectangular block in the upper-left (oldest ages, earliest years) and clears once the source data's age detail becomes complete. When present, grey is given its own "Missing data" legend key via a small dummy point layer, since a continuous color scale cannot show an `na.value` swatch on its own.

3.6 Worked examples

The nine scripts summarized below live in `examples/` and are fully runnable end to end (each retrieves live data with the package's own functions before plotting). They double as integration tests for the indicator and charting functions documented above.

3.6.1 Birth-to-death ratio map (`birth_death_ratio_map.R`)

Replicates a reference choropleth style (binned, pink-to-green diverging scale) at the finest granularity Spain's public birth/death data actually supports: province (municipality-level birth/death tables do not exist in INE's MNPN/MNPD operations).

```
library(inedemogR)

vitals <- get_ine_demog(indicator = c("births_total", "deaths_total"), year = 2023)
```

```r
ratio <- birth_death_ratio(vitals)

map_indicator(
  ratio, value_col = "birth_death_ratio", geo_level = "province",
  breaks = c(0.25, 0.5, 0.75, 1, 1.5, 2, 3), palette = "PiYG",
  legend_title = "Births per death",
  title = "Birth-to-death ratio by province, Spain (2023)"
)
```

3.6.2 Andalusian population pyramids (`andalusia_pyramids.R`)

Builds an eight-panel `facet_wrap()` grid of population pyramids for Andalusia's eight provinces (all sharing `nuts2_code == "ES61"`), directly in **ggplot2** syntax rather than calling `plot_population_pyramid()` eight times.

```r
library(inedemogR); library(dplyr); library(tidyr); library(ggplot2)

pop <- get_ine_population()
latest_year <- 2024
andalusia <- pop$data |> filter(nuts2_code == "ES61", year == latest_year)

pyramid_data <- andalusia |>
  select(province_name, age, female, male) |>
  pivot_longer(c(female, male), names_to = "sex", values_to = "count") |>
  mutate(count = if_else(sex == "male", -count, count))

ggplot(pyramid_data, aes(x = age, y = count, fill = sex)) +
  geom_col(width = 1) +
  coord_flip() +
  facet_wrap(~province_name, nrow = 2, ncol = 4, scales = "free_x") +
  scale_y_continuous(labels = abs) +
  scale_fill_manual(values = c(female = "#D4667A", male = "#4477AA")) +
  labs(x = "Age", y = "Population", fill = "Sex",
       title = paste("Population pyramids, Andalusian provinces,", latest_year)) +
  theme_minimal()
```

3.6.3 Madrid/Barcelona Lexis diagrams (`madrid_barcelona_lexis.R`)

`plot_lexis_diagram()` handles one province per call; a two-panel comparison is built directly with the pipeline functions plus `facet_wrap()`.

```r
library(inedemogR); library(dplyr); library(ggplot2)

pop <- get_ine_population()
deaths <- get_ine_deaths()
exposure <- compute_exposure(pop$data, deaths$data_provinces)
rates <- compute_death_rates(deaths$data_provinces, exposure$data)

mx <- rates$mx_1x1 |> filter(province_name %in% c("Madrid", "Barcelona")) |>
  mutate(mx_per_1000 = mx_total * 1000)

ggplot(mx, aes(x = year, y = age, fill = mx_per_1000)) +
  geom_tile() +
  scale_fill_viridis_c(
    name = "Mortality rate\n(per 1,000, log scale)", trans = "log10",
```

```
    na.value = "grey50", labels = scales::label_number(accuracy = 0.1)
  ) +
  facet_wrap(~province_name, nrow = 1, ncol = 2) +
  labs(x = "Year", y = "Age", title = "Lexis diagrams: Madrid and Barcelona") +
  theme_minimal()
```

3.6.4 Life expectancy comparison (`life_expectancy_comparison.R`)

Combines a multi-province time series with a national choropleth for one year, both from the same life-table pipeline run once for every province.

```
library(inedemogR); library(dplyr); library(ggplot2)

pop <- get_ine_population()
deaths <- get_ine_deaths()
exposure <- compute_exposure(pop$data, deaths$data_provinces)
rates <- compute_death_rates(deaths$data_provinces, exposure$data)
lt <- build_life_tables(rates$mx_1x1)
le <- life_expectancy_summary(lt$fltper, sex = "female")

provinces_to_compare <- c("Madrid", "Barcelona", "Sevilla", "A Coruna")
ggplot(le |> filter(province_name %in% provinces_to_compare),
       aes(x = year, y = e0, color = province_name)) +
  geom_line(linewidth = 0.8) + geom_point(size = 1.5) +
  labs(x = "Year", y = "Life expectancy at birth (e0)", color = "Province") +
  theme_minimal()

map_life_expectancy(le, year = 2024, sex = "female",
                    title = "Life expectancy at birth by province, Spain")
```

3.6.5 Natural increase analysis (`natural_increase_analysis.R`)

Visualizes the "demographic crossover" (CDR overtaking CBR) and the resulting rate of natural increase.

```
library(inedemogR); library(dplyr); library(ggplot2)

births <- get_ine_births(); deaths <- get_ine_deaths(); pop <- get_ine_population()
cbr <- crude_birth_rate(births$data, pop$data)
cdr <- crude_death_rate(deaths$data_provinces, pop$data)
rni <- rate_of_natural_increase(cbr, cdr)

provinces_to_compare <- c("Madrid", "Barcelona", "Sevilla", "A Coruna")
ggplot(rni |> filter(province_name %in% provinces_to_compare),
       aes(x = year, y = rni, color = province_name)) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "grey50") +
  geom_line(linewidth = 0.8) + geom_point(size = 1.5) +
  labs(x = "Year", y = "Rate of natural increase (per 1,000)", color = "Province") +
  theme_minimal()
```

3.6.6 Fertility and infant mortality analysis (`fertility_infant_mortality_analysis.R`)

```
library(inedemogR); library(dplyr); library(ggplot2)

births <- get_ine_births(); deaths <- get_ine_deaths(); pop <- get_ine_population()
```

```
gfr <- general_fertility_rate(births$data, pop$data)
imr <- infant_mortality_rate(deaths$data_provinces, births$data)

provinces_to_compare <- c("Madrid", "Barcelona", "Sevilla", "A Coruna")
ggplot(gfr |> filter(province_name %in% provinces_to_compare),
       aes(x = year, y = gfr, color = province_name)) +
  geom_line(linewidth = 0.8) + geom_point(size = 1.5) +
  labs(x = "Year", y = "General fertility rate (per 1,000 women 15-49)",
       color = "Province") +
  theme_minimal()
```

3.6.7 Fertility indicators analysis (`fertility_indicators_analysis.R`)

Exercises the full ASFR-based fertility schedule (Section 3.1.6): the age schedule itself, TFR/MAC trends, and the GRR-vs-NRR mortality-adjustment gap, for a comparison set of provinces.

```
library(inedemogR); library(dplyr); library(ggplot2)

births_age <- get_ine_births_by_age()
pop <- get_ine_population()
deaths <- get_ine_deaths()
asfr <- age_specific_fertility_rate(births_age$data, pop$data)
tfr <- total_fertility_rate(asfr)
mac <- mean_age_at_childbearing(asfr)
grr <- gross_reproduction_rate(asfr)

exposure <- compute_exposure(pop$data, deaths$data_provinces)
rates <- compute_death_rates(deaths$data_provinces, exposure$data)
lt <- build_life_tables(rates$mx_1x1)
nrr <- net_reproduction_rate(asfr, lt$fltper)

latest_year <- 2024
provinces_to_compare <- c("Madrid", "Barcelona", "Sevilla", "A Coruna")

ggplot(asfr |> filter(province_name %in% provinces_to_compare, year == latest_year),
       aes(x = age, y = asfr_per_1000, color = province_name)) +
  geom_line(linewidth = 0.8) + geom_point(size = 1.2) +
  labs(x = "Age of mother", y = "ASFR (births per 1,000 women of that age)",
       color = "Province", title = paste("ASFR schedule,", latest_year)) +
  theme_minimal()

ggplot(tfr |> filter(province_name %in% provinces_to_compare),
       aes(x = year, y = tfr, color = province_name)) +
  geom_line(linewidth = 0.8) + geom_point(size = 1.5) +
  geom_hline(yintercept = 2.1, linetype = "dashed", color = "grey40") +
  labs(x = "Year", y = "Total fertility rate", color = "Province",
       subtitle = "Dashed line: replacement-level TFR (2.1)") +
  theme_minimal()
```

Live-verified results: TFR ranges roughly 1.0–1.3 across the four provinces (Madrid highest, A Coruña lowest), well below replacement, and Madrid's mean age at childbearing comes to 32.8 years — consistent with Spain's known late-childbearing pattern.

3.6.8 Mortality decomposition analysis (`mortality_decomposition_analysis.R`)

Exercises `decompose_life_expectancy()` (Section 3.2) on both a cross-sectional gap (two provinces, one year) and a temporal gap (one province, two years), comparing Arriaga's and Pollard's methods.

```
library(inedemogR); library(dplyr); library(ggplot2)

pop <- get_ine_population(); deaths <- get_ine_deaths()
exposure <- compute_exposure(pop$data, deaths$data_provinces)
rates <- compute_death_rates(deaths$data_provinces, exposure$data)
lt <- build_life_tables(rates$mx_1x1)

latest_year <- 2024; earliest_year <- 2006 # fixed years: results do not change with new INE
    releases
lt_a <- lt$fltper |> filter(province_name == "A Coruna", year == latest_year)
lt_b <- lt$fltper |> filter(province_name == "Madrid", year == latest_year)
decomp <- decompose_life_expectancy(lt_a, lt_b, method = "arriaga")

ggplot(decomp, aes(x = age, y = contribution, fill = contribution > 0)) +
  geom_col(width = 1) +
  scale_fill_manual(values = c(`TRUE` = "steelblue", `FALSE` = "firebrick"), guide = "none") +
  labs(x = "Age", y = "Contribution to e0 gap (years)",
       title = "Madrid vs. A Coruna life expectancy gap by age") +
  theme_minimal()
```

With the frozen inputs: A Coruña 2024 female $e_0 = 86.95$, Madrid 2024 $e_0 = 87.86$, a gap of 0.91 years; the temporal comparison (A Coruña, 2006–2024) gives a gain of 2.85 years.

3.6.9 Abridged life tables analysis (`abridged_life_tables_analysis.R`)

Cross-checks `build_abridged_life_tables()` (Section 2.3.4) against the exact single-year `build_life_tables()` on the same underlying rates, then charts the abridged survivorship curve and mortality schedule.

```
library(inedemogR); library(dplyr); library(ggplot2)

pop <- get_ine_population(); deaths <- get_ine_deaths()
exposure <- compute_exposure(pop$data, deaths$data_provinces)
rates <- compute_death_rates(deaths$data_provinces, exposure$data)

lt_1x1 <- build_life_tables(rates$mx_1x1)
lt_abridged <- build_abridged_life_tables(rates$mx_1x1, rates$mx_5x1)

latest_year <- 2024
lt_1x1_one <- lt_1x1$fltper |> filter(province_name == "A Coruna", year == latest_year)
lt_abridged_one <- lt_abridged$fltper |> filter(province_name == "A Coruna", year == latest_year)

survivorship <- bind_rows(
  lt_1x1_one |> transmute(age = age, lx = lx, table = "Single-year (1x1)"),
  lt_abridged_one |> transmute(age = age_start, lx = lx, table = "Abridged (5x1)")
)
ggplot(survivorship, aes(x = age, y = lx, color = table)) +
  geom_line(linewidth = 0.8) +
  labs(x = "Age", y = "Survivors (out of 100,000 births)", color = NULL) +
  theme_minimal()
```

With the frozen inputs, A Coruña 2024 abridged female $e_0 = 86.95$, within 0.01 years of the single-year

table. The two tables share their inputs, so this is a consistency check rather than external validation.

Appendix

A.1 Function reference

Function	Purpose
`get_ine_demog()`	System A: totals for one or more indicators, any geography
`get_ine_geo()`	Province/municipality boundary geometries
`list_ine_indicators()`	The indicator registry
`update_ine_data()`	Refresh cached System A indicators if INE published newer data
`plot_ine_map()`	Highlight-region diagnostic map
`map_indicator()`	General choropleth for any `GEOID`-keyed tibble
`get_ine_population()`	Province population by single-year age/sex
`get_ine_births()`	Province births by sex
`get_ine_births_by_age()`	Province births by single year of age of mother and sex of newborn
`get_ine_deaths()`	Province deaths by sex, age-specific and age-less
`compute_exposure()`	Exposure-to-risk $E(x,t)$
`compute_death_rates()`	Central death rates: 1x1, 5x1, ASDR
`andreev_kingkade_a0()`	Age-0 life-table parameter a_0
`build_life_table()` / `build_life_tables()`	Period life table, one sex/one province, or all
`build_abridged_life_table()` / `build_abridged_life_tables()`	Abridged (5-year age group) period life table, one sex/one province, or all
`download_ine_data()`	Export the whole pipeline to CSV/HMD-txt files
`validate_population()`, `validate_births()`, `validate_births_by_age()`, `validate_deaths()`, `validate_deaths_age()`, `validate_exposure()`, `validate_death_rates()`, `validate_life_table()`, `validate_abridged_life_table()`	Quality-control checks for each pipeline stage
`age_dependency_ratio()`	Youth/old-age/total dependency ratios
`aging_index()`	Elderly per 100 children
`sex_ratio()`	Males per 100 females
`crude_birth_rate()` / `crude_death_rate()`	CBR / CDR
`general_fertility_rate()`	GFR (births per 1,000 women 15-49)
`age_specific_fertility_rate()`	ASFR/ASFR-female schedule by single year of age
`total_fertility_rate()`	TFR (children per woman)
`mean_age_at_childbearing()`	MAC
`gross_reproduction_rate()`	GRR (daughters per woman, ignoring mortality)
`net_reproduction_rate()`	NRR (daughters per woman, mortality-adjusted)
`infant_mortality_rate()`	IMR (infant deaths per 1,000 live births)
`rate_of_natural_increase()`	RNI = CBR − CDR

Function	Purpose
`birth_death_ratio()`	Births per death, System A totals
`life_expectancy_summary()`	e_0/e_{65} from a full life-table output
`life_expectancy()`	e_0/e_{65}, one-province convenience wrapper
`decompose_life_expectancy()`	Age-specific decomposition of an e_0 gap (Arriaga/Pollard)
`plot_population_pyramid()`	Mirrored age/sex pyramid
`plot_demog_trend()`	Generic indicator time series
`plot_lexis_diagram()`	Age $\times$ year mortality surface with cohort diagonals
`map_life_expectancy()`	Life-expectancy choropleth (bridges System B to System A geometry)

A.2 INE data sources used

Series	Table ID	Operation
Population (municipality)	29005	Cifras oficiales del padrón (DPOP)
Population (province, by age/sex)	56945	Padrón Municipal Continuo (ECP)
Births (province)	6506	Movimiento Natural de la Población — Nacimientos (MNPN)
Births (province, age of mother)	6508	Movimiento Natural de la Población — Nacimientos (MNPN)
Deaths (province, age-less)	6545	Movimiento Natural de la Población — Defunciones (MNPD)
Deaths (province, age-specific)	6547	Movimiento Natural de la Población — Defunciones (MNPD)

A.3 Methodology references

- Human Mortality Database. *Methods Protocol for the Human Mortality Database*, Version 6. University of California, Berkeley, and Max Planck Institute for Demographic Research.
- Andreev, E. M., & Kingkade, W. W. (2015). Average age at death in infancy and infant mortality level: Reconsidering the Coale-Demeny formulas at current levels of low mortality. *Demographic Research*, 33, 363–390.
- Eurostat. *Revision of the European Standard Population* (2013 edition).
- Preston, S. H., Heuveline, P., & Guillot, M. (2001). *Demography: Measuring and Modeling Population Processes*. Blackwell Publishing. (Abridged life-table construction, Ch. 3.)
- Arriaga, E. E. (1984). Measuring and explaining the change in life expectancies. *Demography*, 21(1), 83–96.
- Pollard, J. H. (1988). On the decomposition of changes in expectation of life and differentials in life expectancy. *Demography*, 25(2), 265–276.
- Instituto Nacional de Estadística (INE). *Fenómenos demográficos*. https://www.ine.es.