Demographic Projections in the Colombian Context with the PyCCM Library

Juliana Echeverri Jaramillo, Charlie Rahal, and Jiani Yan

Banco de la República and the University of Oxford

Populorum, April 15th, 2026

Who are we?

Who are we?

Who are we?

Who are we?

Who are we?

Between us, we have complementary expertise in:

- Demography and the Colombian population profile (Juliana)

Who are we?

Between us, we have complementary expertise in:

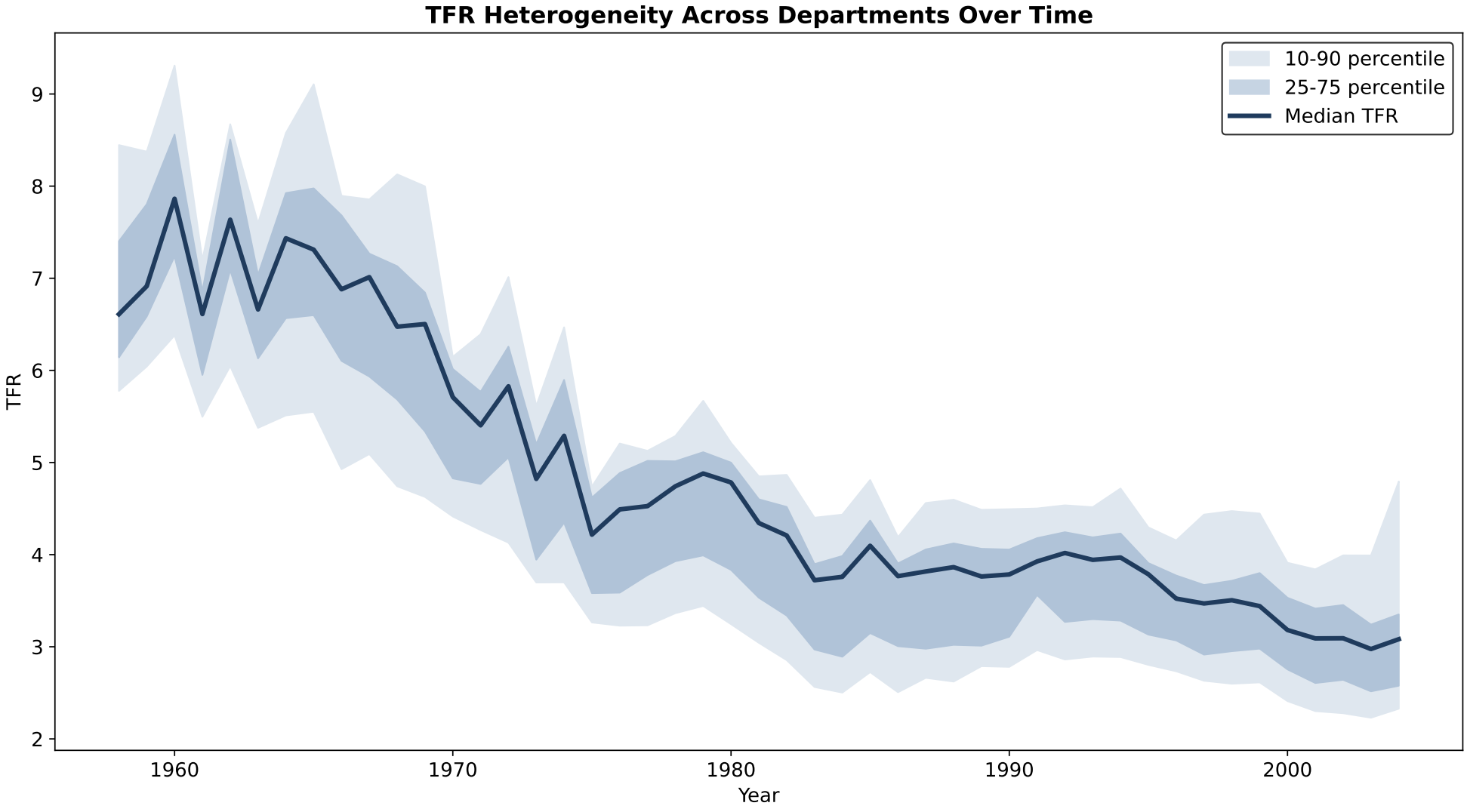
- Demography and the Colombian population profile (Juliana)
- Demography and population data science (Charlie)

Who are we?

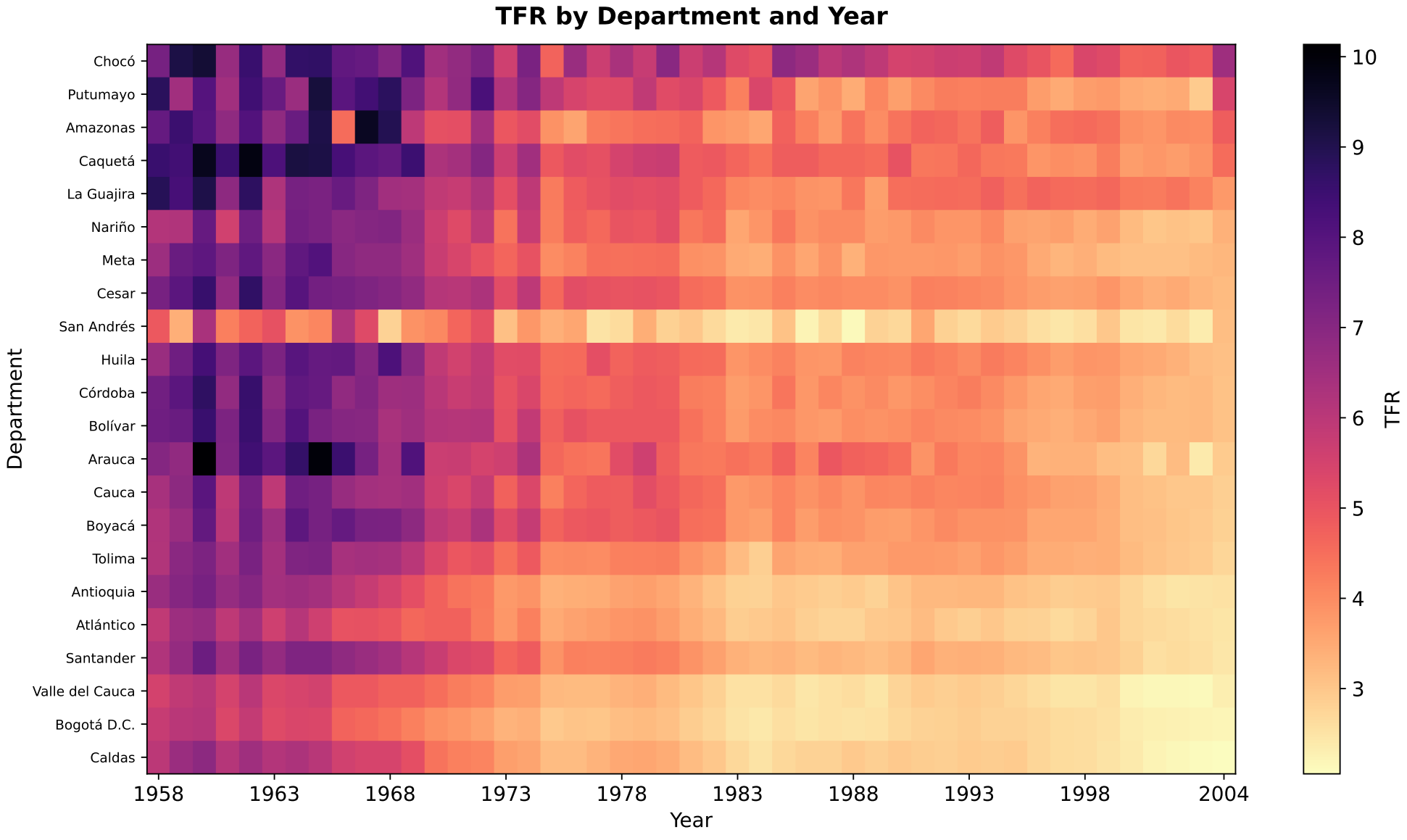
Between us, we have complementary expertise in:

- Demography and the Colombian population profile (Juliana)
- Demography and population data science (Charlie)
- Demography and software engineering (Jiani)

What's the Motivation?



What's the Motivation?



The Brief

Build an original library to create projections of the Colombian population structure, which:

The Brief

Build an original library to create projections of the Colombian population structure, which:

- Accounted for various types of data provision, including:

The Brief

Build an original library to create projections of the Colombian population structure, which:

- Accounted for various types of data provision, including:
  - Time-varying measurement of things (e.g. 2018, ..., 2024)

The Brief

Build an original library to create projections of the Colombian population structure, which:

- Accounted for various types of data provision, including:
  - Time-varying measurement of things (e.g. 2018, ..., 2024)
  - Multiple different data originators (e.g. census and vitals)

The Brief

Build an original library to create projections of the Colombian population structure, which:

- Accounted for various types of data provision, including:
  - Time-varying measurement of things (e.g. 2018, ..., 2024)
  - Multiple different data originators (e.g. census and vitals)
- Allows for user-specified assumptions about key processes.

The Brief

Build an original library to create projections of the Colombian population structure, which:

- Accounted for various types of data provision, including:
  - Time-varying measurement of things (e.g. 2018, ..., 2024)
  - Multiple different data originators (e.g. census and vitals)
- Allows for user-specified assumptions about key processes.
- Makes projections for an indefinite horizon.

The Brief

Build an original library to create projections of the Colombian population structure, which:

- Accounted for various types of data provision, including:
  - Time-varying measurement of things (e.g. 2018, ..., 2024)
  - Multiple different data originators (e.g. census and vitals)
- Allows for user-specified assumptions about key processes.
- Makes projections for an indefinite horizon.
- Projects at 34 levels: 32 DPTOs, Bogota, and Total.

The Brief

Build an original library to create projections of the Colombian population structure, which:

- Accounted for various types of data provision, including:
  - Time-varying measurement of things (e.g. 2018, ..., 2024)
  - Multiple different data originators (e.g. census and vitals)
- Allows for user-specified assumptions about key processes.
- Makes projections for an indefinite horizon.
- Projects at 34 levels: 32 DPTOs, Bogota, and Total.
- Is *similar* to DANE's apprach, but gives user control.

Importantly, it should be **open**!

Importantly, it should be **open**!

(Including all data inputs: thanks to Juliana, incredible work!)

What we've Done

- We think we've gone above and beyond this!

What we've Done

- We think we've gone above and beyond this!
- Create an open Python 3+ library with minimal dependancies.

What we've Done

- We think we've gone above and beyond this!
- Create an open Python 3+ library with minimal dependancies.
- It relies on as little as possible outside of the 'standard library'.
  - Some things – e.g. `pandas`, `numpy`, and `streamlit` – are unavoidable but well recognised dependancies.

What we've Done

- We think we've gone above and beyond this!
- Create an open Python 3+ library with minimal dependancies.
- It relies on as little as possible outside of the 'standard library'.
  - Some things – e.g. `pandas`, `numpy`, and `streamlit` – are unavoidable but well recognised dependancies.
- As modular as possible for accessibility and customisation.

What we've Done

- We think we've gone above and beyond this!
- Create an open Python 3+ library with minimal dependancies.
- It relies on as little as possible outside of the 'standard library'.
  - Some things – e.g. `pandas`, `numpy`, and `streamlit` – are unavoidable but well recognised dependancies.
- As modular as possible for accessibility and customisation.
- Has (reasonable?) defaults, but allows user-specified choices.

What we've Done

- We think we've gone above and beyond this!
- Create an open Python 3+ library with minimal dependancies.
- It relies on as little as possible outside of the 'standard library'.
  - Some things – e.g. `pandas`, `numpy`, and `streamlit` – are unavoidable but well recognised dependancies.
- As modular as possible for accessibility and customisation.
- Has (reasonable?) defaults, but allows user-specified choices.
- Actively developed, documented, and tracked on GitHub.

What we've Done

- We think we've gone above and beyond this!
- Create an open Python 3+ library with minimal dependancies.
- It relies on as little as possible outside of the 'standard library'.
  - Some things – e.g. `pandas`, `numpy`, and `streamlit` – are unavoidable but well recognised dependancies.
- As modular as possible for accessibility and customisation.
- Has (reasonable?) defaults, but allows user-specified choices.
- Actively developed, documented, and tracked on GitHub.
- Can be publicly released under a GNU GPL 3.0 license.

crahal Some attempted updates for better parralisation (not enough to close ... cbb8bf1 · 4 months ago 69 Commits

course	remove redundant files	6 months ago
data	Add mortality_improvements_example.csv	5 months ago
docs	remove redundant files	6 months ago
src	Some attempted updates for better parralisation (not en...	4 months ago
tests	update the documents to include projections and correct ...	6 months ago
.gitignore	Reorganize documentation structure and update all test s...	6 months ago
LICENSE	Initial commit	10 months ago
README.md	Adding Jiani to the citation! Maybe we could do a .CFT in t...	6 months ago
config.yaml	Fix #24; about age bins being overridden (taken out of ya...	4 months ago
pyproject.toml	source file changes from code reviewing	6 months ago

README GPL-3.0 license

PyCCM

License GNUGPL3.0 | Python 3.7 | 3.8 | 3.9 | 3.10 | 3.11 | Tests 211 passing | Quality 4.7/5 stars | OS Linux | Windows | macOS

Population projection pipeline with single-year ages ("unabridged") and joint parameter sweeps for fertility & mortality assumptions. Supports parallelized scenario runs and consolidated outputs (life tables, ASFR, Leslie matrices, projections).

> **Repo name:** `PyCCM`
> **Entry point:** `src/main_compute.py`
> **Config:** `config.yaml` (root)
> **Data dir:** `data/` (CSV inputs)
> **Outputs:** `results_unabridged/` (default)
> **Status:** Production Ready (211/211 tests passing)

A Python library for making cohort component based projections.

Readme
GPL-3.0 license
Activity
5 stars
0 watching
2 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Contributors 2

crahal
vallerrr J Yan

Languages

Jupyter Notebook 73.1%
Python 26.9%

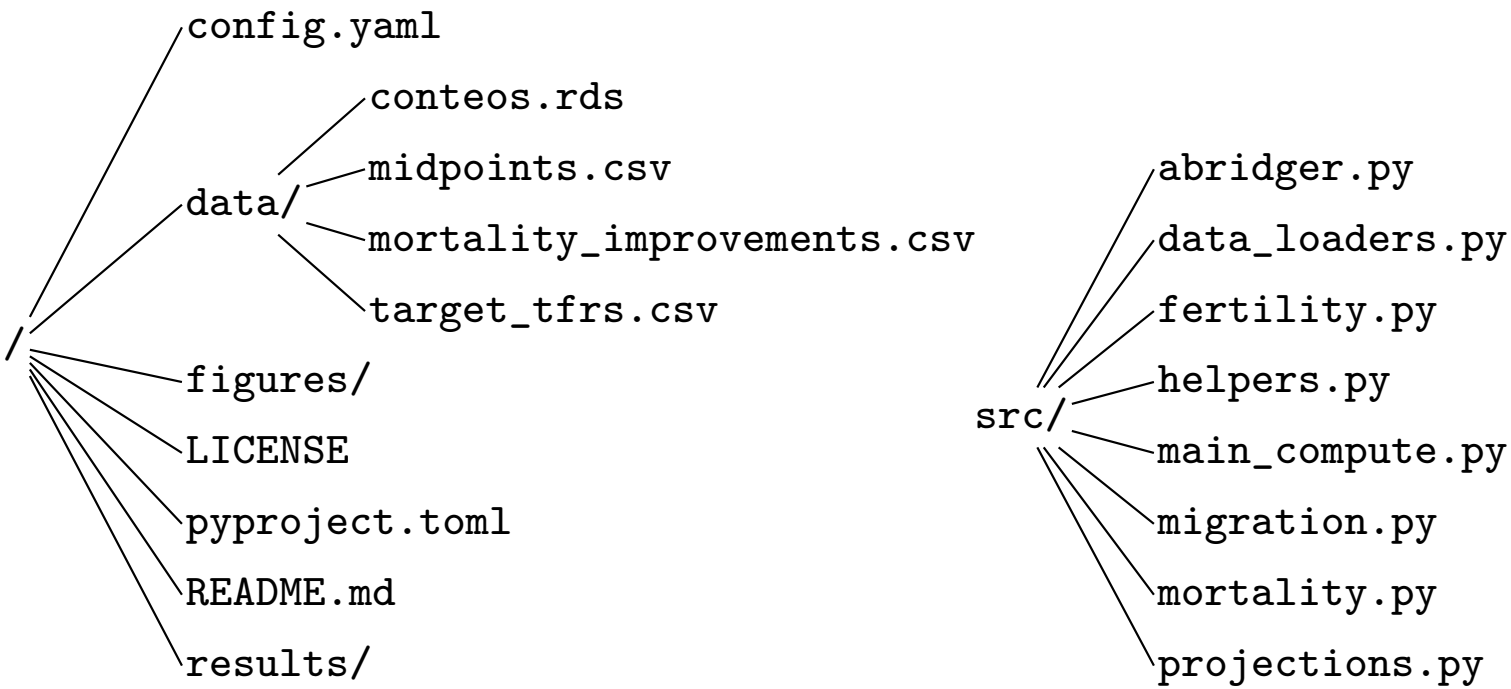
Suggested workflows

Based on your tech stack

Python package Configure

Create and test a Python package on multiple Python versions.

Project Structure



The Data

In a Colombian context, the data has the specific properties:

The Data

In a Colombian context, the data has the specific properties:

- Census data collected in 2018.

The Data

In a Colombian context, the data has the specific properties:

- Census data collected in 2018.
- EEVV data collected through to 2023.

The Data

In a Colombian context, the data has the specific properties:

- Census data collected in 2018.
- EEVV data collected through to 2023.
    - Note: EEVV and Census, do not concur...

The Data

In a Colombian context, the data has the specific properties:

- Census data collected in 2018.
- EEVV data collected through to 2023.
    - Note: EEVV and Census, do not concur...
- Migration data collected through 2024.

The Data

In a Colombian context, the data has the specific properties:

- Census data collected in 2018.
- EEVV data collected through to 2023.
    - Note: EEVV and Census, do not concur...
- Migration data collected through 2024.
- Data is 'semi-abridged':

The Data

In a Colombian context, the data has the specific properties:

- Census data collected in 2018.
- EEVV data collected through to 2023.
    - Note: EEVV and Census, do not concur...
- Migration data collected through 2024.
- Data is 'semi-abridged':
    - It has 0-1, 2-4, and then 5 year intervals through 90+.

The Data (Cont.)

- For some migration statistics, it ends at 70+ or 80+.

The Data (Cont.)

- For some migration statistics, it ends at 70+ or 80+.
- Naturally, immense difficulty with migration measurement.

The Data (Cont.)

- For some migration statistics, it ends at 70+ or 80+.
- Naturally, immense difficulty with migration measurement.
- For some SEXO/FUENTE/DPTO, VALORs are unattributable.

The Data (Cont.)

- For some migration statistics, it ends at 70+ or 80+.
- Naturally, immense difficulty with migration measurement.
- For some SEXO/FUENTE/DPTO, VALORs are unattributable.
- There is – essentially – an OMISSION uncertainty.

The Data (Cont.)

- For some migration statistics, it ends at 70+ or 80+.
- Naturally, immense difficulty with migration measurement.
- For some SEXO/FUENTE/DPTO, VALORs are unattributable.
- There is – essentially – an OMISSION uncertainty.

Immense thanks to Juliana for curating a brilliant `conteos.rds`!

Data Processing

- When we have no EDAD, evenly distribute.

Data Processing

- When we have no EDAD, evenly distribute.
- When we have omisssions, create three different projections:

Data Processing

- When we have no EDAD, evenly distribute.
- When we have omisssions, create three different projections:
    1. Low: Assume omissions are as low as possible within range.

Data Processing

- When we have no EDAD, evenly distribute.
- When we have omisssions, create three different projections:
  1. Low: Assume omissions are as low as possible within range.
  2. Mid: Assume omissions are in middle of range.

Data Processing

- When we have no EDAD, evenly distribute.
- When we have omisssions, create three different projections:
  1. Low: Assume omissions are as low as possible within range.
  2. Mid: Assume omissions are in middle of range.
  3. High: Assume omissions are as high as possible within range.

Data Processing

- When we have no EDAD, evenly distribute.
- When we have omisssions, create three different projections:
  1. Low: Assume omissions are as low as possible within range.
  2. Mid: Assume omissions are in middle of range.
  3. High: Assume omissions are as high as possible within range.
- Allow user to stochastically draw (k) times:

Data Processing

- When we have no EDAD, evenly distribute.
- When we have omisssions, create three different projections:
  1. Low: Assume omissions are as low as possible within range.
  2. Mid: Assume omissions are in middle of range.
  3. High: Assume omissions are as high as possible within range.
- Allow user to stochastically draw (k) times:
  1. From a PERT distribution;

Data Processing

- When we have no EDAD, evenly distribute.
- When we have omisssions, create three different projections:
  1. Low: Assume omissions are as low as possible within range.
  2. Mid: Assume omissions are in middle of range.
  3. High: Assume omissions are as high as possible within range.
- Allow user to stochastically draw (k) times:
  1. From a PERT distribution;
  2. From a Uniform distribution;

Data Processing

- When we have no EDAD, evenly distribute.
- When we have omisssions, create three different projections:
  1. Low: Assume omissions are as low as possible within range.
  2. Mid: Assume omissions are in middle of range.
  3. High: Assume omissions are as high as possible within range.
- Allow user to stochastically draw (k) times:
  1. From a PERT distribution;
  2. From a Uniform distribution;
  3. From a Beta distribution.

YAML configuration file

Key sections control paths, behavior, horizon, and assumptions.

- **paths** – input/output file locations.
- **housekeeping** – clean run, diagnostics toggles.
- **projections** – start/end years, death data choices.
- **fertility** – TFR target, convergence, smoother.
- **mortality** – MA smoothing, improvement targets.
- **age_bins** – expected input bins.
- **runs & parallel** – scenarios, number of processes.

Role. A single declarative record of inputs & parameterse.

Abridger: parsing and infant adjustment

Inputs. Counts with `EDADs` (e.g. `0-1`, `2--4`, ..., `90+`) and totals.

Step 1. Parse ages.

- Intervals: a–$b \mapsto [a, b]$.
- Open-ended: $a+ \mapsto [a, \infty)$ (passed through).
- Convention: a–$(a+1)$ collapsed to point a.

Step 2. Infant adjustment (`poblacion_total`).

- If 0–4 present: distribute into 0,1,2,3,4.
- If 1–4 present: distribute into 1,2,3,4.
- Weights w_a proportional to $nL_a(l_0, \ldots, l_5, a_0)$ (from survivorship).

Abridger: smoothing and tails

Step 3. Constraint system.

- Variables x_a = single-year counts.
- For each input bin B: enforce $\sum_{a \in B} x_a = y_B$ (total abridged count).

Step 4. Smooth solution.

$$\min_{x \geq 0} \ \|D_2 x\|^2 + \rho \|x\|^2 \quad \text{s.t. } Ax = b,$$

with D_2 = 2nd-difference, $\rho \approx 10^{-6}$ (ridge). Negatives clipped to 0.

Step 5. Tail harmonization (to 90+).

- Migration: split 70+, 80+ using population shares.
- Population/deaths: geometric splits, different for pop vs. deaths.

Outputs. Single-year `EDAD`, totals preserved on all input bins.

The Core Concept of a CCM

A CCM requires four key things. Three go into a ‘Leslie Matrix’:

The Core Concept of a CCM

A CCM requires four key things. Three go into a 'Leslie Matrix':

1. Survival rates from a lifetable.

The Core Concept of a CCM

A CCM requires four key things. Three go into a 'Leslie Matrix':

1. Survival rates from a lifetable.
2. Age Standardised Fertility Rates (ASFRs)

The Core Concept of a CCM

A CCM requires four key things. Three go into a 'Leslie Matrix':

1. Survival rates from a lifetable.
2. Age Standardised Fertility Rates (ASFRs)
3. Net migration.

The Core Concept of a CCM

A CCM requires four key things. Three go into a 'Leslie Matrix':

1. Survival rates from a lifetable.
2. Age Standardised Fertility Rates (ASFRs)
3. Net migration.

These go into a Leslie Matrix which is multiplied by:

The Core Concept of a CCM

A CCM requires four key things. Three go into a 'Leslie Matrix':

1. Survival rates from a lifetable.
2. Age Standardised Fertility Rates (ASFRs)
3. Net migration.

These go into a Leslie Matrix which is multiplied by:

- A population vector

Midpoint deaths (EEVV–Censo blend)

For the `death_choice = "midpoint"` option, baseline deaths are a convex combination of the two sources.

For each department d, sex s, age a at $t_0 =$ `START_YEAR`: deaths from EEVV, D^{EEVV}_{dsa}, and from Censo, D^{censo}_{dsa}.

Per–department $w_d \in [0,1]$ from `midpoints.csv`; fallback to YAML `midpoints.default_eevv_weight`. We clip w_d to $[0,1]$.

Blend (deaths only):

$$D^{\text{mid}}_{dsa} \;=\; w_d\, D^{\text{EEVV}}_{dsa} \;+\; (1 - w_d)\, D^{\text{censo}}_{dsa}.$$

Ages are aligned/grouped before blending; the result D^{mid}_{dsa} is used to build the life table for the midpoint run. The blend is fixed at t_0 and reused for all projection years for this option.

Life Tables

We create **period** life tables, allowing user to specify multiple things:

Life Tables

We create **period** life tables, allowing user to specify multiple things:

1. MA window for smoothing, default six periods.

Life Tables

We create **period** life tables, allowing user to specify multiple things:

1. MA window for smoothing, default six periods.
2. Mortality rates which improve by a specific percent:

Life Tables

We create **period** life tables, allowing user to specify multiple things:

1. MA window for smoothing, default six periods.
2. Mortality rates which improve by a specific percent:
    - These converge at a target horizon.

Life Tables

We create **period** life tables, allowing user to specify multiple things:

1. MA window for smoothing, default six periods.
2. Mortality rates which improve by a specific percent:
    - These converge at a target horizon.
    - They allow exponential or logarithmic convergeance.

Life Tables

We create **period** life tables, allowing user to specify multiple things:

1. MA window for smoothing, default six periods.
2. Mortality rates which improve by a specific percent:
    - These converge at a target horizon.
    - They allow exponential or logarithmic convergeance.
    - Allow the specification of a 'frac' convergeance (speed).

Life Tables

We create **period** life tables, allowing user to specify multiple things:

1. MA window for smoothing, default six periods.
2. Mortality rates which improve by a specific percent:
    - These converge at a target horizon.
    - They allow exponential or logarithmic convergeance.
    - Allow the specification of a 'frac' convergeance (speed).

We also allow the user to specify custom targets per DPTO.

Smoothing

Setup. Single-year ages $x \in \mathbb{Z}_{\geq 0}$ with exposures E_x and deaths D_x. Crude rates: $m_x = D_x/E_x$ (with numerical floor on E_x).

Centered log–moving average (window W).

$$\tilde{m}_x \;=\; \exp\!\left(\frac{1}{|\mathcal{N}_W(x)|}\sum_{r\in\mathcal{N}_W(x)} \log m_r\right), \quad \mathcal{N}_W(x) = \{x-\lfloor W/2\rfloor, \ldots, x+\lfloor W/2\rfloor\}$$

(edge windows shrink). Use $\tilde{m}_x$ downstream.

Life table (abridged formulas, $n_x = 1$ except open end).

$$q_x \;=\; \frac{\tilde{m}_x}{1+(1-a_x)\tilde{m}_x}, \quad p_x = 1-q_x, \quad \ell_0 = R, \;\; \ell_{x+1} = \ell_x p_x.$$

Final open interval handled in the usual way via $L_\omega = \ell_\omega/\tilde{m}_\omega$.

Life table detail: a_x for age 0–1

General rule. For most ages,

$$a_x \approx \tfrac{1}{2} n_x,$$

i.e. deaths are assumed uniformly distributed within the interval.

Exception: infants (0–1). Mortality is concentrated near birth, so special Coale–Demeny style formulas are used:

$$a_0 = \begin{cases} 0.14903 - 2.05527\, m_0, & m_0 < 0.01724, \\ 0.04667 + 3.88089\, m_0, & 0.01724 \le m_0 < 0.06891, \\ 0.31411, & m_0 \ge 0.06891, \end{cases}$$

where m_0 is the central death rate in the 0–1 interval.

Rationale. These piecewise linear adjustments better reflect observed timing of infant deaths, avoiding bias in q_0 and survivorship.

Target mortality improvements

Parameters (per unit).

- Total improvement fraction: $\theta \in (0, 1)$ (e.g. $0.10 = 10\%$ reduction).
- Convergence horizon: C years.
- Smoother type: exponential (default) or logistic.

Exponential smoother.

$$G = -\ln(1-\theta), \qquad S_t = 1 - \exp\left(-\frac{-\ln(1-\rho)}{C}\, t\right), \qquad \phi_t = e^{-GS_t},$$

with $\rho \approx 0.99$ target convergence fraction.

For year t after baseline, multiply each mortality rate by factor

$$m_x(t) \;\leftarrow\; \phi_t \cdot m_x(t).$$

Headline e0 statistics

Life expectancies are birth are *marginally* lower than NIH estimates:

Headline e0 statistics

Life expectancies are birth are *marginally* lower than NIH estimates:

- F: 78.47 (censo_2018), 78.15 (EEV), 78.36 ($\frac{1}{2}$ midpoint)
- M: 72.80 (censo_2018), 71.35 (EEV), 71.70 ($\frac{1}{2}$ midpoint)
- T: 75.23 (censo_2018), 74.41 (EEV), 74.83 ($\frac{1}{2}$ midpoint)

Headline e0 statistics

Life expectancies are birth are *marginally* lower than NIH estimates:

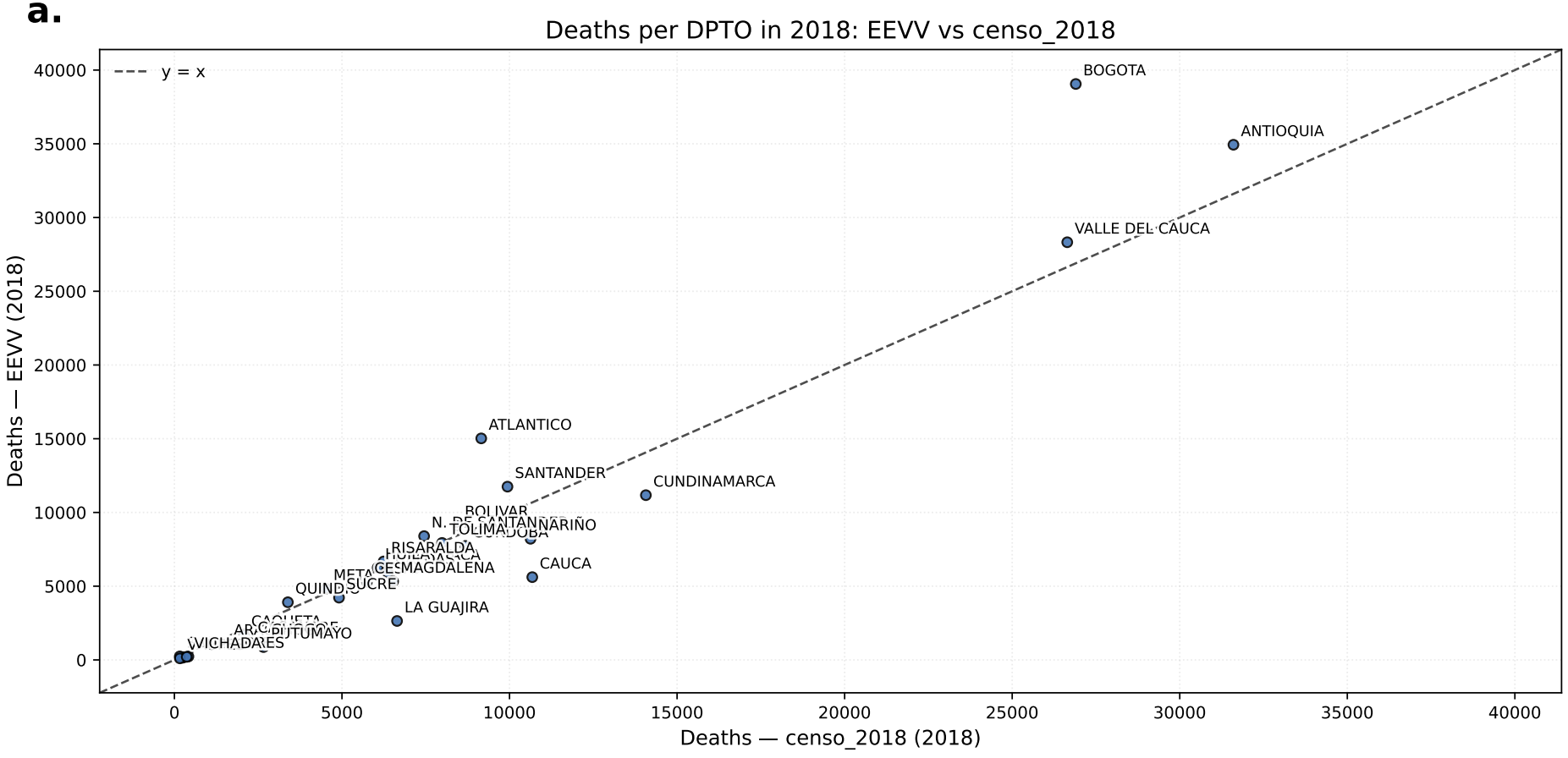
- F: 78.47 (censo_2018), 78.15 (EEV), 78.36 ($\frac{1}{2}$ midpoint)
- M: 72.80 (censo_2018), 71.35 (EEV), 71.70 ($\frac{1}{2}$ midpoint)
- T: 75.23 (censo_2018), 74.41 (EEV), 74.83 ($\frac{1}{2}$ midpoint)

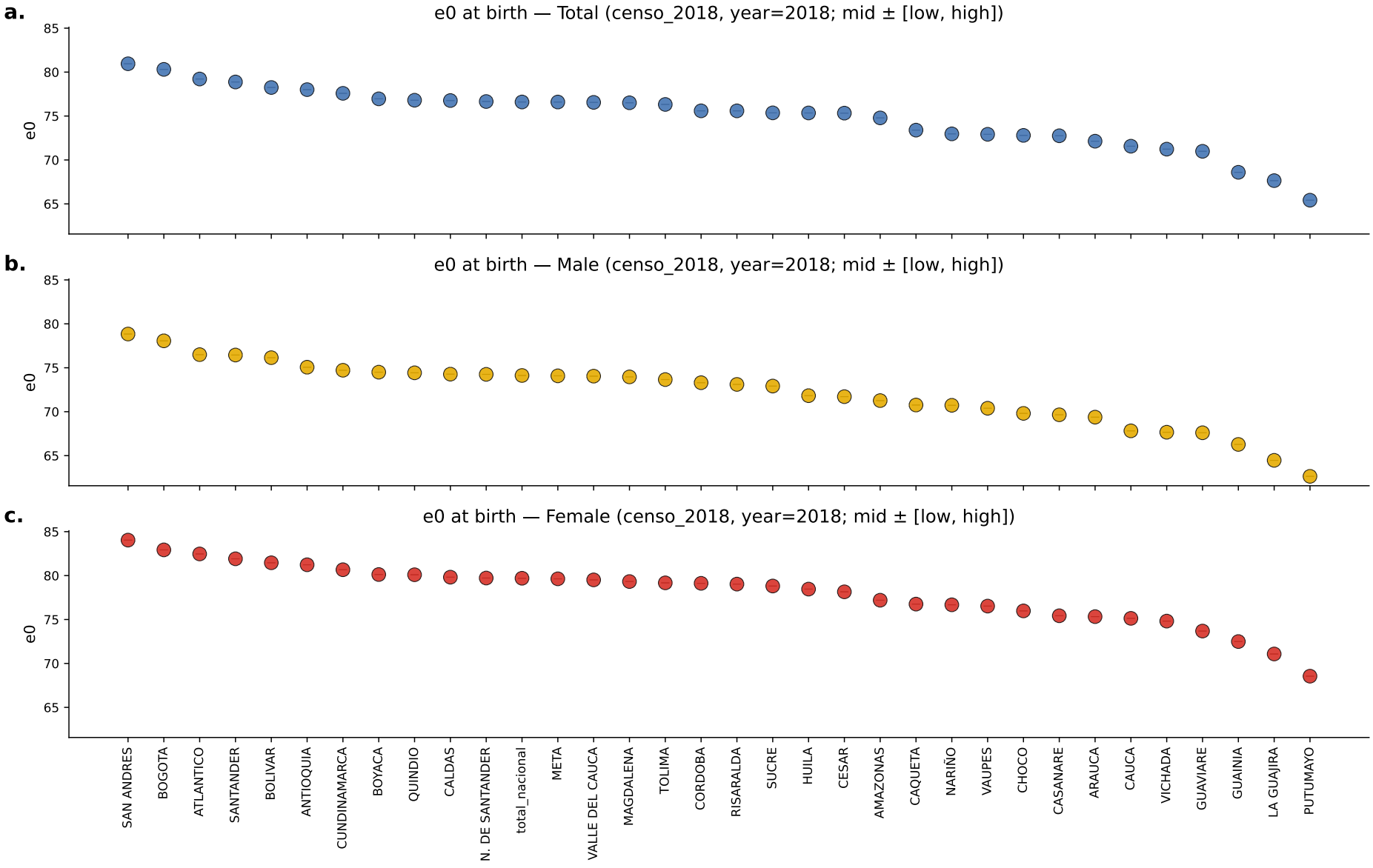
OMISSIONs relatively little impact as typically same for deaths/population: an invariant transformation unless the values differ.

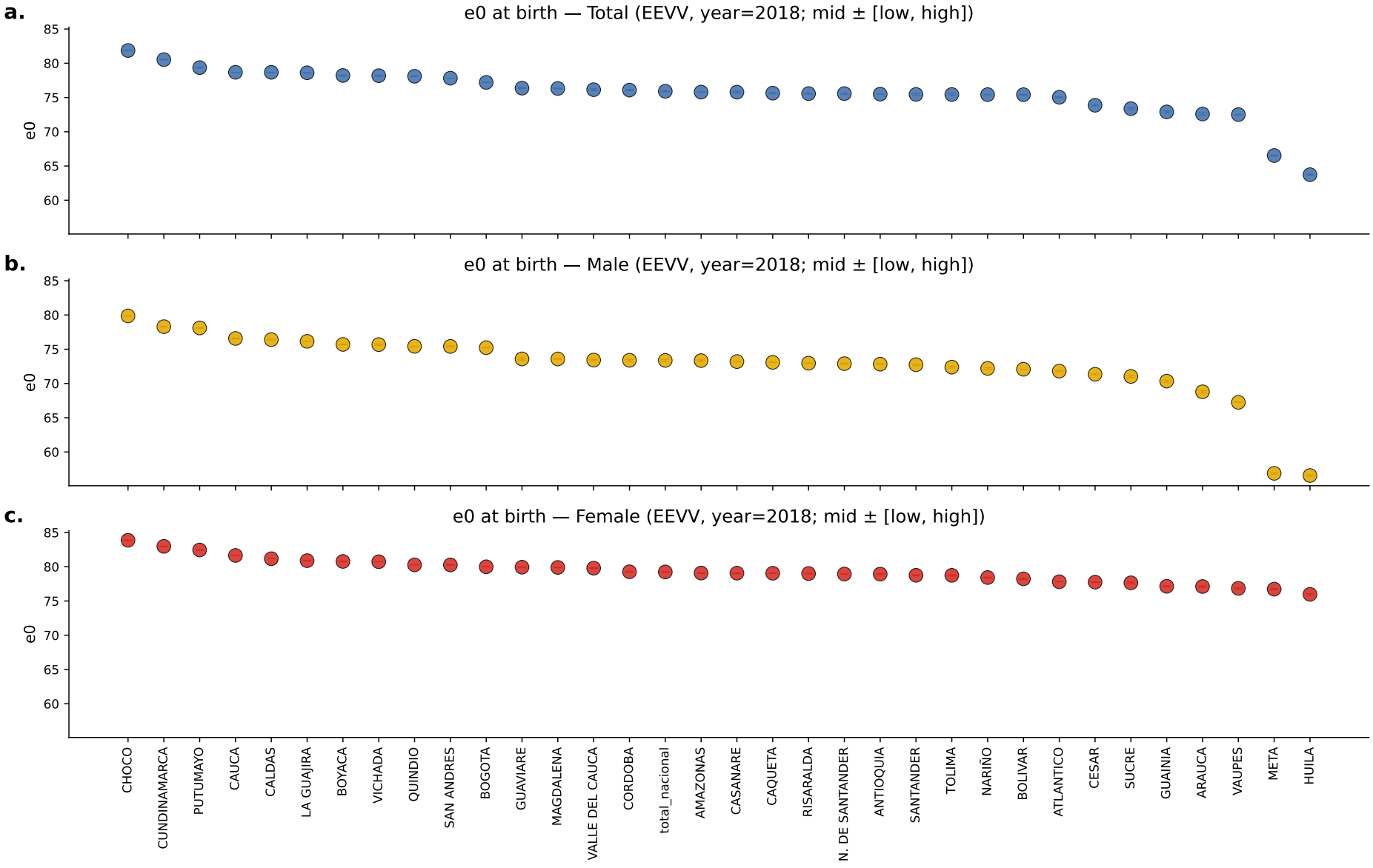
Life Table Extract (2018, Female, total_nacional)

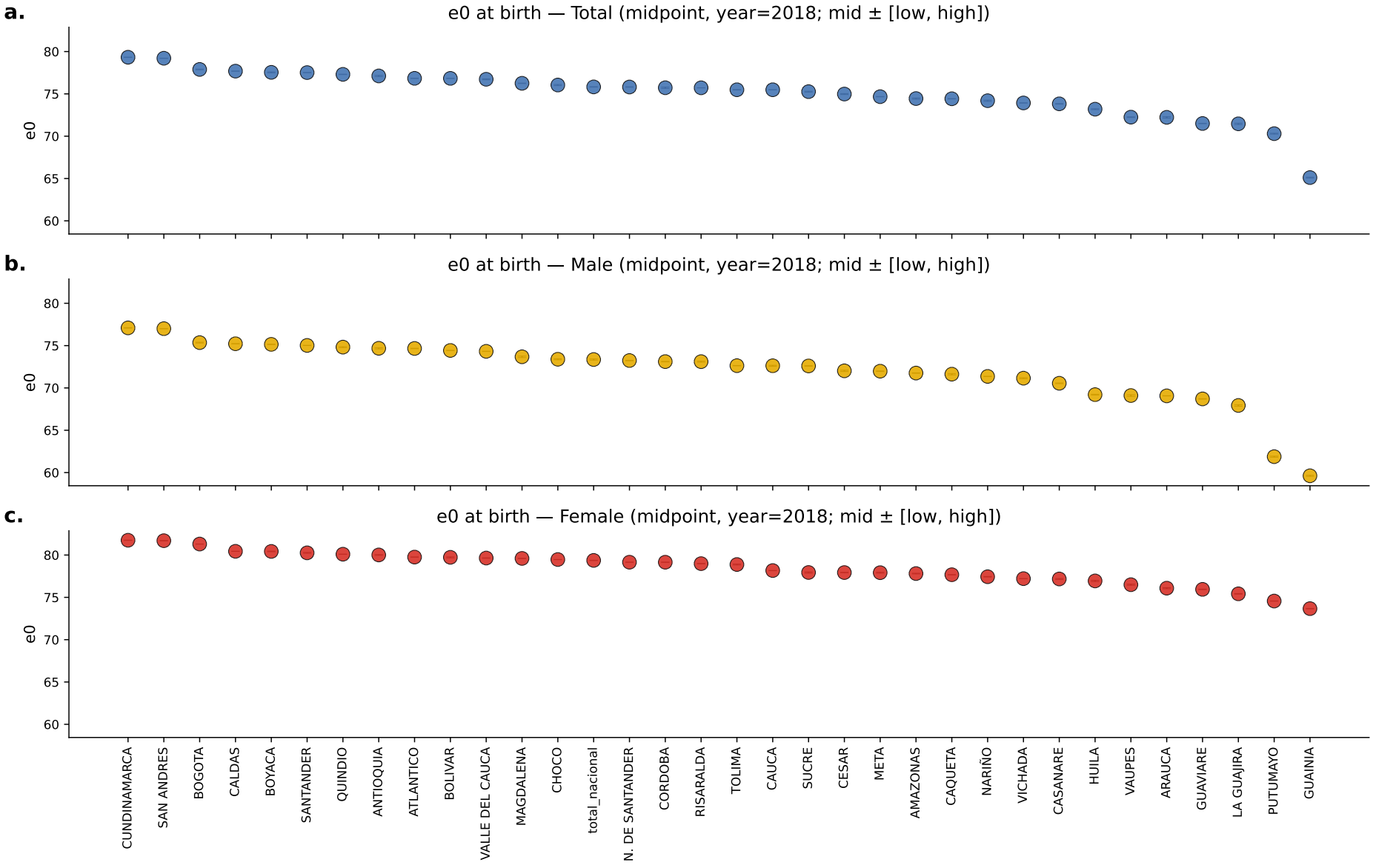
DPTO	death	impr	ma	year	Sex	EDAD	n	mx	ax	qx	px	lx	dx	Lx	Tx	ex
total	mid	0.25	3	2018	F	0	1	0.0183	0.498	0.0181	0.982	100000	1808	99093	7798074	78.36
total	mid	0.25	3	2018	F	1	1	0.0121	0.500	0.0121	0.988	98192	1184	97599	7698981	78.41
total	mid	0.25	3	2018	F	2	1	0.00173	0.500	0.00173	0.99827	97008	168	96924	7601381	78.36
total	mid	0.25	3	2018	F	3	1	0.0000521	0.500	0.0000521	0.99995	96840	5	96837	7504457	77.49
total	mid	0.25	3	2018	F	4	1	0.000000885	0.500	0.000000885	0.999999	96835	0	96835	7407620	76.50
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
total	mid	0.25	3	2018	F	85	1	0.0704	0.500	0.0680	0.932	49770	3384	48078	235871	4.74
total	mid	0.25	3	2018	F	86	1	0.0785	0.500	0.0755	0.924	46386	3504	44634	187793	4.05
total	mid	0.25	3	2018	F	87	1	0.0886	0.500	0.0848	0.915	42882	3637	41064	143159	3.34
total	mid	0.25	3	2018	F	88	1	0.102	0.500	0.0969	0.903	39246	3802	37345	102095	2.60
total	mid	0.25	3	2018	F	89	1	0.190	0.500	0.173	0.827	35443	6136	32375	64750	1.83
total	mid	0.25	3	2018	F	90+	5	0.905	2.500	1.000	0.000	29307	29307	32375	32375	1.10

a.


Deaths per DPTO in 2018: EEVV vs censo_2018
y = x
Deaths — EEVV (2018)
Deaths — censo_2018 (2018)
BOGOTA
ANTIOQUIA
VALLE DEL CAUCA
ATLANTICO
SANTANDER
CUNDINAMARCA
BOLIVAR
TOLIMA
RISARALDA
CAUCA
MAGDALENA
QUINDIO
SUCRE
LA GUAJIRA
PUTUMAYO
0
5000
10000
15000
20000
25000
30000
35000
40000


a.
e0 at birth — Total (censo_2018, year=2018; mid ± [low, high])
b.
e0 at birth — Male (censo_2018, year=2018; mid ± [low, high])
c.
e0 at birth — Female (censo_2018, year=2018; mid ± [low, high])
e0
85
80
75
70
65
SAN ANDRES
BOGOTA
ATLANTICO
SANTANDER
BOLIVAR
ANTIOQUIA
CUNDINAMARCA
BOYACA
QUINDIO
CALDAS
N. DE SANTANDER
total_nacional
META
VALLE DEL CAUCA
MAGDALENA
TOLIMA
CORDOBA
RISARALDA
SUCRE
HUILA
CESAR
AMAZONAS
CAQUETA
NARIÑO
VAUPES
CHOCO
CASANARE
ARAUCA
CAUCA
VICHADA
GUAVIARE
GUAINIA
LA GUAJIRA
PUTUMAYO


a.
e0 at birth — Total (EEVV, year=2018; mid ± [low, high])
b.
e0 at birth — Male (EEVV, year=2018; mid ± [low, high])
c.
e0 at birth — Female (EEVV, year=2018; mid ± [low, high])
e0
85
80
75
70
65
60
CHOCO
CUNDINAMARCA
PUTUMAYO
CAUCA
CALDAS
LA GUAJIRA
BOYACA
VICHADA
QUINDIO
SAN ANDRES
BOGOTA
GUAVIARE
MAGDALENA
VALLE DEL CAUCA
CORDOBA
total_nacional
AMAZONAS
CASANARE
CAQUETA
RISARALDA
N. DE SANTANDER
ANTIOQUIA
SANTANDER
TOLIMA
NARIÑO
BOLIVAR
ATLANTICO
CESAR
SUCRE
GUAINIA
ARAUCA
VAUPES
META
HUILA


a.
e0 at birth — Total (midpoint, year=2018; mid ± [low, high])
b.
e0 at birth — Male (midpoint, year=2018; mid ± [low, high])
c.
e0 at birth — Female (midpoint, year=2018; mid ± [low, high])
e0
80
75
70
65
60
CUNDINAMARCA
SAN ANDRES
BOGOTA
CALDAS
BOYACA
SANTANDER
QUINDIO
ANTIOQUIA
ATLANTICO
BOLIVAR
VALLE DEL CAUCA
MAGDALENA
CHOCO
total_nacional
N. DE SANTANDER
CORDOBA
RISARALDA
TOLIMA
CAUCA
SUCRE
CESAR
META
AMAZONAS
CAQUETA
NARIÑO
VICHADA
CASANARE
HUILA
VAUPES
ARAUCA
GUAVIARE
LA GUAJIRA
PUTUMAYO
GUAINIA

Target TFR: inputs, identification, and baseline

Files and configuration.

- Read `target_tfr_csv` with columns: DPTO_NOMBRE, Target_TFR, and (optionally) convergence years.
- If absent, use global defaults from YAML:
  $\tau_{\text{glob}} = \texttt{fertility.default\_tfr\_target}$,
  $C_{\text{glob}} = \texttt{fertility.convergence\_years}$.

Departmental targets and convergence.

$$\tau_d = \begin{cases} \text{CSV value} & \text{if present} \\ \tau_{\text{glob}} & \text{otherwise,} \end{cases} \qquad C_d = \begin{cases} \text{CSV value} & \text{if present,} \\ C_{\text{glob}} & \text{otherwise.} \end{cases}$$

National target if not given

If $\tau_{\texttt{total\_nacional}}$ is missing, compute it as an exposure–weighted mean over departments (exposures restricted to ASFR ages):

$$\tau_{\text{nat}}\big(t,\ \texttt{death\_choice}\big) \;=\; \sum_{d \neq \text{nat}} w_d(t)\, \tau_d\big(t,\ \texttt{death\_choice}\big)$$

and

$$w_d(t) \;=\; \frac{E_d(t)}{\displaystyle\sum_{j \neq \text{nat}} E_j(t)}$$

where $E_d(t)$ is female exposure by department at $t = \texttt{START\_YEAR}$.

Baseline ASFR and weights

For each unit $u \in \{d, \text{nat}\}$ and a cutoff year $t_0(u)$ (last observed for the chosen death series):

$$\text{ASFR}_a^{(0)}(u) \;=\; \frac{B_a(u)}{P_a(u)}, \qquad \text{TFR}_0(u) \;=\; \sum_a \text{ASFR}_a^{(0)}(u)\, n_a,$$

$$w_a(u) \;=\; \frac{\text{ASFR}_a^{(0)}(u)}{\text{TFR}_0(u)} \quad \text{(stored and reused as the shape over ages).}$$

If $\text{TFR}_0(u)$ is not finite, code falls back to previously stored weights for u, or to national weights if departmental ones unavailable.

Target TFR: smoothing path/ASFR reconstruction

For year $t \geq t_0(u)$, let the fractional progress to target be $S_t \in [0, 1]$.

Exponential parameters:
C_u and $\phi \in (0, 1) =$ `smoother.converge_frac`:

$$\kappa \;=\; -\frac{\ln(1-\phi)}{C_u}, \qquad S_t \;=\; 1 - e^{-\kappa(t-t_0)}.$$

Logistic parametres:
C_u, mid-fraction $m \in (0, 1)$, steepness $s > 0$:

$$x \;=\; \frac{t - t_0}{C_u} - m, \qquad S_t \;=\; \frac{1}{1 + e^{-sx}},$$

with a default s chosen so that $S_{t_0+C_u} \approx 0.99$ if not provided.

TFR trajectory and ASFRs. Let τ_u be the (departmental or national) target TFR and $\mathrm{TFR}_0(u)$ the baseline:

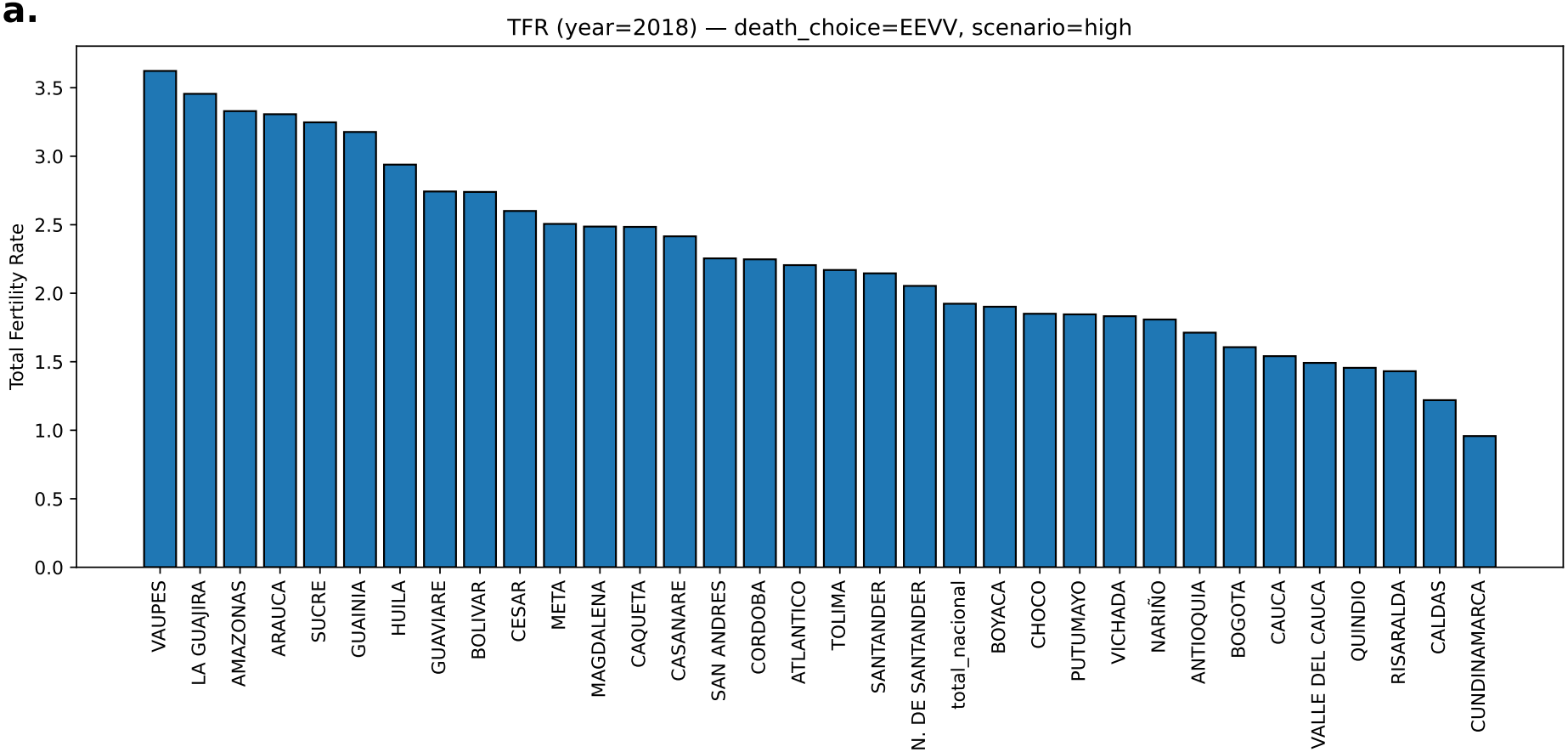
$$\mathrm{TFR}_t(u) \;=\; (1 - S_t)\,\mathrm{TFR}_0(u) \;+\; S_t\,\tau_u.$$

ASFRs annually rebuilt, rescaling fixed $w_a(u)$ to $\mathrm{TFR}_t(u)$:

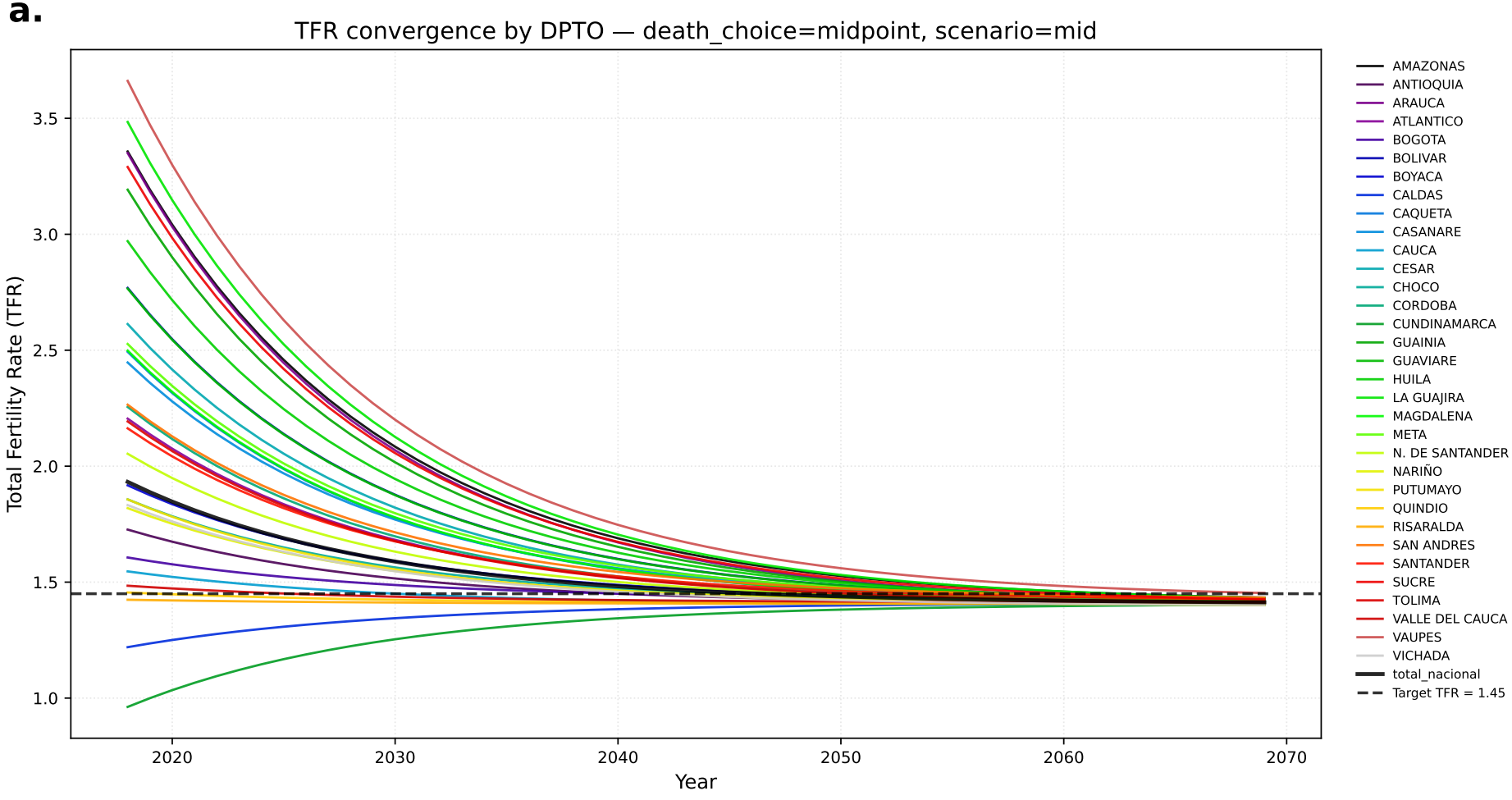
$$\widehat{\mathrm{ASFR}}_a(u,t) \;=\; \widetilde{w}_a(u;\{n_a\})\,\mathrm{TFR}_t(u), \quad \text{with } \sum_a \widetilde{w}_a(u)\,n_a = 1.$$

The implementation normalizes w_a to the age index. It also checks $\sum_a \widehat{\mathrm{ASFR}}_a n_a = \mathrm{TFR}_t$ to 10^{-6} tolerance, clipping non-finite values.

a.


TFR (year=2018) — death_choice=EEVV, scenario=high
Total Fertility Rate
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
VAUPES
LA GUAJIRA
AMAZONAS
ARAUCA
SUCRE
GUAINIA
HUILA
GUAVIARE
BOLIVAR
CESAR
META
MAGDALENA
CAQUETA
CASANARE
SAN ANDRES
CORDOBA
ATLANTICO
TOLIMA
SANTANDER
N. DE SANTANDER
total_nacional
BOYACA
CHOCO
PUTUMAYO
VICHADA
NARIÑO
ANTIOQUIA
BOGOTA
CAUCA
VALLE DEL CAUCA
QUINDIO
RISARALDA
CALDAS
CUNDINAMARCA

a.


TFR convergence by DPTO — death_choice=midpoint, scenario=mid
Total Fertility Rate (TFR)
Year
3.5
3.0
2.5
2.0
1.5
1.0
2020
2030
2040
2050
2060
2070
AMAZONAS
ANTIOQUIA
ARAUCA
ATLANTICO
BOGOTA
BOLIVAR
BOYACA
CALDAS
CAQUETA
CASANARE
CAUCA
CESAR
CHOCO
CORDOBA
CUNDINAMARCA
GUAINIA
GUAVIARE
HUILA
LA GUAJIRA
MAGDALENA
META
N. DE SANTANDER
NARIÑO
PUTUMAYO
QUINDIO
RISARALDA
SAN ANDRES
SANTANDER
SUCRE
TOLIMA
VALLE DEL CAUCA
VAUPES
VICHADA
total_nacional
Target TFR = 1.45

Migration module: goal and inputs

Objective: Construct age–sex migration panels for projection.

Inputs from `conteos`:

- `poblacion_total` $\longrightarrow P_{a,s}(y)$.
- `flujo_inmigracion` (IN), `flujo_emigracion` (OUT).
- Ages in 5-year bins, 0–$4, \ldots, 80+$; sex $\in \{1, 2\}$.

Preprocessing:

- Coerce counts ≥ 0; ensure ordered categories for age, sex, year.

Constructing migration measures (2018 slice)

Population aggregation:

$$P_{a,s}(y) = \sum_{(\text{ANO}=y,\ \text{EDAD}=a,\ \text{SEXO}=s)} \text{VALOR}.$$

Flows:

$$I_{a,s}(y) = \sum \mathbf{1}_{\text{IN}}\text{VALOR}, \qquad E_{a,s}(y) = \sum \mathbf{1}_{\text{OUT}}\text{VALOR}.$$

Derived:

$$\text{Net}_{a,s}(y) = I_{a,s}(y) - E_{a,s}(y), \quad r^{\text{net}}_{a,s}(y) = \frac{\text{Net}_{a,s}(y)}{P_{a,s}(y)}.$$

Return panel for $y = 2018$: $\{a, s, I, E, \text{Net}, P, r^{\text{net}}\}$.

National flows and departmental scaling

Recency rule. For projection year t:

$$y^{\text{flows}} = \min(t,\ \texttt{FLOWS\_LATEST\_YEAR}),$$

so migration profiles are frozen at the most recent observed year.

National totals:

$$I^{\text{nat}}_{a,s} = \sum_d I_{a,s}(d, y^{\text{flows}}), \quad E^{\text{nat}}_{a,s} = \sum_d E_{a,s}(d, y^{\text{flows}}).$$

Department shares:

$$\rho_{a,s}(d,t) = \frac{P_{a,s}(d,t)}{P^{\text{nat}}_{a,s}(t)}.$$

Scaled flows:

$$\Delta_{a,s}(d,t) = \texttt{PERIOD\_YEARS} \cdot \rho_{a,s}(d,t) \left(I^{\text{nat}}_{a,s} - E^{\text{nat}}_{a,s}\right).$$

Half-flow convention and state update

Mid-interval exposure:

$$P^*_{a,s}(d,t) = P_{a,s}(d,t) + \tfrac{1}{2}\Delta_{a,s}(d,t),$$

used only as denominator for ASFRs and mortality rates.

End-of-period stock:

$$P_{a,s}(d,t+\Delta) = \mathcal{L}_{a,s}\big(P^*(d,t)\big) + \Delta_{a,s}(d,t).$$

Thus: half of Δ enters via exposures (rates), the full Δ updates the stock. No double-counting.

Leslie construction for 1-year ages (female block)

Setup. Ages $a = 0, 1, \ldots, A$ (e.g. up to $90+$).
Let $\mathcal{F} = \{a_{\min}, \ldots, a_{\max}\}$ be fertile ages (e.g. $12:49$). With 1-year steps, widths $n_a \equiv 1$.

Ingredients at year t:

- ASFR shape $\widehat{f}_a(t)$ for $a \in \mathcal{F}$ (else 0), normalized to TFR_t.
- Survival $p_a^F(t) = 1 - q_a^F(t)$ from life table (female).
- Sex-at-birth fraction κ_F (female share of births).

Fertile ages $\mathcal{F} = \{a_{\min}, \ldots, a_{\max}\}$. Define

$$b_a(t) = \begin{cases} \kappa_F \widehat{f}_a(t)\, p_0^F(t), & a \in \mathcal{F}, \\ 0, & \text{otherwise.} \end{cases}$$

$$L^{(F)}(t) = \begin{pmatrix} b_0 & b_1 & b_2 & \cdots & b_{A-1} & b_A \\ p_0^F & 0 & 0 & \cdots & 0 & 0 \\ 0 & p_1^F & 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & p_{A-2}^F & 0 & 0 \\ 0 & \cdots & 0 & 0 & p_{A-1}^F & r_A^F \end{pmatrix}$$

Notes: (i) $b_0 = 0$, so newborns are not fertile; (ii) sub-diagonal entries are one-year survivals $p_a^F(t)$; (iii) the open age uses a diagonal retention $r_A^F(t) \in [0, 1]$.

Blocks. With male survival $p_a^M(t)$ and male share $\kappa_M = 1 - \kappa_F$,

$$L_{MF}(t) = \Big[\text{first row } b_a^{(M)}(t) = \kappa_M \widehat{f}_a(t)\, p_0^M(t);\ \text{subdiag } 0\Big],$$

$L_{MM}(t) =$ male survival sub-diagonal like L_{FF} but no fertility row.

Projection with additive migration. For departmental vectors $N^F(t), N^M(t)$ (by single year),

$$\begin{bmatrix} N^F(t{+}1) \\ N^M(t{+}1) \end{bmatrix} = \begin{bmatrix} L_{FF}(t) & 0 \\ L_{MF}(t) & L_{MM}(t) \end{bmatrix} \begin{bmatrix} N^F(t) \\ N^M(t) \end{bmatrix} + \begin{bmatrix} \Delta^F(t) \\ \Delta^M(t) \end{bmatrix}.$$

Here $\Delta^{\cdot}(t)$ are the *full-period* net migration vectors (age–sex specific). The "half-flow" $\frac{1}{2}\Delta$ is used only to adjust exposures for rate computation; it is *not* inserted into L.

Running the Code:

- Running the code is as simple as:

Running the Code:

- Running the code is as simple as:

  1. Clone the GitHub repository:

     ```
     git clone https://github.com/crahal/pyccm.git
     ```

Running the Code:

- Running the code is as simple as:

  1. Clone the GitHub repository:

     ```
     git clone https://github.com/crahal/pyccm.git
     ```

  2. Install the package and dependencies:

     ```
     pip install -e .[all]
     ```

Running the Code:

- Running the code is as simple as:

  1. Clone the GitHub repository:

     ```
     git clone https://github.com/crahal/pyccm.git
     ```

  2. Install the package and dependencies:

     ```
     pip install -e .[all]
     ```

  3. Put the 'conteos.rds' file in `./data`.

Running the Code:

- Running the code is as simple as:

1. Clone the GitHub repository:

```
git clone https://github.com/crahal/pyccm.git
```

2. Install the package and dependencies:

```
pip install -e .[all]
```

3. Put the 'conteos.rds' file in `./data`.

4. Edit the `config.yaml` file to specify desired settings.

Running the Code:

- Running the code is as simple as:

1. Clone the GitHub repository:

```
git clone https://github.com/crahal/pyccm.git
```

2. Install the package and dependencies:

```
pip install -e .[all]
```

3. Put the 'conteos.rds' file in `./data`.

4. Edit the `config.yaml` file to specify desired settings.

5. Run the projection code:

```
cd src; python main_compute.py
```

Running the Code:

- Running the code is as simple as:
  1. Clone the GitHub repository:
     ```
     git clone https://github.com/crahal/pyccm.git
     ```
  2. Install the package and dependencies:
     ```
     pip install -e .[all]
     ```
  3. Put the 'conteos.rds' file in `./data`.
  4. Edit the `config.yaml` file to specify desired settings.
  5. Run the projection code:
     ```
     cd src; python main_compute.py
     ```
- Concatenated outputs: Produces `all_lifetables`, `all_asfr`, `all_leslie_matrices`, and `all_projections` in CSV/Parquet.

Unit Testing in PyCCM:

- PyCCM currently has **211 automated tests** across the core modules.
- Coverage spans mortality, fertility, migration, abridging, helpers, and pipeline orchestration.
- All tests are run with `pytest` and include unit, integration, and edge-case checks.
- Current module-level test counts are:
  1. `test_mortality.py`: 37 tests
  2. `test_fertility.py`: 40 tests
  3. `test_migration.py`: 24 tests
  4. `test_abridger.py`: 46 tests
  5. `test_main_compute.py`: 24 tests
  6. `test_helpers.py`: 40 tests
- Latest documented run: **100% passing** (211/211).

Documentation in PyCCM:

- PyCCM is supported by **extensive technical documentation** across code, tests, and methods.
- Current docs include roughly **110,000 words** spanning module explainers, assessments, and reports.
- Each core module has dedicated material covering:
  1. algorithmic logic and implementation details,
  2. demographic assumptions and methodological choices,
  3. test summaries, known limitations, and recommended improvements.
- Consolidated reports provide both executive summaries and deep-dive validation notes.
- This documentation supports reproducibility, onboarding, and transparent review of the full pipeline.

Root `config.yaml` Defaults (1/7):

- Exact defaults currently in the root YAML:
  1. **Theme: Paths**
     - 1.1 `paths.data_dir = "./data"`
     - 1.2 `paths.results_dir = "./results"`
     - 1.3 `paths.target_tfr_csv = "./data/target_tfrs.csv"`
     - 1.4 `paths.midpoints_csv = "./data/midpoints.csv"`
     - 1.5 `paths.mortality_improvements_csv = "./data/mortality_improvements_in_class.csv"`

Root `config.yaml` Defaults (2/7):

- Exact defaults currently in the root YAML:
  2. **Theme: Diagnostics and Maintenance**
     - 2.1 `diagnostics.print_target_csv = true`
     - 2.2 `diagnostics.mortality_improvements_debug = true`
     - 2.3 `maintenance.clean_run = true`
     - 2.4 `unabridging.enabled = true`

Root `config.yaml` Defaults (3/7):

- Exact defaults currently in the root YAML:

3. **Theme: Projection Horizon and Inputs**

3.1 `projections.start_year = 2018`
3.2 `projections.end_year = 2069`
3.3 `projections.step_years = 1`
3.4 `projections.death_choices = ["EEVV"]`
3.5 `projections.period_years = 1`
3.6 `projections.flows_latest_year = 2024`
3.7 `last_observed_year_by_death.EEVV = 2023`
3.8 `last_observed_year_by_death.censo_2018 = 2018`
3.9 `last_observed_year_by_death.midpoint = 2018`

Root `config.yaml` Defaults (4/7):

- Exact defaults currently in the root YAML:

4. **Theme: Fertility Settings**

4.1 `fertility.default_tfr_target = 1.45`
4.2 `fertility.convergence_years = 52`
4.3 `fertility.smoother.kind = "exp"`
4.4 `fertility.smoother.converge_frac = 0.99`
4.5 `fertility.smoother.logistic.mid_frac = 0.5`
4.6 `fertility.smoother.logistic.steepness = null`
4.7 `fertility.default_tfr_target_range.start = 1.45`
4.8 `fertility.default_tfr_target_range.stop = 1.45`
4.9 `fertility.default_tfr_target_range.step = 0.05`

Root `config.yaml` Defaults (5/7):

- Exact defaults currently in the root YAML:
  5. **Theme: Midpoint Settings**
     5.1 `midpoints.default_eevv_weight = 0.5`

Root `config.yaml` Defaults (6/7):

- Exact defaults currently in the root YAML:

6. **Theme: Mortality Settings**

6.1 `mortality.use_ma = true`
6.2 `mortality.ma_window = 5`
6.3 `mortality.improvement_total = 0.2`
6.4 `mortality.convergence_years = 52`
6.5 `mortality.smoother.kind = "exp"`
6.6 `mortality.smoother.converge_frac = 0.99`
6.7 `mortality.smoother.logistic.mid_frac = 0.5`
6.8 `mortality.smoother.logistic.steepness = null`
6.9 `mortality.improvement_total_range.start = 0.25`
6.10 `mortality.improvement_total_range.stop = 0.25`
6.11 `mortality.improvement_total_range.step = 0.05`
6.12 `mortality.ma_window_range.start = 5`
6.13 `mortality.ma_window_range.stop = 5`
6.14 `mortality.ma_window_range.step = 1`

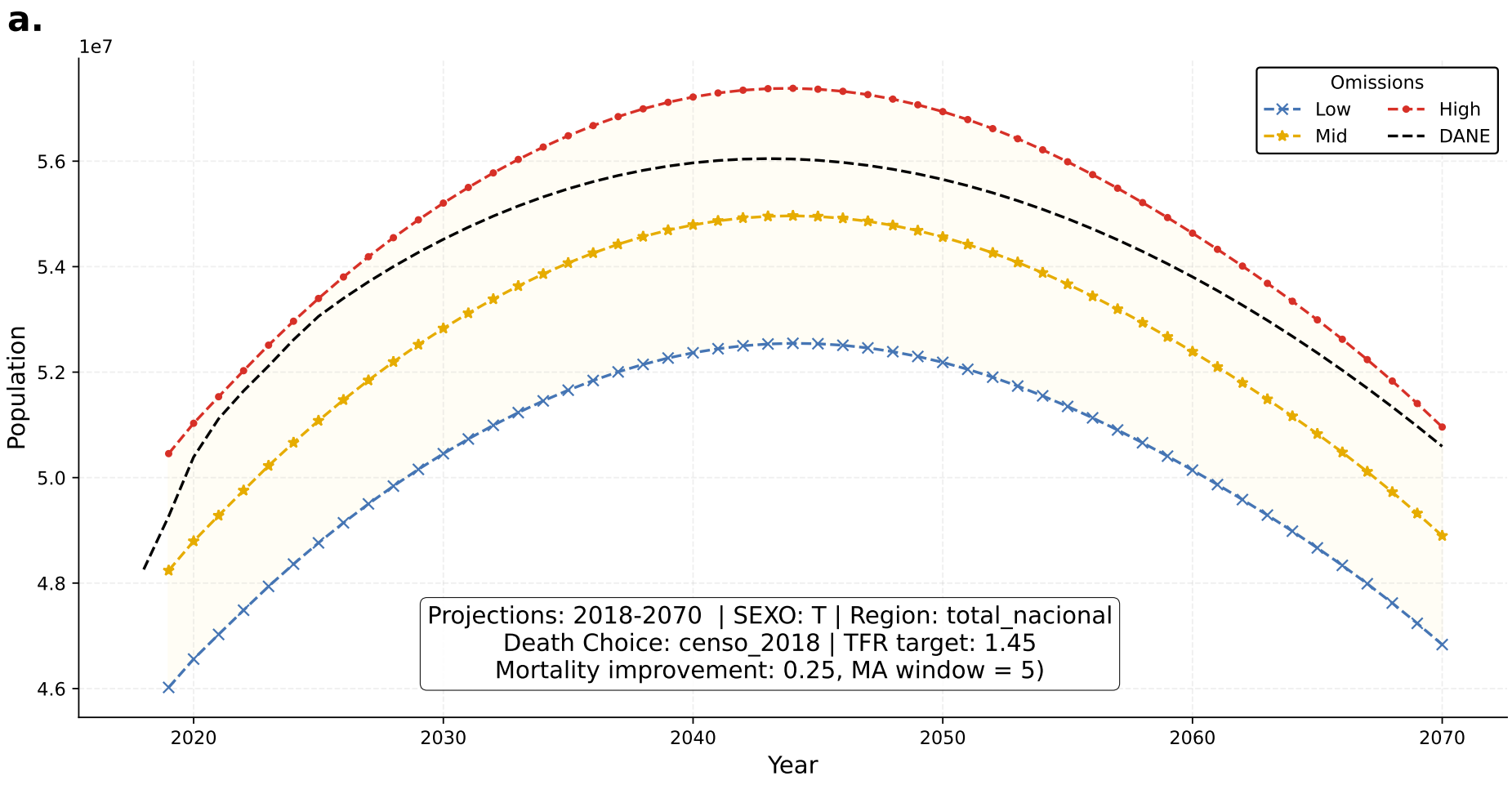
Root `config.yaml` Defaults (7/7):

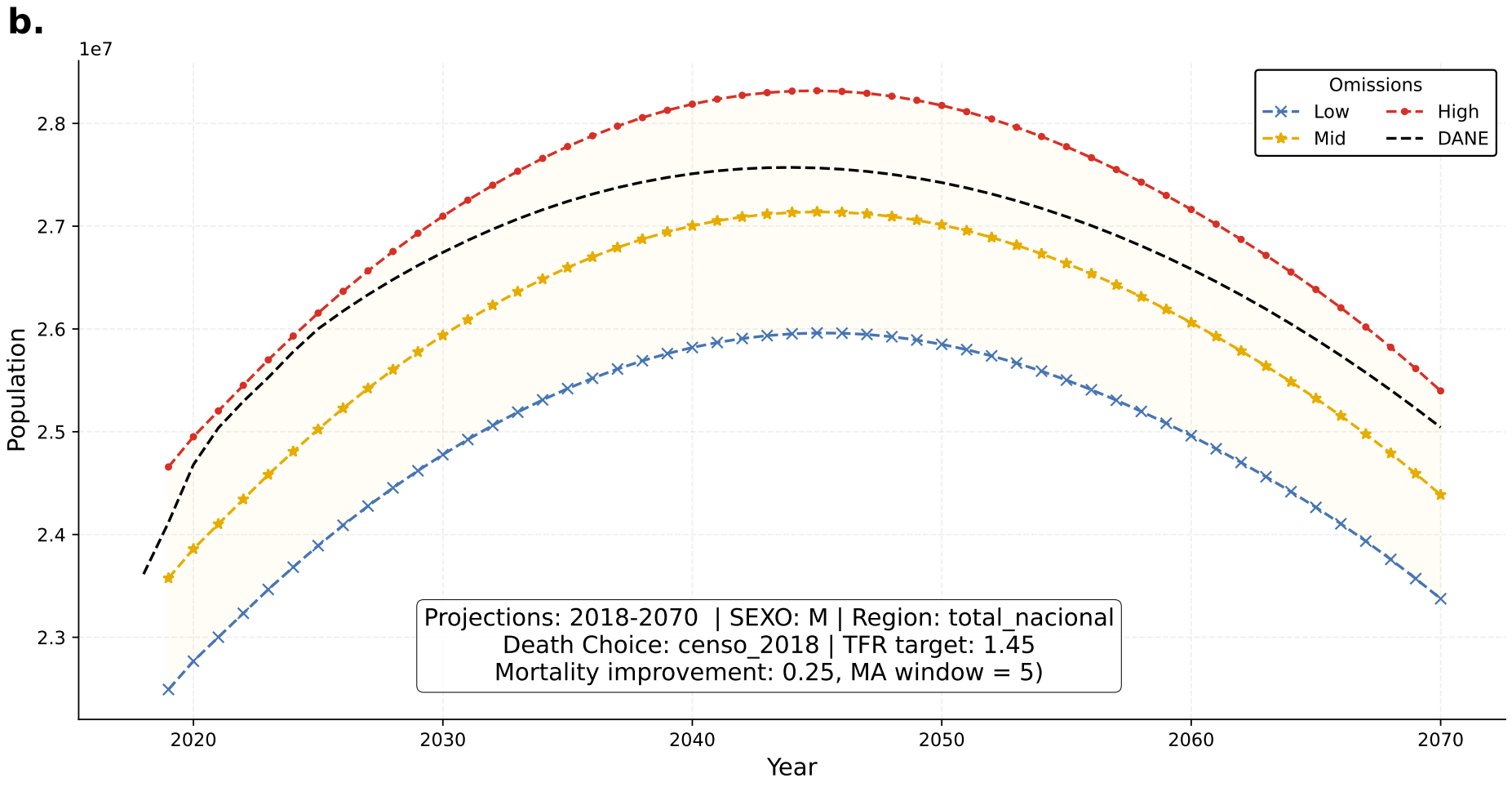
- Exact defaults currently in the root YAML:

7. **Theme: Runs, Parallel, and Filenames**

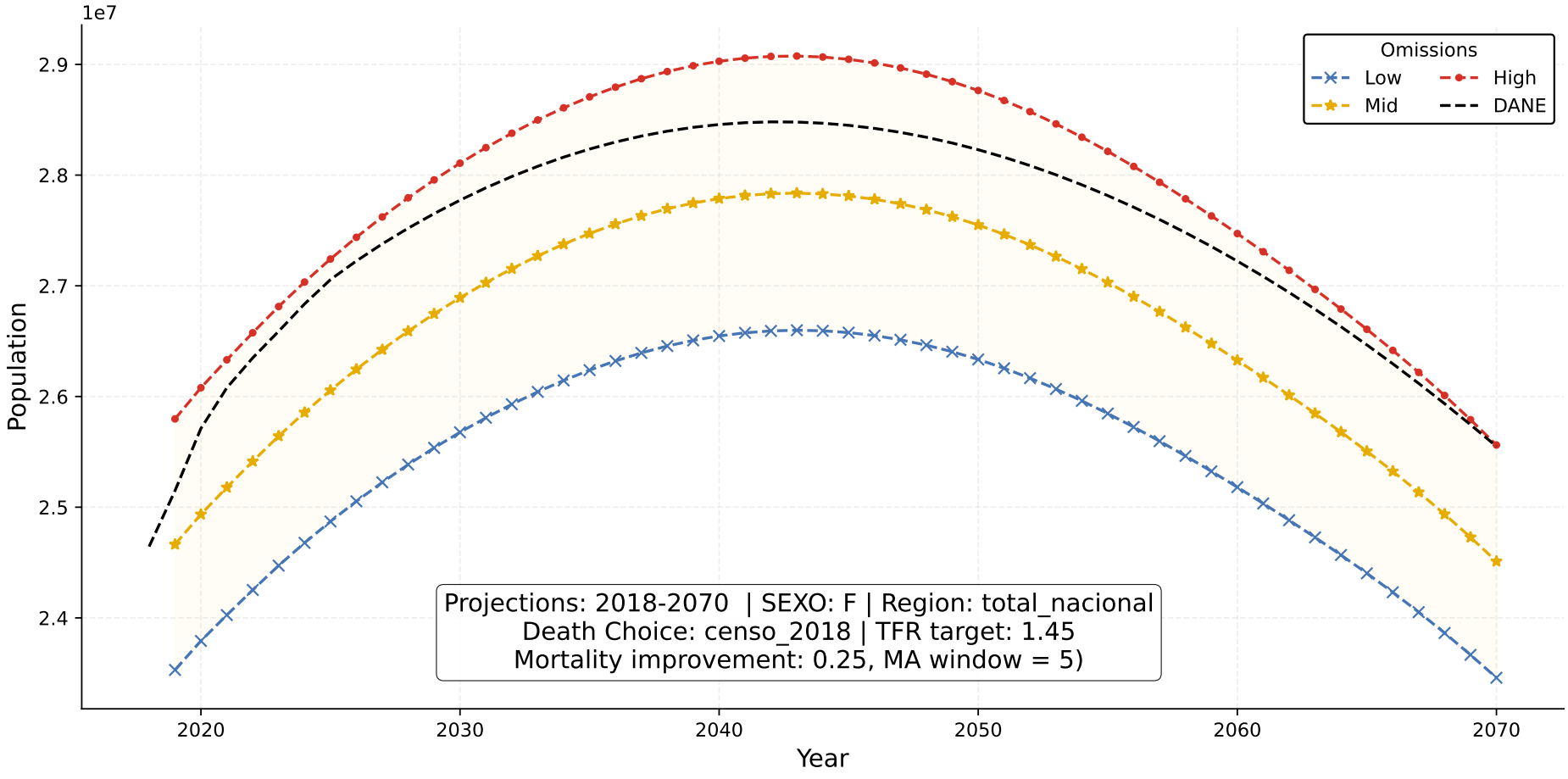
7.1 `runs.mode = "no_draws"`
7.2 `runs.no_draws_tasks = [mid_omissions, low_omissions, high_omissions]`
7.3 `runs.draws.num_draws = 1000`
7.4 `runs.draws.dist_types = ["uniform"]`
7.5 `runs.draws.label_pattern = "{dist}_draw_{i}"`
7.6 `parallel.processes = 20`
7.7 `filenames.asfr = "asfr.csv"`
7.8 `filenames.lt_M = "lt_M_t.csv"`
7.9 `filenames.lt_F = "lt_F_t.csv"`
7.10 `filenames.lt_T = "lt_T_t.csv"`

Lets finally see what these (very conservative) defaults look like...!

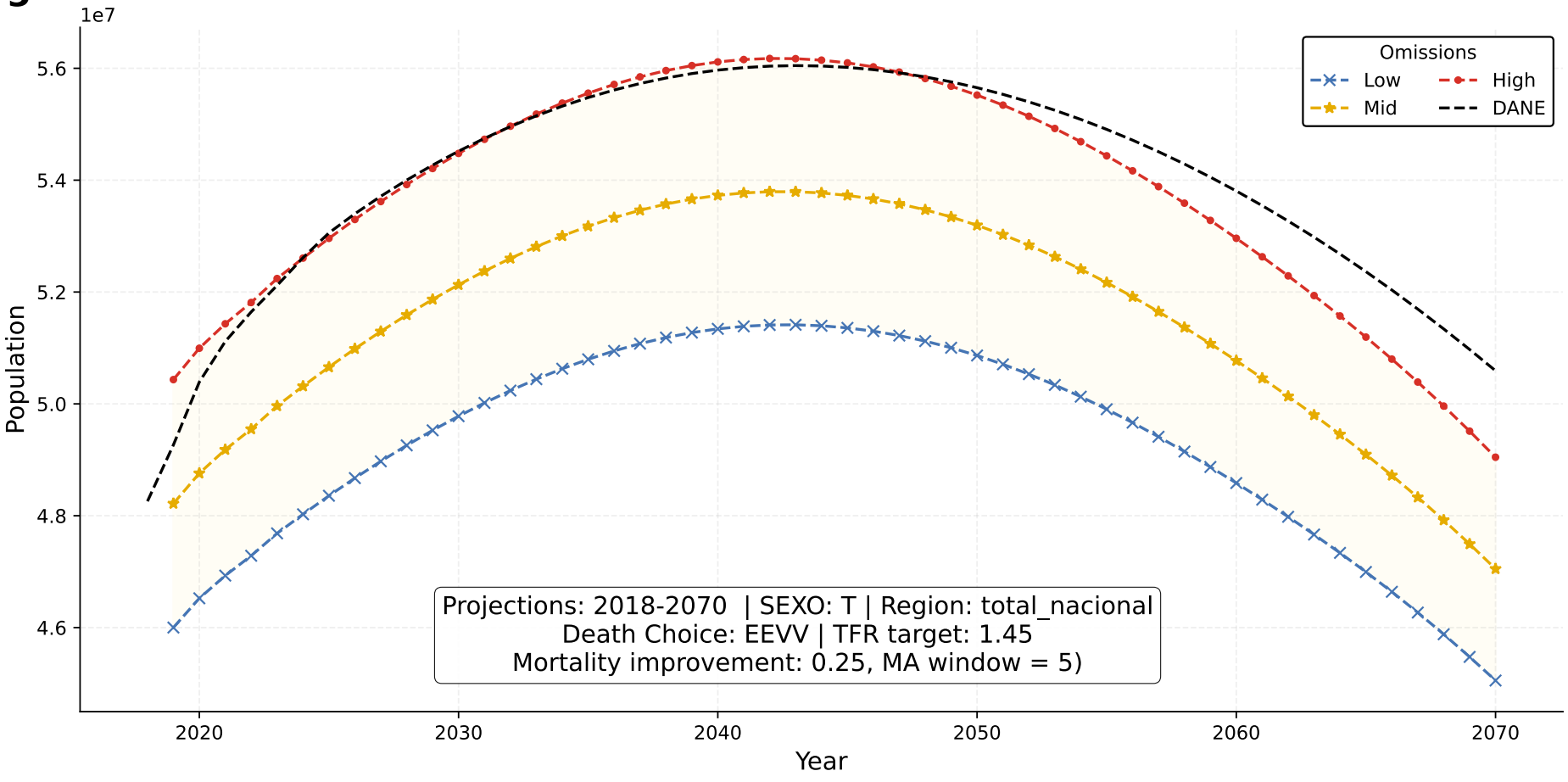

a.
1e7
5.6
5.4
5.2
5.0
4.8
4.6
Population
2020
2030
2040
2050
2060
2070
Year
Omissions
Low
High
Mid
DANE
Projections: 2018-2070 | SEXO: T | Region: total_nacional
Death Choice: censo_2018 | TFR target: 1.45
Mortality improvement: 0.25, MA window = 5)

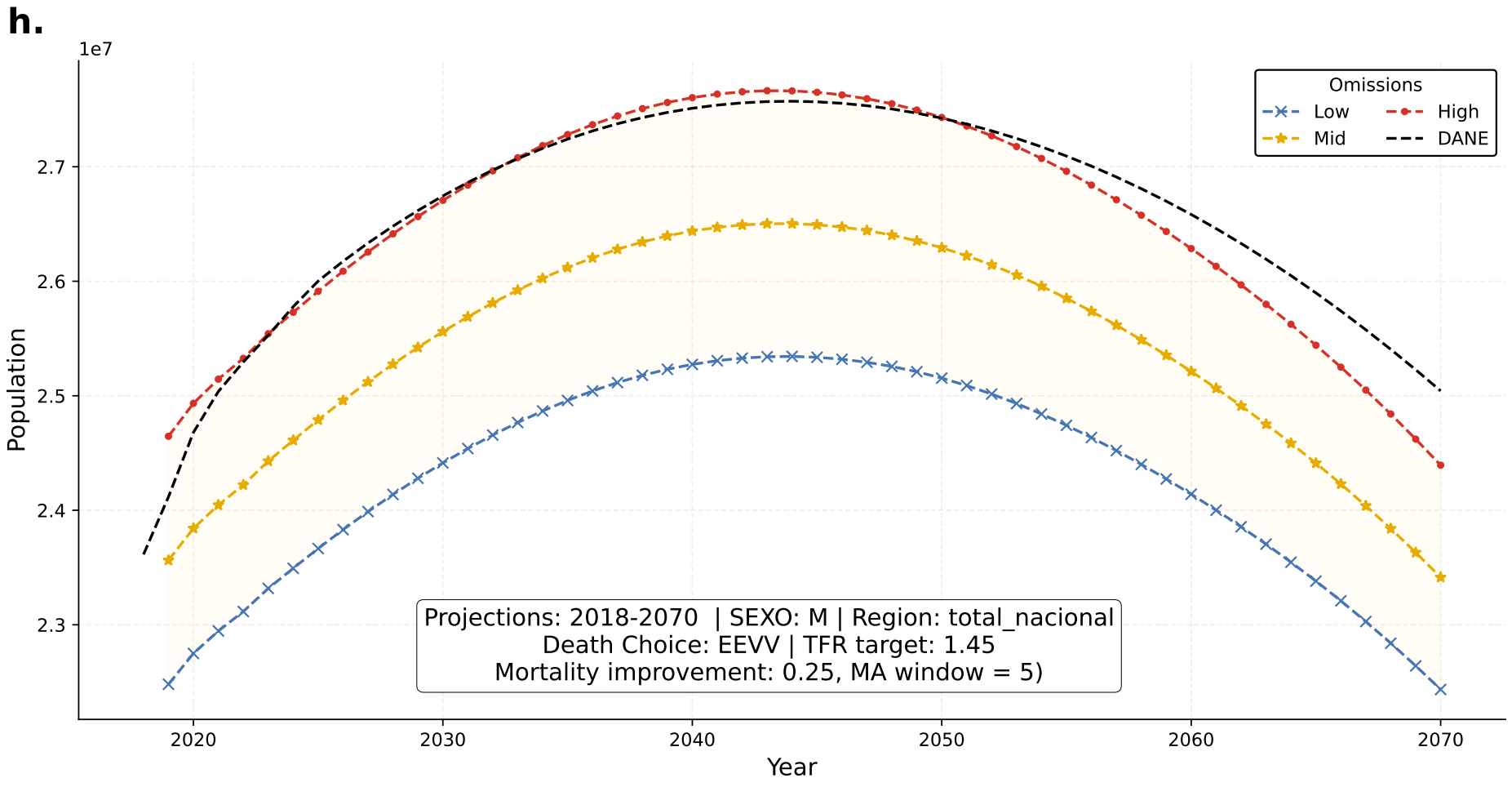

b.
1e7
2.8
2.7
2.6
2.5
2.4
2.3
Population
2020
2030
2040
2050
2060
2070
Year
Omissions
Low
High
Mid
DANE
Projections: 2018-2070 | SEXO: M | Region: total_nacional
Death Choice: censo_2018 | TFR target: 1.45
Mortality improvement: 0.25, MA window = 5)

c.

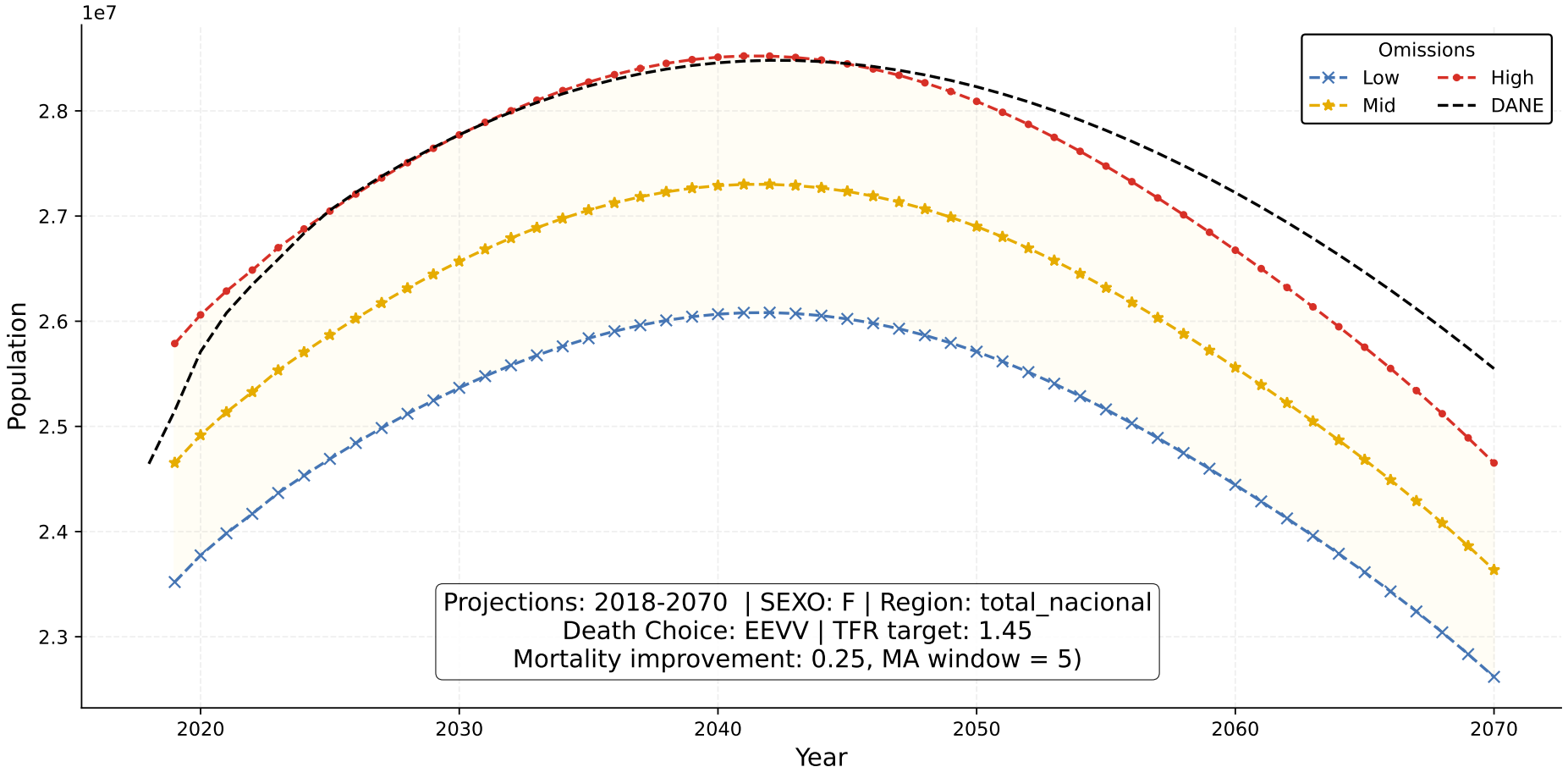

1e7
2.9
2.8
2.7
2.6
2.5
2.4
Population
2020
2030
2040
2050
2060
2070
Year
Omissions
Low
High
Mid
DANE
Projections: 2018-2070 | SEXO: F | Region: total_nacional
Death Choice: censo_2018 | TFR target: 1.45
Mortality improvement: 0.25, MA window = 5)

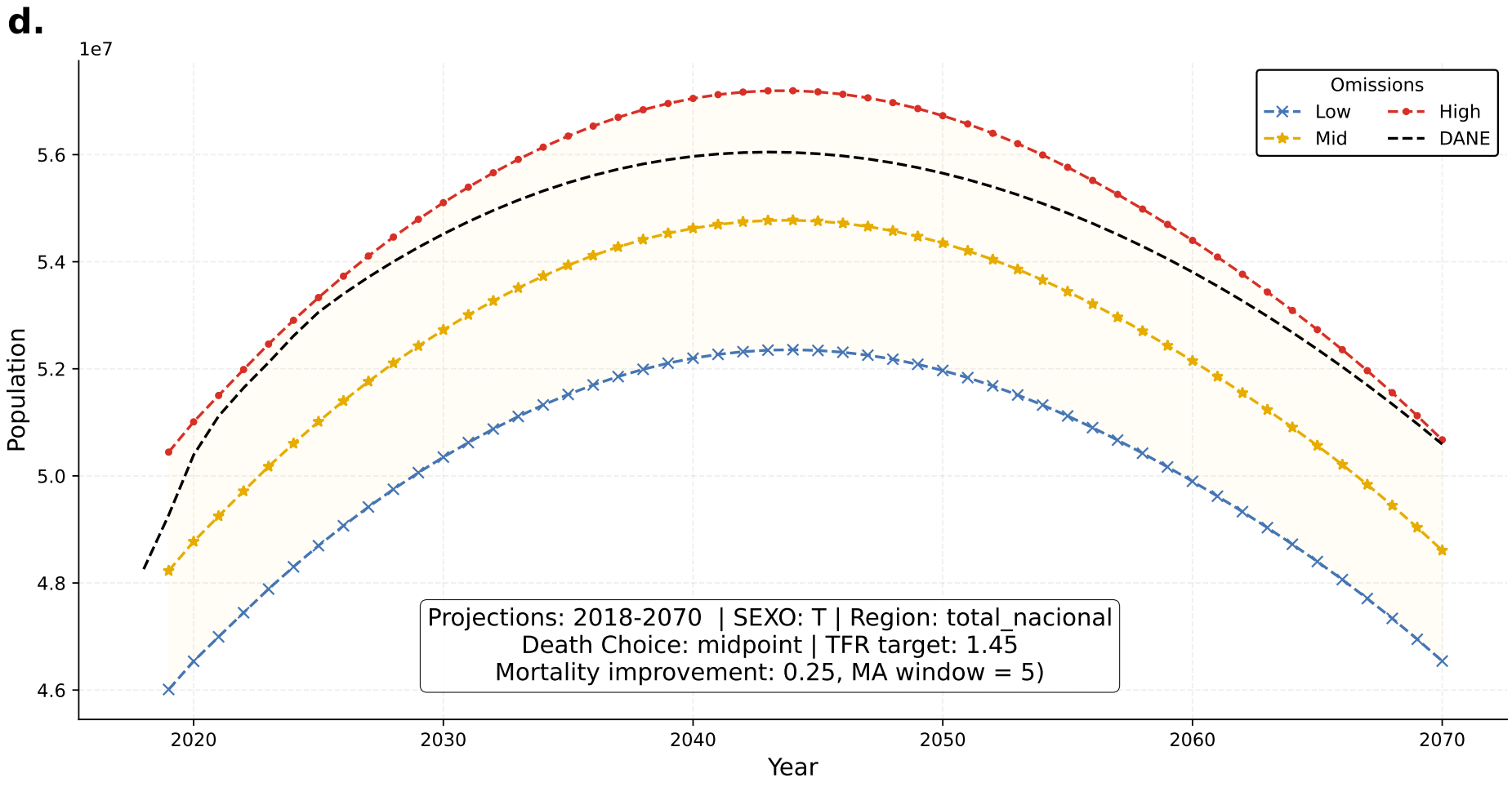
g.


1e7
5.6
5.4
5.2
5.0
4.8
4.6
Population
2020
2030
2040
2050
2060
2070
Year
Omissions
Low
High
Mid
DANE
Projections: 2018-2070 | SEXO: T | Region: total_nacional
Death Choice: EEVV | TFR target: 1.45
Mortality improvement: 0.25, MA window = 5)

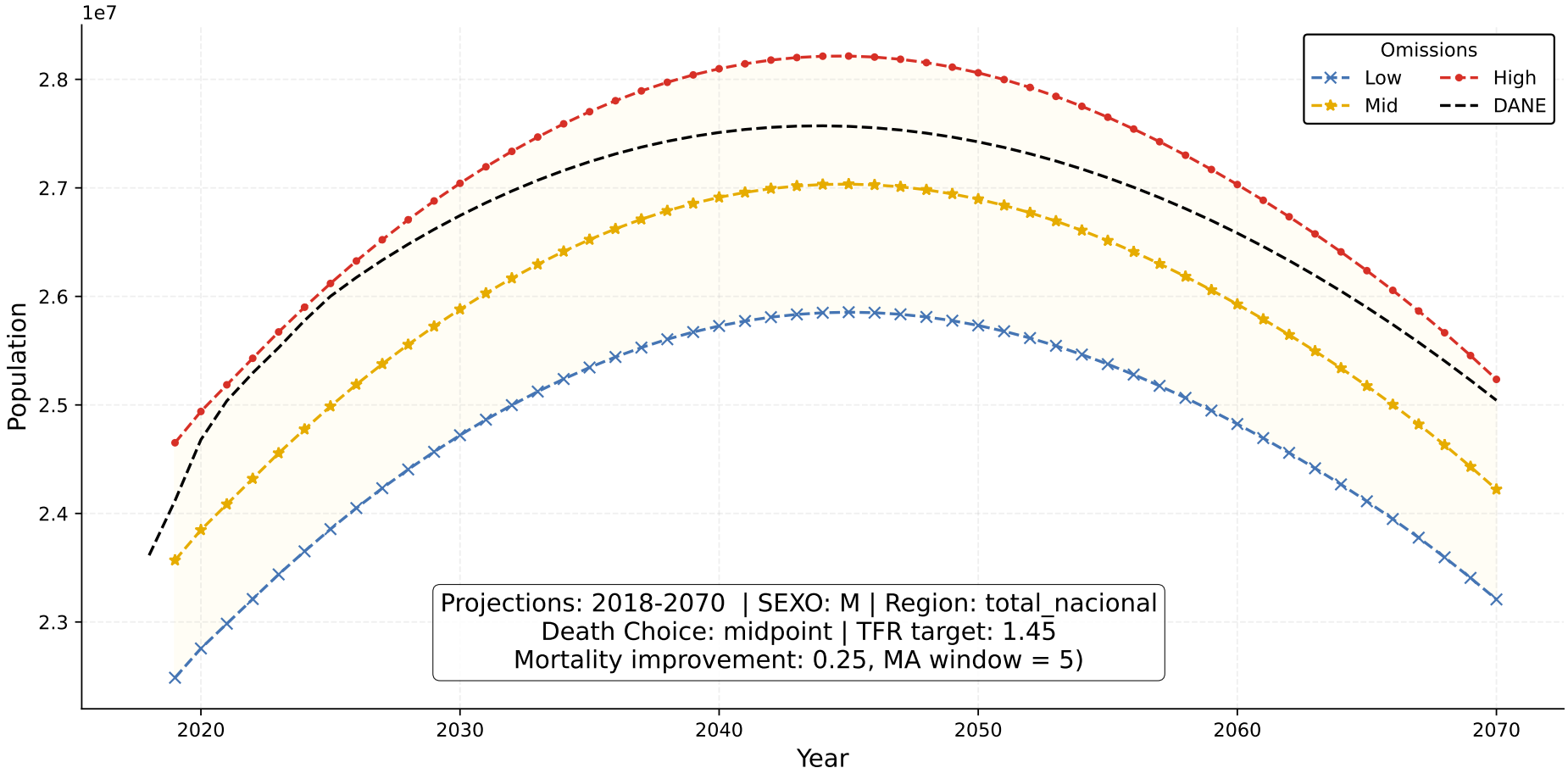

h.
1e7
Population
2.7
2.6
2.5
2.4
2.3
2020
2030
2040
2050
2060
2070
Year
Omissions
Low
High
Mid
DANE
Projections: 2018-2070 | SEXO: M | Region: total_nacional
Death Choice: EEVV | TFR target: 1.45
Mortality improvement: 0.25, MA window = 5)

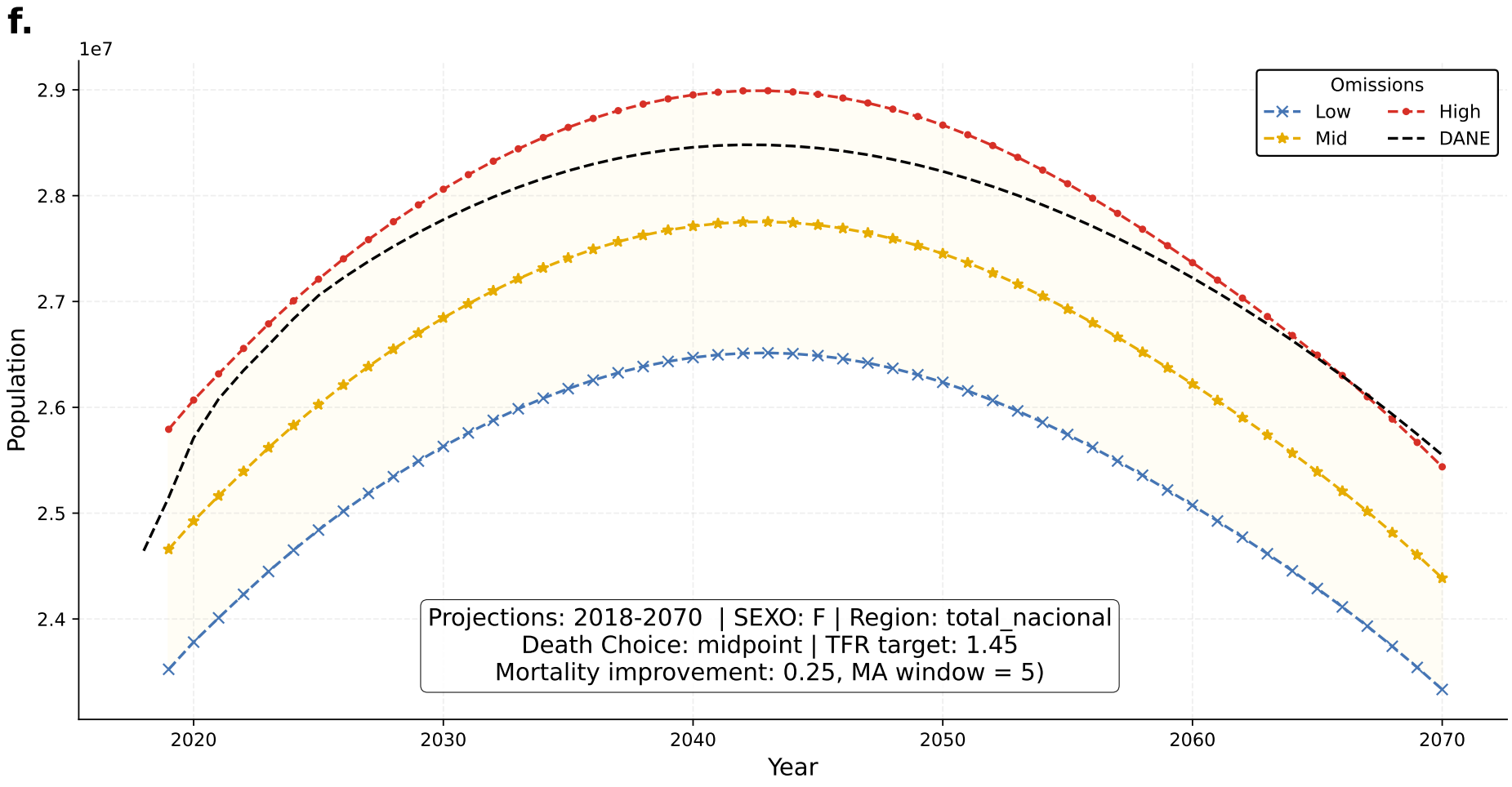
i.

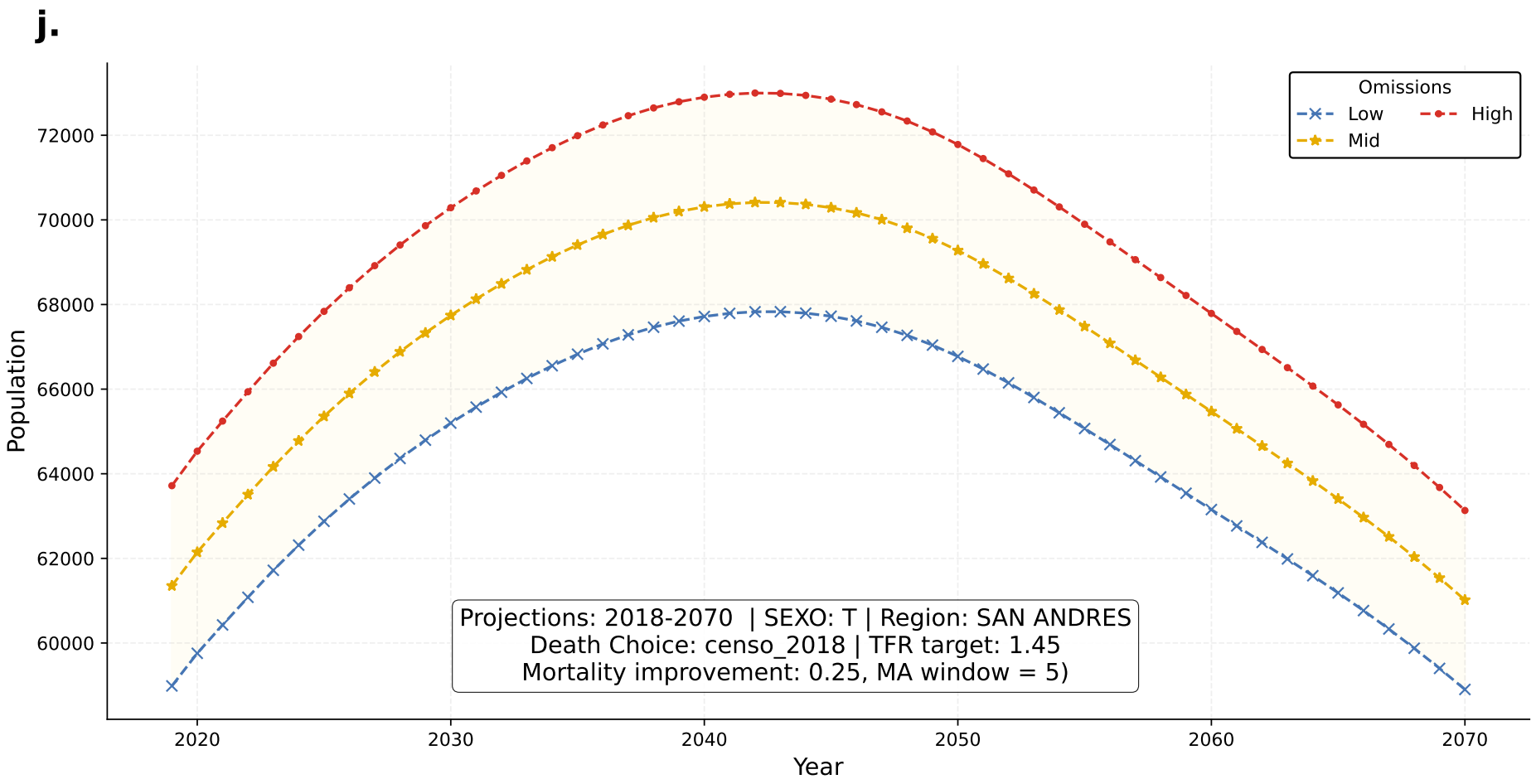

1e7
2.8
2.7
2.6
2.5
2.4
2.3
Population
2020
2030
2040
2050
2060
2070
Year
Omissions
Low
High
Mid
DANE
Projections: 2018-2070 | SEXO: F | Region: total_nacional
Death Choice: EEVV | TFR target: 1.45
Mortality improvement: 0.25, MA window = 5)


d.
1e7
5.6
5.4
5.2
5.0
4.8
4.6
Population
2020
2030
2040
2050
2060
2070
Year
Omissions
Low
High
Mid
DANE
Projections: 2018-2070 | SEXO: T | Region: total_nacional
Death Choice: midpoint | TFR target: 1.45
Mortality improvement: 0.25, MA window = 5)

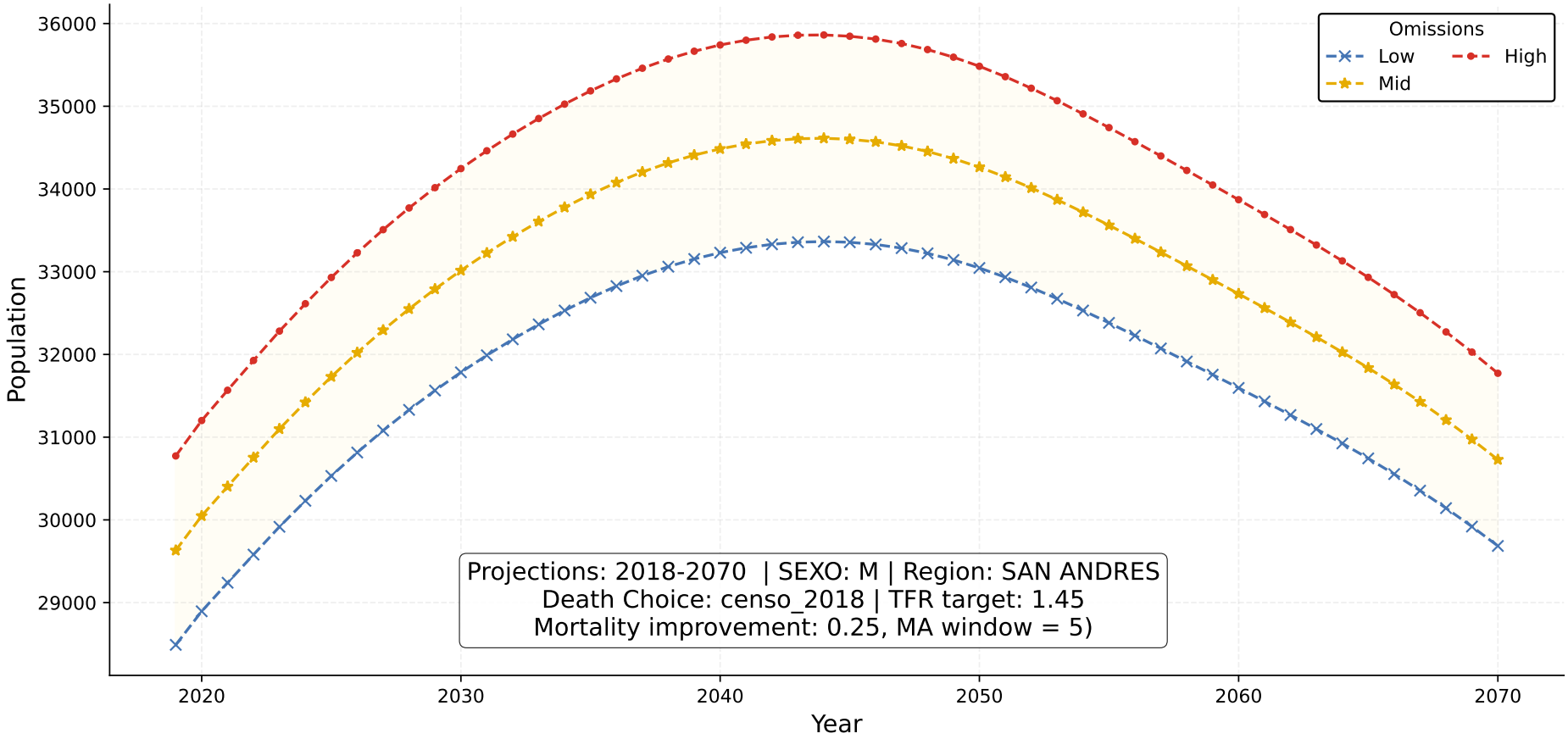
e.


1e7
2.8
2.7
2.6
2.5
2.4
2.3
Population
2020
2030
2040
2050
2060
2070
Year
Omissions
Low
High
Mid
DANE
Projections: 2018-2070 | SEXO: M | Region: total_nacional
Death Choice: midpoint | TFR target: 1.45
Mortality improvement: 0.25, MA window = 5)

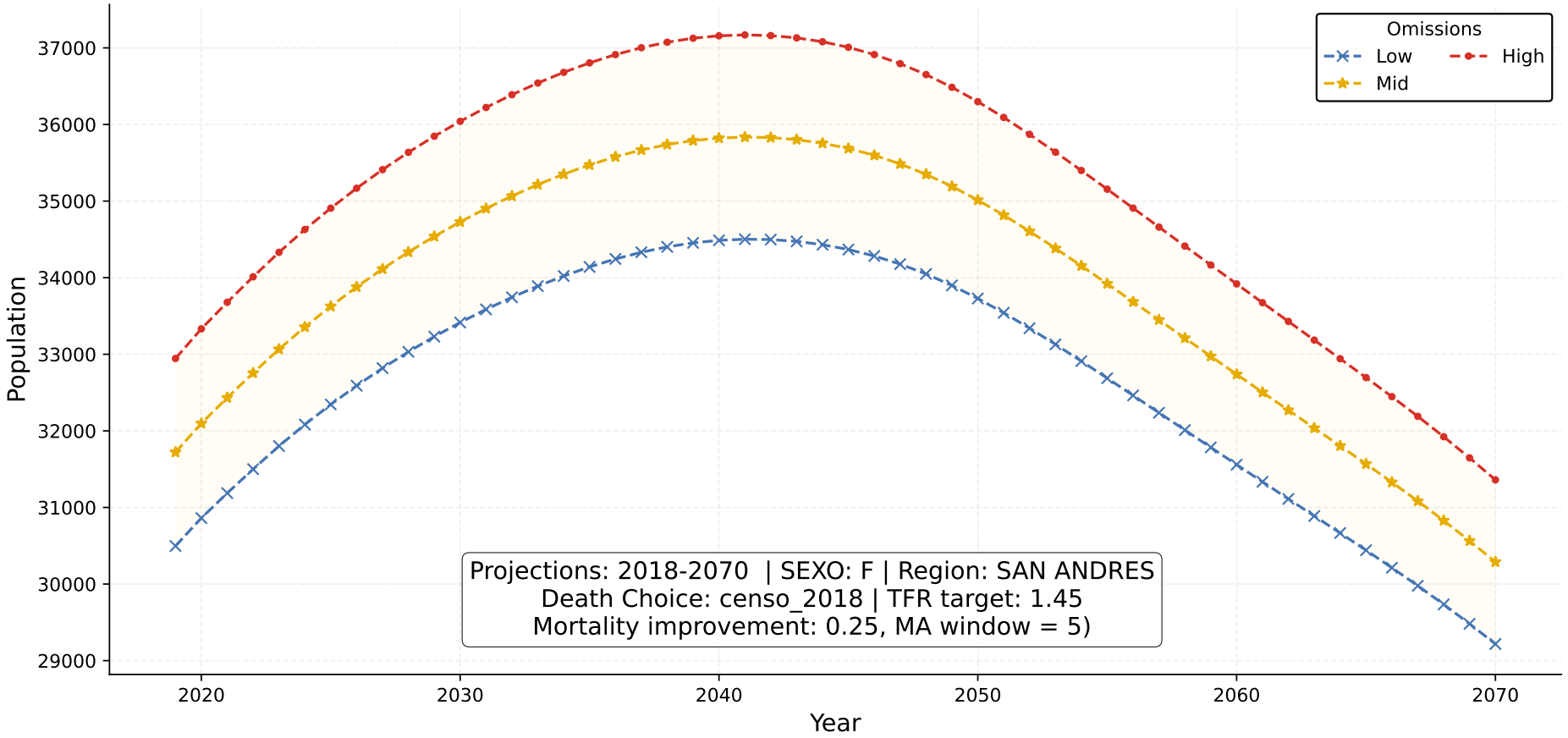

f.
1e7
2.9
2.8
2.7
2.6
2.5
2.4
Population
2020
2030
2040
2050
2060
2070
Year
Omissions
Low
High
Mid
DANE
Projections: 2018-2070 | SEXO: F | Region: total_nacional
Death Choice: midpoint | TFR target: 1.45
Mortality improvement: 0.25, MA window = 5)

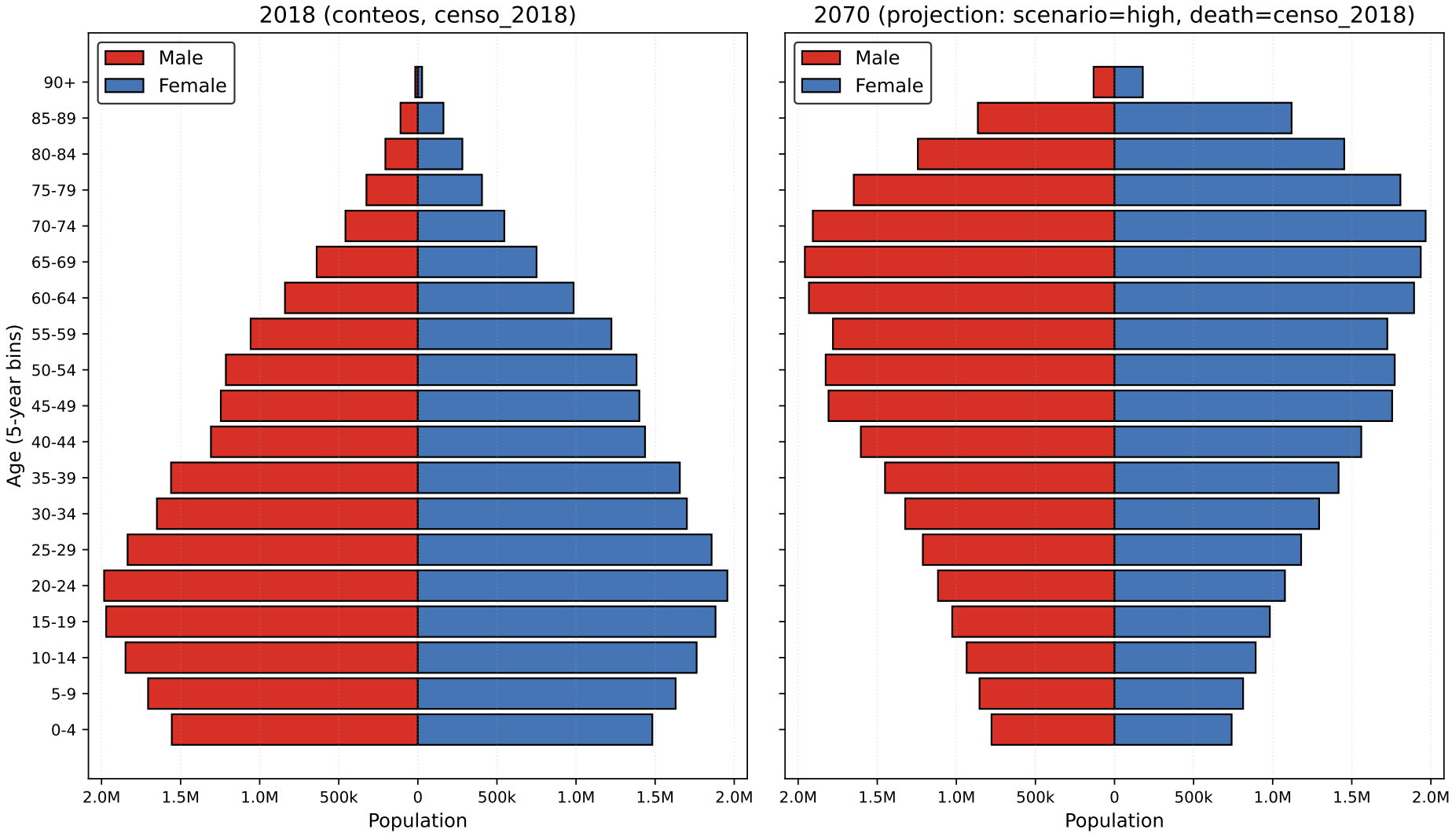

j.
Omissions
Low
Mid
High
Population
72000
70000
68000
66000
64000
62000
60000
2020
2030
2040
2050
2060
2070
Year
Projections: 2018-2070 | SEXO: T | Region: SAN ANDRES
Death Choice: censo_2018 | TFR target: 1.45
Mortality improvement: 0.25, MA window = 5)

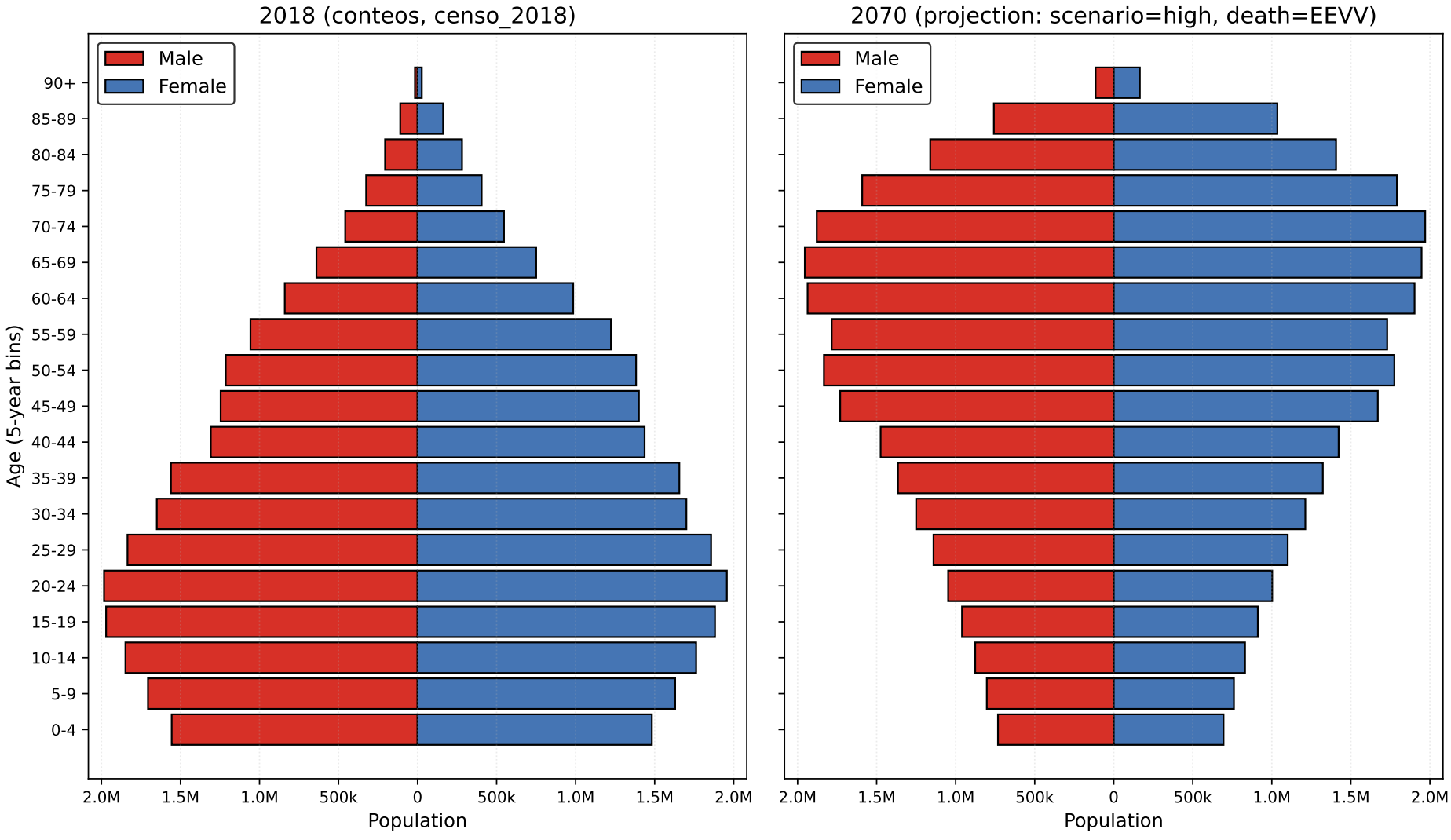
k.

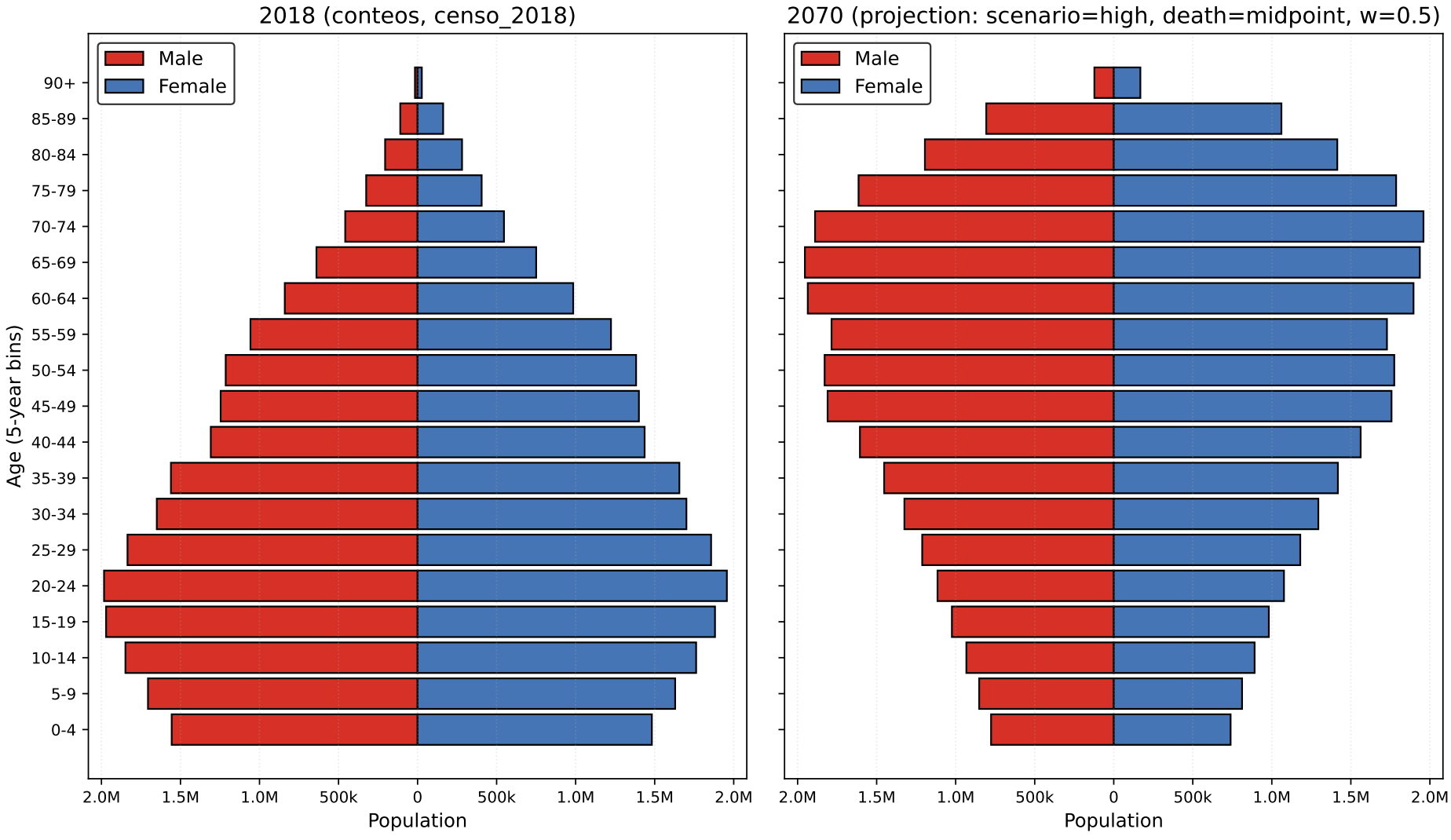

Omissions
Low
Mid
High
Population
Year
36000
35000
34000
33000
32000
31000
30000
29000
2020
2030
2040
2050
2060
2070
Projections: 2018-2070 | SEXO: M | Region: SAN ANDRES
Death Choice: censo_2018 | TFR target: 1.45
Mortality improvement: 0.25, MA window = 5)

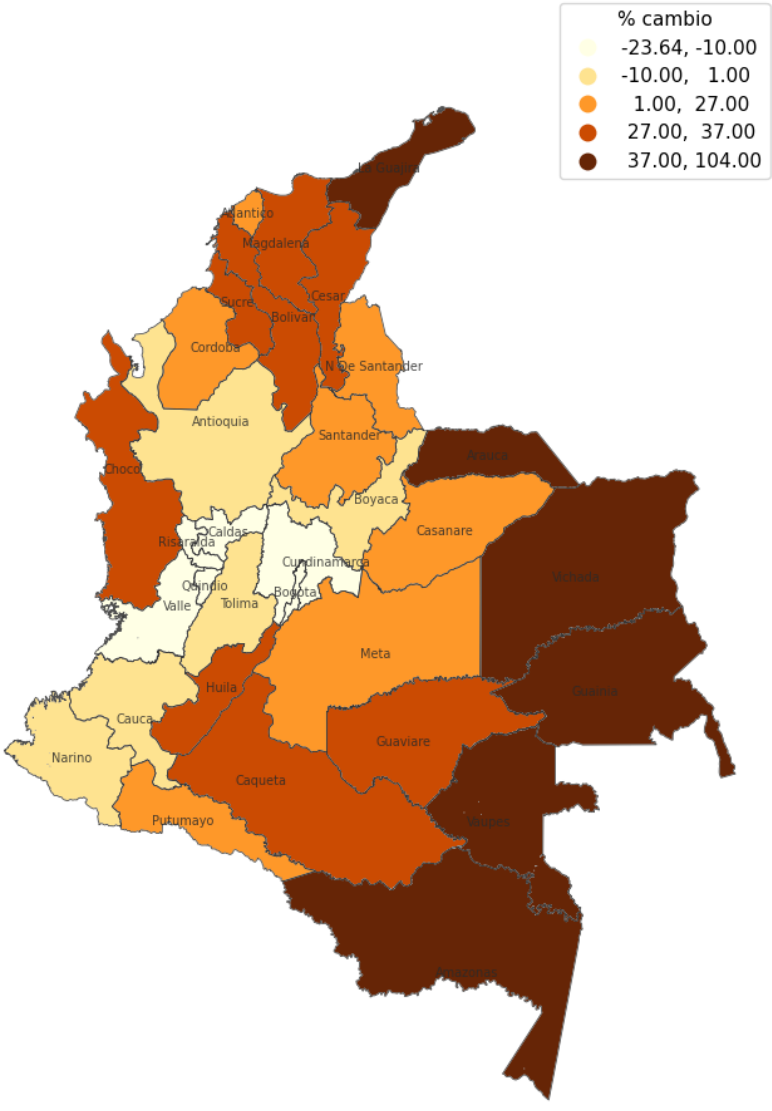
I.

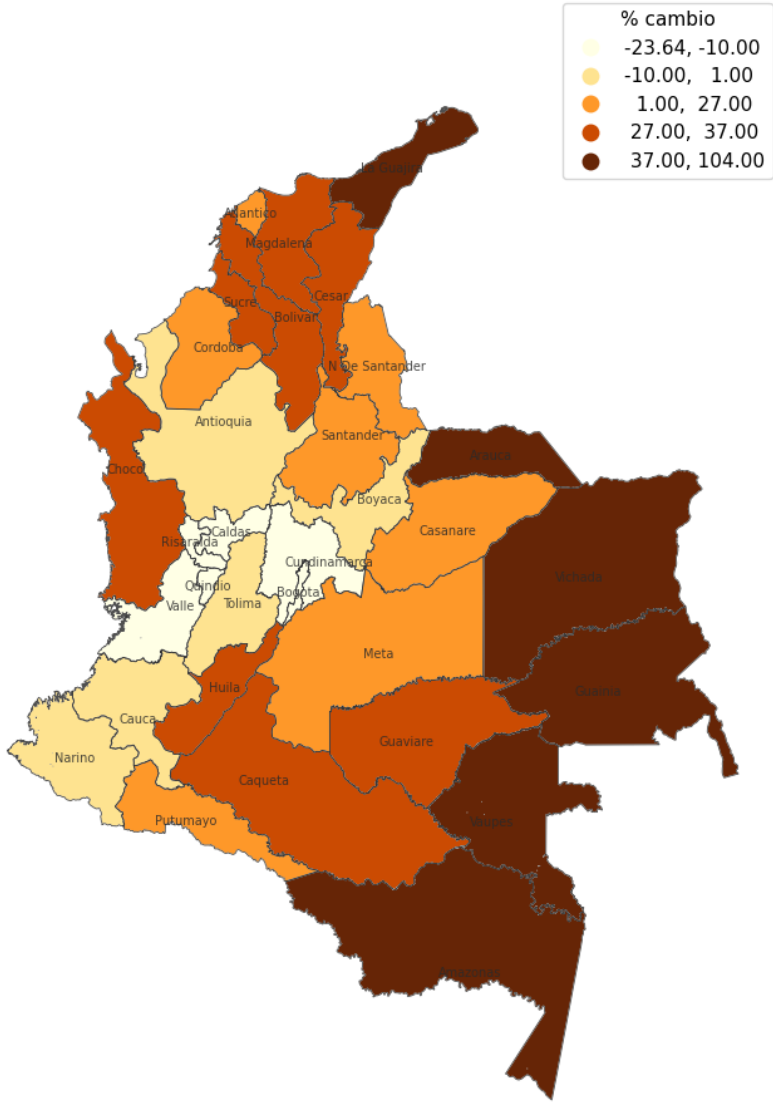

Omissions
Low
Mid
High
Population
Year
37000
36000
35000
34000
33000
32000
31000
30000
29000
2020
2030
2040
2050
2060
2070
Projections: 2018-2070 | SEXO: F | Region: SAN ANDRES
Death Choice: censo_2018 | TFR target: 1.45
Mortality improvement: 0.25, MA window = 5)


2018 (conteos, censo_2018)
2070 (projection: scenario=high, death=censo_2018)
Male
Female
Age (5-year bins)
90+
85-89
80-84
75-79
70-74
65-69
60-64
55-59
50-54
45-49
40-44
35-39
30-34
25-29
20-24
15-19
10-14
5-9
0-4
2.0M
1.5M
1.0M
500k
0
500k
1.0M
1.5M
2.0M
Population


2018 (conteos, censo_2018)
2070 (projection: scenario=high, death=EEVV)
Male
Female
Age (5-year bins)
90+
85-89
80-84
75-79
70-74
65-69
60-64
55-59
50-54
45-49
40-44
35-39
30-34
25-29
20-24
15-19
10-14
5-9
0-4
2.0M
1.5M
1.0M
500k
0
Population


2018 (conteos, censo_2018)
2070 (projection: scenario=high, death=midpoint, w=0.5)
Male
Female
Age (5-year bins)
90+
85-89
80-84
75-79
70-74
65-69
60-64
55-59
50-54
45-49
40-44
35-39
30-34
25-29
20-24
15-19
10-14
5-9
0-4
2.0M
1.5M
1.0M
500k
0
Population


% cambio
-23.64, -10.00
-10.00, 1.00
1.00, 27.00
27.00, 37.00
37.00, 104.00
Atlantico
Magdalena
Cesar
Sucre
Bolivar
Cordoba
N de Santander
Antioquia
Santander
Choco
Boyaca
Caldas
Risaralda
Cundinamarca
Casanare
Arauca
Vichada
Quindio
Valle
Tolima
Bogota
Meta
Huila
Guainia
Cauca
Guaviare
Narino
Caqueta
Putumayo
Vaupes


% cambio
-23.64, -10.00
-10.00, 1.00
1.00, 27.00
27.00, 37.00
37.00, 104.00
Atlantico
Magdalena
Cesar
Sucre
Bolivar
Cordoba
N de Santander
Antioquia
Santander
Choco
Boyaca
Caldas
Risaralda
Cundinamarca
Casanare
Arauca
Vichada
Quindio
Valle
Tolima
Bogota
Meta
Huila
Guainia
Cauca
Guaviare
Narino
Caqueta
Putumayo
Vaupes

Lets now see a way by which PyCCM can be used to do scenario analysis

Lets now see a way by which PyCCM can be used to do scenario analysis

(Thanks again to Juliana... incredible work as always!)

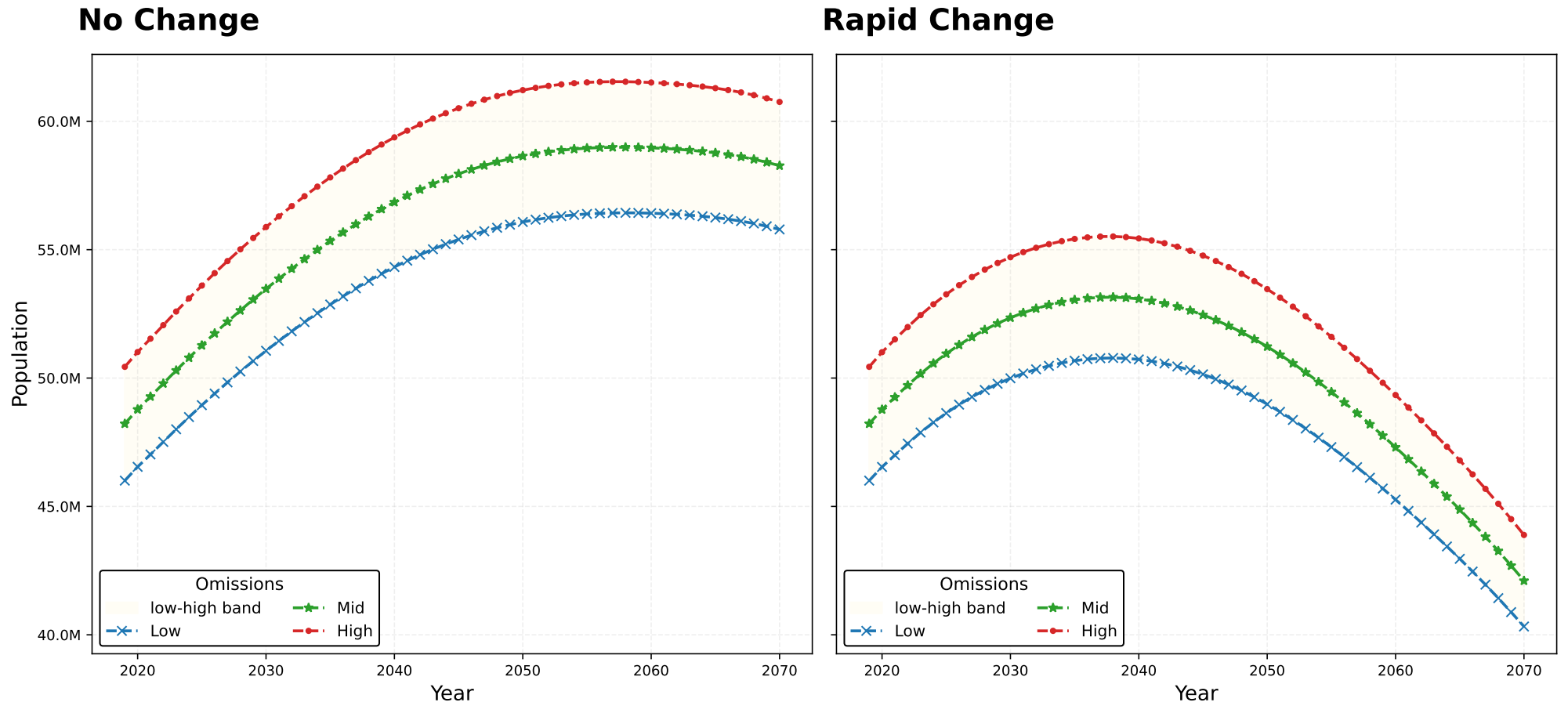
Scenario Design Summary:

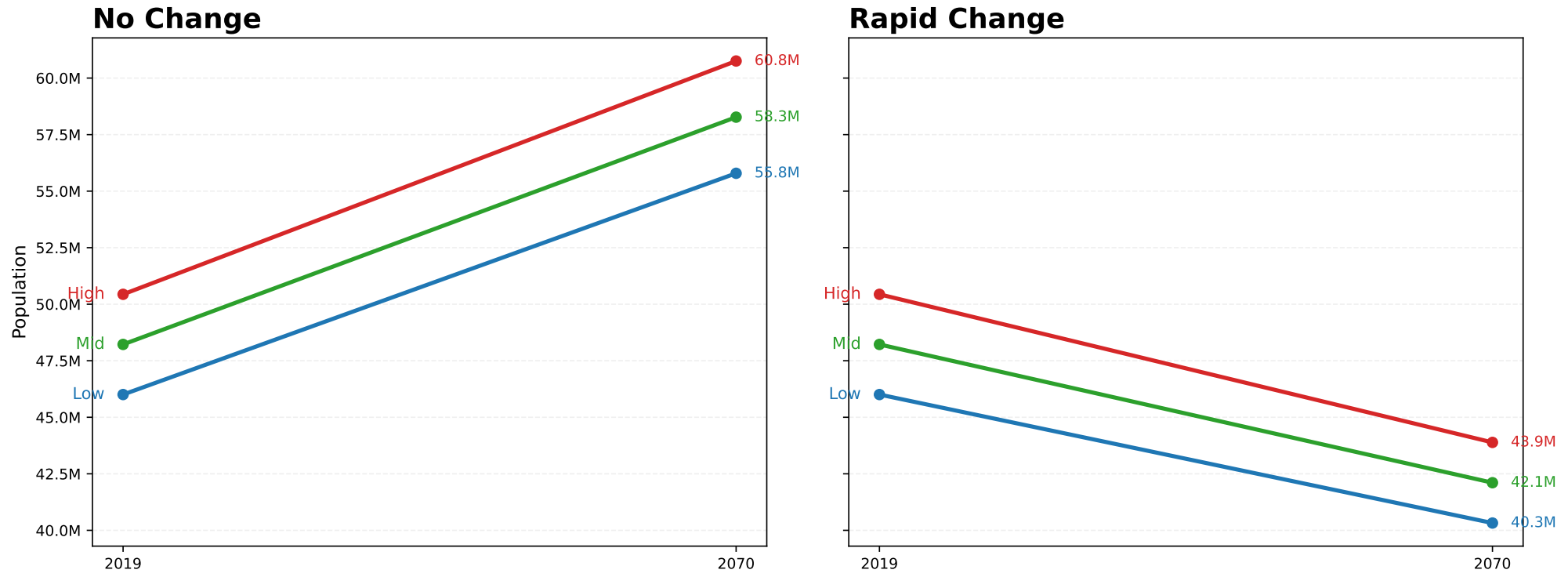
1. No Demographic Change Scenario

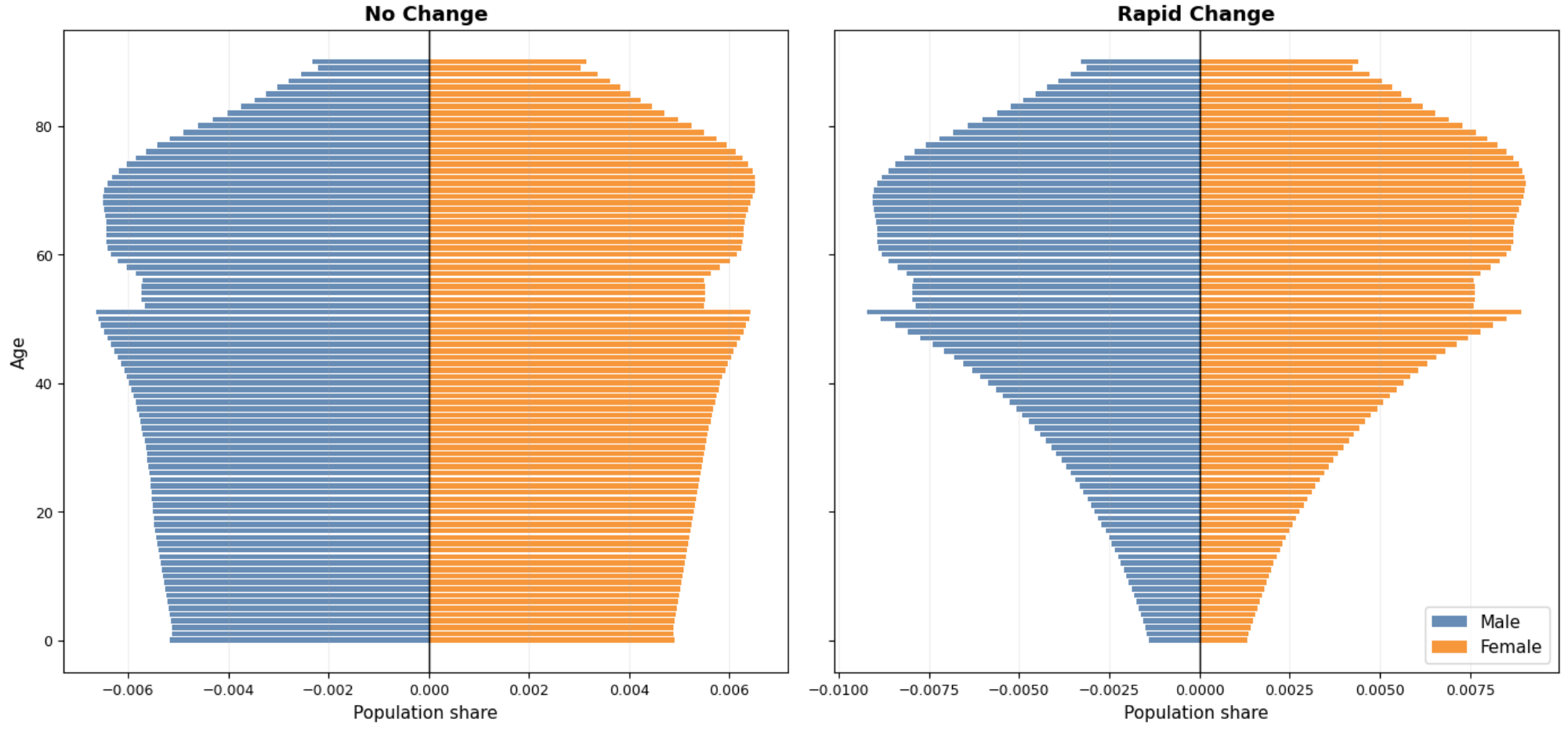
- **Mortality:** No long-run mortality improvement.
- `improvement_total = 0` for all departments in `mortality_improvements_no_change.csv`.
- **Fertility:** Department TFR fixed at observed 2018 DANE level.
- In `target_tfrs_no_change.csv`, `Target_TFR` is the department-specific TFR from the DANE 2005–2018 report.

2. Accelerated Demographic Change Scenario

- **Mortality:** Heterogeneous improvements by department.
- `improvement_total` ranges from 0.3 to 0.8; 10-year convergence.
- **Fertility:** Department TFR converges over 5 years to projected 2024 TFR minus 0.2 children/woman.
- `Target_TFR` is set from department-specific DANE 2024 projected TFR minus 0.2.


No Change
Rapid Change
Population
60.0M
55.0M
50.0M
45.0M
40.0M
2020
2030
2040
2050
2060
2070
Year
Omissions
low-high band
Low
Mid
High


Start vs End Population by Omission Scenario
No Change
Rapid Change
Population
60.0M
57.5M
55.0M
52.5M
50.0M
47.5M
45.0M
42.5M
40.0M
High
Mid
Low
60.8M
58.3M
55.8M
43.9M
42.1M
40.3M
2019
2070


Population Pyramid, total_nacional, Year 2070 (mid scenario)
No Change
Rapid Change
Age
0
20
40
60
80
−0.006
−0.004
−0.002
0.000
0.002
0.004
0.006
−0.0100
−0.0075
−0.0050
−0.0025
0.0000
0.0025
0.0050
0.0075
Population share
Population share
Male
Female

Still TODO....:

A few things we are still working on:

Still TODO....:

A few things we are still working on:

1. Fixing a small issue with graduating 0-1 infants who've survived

Still TODO....:

A few things we are still working on:

1. Fixing a small issue with graduating 0-1 infants who've survived
2. Precomputing all feasible grids of parameterisations

Still TODO....:

A few things we are still working on:

1. Fixing a small issue with graduating 0-1 infants who've survived
2. Precomputing all feasible grids of parameterisations
3. Turning this grid into a live, interactive dashboard and accompanying academic paper

ICSC
INTERNATIONAL CONFERENCE ON SOCIAL COMPUTING

THE 2026 EDITION OF THE

INTERNATIONAL CONFERENCE ON SOCIAL COMPUTING

SCAN TO SUBMIT BEFORE MAY 25TH

ICSC is a long-running research conference that connects computational methods with social science insights to better understand human behavior, social networks, and societal change. Held in the United Kingdom for the first time, this conference aims to bring together leading scholars and experts from around the world, thereby fostering a unique environment that promotes cross-cultural exchange and interdisciplinary collaboration. **This year's conference features single- and multi-track sessions dedicated to Digital and Computational Demography.**

KEY DATES ICSC 2026

- **Submission Deadline:** 23:59 GMT May 25th, 2026
- **Notification Date:** June 15th, 2026
- **Camera Ready:** August 17th, 2026
- **Taught workshop:** September 2nd, 2026
- **Conference Dates:** September 3rd-4th, 2026

CONFERENCE HIGHLIGHTS

Proceedings: Papers will be considered for optional publication in our Conference Proceedings. A selection of outstanding papers will be fast-tracked to ACM Transactions on Social Computing or the Journal of Social Computing. If publication in the proceedings is not sought, papers - or extended abstracts - can be submitted in any format.

We invite original research on all aspects of social computing. Example topics include:

- Digital and Computational Demography
- Social applications of Large Language Models
- Large-scale social media analytics and intelligence
- Digital inclusion in the Global South
- The Science of (Open) Science
- Applied social computing applications in diverse areas such as health and finance

清華大學出版社
Tsinghua University Press

UNIVERSITY OF OXFORD

Nuffield College
UNIVERSITY OF OXFORD

NUFFIELD COLLEGE, UNIVERSITY OF OXFORD
Oxford OX1 1NF, United Kingdom

More Info: https://icsc-conf.github.io/

Points for discussion:

1. Are our defaults reasonable?
2. How can we include ‘tempo’ in our analysis, given data constraints?
3. Are our analytical choices – in terms of how to build the CCM – reasonable?
4. What other functionality should PyCCM include?