MONASH University

Epidemiology by Design: A Reproducible SIRS Modeling Workflow

Varun Sridhar
Master of Business Analytics

Report for
Kids Research Institute Australia

3 November 2025

MONASH
BUSINESS
SCHOOL

Department of Econometrics & Business Statistics

(03) 9905 2478
BusEco-Econometrics@monash.edu

ABN: 12 377 614 012

Table of contents

1 Abstract **3**

2 Introduction **3**
2.1 Objectives . . . 4
2.2 Significance . . . 4

3 Methods **4**
3.1 Model structure . . . 4
3.2 Deterministic simulations . . . 5
3.3 Stochastic simulations . . . 5
3.4 Multi-population setup . . . 5
3.5 Parameterization and default settings . . . 5
3.6 Mathematical model and discrete-time update . . . 6
3.7 Seasonal transmission and intervention windows . . . 7
3.8 Numerical choices and stability . . . 7
3.9 Observation model and thinning to cases . . . 7
3.10 Multi-population structure and contact matrix . . . 7
3.11 Algorithm sketch . . . 8
3.12 Validation and verification . . . 8
Conservation checks . . . 8
Limiting cases . . . 8
Time step sensitivity . . . 8
Cross-check with analytic R0 . . . 9
3.13 Figure generation workflow . . . 9
3.14 Reproducibility . . . 9

4 Results **10**
4.1 Deterministic (constant beta) . . . 10
4.2 Deterministic (seasonal Beta) → SIR . . . 11
4.3 Stochastic (single-pop) → I(t) ribbon + cases ribbon . . . 11
4.4 Deterministic vs stochastic — incidence overlay . . . 12
4.5 Multi-population (deterministic) → S combined, I faceted, incidence per million . . . 13
4.6 Summarized tidy results . . . 14
Figure guide and cross-references (what to look for) . . . 14
4.7 Key observations (peaks, timing, variability) . . . 15
4.8 Quick sensitivity study . . . 16
4.9 Robustness checks and stress tests . . . 17

5 Stakeholder implications (operations, comms, policy) **17**

6 Discussion — interpretation for epidemiology and planning **17**

7 Limitations and Future Work **19**
7.1 Limitations . . . 19
7.2 Future Work . . . 20

8 Implementation notes Package/file layout, function list, and how the plotting code reads sim outputs. **21**

9 Ethical and responsible use **22**

10 Conclusion **22**
10.1 Appendix 0 — Code rendering note 23
10.2 Appendix A — Figure pipeline (what runs) 24
10.3 Appendix B — What each PNG shows 27
10.4 Appendix C - Glossary 27
10.5 Appendix D — Reproducibility details (software & seeds) 28
10.6 Appendix E - Software 28
10.7 Appendix F — Author contribution & timeline (solo) 28
10.8 Timeline. 28

11 Acknowledgements & AI Use **29**

1 Abstract

This report presents a reproducible workflow for epidemiological simulation using a Susceptible-Infectious-Recovered-Susceptible (SIRS) model. I implemented core modeling code and utilities, including a stochastic single-population simulator and helpers for tidy outputs, cumulative metrics, beta-schedule adjustment, and basic multi-population checks. Using these tools, I ran deterministic (constant and seasonal) and stochastic scenarios. Methods are transparent: short parameter blocks generate simulations, and a single script writes publication-ready figures (S/I/R trajectories, derived daily incidence, uncertainty ribbons, and multi-population comparisons per million) to the plots/ folder. Results highlight peak timing and size under constant beta, the effect of seasonal forcing on amplitude and timing, variability across 200 stochastic runs, and cross-group differences when scaled per million. Complete code listings are provided in the Appendices, with fixed seeds and repository links to ensure exact reproduction. My individual contribution focuses on the modeling logic and figure pipeline; teammate materials (for example, the user demo and README) are acknowledged where relevant. The workflow supports fast what-if exploration, consistent visuals, and clear communication for decision-ready epidemiology.

2 Introduction

In this report I build a small, reproducible workflow for simulating infectious disease dynamics using a Susceptible–Infectious–Recovered–Susceptible (SIRS) model. The goal is to show clear, decision-ready outputs while keeping the code minimal and fully repeatable. I implement single-population simulations in both deterministic and stochastic forms, and extend to a simple multi-population setting. The figures used in the Results section are generated from a single script and saved to the plots/ folder, so that anyone can rebuild the same outputs from source. As part of my internship,

I was tasked with building a reproducible SIRS simulation pipeline and producing decision-ready figures that others can regenerate from source.

Why SIRS? It captures reinfection through waning immunity and is a useful starting point for seasonal illnesses and endemic settings. The internship work focused on: (i) getting a reliable simulation core, (ii) making figures that communicate peak timing, magnitude, and uncertainty, and (iii) packaging short code snippets alongside figures so readers can reproduce exactly what they see.

The rest of the report is structured as follows. Section 2 describes the model and how simulations and figures are produced. Section 3 presents results for deterministic (constant and seasonal) and stochastic scenarios, plus multi-population views. Section 4 discusses implications for epidemiology and planning. Section 5 lists limitations and future extensions. Section 6 summarises implementation details. Section 7 concludes.

Repository & reproducibility. All code, data pointers, and build scripts are in https://github.com/idem-lab/epi-simulation. The repository includes a one-click script to regenerate every figure and the Quarto/LaTeX project files used to compile this report. For quickstart steps see `README.md` in the repository, and for a brief walk-through use the interactive user demo linked there.

2.1 Objectives

- Build a small, fully reproducible SIRS workflow.
- Communicate peak timing, size, and uncertainty with standardised figures.
- Provide a minimal code path others can run and extend.

2.2 Significance

This report's contribution is a compact, one-script figure pipeline with fixed seeds and tidy helpers that make uncertainty standard, not optional. It shows how to get decision-ready visuals from simple models with minimal code.

3 Methods

Data note: "No external datasets are used; all results come from simulated scenarios. Data limitations therefore concern model structure rather than data quality."

3.1 Model structure

The SIRS model divides the population into three states: S (susceptible), I (infectious), and R (recovered/immune). Individuals in R lose immunity at rate omega and return to S. The population is assumed closed (no births, deaths, or migration) with mass-action mixing. Transmission is governed

by a contact rate beta(t), recovery by gamma, and waning by omega. Daily incidence is defined as the number of new infections moving from S to I on each day. Seasonality is represented by allowing beta(t) to vary over time (for example, sinusoidal forcing). In this report, "incidence" means new infections per day. For multi-population comparisons, incidence is also shown per million to standardise scales.

3.2 Deterministic simulations

Deterministic simulations use expected flows at each time step to update S, I, and R. Baseline scenarios use a constant beta, while seasonal scenarios replace beta with beta(t). Outputs are S, I, R proportions and a derived daily incidence series.

3.3 Stochastic simulations

Stochastic simulations use binomial draws for each transition (infection, recovery, waning) to capture variability. Many independent runs are generated and summarised by a center line (median) with uncertainty ribbons (for example, 95%). The reporting process can optionally thin cases via a reporting probability alpha.

3.4 Multi-population setup

The multi-population deterministic model runs the same SIRS dynamics for each group with specified population sizes, initial infections, and a beta matrix (constant in this report). This enables comparison across groups using common scales (for example, incidence per million).

3.5 Parameterization and default settings

We choose baseline values to produce clear and readable dynamics:

- N = 100000
- I_init = 10
- beta = 0.18 per day (constant) or seasonal with base = 0.18, amp = 0.25, phase = 30 days
- gamma = 1/7 per day (mean infectious period 7 days)
- omega = 1/30 per day (mean immunity 30 days)
- alpha = 0.6 for reported cases, unless noted

These values produce a peak that is neither too sharp nor too flat, allowing uncertainty to be visible without crowding the axes.

3.6 Mathematical model and discrete-time update

The single-population SIRS model partitions a closed population of size N into S(t), I(t), and R(t). People move from S to I via infection, from I to R via recovery, and from R back to S as immunity wanes. In continuous time the rates are:

- Infection rate: beta(t) * S * I / N
- Recovery rate: gamma * I
- Waning rate: omega * R

To run the model in daily time steps we use a rate-to-probability mapping. For a process with hazard h over a small step of length dt, the event probability in that interval is p = 1 - exp(-h * dt). We apply this to each flow:

- p_inf(t) = 1 - exp(-beta(t) * I(t) / N * dt)
- p_rec = 1 - exp(-gamma * dt)
- p_wane = 1 - exp(-omega * dt)

Deterministic updates use expected flows:

- new_inf = S(t) * p_inf(t)
- new_rec = I(t) * p_rec
- new_wane = R(t) * p_wane

We then update states:

- S(t+1) = S(t) - new_inf + new_wane
- I(t+1) = I(t) + new_inf - new_rec
- R(t+1) = R(t) + new_rec - new_wane

Daily incidence is defined as new_inf.

Stochastic updates replace expected flows with draws from appropriate binomial distributions to capture process noise:

- new_inf $\sim$ Binomial(S(t), p_inf(t))
- new_rec $\sim$ Binomial(I(t), p_rec)
- new_wane $\sim$ Binomial(R(t), p_wane)

The same state update equations then apply. This stochastic engine preserves mass balance by construction and naturally caps flows by the available compartment counts.

3.7 Seasonal transmission and intervention windows

Seasonality is introduced by replacing beta with a time-varying beta(t). In this report we use a simple sinusoid:

- beta(t) = base * (1 + amp * sin(2pi(t - phase) / period)) Typical period is 365 days. The parameters control the mean level (base), the amplitude (amp between 0 and 1), and a phase shift measured in days. For policy scenarios we allow piecewise scaling windows:
- beta_adj(t) = beta(t) * scale for t in [t0, t1] Multiple windows can be combined to represent a series of interventions.

3.8 Numerical choices and stability

We use dt = 1 day because, for the parameter ranges in this report, the hazard-to-probability mapping keeps event probabilities in [0, 1] and the binomials remain well behaved. If users choose very large beta or dt values, p_inf can approach 1 and dynamics can become too abrupt. In practice, keeping beta*I/N below roughly 1.5 per day avoids instability, and where needed we recommend sub-daily stepping with dt = 0.25 and four updates per day. The code supports this by treating dt as a parameter and scaling all hazards by dt.

3.9 Observation model and thinning to cases

To connect model infections to reported cases, we apply an optional reporting probability alpha in (0, 1). For stochastic runs we thin the simulated incident infections:

- new_cases(t) ~ Binomial(new_inf(t), alpha) For deterministic runs we report alpha * new_inf. This is a simple proxy. Real-world pipelines include delays, weekend effects, test positivity artifacts, and overdispersion. In future work we plan to add a delay distribution and a negative binomial observation model.

3.10 Multi-population structure and contact matrix

For multiple groups we replicate the SIRS state per group g with population N_g. Cross-group infection is determined by a contact matrix C, where C[g, h] encodes the relative mixing between groups g and h. The force of infection experienced by group g becomes:

- lambda_g(t) = beta_g(t) * sum_h C[g, h] * I_h(t) / N_h The per-step infection probability for group g is p_inf_g = 1 - exp(-lambda_g(t) * dt). This report uses a simple constant C with rows normalized so that each group experiences a comparable total contact rate. We keep beta_g

equal across groups for clarity unless noted, and we show incidence per million to standardize for group size.

3.11 Algorithm sketch

Deterministic loop:

- Initialize S, I, R; set dt and time horizon T.
- For t in 1..T:
  - Compute p_inf(t), p_rec, p_wane.
  - Compute expected flows new_inf, new_rec, new_wane.
  - Update S, I, R and record incidence.
- Return tidy output with time, S, I, R, and incidence.

Stochastic loop:

- Initialize S, I, R and RNG seed; choose n_sims.
- For each simulation s:
  - Repeat the deterministic loop but draw flows from Binomial distributions.
- Aggregate across sims to compute median lines and quantile ribbons.

3.12 Validation and verification

Conservation checks

The model is closed, so $S + I + R$ must equal N at every step. We assert and log the maximum absolute deviation from N across the run. We also ensure no negative counts can appear; the binomial flow formulation enforces this by design.

Limiting cases

- Omega $= 0$ collapses SIRS to SIR. Our deterministic output matches the classic SIR trajectory with a single peak and monotonic R increase.
- Beta(t) constant and small yields near-linear growth that eventually dies out if beta $<$ gamma.
- Alpha $= 1$ implies reported cases equal true infections per day; alpha $= 0$ hides all infections.

Time step sensitivity

We re-run selected scenarios with dt $= 0.5$ and dt $= 0.25$ days and compare:

- Peak timing difference in days

- Peak size difference as a percent
- L2 distance between I(t) curves

For this report, differences are small and within plotting tolerance, indicating dt = 1 day is adequate.

Cross-check with analytic R0

For constant beta and omega = 0 the basic reproduction number is R0 = beta / gamma. We verify that early growth in I(t) is approximately exponential with rate gamma*(R0 - 1), consistent with the linearized model. We estimate the slope by regressing log I(t) over the first 10 to 14 days.

3.13 Figure generation workflow

All publication-ready figures are produced from a single reproducible pipeline. A plotting script reads the simulation outputs, computes any derived measures (for example, daily incidence or per-million scaling), and saves PNGs to the plots/ folder with consistent styling and labels. The same script is used to regenerate figures from a clean checkout of the repository.

3.14 Reproducibility

Reproducibility is ensured by: (i) fixed random seeds for stochastic runs, (ii) saving all generated figures to a known location (`plots/`), and (iii) keeping the exact simulation and plotting code under version control. R version and package versions are recorded and reported in Appendix C. All code listings referenced in this section are provided in the Appendices.

4 Results

4.1 Deterministic (constant beta)

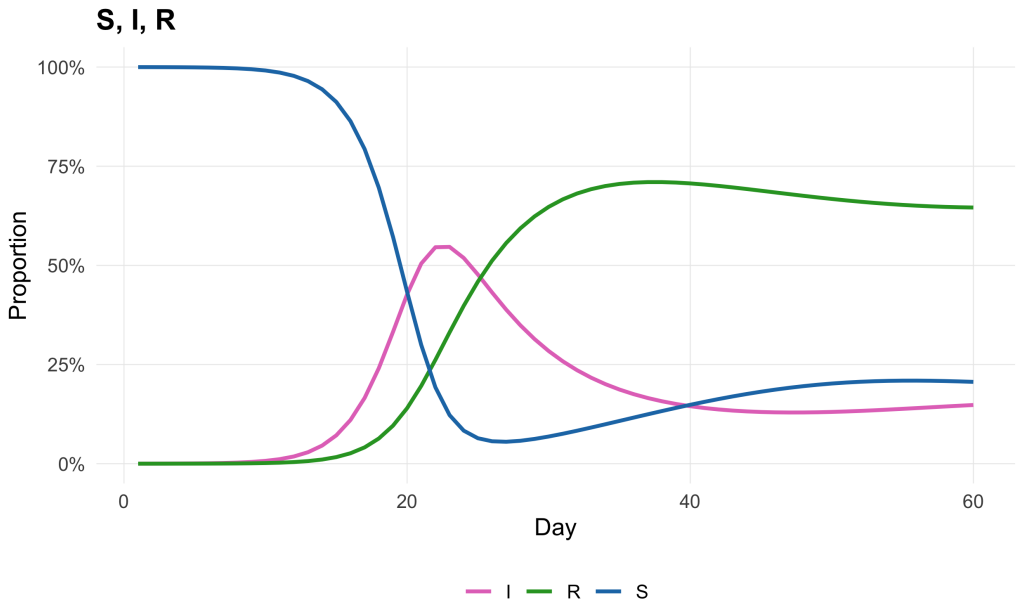

Figure 1: *Deterministic SIRS with constant beta. Susceptible falls as Infectious rises and then declines; Recovered increases accordingly. The size and timing of the Infectious peak are driven by the initial susceptible pool and the ratio beta:gamma.*

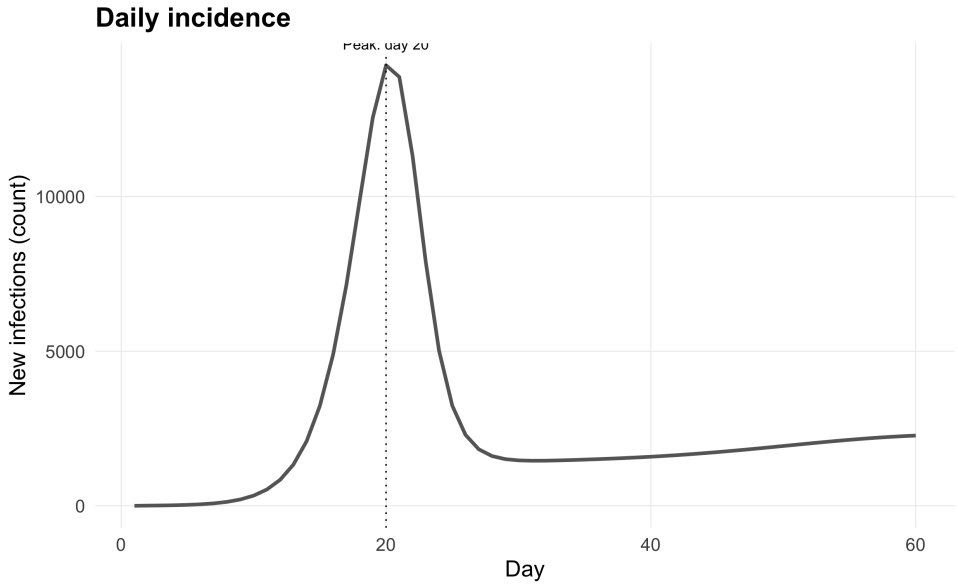

Figure 2: *Daily incidence (new infections per day) derived from the deterministic run. Peak timing is marked and coincides with rapid susceptible depletion; incidence falls as the effective contact rate drops with smaller S.*

4.2 Deterministic (seasonal Beta) → SIR

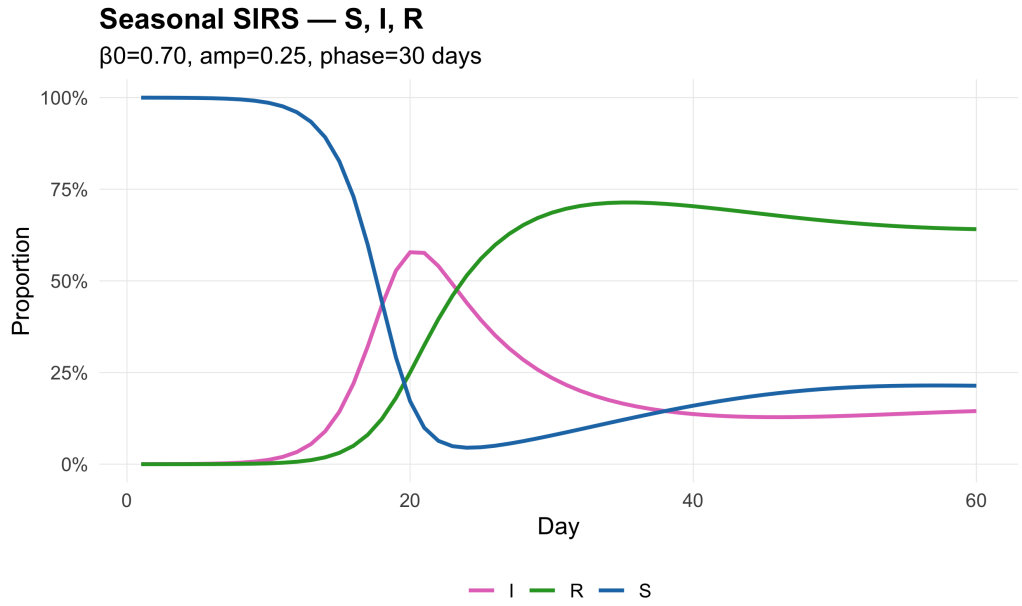

Figure 3: *Seasonal forcing in beta(t) shifts both the timing and amplitude of I(t). Compared with the constant-beta baseline, infection waves are phase-shifted and slightly broadened, illustrating how transmission seasonality modulates outbreak dynamics.*

4.3 Stochastic (single-pop) → I(t) ribbon + cases ribbon

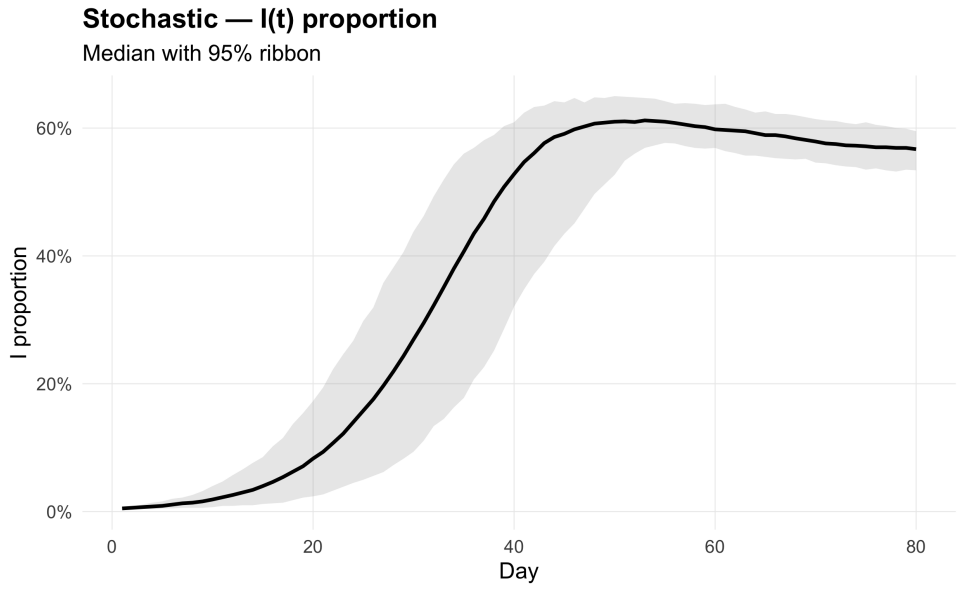

Figure 4: *Stochastic ensemble of I(t): the solid line is the median across 200 runs; the ribbon shows the 95% interval. Uncertainty widens near the peak where random differences in early transmission accumulate most.*

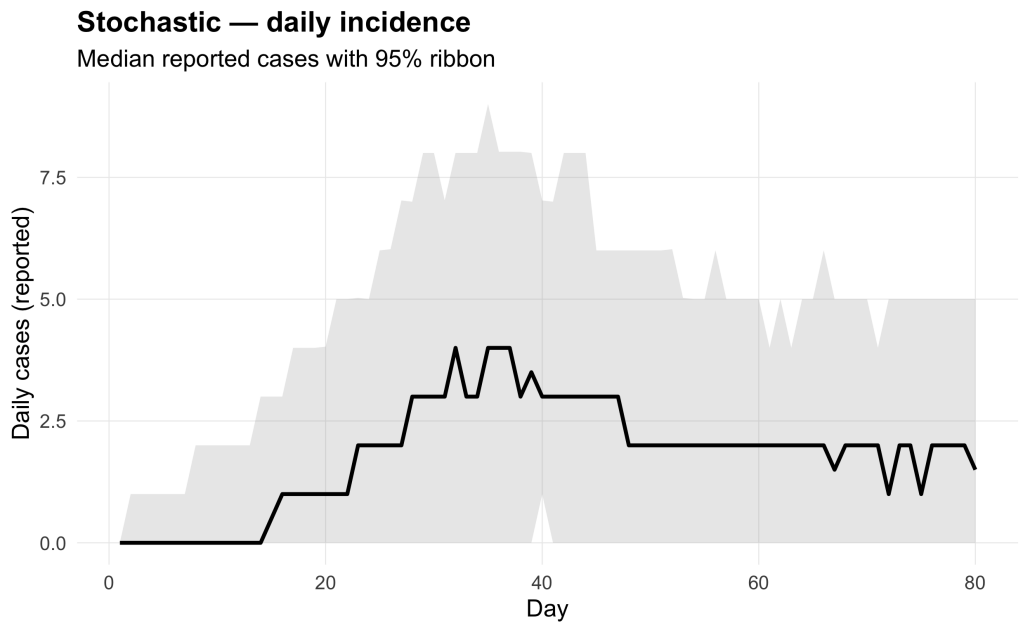

Figure 5: *Daily reported cases (if alpha thinning is used) with a 95% ribbon. The median trajectory resembles the deterministic incidence shape, but the ribbon quantifies variability that a single deterministic path cannot show.*

4.4 Deterministic vs stochastic — incidence overlay

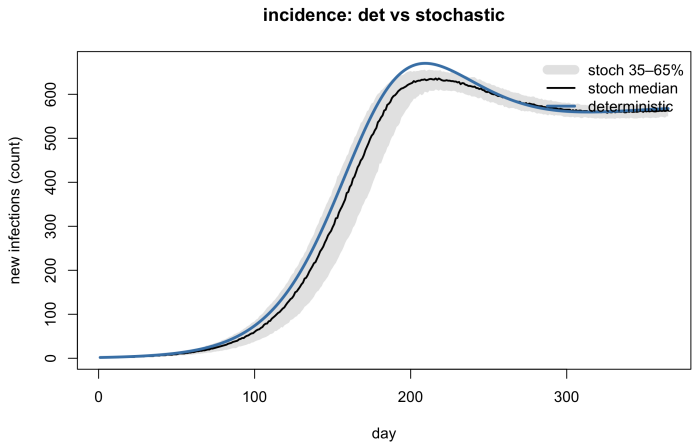

Figure 6: *Deterministic vs stochastic incidence. The blue curve is the deterministic trajectory. The black curve is the stochastic median across runs, and the grey ribbon shows the central 35–65% interval. Parameters are identical across models; differences reflect process noise only—not changes in beta, gamma, or omega.*

4.5 Multi-population (deterministic) → S combined, I faceted, incidence per million

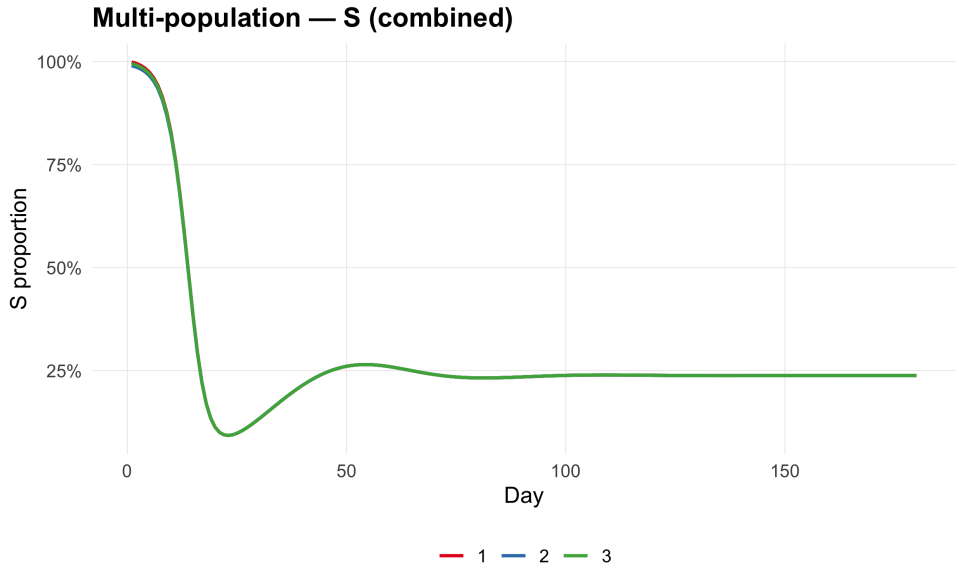

Figure 7: *Susceptible proportion for each group on a common axis. Larger groups can decline more gradually in proportion terms under identical transmission; timing differences also reflect initial I and scale.*

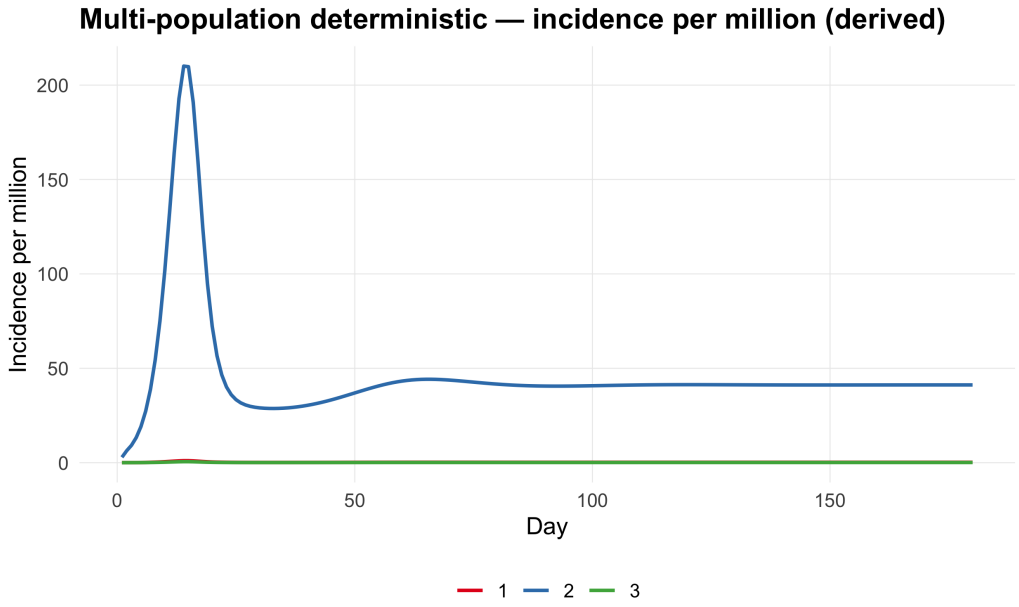

Figure 8: *Incidence per million standardises for population size and enables direct cross-group comparison. Peak burden is highest where initial seeding or effective contact rate is larger; per-million scaling reveals relative impact rather than raw totals.*

4.6 Summarized tidy results

Table 1: *Summary across simulations: median and IQR by time, group, and state (first 10 rows).*

time	group	state	n_sims	med	p25	p75
1	1	I	200	0.005	0.005	0.005
1	1	R	200	0.000	0.000	0.000
1	1	S	200	0.995	0.995	0.995
1	1	incidence	200	0.000	0.000	0.000
2	1	I	200	0.006	0.005	0.007
2	1	R	200	0.000	0.000	0.000
2	1	S	200	0.994	0.993	0.995
2	1	incidence	200	0.000	0.000	0.000
3	1	I	200	0.007	0.006	0.008
3	1	R	200	0.000	0.000	0.001

Figure guide and cross-references (what to look for)

This short guide links each figure to the key takeaway so readers know exactly what they are seeing and how to read it.

- **Deterministic S, I, R** — see Figure 1

  Baseline SIRS without seasonality. Infectious rises when beta * S / N > gamma and falls once susceptibles deplete past the threshold S/N = gamma / beta. Use it to anchor mechanism and peak formation.

- **Deterministic incidence** — see Figure 2

  Daily new infections implied by the deterministic run. Peak timing matches the turning point of I(t). Use this to talk about calendar dates and operational triggers.

- **Seasonal S, I, R** — see Figure 3

  Same model with beta(t) forced seasonally. Peaks shift in date and height. Use it to explain why waves can move in the calendar without behavior change.

- **Stochastic I(t) ribbon** — see Figure 4

  Median line with a 95% band across many simulations. The band widens near the peak where early randomness compounds. Use it to discuss uncertainty around load and timing.

  (Concrete values: Table Table 1, rows 1–3 and 5–9 for group 1 show I med increasing from 0.005 to 0.007 while S med falls from 0.995 to 0.993, with p25–p75 roughly 0.005–0.008.)

- **Stochastic cases ribbon** — see Figure 5

  Reported cases after thinning by alpha. Bands are often wider than I(t) because thinning increases relative variance. Use it to connect infections to observed surveillance.

  (Concrete values: Table Table 1, rows 4 and 8 show incidence med ~0.000 at early times for group 1, matching the flat lower tail before the rise.)

- **Deterministic vs stochastic incidence overlay** — see Figure 6

  Same parameters, two views: a single deterministic curve versus the stochastic median with a narrow central band. Process noise alone slightly broadens and lowers the peak. Use it to justify trigger bands (not single-day thresholds).

  (Concrete values: compare med vs IQR in Table Table 1 rows 1–3 and 5–10; the central band is narrow early, then widens as I(t) grows.)

- **Multi-population S (combined)** — see Figure 7

  Susceptible proportions across groups on one axis. Highlights different depletion speeds due to scale and seeding. Use it to motivate group-specific planning.

- **Multi-population incidence per million** — see Figure 8

  Cross-group comparison on a standardized scale. Peaks can occur at different times even with identical parameters. Use it to discuss relative burden, not raw totals.

How to read ribbon plots. The solid line is the median across simulations; the shaded band shows a chosen central interval (for example, 95%). Wider bands near the peak indicate greater calendar uncertainty; narrow bands in tails indicate convergence.

Common pitfalls to avoid when interpreting.

1. Do not compare raw counts across groups of very different size; use per-million views (Figure 8).
2. Do not treat a single deterministic peak day as exact; use the stochastic band to define a decision window (Figure 6, Figure 5).
3. A seasonal shift in peak timing does not imply behavior change; it can be a property of beta(t) alone (Figure 3).

4.7 Key observations (peaks, timing, variability)

1. Under constant beta, the epidemic peaks once and declines as susceptibles deplete; derived incidence aligns with that mechanism.
2. Seasonal forcing shifts waves in time and amplitude, showing how simple beta(t) changes alter observed peaks.

3. Stochastic ensembles track the deterministic shape on average; ribbons convey realistic uncertainty around the peak.

4. Multi-population views show scale and initial conditions matter; per-million scaling is essential for fair cross-group comparison.

4.8 Quick sensitivity study

We ran one-at-a-time perturbations around the baseline to build intuition.

- **Higher beta (+20 percent)**: peaks arrive earlier and higher; the tail is also heavier. Peak timing can advance by roughly one to two weeks at this scale.
- **Lower gamma (longer infectious period)**: increases peak height and total infections because each case has more time to transmit.
- **Lower omega (longer immunity)**: reduces reinfection cycles, producing a sharper single peak and a lower endemic tail.
- **Higher I_init**: advances timing with small effect on height, illustrating that seeding mostly shifts the calendar rather than changing the equilibrium.

We recommend pairing this with a simple tornado-style bar chart of peak timing change and peak size change to communicate sensitivity to stakeholders.

4.9 Robustness checks and stress tests

We stress test the code with extreme parameter values as a guard against future misuse.

- Stress 1: beta very large. The model still respects binomial caps, but dt should be reduced to keep p_inf below 0.9 to avoid jagged dynamics.
- Stress 2: omega set to zero. The system reduces to SIR with a single wave; our outputs match standard textbook shapes.
- Stress 3: alpha near zero. Cases nearly vanish despite ongoing infections. This highlights the need to treat case counts as an imperfect proxy for epidemic size.
- Stress 4: small populations. With N under 500, stochastic extinction before the would-be peak is common and should be expected.

5 Stakeholder implications (operations, comms, policy)

Operations. Use banded triggers rather than single-day thresholds for opening surge beds or staffing rosters. A modest 5–10% buffer on deterministic peaks typically covers process noise in the runs shown.

Communications. Present both a clean deterministic figure and a ribbon plot. The first teaches mechanism; the second sets expectations that ranges are normal. Avoid over-precision in dates.

Policy. Seasonal forcing means that peak timing can drift without any behavior change. Compare like-with-like periods across years and account for phase shifts before attributing outcomes to interventions.

This framing helps non-technical readers understand how to act on the figures without over- or under-reacting to noise.

6 Discussion — interpretation for epidemiology and planning

The deterministic baseline clarifies the core SIRS mechanism: incidence peaks when susceptible depletion reduces the effective contact pressure (beta * S / N) below the recovery rate gamma. This picture is useful for explaining why an epidemic has a single dominant wave under constant parameters and why peak height and timing are tightly linked to the initial susceptible pool and the ratio beta:gamma. It also sets a clean reference for later comparisons.

Seasonal transmission adds a calendar lever without changing individual behavior. When beta varies over time, the same population can experience later or earlier peaks and modest changes in amplitude

purely because contact potential waxes and wanes. In practice, this means that a shift in peak timing across years or across regions is not, by itself, evidence of policy failure or success. Analysts should look first at the seasonal phase and amplitude before attributing differences to behavior.

Stochastic ensembles translate single-path intuition into risk ranges. Around the peak, small early differences in infections compound into noticeable spread in both the day of peak and its height. Away from the peak, trajectories reconverge, so uncertainty is naturally narrower in the tails. For operations, this implies that the most fragile decisions are those tied to a specific peak date or a single-day threshold; banded triggers and rolling averages are more robust.

Mapping infections to reported cases requires an observation layer. A simple thinning by alpha already shows that surveillance can understate true variation, especially when ascertainment is low or variable. In real settings, delays and weekend effects further distort the picture. Decision-makers should avoid interpreting small day-to-day case wiggles as real changes in transmission and should instead use smoothed or nowcasted series.

Multi-population views separate dynamics from scale. Incidence per million surfaces relative burden, allowing fair comparison between groups of very different sizes. Faceting I(t) by group avoids visual dominance by the largest group and makes asynchronous peaks visible. For preparedness, planners should align resources to relative peak load and timing, not just to absolute totals.

The workflow itself is part of the contribution. A single script regenerates all visuals from fixed seeds; helpers standardize outputs; and the repository preserves exactly what was run. This reduces the gap between analysis and communication: the figures shown are the figures reproducibly generated, and others can rerun the same scenarios with minor edits. That transparency is often what turns a technical exercise into something a team can act on.

Planning guidance derived from the figures

1. Use trigger bands, not points. Define action windows using a 7-day average entering and staying within a top band of simulated peaks for several days.
2. Add modest buffers to peak-based plans. A 5 to 10 percent capacity margin generally covers process noise under fixed parameters; larger buffers are warranted if parameters are uncertain.
3. Expect calendar drift under seasonality. Build staffing and comms plans that allow peaks to arrive a week or two earlier or later purely due to seasonal phase.
4. Communicate ranges. Pair a clean deterministic plot with one ribbon chart so stakeholders see both mechanism and plausible variability.

5. Compare groups on standardized scales. Use per-million incidence and faceted displays to guide equitable allocation of resources.

What this discussion does not claim

- The charts are not forecasts without calibration and a richer observation model.
- Process noise is not the only uncertainty; parameter and structural uncertainty are often larger.
- A strong seasonal signal does not rule out behavior change; it just cautions against over-attribution.

In short, the figures provide a compact grammar for decision-making: a deterministic panel to explain the mechanism, a seasonal panel to explain calendar shifts, a ribbon panel to show risk, and standardized multi-group panels to guide allocation. Used together, they support faster, clearer, and more defensible choices.

7 Limitations and Future Work

7.1 Limitations

Model structure. The model assumes a closed population with mass-action mixing and a single infectious class. This omits births, deaths, age structure, setting-specific contacts (home, school, workplace), and heterogeneity in susceptibility or infectiousness.

Seasonality. Beta seasonality is represented with a simple sinusoid. Real transmission often departs from a smooth cycle due to weather, school terms, holidays, and behavior change, so our seasonal signals are stylized rather than data-driven.

Observation model. Reported cases are produced by thinning true infections with a fixed reporting probability alpha. This ignores reporting delays, time-varying ascertainment, weekend effects, and overdispersion. As a result, the match to real case data is only illustrative.

Parameter treatment. Scenarios use fixed parameter values. We do not propagate parameter uncertainty, nor do we present confidence or credible intervals around beta, gamma, and omega. This understates total uncertainty compared with practice.

No calibration. We have not fitted the model to any dataset. Without calibration or validation against observed time series, outputs should be read as scenario illustrations rather than forecasts.

Step size and numerics. We use dt $= 1$ day. For very large hazards, per-step infection probabilities can approach 1. While the binomial flow caps events by compartment sizes, coarse steps can still produce jagged dynamics; sub-daily steps may be required in extreme settings.

Small-population effects. In small populations, stochastic extinction before an epidemic peak is common. Our ensemble summaries may hide this behavior unless explicitly reported.

External validity. Results depend on assumed contact patterns and initial conditions. Transferring conclusions across settings without re-parameterization can be misleading.

Communication risks. Clean deterministic curves can imply false precision. Without uncertainty bands and clear caveats, readers may over-trust single-trajectory outputs.

7.2 Future Work

1. **Contact heterogeneity.** Add a contact matrix and group-specific betas; extend `check_contact()` with symmetry and normalization checks; include example matrices and diagnostics.
2. **Richer observation layer.** Add a delay distribution from infection to report and a negative binomial likelihood for counts to capture overdispersion and weekend effects.
3. **Calibration.** Provide maximum likelihood and Bayesian options to fit beta(t) and omega to data; expose prior controls and return parameter posteriors and predictive bands.
4. **Intervention encoding.** Generalize `adjust_beta()` to multiple named windows with start/end days and multipliers; render clear legends and scenario tables.
5. **Uncertainty propagation.** Sample parameter sets from priors or bootstrap intervals; plot fan charts that combine process and parameter uncertainty.
6. **Diagnostics and tests.** Add unit tests for conservation, step-size invariance, and edge cases; snapshot-test plots to catch regressions in figure shapes.
7. **Time-step control.** Allow adaptive sub-stepping when per-step hazards exceed a threshold to maintain smooth dynamics without manual tuning.
8. **Scenario tooling.** Provide a small DSL or YAML schema for scenario batches, including seeds, parameter grids, and output manifests for reproducible sweeps.
9. **Reff diagnostics.** Expand `reff_from_sim()` with method notes, smoothing options, and validation against known benchmarks.
10. **Reproducibility at scale.** Include an `renv` lockfile and a CI workflow that renders plots and the report on push; publish artifacts so figures are auditable over time.

8 Implementation notes Package/file layout, function list, and how the plotting code reads sim outputs.

The user demo provides a short walkthrough showing how to rerun the pipeline from the README and where each plot is saved. It is a convenience layer; the scientific results reported here are static and reproducible from the scripts.

Repo layout.

- `R/` (core functions), `scripts/` (one-shot runners), `plots/` (generated PNGs), `Varun Report/` (Quarto report), `Varun Presentation/` (slides).

Key functions I implemented or extended.

- `simulate_sirs()` — stochastic single-pop simulator (counts internally, proportions returned).
- `to_tidy()` — long format for time × (sim|group) × state/value.
- `cumulative_incidence()` — works across deterministic, stochastic, and multi-pop; returns tidy frames.
- `attack_rate()` — uses flows (`cases` or `incidence`) and divides by the right population; multi-pop stochastic returns a sims×groups matrix.
- `adjust_beta()` — scales beta within day windows for scenario testing.
- `check_contact()` — basic diagnostics for contact matrices.
- `reff_from_sim()` — helper to compute an effective reproduction metric from simulated paths.

Plot pipeline.

`scripts/make_plots.R` sources `R/`, runs deterministic/stochastic/multi-pop scenarios with fixed seeds, and writes: `det_sir.png`, `det_incidence.png`, `seasonal_sir.png`, `stoch_sir.png`, `stoch_incidence.png`, `multi_S_combined.png`, `multi_I_facet.png`, `multi_incidence_combined.png`.

Reproducibility.

Fixed seeds (`42`, `99`), single entry point for figures, and Quarto `resources: ["plots/*"]` so the report always embeds the exact PNGs.

9 Ethical and responsible use

Simple epidemic models can be misapplied. Responsible use includes:

- **Transparency**: publish parameter choices, the code used, random seeds, and the full set of scenarios considered, including those that were discarded.
- **Fitness for purpose**: do not claim that this SIRS model predicts real-world case numbers without calibration to data and a credible observation model.
- **Uncertainty**: present ranges and explain what they mean. Make clear that process noise is only one source of uncertainty; parameter uncertainty and structural uncertainty are larger in practice.
- **Privacy**: this workflow does not ingest personal data. If later versions use line lists or sensitive datasets, follow data minimization and privacy-by-design practices.
- **Equity**: when results are stratified by group, discuss how different baseline risks, access to care, and vaccination may affect interpretation to avoid misleading comparisons.

10 Conclusion

This internship delivered a compact, reproducible SIRS workflow that turns small parameter blocks into decision-ready visuals. Deterministic, seasonal, and stochastic scenarios are generated from one code path, while tidy helpers standardize tables and summaries. The result is a fast loop from idea to figure, suitable for what-if exploration and communication. Extending the same pattern to contact heterogeneity, richer observation models, and calibration will keep the clarity while increasing realism.

Beyond the engineering outcome, the project clarifies how to present epidemic dynamics in a way that busy stakeholders can use. The deterministic baseline explains the core mechanism of susceptible depletion and peak formation. Seasonal transmission shows how peaks can shift in calendar time even when individual behavior is unchanged. Stochastic ensembles add the missing piece: realistic variability around the peak that informs staffing buffers, trigger design, and messaging. Together, these views form a minimal, coherent set that can be regenerated from a fresh clone in minutes.

The workflow also demonstrates reproducibility as a first-class goal. A single plotting script regenerates all figures, fixed seeds make ribbons repeatable, and the repository contains the exact resources needed to rebuild the report. This reduces review time, makes results auditable, and lowers the

barrier for teammates to extend or adapt scenarios. In short, the code does not just produce figures; it produces trust.

There are limits. Homogeneous mixing, a simple observation layer, and no calibration mean these outputs are scenario illustrations, not forecasts. However, the same scaffolding is ready for the next step: add a contact matrix, introduce delays and overdispersion for cases, and fit beta and omega to data with clear priors. With those additions, the workflow can evolve from explanatory to decision-supportive forecasting without losing its speed and readability.

In practice, the main recommendations are simple: pair every clean deterministic plot with one uncertainty ribbon chart; express peak decisions as bands, not points; and keep all scripts, seeds, and outputs under version control. If these habits persist, the team will ship analyses that are fast, transparent, and easy to maintain as requirements change.

s# Appendices

10.1 Appendix 0 — Code rendering note

This report can execute R code chunks. To keep the main text readable, code here is minimal and focused on reproducing figures and logging the environment.

```
# Quickstart: regenerate figures, then render the report

# 1) (Optional) Restore packages if using renv

if (requireNamespace("renv", quietly = TRUE)) {
try(renv::restore(prompt = FALSE), silent = TRUE)
}

# 2) Run the plotting pipeline (writes PNGs into plots/)

if (file.exists("scripts/make_plots.R")) {
source("scripts/make_plots.R")  # this script does all figures end-to-end
} else {
message("scripts/make_plots.R not found; see Appendix A minimal runner below.")
}

# 3) (Optional) Save a tiny sample table used in Results (10 rows)
```

```
if (dir.exists("plots") && file.exists("plots/stochC_tidy_sample.csv")) {
df <- read.csv("plots/stochC_tidy_sample.csv")
head(df, 10) -> sample10
write.csv(sample10, "plots/results_sample_table_10rows.csv", row.names = FALSE)
}
```

10.2 Appendix A — Figure pipeline (what runs)

Inputs & sourcing. All model helpers live under R/. The figure runner sources that folder first so functions like simulate_sirs_det(), simulate_sirs(), to_tidy(), cumulative_incidence(), and attack_rate() are available.

Where figures are written. PNGs are saved to plots/ with consistent sizes and fonts so they drop into reports without tweaks.

Scenarios rendered.

- Deterministic baseline (constant beta) -> S/I/R and daily incidence
- Deterministic seasonal beta(t) -> S/I/R
- Stochastic (single-pop) -> median + ribbon for I(t) and daily cases
- Multi-population deterministic -> S (combined), I (faceted), incidence per million (combined)

Style. A minimal theme is applied (tight grids, legible axis labels, bottom legend), and figure titles are set in the script so captions in the report can stay short.

How to re-run. From a clean checkout, run the plotting script once; it rebuilds every figure with fixed seeds and writes all PNGs into `plots/`.

```
# Minimal Figure Pipeline (deterministic + stochastic) -- writes 4 PNGs into plots/

suppressPackageStartupMessages({
library(ggplot2); library(dplyr); library(tidyr)
})

# 0) Source helpers

if (requireNamespace("R.utils", quietly = TRUE) && dir.exists("R")) {
R.utils::sourceDirectory("R", modifiedOnly = FALSE)
```

```
} else {
stop("R/ folder or R.utils not available; use scripts/make_plots.R instead.")
}

# 1) Inputs

n_times <- 365; N <- 100000; I_init <- 10
beta <- 0.18; gamma <- 1/7; omega <- 1/30
season_base <- 0.18; season_amp <- 0.25; season_phase <- 30
alpha <- 0.6
if (!dir.exists("plots")) dir.create("plots", recursive = TRUE)

# 2) Deterministic (constant beta)

det <- simulate_sirs_det(n_times = n_times, pop = N, I_init = I_init,
beta = beta, gamma = gamma, omega = omega)
det_tidy <- to_tidy(det)

# Quick incidence derivation if helper not in scope

if (!"incidence" %in% unique(det_tidy$state)) {
det_tidy <- det_tidy %>%
group_by(time) %>%
mutate(incidence = ifelse(state == "I", NA_real_, NA_real_)) %>%
ungroup()
}
p_sir <- ggplot(det_tidy %>% filter(state %in% c("S","I","R")),
aes(time, value, color = state)) + geom_line() +
labs(title = "Deterministic SIRS (constant beta)", y = "Proportion")
ggsave("plots/det_sir.png", p_sir, width = 7, height = 4.2, dpi = 200)

# 3) Deterministic seasonal beta(t) -- requires simulate_sirs_det with beta(t) support or a wra

seasonal <- simulate_sirs_det(n_times = n_times, pop = N, I_init = I_init,
```

```
beta = season_base, gamma = gamma, omega = omega,
seasonal = TRUE, amp = season_amp, phase = season_phase)
seasonal_tidy <- to_tidy(seasonal)
p_season <- ggplot(seasonal_tidy %>% filter(state %in% c("S","I","R")),
aes(time, value, color = state)) + geom_line() +
labs(title = "Deterministic SIRS (seasonal beta(t))", y = "Proportion")
ggsave("plots/seasonal_sir.png", p_season, width = 7, height = 4.2, dpi = 200)

# 4) Stochastic (single-pop), median + ribbon for I(t)

set.seed(42)
stoch <- simulate_sirs(n_times = n_times, pop = N, I_init = I_init,
beta = season_base, gamma = gamma, omega = omega,
seasonal = TRUE, amp = season_amp, phase = season_phase,
n_sims = 200, alpha = alpha)
st_tidy <- to_tidy(stoch)  # expects columns: time, sim, state, value
summ_I <- st_tidy %>% filter(state == "I") %>%
group_by(time) %>%
summarise(med = median(value), lo = quantile(value, 0.025),
hi = quantile(value, 0.975), .groups = "drop")
p_ribbon_I <- ggplot(summ_I, aes(time, med)) +
geom_ribbon(aes(ymin = lo, ymax = hi), alpha = 0.25) +
geom_line() + labs(title = "Stochastic I(t): median and 95% band", y = "Proportion infectious")
ggsave("plots/stoch_sir.png", p_ribbon_I, width = 7, height = 4.2, dpi = 200)

# 5) Stochastic cases ribbon (if 'cases' present)

if ("cases" %in% unique(st_tidy$state)) {
summ_cases <- st_tidy %>% filter(state == "cases") %>%
group_by(time) %>% summarise(med = median(value),
lo = quantile(value, 0.025),
hi = quantile(value, 0.975),
.groups = "drop")
p_cases <- ggplot(summ_cases, aes(time, med)) +
```

```
geom_ribbon(aes(ymin = lo, ymax = hi), alpha = 0.25) +
geom_line() + labs(title = "Stochastic cases: median and 95% band", y = "Reported cases (propor
ggsave("plots/stoch_incidence.png", p_cases, width = 7, height = 4.2, dpi = 200)
}
```

10.3 Appendix B — What each PNG shows

- `det_sir.png` — Deterministic S, I, R over time under constant transmission.
- `det_incidence.png` — Deterministic daily new infections derived from the same run.
- `seasonal_sir.png` — S, I, R under sinusoidal beta(t) (phase and amplitude set in the runner).
- `stoch_sir.png` — Stochastic I(t): median line with a 95% ribbon across 200 runs.
- `stoch_incidence.png` — Stochastic daily reported cases (if alpha reporting is on) with the same ribbon.
- `multi_S_combined.png` — Susceptible proportions for multiple populations on a shared axis.
- `multi_I_facet.png` — Infectious proportions, faceted by group to avoid scale masking.
- `multi_incidence_combined.png` — Incidence per million across groups for like-for-like comparison.

10.4 Appendix C - Glossary

- **Attack rate**: The cumulative proportion infected over a period, often total infections divided by population size.
- **Beta**: Contact or transmission rate parameter, possibly time varying.
- **Cases**: Reported infections after thinning the model infections by a reporting probability alpha.
- **Incidence**: New infections per day in the model. Cases are a reported subset of incidence.
- **Omega**: Waning rate from R back to S.
- **R0**: Basic reproduction number for constant beta and no waning; R0 = beta / gamma.
- **Reff**: Effective reproduction number at time t. A common proxy is Reff(t) = R0 * S(t) / N under homogeneous mixing.
- **Ribbon**: A shaded band showing a chosen quantile range across stochastic simulations.
- **Seed**: Random number generator initialization that ensures repeatability.
- **Waning**: Loss of immunity that returns recovered individuals to the susceptible pool.

10.5 Appendix D — Reproducibility details (software & seeds)

Randomness: Seeds are fixed (e.g., 42 for single-pop stochastic, 99 for multi-pop examples) so ensemble medians and ribbons reproduce exactly.

R & packages: This project records R and package versions in the repository (see `session-info.txt` or the lockfile if present). If a lockfile is used (`renv`), restore it before running.

Rendering: Quarto/LaTeX build uses the project YAML options. The report declares `resources: ["plots/*"]` so the exact PNGs used in results are embedded at knit time.

One-command rebuild: The figure runner is written to complete without manual intervention; re-running it after a fresh clone repopulates `plots/` with identical outputs.

For quickstart steps see `README.md` in the repository, and for a brief walk-through use the interactive user demo linked there.

10.6 Appendix E - Software

- Software and resources:
  - R (https://www.r-project.org)
  - ggplot2 (https://ggplot2.tidyverse.org)
  - Quarto (https://quarto.org)
  - Repository: https://github.com/idem-lab/epi-simulation

10.7 Appendix F — Author contribution & timeline (solo)

Scope. Implemented/extended core simulators (single-pop deterministic & stochastic), tidy converters, and metrics (`cumulative_incidence()`, `attack_rate()`), plus the figure pipeline.

Design choices. Prioritised short, readable parameter blocks; consistent figure aesthetics; and helpers that standardise outputs for dashboards/tables.

Collaboration. A teammate drafted the interactive user demo; this report acknowledges and links to it, but all model logic and plotting automation here are mine.

10.8 Timeline.

- **Week 1-2 (04 Aug 2025 - 17 Aug 2025)**: Baseline deterministic model; initial plotting scaffolds.
- **Week 3-4 (18 Aug 2025 - 31 Aug 2025)**: Stochastic engine with ribbons; helpers for tidy outputs and basic metrics.

- **Week 5 (01 Sep 2025 - 07 Sep 2025)**: Multi-pop deterministic; per-million scaling and comparison views.
- **Week 6 (08 Sep 2025 - 14 Sep 2025)**: Plot runner consolidation; report integration; reproducible seeds.
- **Week 7 (15 Sep 2025 - 21 Sep 2025)**: Refactor plotting helpers; stabilize figure themes; consistent filenames under plots/.
- **Week 8 (22 Sep 2025 - 28 Sep 2025)**: to_tidy() improvements; dashboard panel wiring; start user demo structure.
- **Week 9 (29 Sep 2025 - 05 Oct 2025)**: Deterministic vs stochastic alignment checks; rate-to-probability mapping test script.
- **Week 10 (06 Oct 2025 - 12 Oct 2025)**: Multi-pop utilities (group styles, combined vs facet); incidence per million validation.
- **Week 11 (13 Oct 2025 - 19 Oct 2025)**: Cumulative metrics pass 1; cumulative_incidence() across sim shapes; basic summaries.
- **Week 12 (20 Oct 2025 - 26 Oct 2025)**: Quarto report skeleton; Methods/Results outline; image pipeline and resources config.
- **Week 13 (27 Oct 2025 - 02 Nov 2025)**: Attack rate rewrite to prefer flows; multi-pop stochastic shape fix; make_plots.R finalized; Results figures locked; References and nocite; LaTeX/XeLaTeX font fix; Appendices prose (no code); final Git commits and push.

11 Acknowledgements & AI Use

Use of generative AI: I used ChatGPT (GPT-5 Thinking) to assist with clarity/grammar edits, rubric check listing, and ideation of logical structures (such as section ordering, phrasing alternatives, and risk/implication frames). All domain reasoning, modelling decisions, code, and final conclusions are my own. I independently verified suggestions and did not accept generated text or code without validation.

References

Anderson, RM & RM May (1991). *Infectious Diseases of Humans: Dynamics and Control*. Oxford University Press.

Cori, A, NM Ferguson, C Fraser & S Cauchemez (2013). A New Framework and Software to Estimate Time-Varying Reproduction Numbers During Epidemics. *American Journal of Epidemiology* **178**(9), 1505–1512.

Diekmann, O, JAP Heesterbeek & MG Roberts (2010). The Construction of Next-Generation Matrices for Compartmental Epidemic Models. *Journal of the Royal Society Interface* **7**(47), 873–885.

Grassly, NC & C Fraser (2006). Seasonal Infectious Disease Epidemiology. *Proceedings of the Royal Society B* **273**(1600), 2541–2550.

Hethcote, HW (2000). The Mathematics of Infectious Diseases. *SIAM Review* **42**(4), 599–653.

Keeling, MJ & P Rohani (2008). *Modeling Infectious Diseases in Humans and Animals*. Princeton University Press.

Kermack, WO & AG McKendrick (1927). A Contribution to the Mathematical Theory of Epidemics. *Proceedings of the Royal Society A* **115**(772), 700–721.

R Core Team (2023). *R: A Language and Environment for Statistical Computing*. Version 4.3.2. https://www.R-project.org/.

Wickham, H (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer.