Lesson 5: Case Study — Measles in Large and Small Towns

Aaron A. King   Edward L. Ionides   Kunyang He

Sunday, March 29, 2026

Objectives

- To display a published case study using plug-and-play methods with non-trivial model complexities.
- To show how extra-demographic stochasticity can be modeled.
- To demonstrate the use of covariates in `pypomp`.
- To demonstrate the use of profile likelihood in scientific inference.
- To discuss the interpretation of parameter estimates.
- To emphasize the potential need for extra sources of stochasticity in modeling.

Challenges in inference from disease dynamics

Understanding, forecasting, and managing epidemiological systems increasingly depends on models. Dynamic models can be used to test causal hypotheses.

Real epidemiological systems:

- are nonlinear
- are stochastic
- are nonstationary
- evolve in continuous time
- have hidden variables
- can be measured only with (large) error

Measles is the paradigm for a nonlinear ecological system that can be well described by low-dimensional nonlinear dynamics.

A tradition of careful modeling studies have proposed and found evidence for a number of specific mechanisms, including

- a high value of $\mathscr{R}_0$ (c. 15–20)
- under-reporting
- seasonality in transmission rates associated with school terms
- response to changing birth rates
- a birth-cohort effect
- metapopulation dynamics
- fadeouts and reintroductions that scale with city size
- spatial traveling waves

Much of this evidence has been amassed from fitting models to data, using a variety of methods. See Rohani and King (2010) for a review of some of the high points.

Measles in England and Wales

We revisit a classic measles data set, weekly case reports in 954 urban centers in England and Wales during the pre-vaccine era (1950–1963).

We examine questions regarding:

- measles extinction and recolonization
- transmission rates
- seasonality
- resupply of susceptibles

We use a model that

1. expresses our current understanding of measles dynamics
2. includes a long list of mechanisms that have been proposed and demonstrated in the literature
3. cannot be fit by previous likelihood-based methods

We examine data from large and small towns using the same model, something no existing methods have been able to do.

We ask: does our perspective on this disease change when we expect the models to explain the data in detail? What bigger lessons can we learn regarding inference for dynamical systems?

Data sets

He et al. (2010) studied twenty towns, including

- 10 largest cities in England and Wales
- 10 smaller towns, chosen at random

Population sizes ranged from about 2,000 to 3.4 million. Weekly case reports span 1950–1963, with annual birth records and population sizes from 1944–1963.

```
all_data = UKMeasles.subset()
measles = all_data["measles"]
demog = all_data["demog"]
```

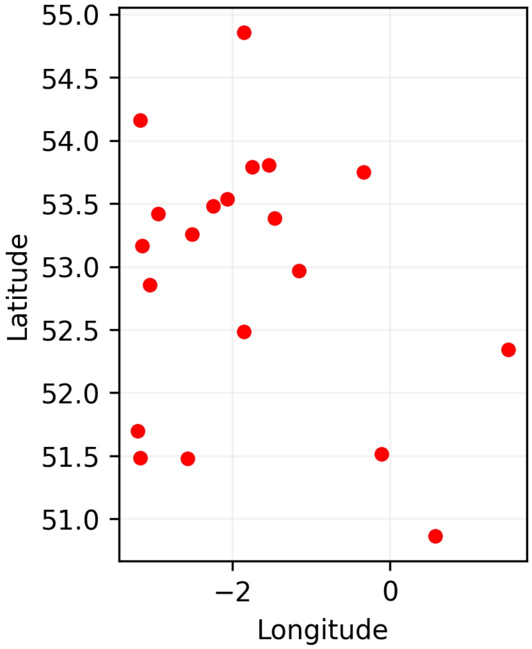
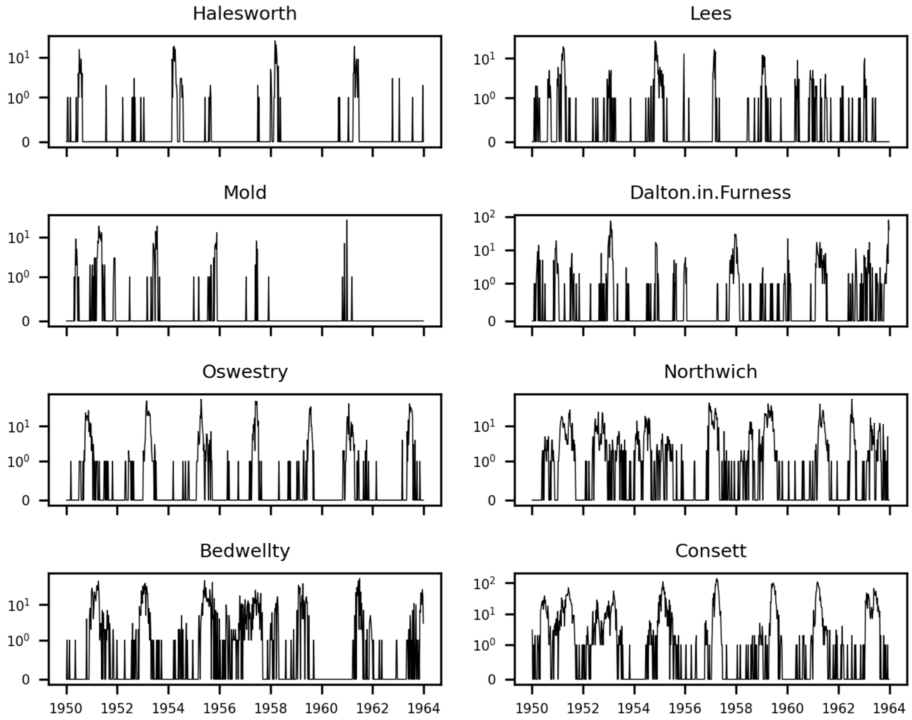


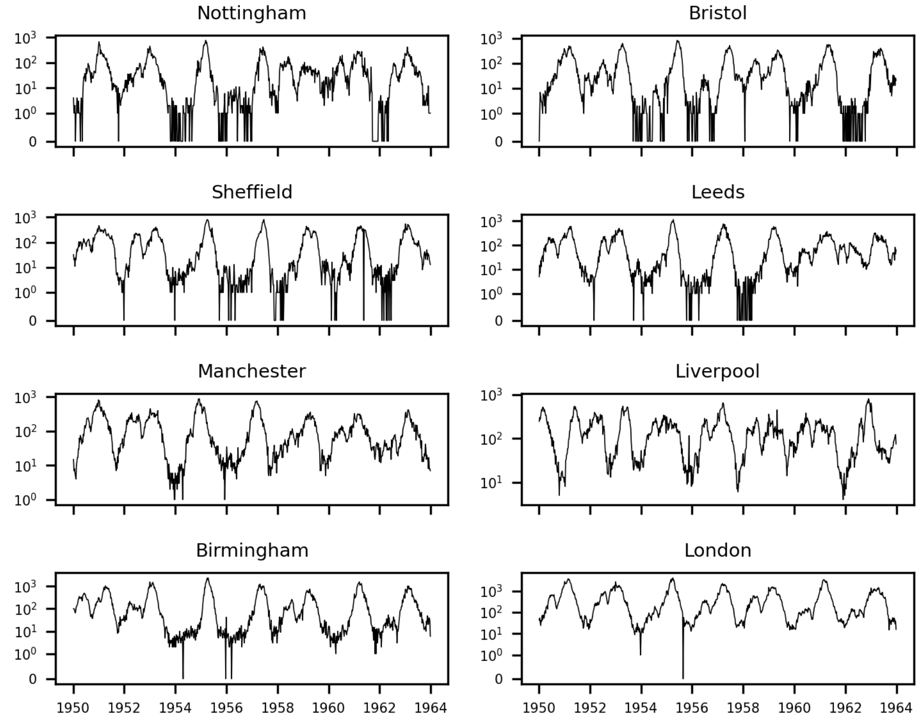
Figure 1: Locations of the 20 cities in the He et al. (2010) analysis.

Map of cities in the analysis

City case counts: smallest 8 cities

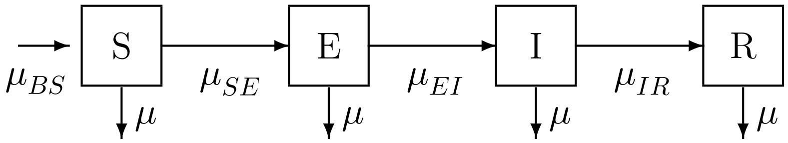


City case counts: largest 8 cities



Continuous-time Markov process model

The model is a SEIR (susceptible–exposed–infectious–recovered) compartment model with births and deaths:



Model specification

Covariates (from data):

- $B(t)$: birth rate
- $N(t)$: population size

Entry into susceptible class (birth-cohort effect):

$$\mu_{BS}(t) = (1-c)\,B(t-\tau) + c\,\delta(t-\lfloor t \rfloor)\int_{t-1}^{t} B(t-\tau-s)\,ds$$

- c: cohort effect (fraction entering at school start)
- τ: school-entry delay
- $\lfloor t \rfloor$: most recent September~1 before t

Force of infection:

$$\mu_{SE}(t) = \frac{\beta(t)}{N(t)}\,(I+\iota)\,\zeta(t)$$

- ι: imported infections
- $\zeta(t)$: Gamma white noise with intensity σ_{SE} (He et al., 2010; Bhadra et al., 2011)

School-term transmission:

$$\beta(t) = \begin{cases} \beta_0\,(1+a(1-p)/p) & \text{during term} \\ \beta_0\,(1-a) & \text{during vacation} \end{cases}$$

- a: amplitude of seasonality
- $p = 0.7589$: fraction of the year children are in school
- The factor $(1-p)/p$ ensures the average transmission rate is β_0.

Overdispersed measurement model:

$$\text{cases}_t \,\big|\, \Delta N_{IR} = C_t \sim \text{Normal}\big(\rho\,C_t,\ \rho\,(1-\rho)\,C_t + (\psi\,\rho\,C_t)^2\big)$$

- ρ: reporting rate
- ψ: overdispersion parameter
- When $\psi = 0$, the variance–mean relation reduces to that of the binomial distribution.

The partially observed Markov process model

We require a simulator for our model. Notable complexities include:

1. Incorporation of the known birthrate.
2. The birth-cohort effect: a specified fraction of the cohort enters the susceptible pool all at once.
3. Seasonality in the transmission rate: higher during school terms than holidays.
4. Extra-demographic stochasticity in the form of a Gamma white-noise term acting multiplicatively on the force of infection.
5. Demographic stochasticity implemented using Euler-multinomial distributions.

Implementation of the process model

The process model is implemented in `pypomp.measles.model_001b`. Let's walk through it.

State variables and parameters:

```
statenames = ["S", "E", "I", "R", "W", "C"]
accumvars = ["W", "C"]

param_names = (
    "R0", "sigma", "gamma", "iota", "rho",
    "sigmaSE", "psi", "cohort", "amplitude",
    "S_0", "E_0", "I_0", "R_0",
)
```

- C is the true incidence (accumulator for new infections), reset each observation interval.
- W is cumulative white noise, useful for diagnostics.

Cohort effect: at the start of the school year (day~251), a fraction `cohort` of the year's births enter susceptibles all at once:

```
t_mod = t - jnp.floor(t)
is_cohort_time = jnp.abs(t_mod - 251.0/365.0) < 0.5*dt
br = jnp.where(
    is_cohort_time,
    cohort * birthrate / dt + (1 - cohort) * birthrate,
    (1 - cohort) * birthrate,
)
```

Term-time seasonality:

```
t_days = t_mod * 365.25
in_term_time = (
    ((t_days >= 7) & (t_days <= 100))
    | ((t_days >= 115) & (t_days <= 199))
    | ((t_days >= 252) & (t_days <= 300))
    | ((t_days >= 308) & (t_days <= 356))
)
seas = jnp.where(
    in_term_time,
    1.0 + amplitude * 0.2411 / 0.7589,
    1 - amplitude
)
beta = R0 * seas * (1.0 - jnp.exp(-(gamma + mu)*dt)) / dt
```

Extra-demographic stochasticity and transitions:

```
# Force of infection
foi = beta * (I + iota) / pop

# Gamma white noise
dw = fast_approx_rgamma(keys[0], dt/sigmaSE**2) * sigmaSE**2

# Rates for Euler-multinomial transitions
rate = jnp.array([foi*dw/dt, mu, sigma, mu, gamma, mu])

# Poisson births
births = fast_approx_rpoisson(keys[1], br * dt)

# Euler-multinomial transitions
transitions = fast_approx_rmultinom(keys[2], populations, rt_final)
```

- `fast_approx_rgamma`, `fast_approx_rpoisson`, and `fast_approx_rmultinom` are JAX-compatible random variate generators provided by `pypomp`.

State update:

```
S = S + births - trans_S[0] - trans_S[1]
E = E + trans_S[0] - trans_E[0] - trans_E[1]
I = I + trans_E[0] - trans_I[0] - trans_I[1]
R = pop - S - E - I
W = W + (dw - dt) / sigmaSE
C = C + trans_I[0]
```

- Since recognized measles cases are quarantined, most infection occurs before case recognition.

True incidence C counts individuals progressing from I to R.

State initializations

The initial state allocates the population across compartments according to proportions S_0, E_0, I_0, R_0:

```
def rinit(theta_, key, covars, t0):
    S_0, E_0, I_0, R_0 = (
        theta_["S_0"], theta_["E_0"],
        theta_["I_0"], theta_["R_0"])
    m = covars["pop"] / (S_0 + E_0 + I_0 + R_0)
    S = jnp.round(m * S_0)
    E = jnp.round(m * E_0)
    I = jnp.round(m * I_0)
    R = jnp.round(m * R_0)
    return {"S": S, "E": E, "I": I, "R": R,
        "W": 0, "C": 0}
```

Measurement model

We model both under-reporting and measurement error. We want $\mathbb{E}[\text{cases}|C] = \rho\, C$ and $\text{Var}[\text{cases}|C] = \rho\,(1-\rho)\,C + (\psi\,\rho\,C)^2$.

Specifically, cases $\mid C \sim f(\cdot \mid \rho, \psi, C)$, where

$$\begin{aligned} f(c \mid \rho, \psi, C) \quad &= \quad \Phi\big(c + \tfrac{1}{2},\, \rho C,\, \rho(1-\rho)C + (\psi\rho C)^2\big) \\ & \quad\quad -\Phi\big(c - \tfrac{1}{2},\, \rho C,\, \rho(1-\rho)C + (\psi\rho C)^2\big)\,. \end{aligned}$$

Here, $\Phi(x, \mu, \sigma^2)$ is the c.d.f. of the normal distribution with mean μ and variance σ^2.

The measurement density `dmeas`:

```
def dmeas(Y_, X_, theta_, covars, t):
    rho, psi, C = theta_["rho"], theta_["psi"], X_["C"]
    m = rho * C
    v = m * (1.0 - rho + psi**2 * m)
    sqrt_v = jnp.sqrt(v) + 1e-18
    y = Y_["cases"]
    upper = jax.scipy.stats.norm.cdf(y + 0.5, m, sqrt_v)
    lower = jax.scipy.stats.norm.cdf(y - 0.5, m, sqrt_v)
    lik = jnp.where(y > 1e-18, upper - lower, upper) + 1e-18
    return jnp.log(lik)
```

The measurement simulator `rmeas`:

```
def rmeas(X_, theta_, key, covars, t):
    rho, psi, C = theta_["rho"], theta_["psi"], X_["C"]
    m = rho * C
    v = m * (1.0 - rho + psi**2 * m)
    cases = jax.random.normal(key) * (jnp.sqrt(v) + 1e-18) + m
    return jnp.where(cases > 0.0, jnp.round(cases), 0.0)
```

Data and covariates

The `pypomp` package includes the UK measles data from He et al. (2010). The `UKMeasles` class provides convenient access.

We illustrate using London:

```
london_data = UKMeasles.subset(units=["London"])
dat = london_data["measles"]
print(f"Date range: {dat['date'].min()}",
    f" to {dat['date'].max()}")
print(f"Number of observations: {len(dat)}")
```

```
Date range: 1944-01-07 00:00:00  to 1965-03-26 00:00:00
Number of observations: 1108
```

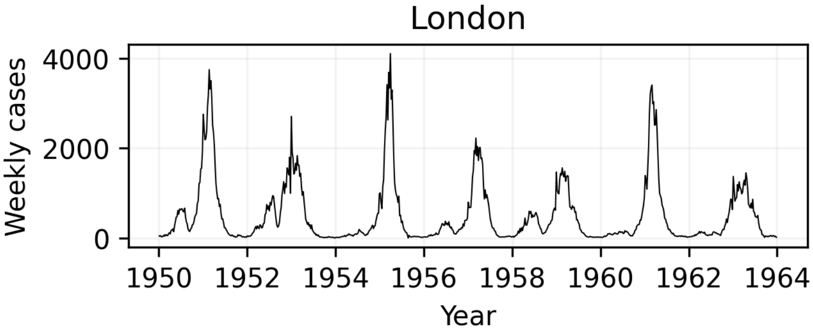


Figure 2: Weekly measles case reports, London 1950–1963.

We smooth the covariates and delay the entry of newborns into the susceptible pool. This is handled internally by `UKMeasles.Pomp()`, which uses smoothing splines for population interpolation and shifts the birth rate by 4 years (the school-entry delay τ).

Constructing the POMP object

```
theta_london = mles["London"].to_dict()

m1 = UKMeasles.Pomp(
    unit=["London"],
    theta=theta_london,
    model="001b",
    interp_method="shifted_splines",
    first_year=1950,
    last_year=1963,
    dt=1/365.25,
    clean=True
)
```

```
State names: ['S', 'E', 'I', 'R', 'W', 'C']
```

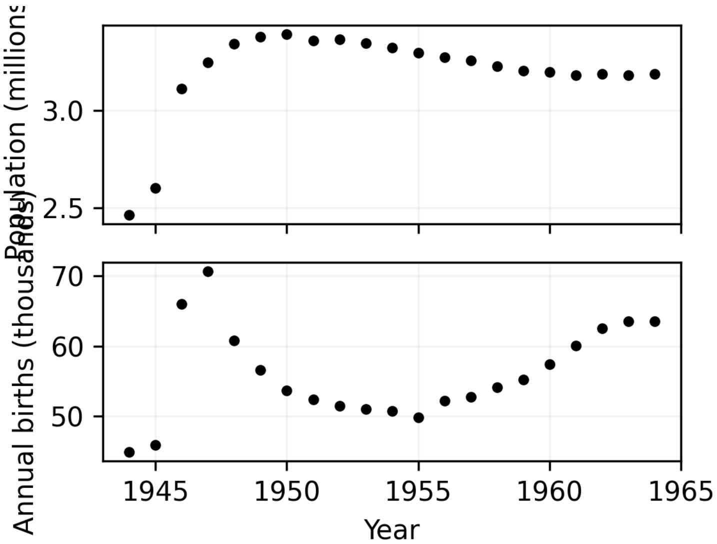


Figure 3: Population and birth-rate covariates for London.

```
Time range: 1950.01 to 1963.98
Observations: 730
```

Estimates from He et al. (2010)

He et al. (2010) estimated the parameters of this model for all 20 cities. We verify that we get the same likelihood.

```
key = jax.random.key(998468235)
cache_file = cache_dir + "/london_pf.pkl"
if os.path.exists(cache_file):
    with open(cache_file, 'rb') as f:
        london_pf = pickle.load(f)
else:
    m1.pfilter(J=[50,500,5000][RL],
        reps=[2,5,10][RL],
        key=key, thresh=0, CLL=True)
    london_pf = m1.results_history[-1]
    with open(cache_file, 'wb') as f:
        pickle.dump(london_pf,f)

london_logliks = london_pf.logLiks.values.flatten()
ll = logmeanexp(london_logliks)
ll_se = logmeanexp_se(london_logliks)
print(f"Log-likelihood: {ll:.1f} (SE {ll_se:.2f})")
```

```
Log-likelihood: -3806.1 (SE 0.25)
```

Simulations at the MLE

```
key = jax.random.key(42)
X_sims, Y_sims = m1.simulate(key=key, nsim=3)
```

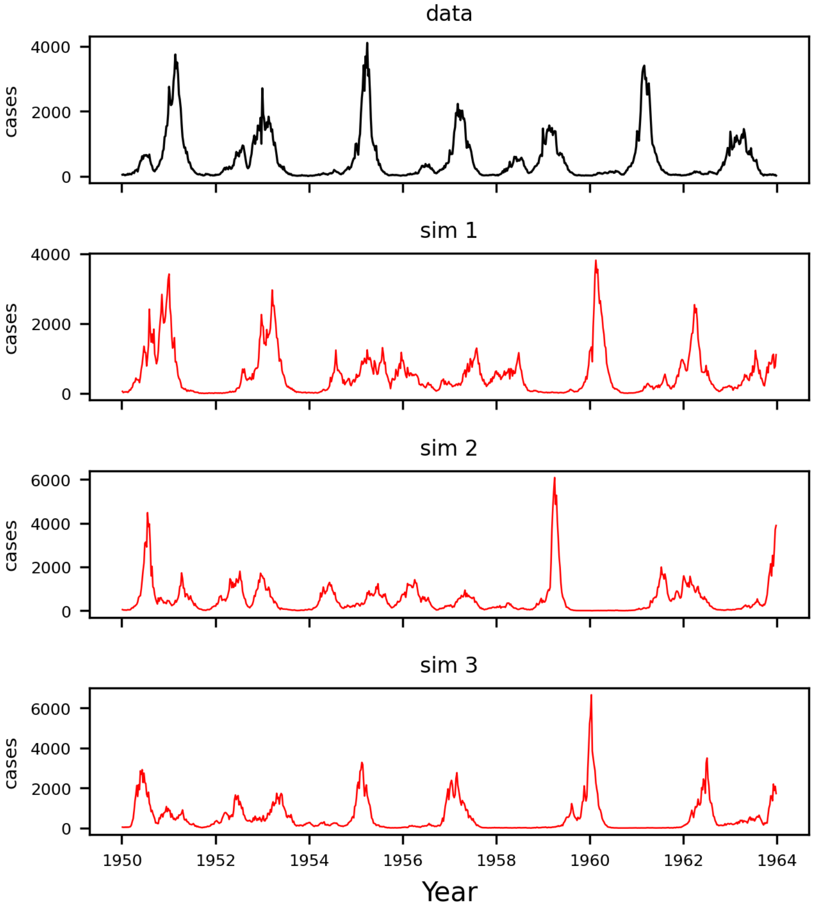


Figure 4: Simulations (red) versus data (black) at the London MLE.

Parameter transformations

The parameters are constrained to be positive, and some lie between 0 and 1. We turn the likelihood maximization problem into an unconstrained one by transforming:

- **Positive parameters** (R_0, σ, γ, ι, σ_{SE}, ψ): log transform
- **Parameters in** $(0, 1)$ (ρ, cohort, amplitude): logit transform
- **Initial proportions** (S_0, E_0, I_0, R_0): log-barycentric transform

These are implemented in `model_001b.to_est` and `model_001b.from_est`.

ARMA benchmark

Linear, Gaussian auto-regressive moving-average (ARMA) models provide a flexible non-mechanistic benchmark.

```
from statsmodels.tsa.arima.model import ARIMA

cases = m1.ys["cases"].values
```

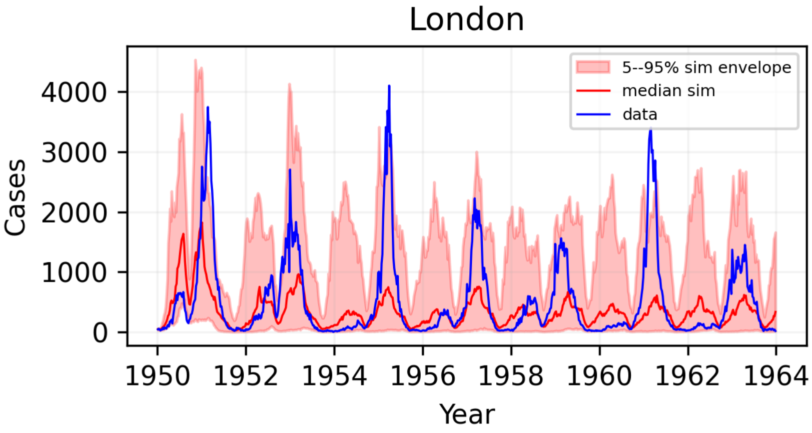


Figure 5: Simulation envelope (100 sims) versus data.

```
log_y = np.log(np.maximum(cases, 1))

arma_fit = ARIMA(log_y, order=(2, 0, 2)).fit()
arma_loglik = arma_fit.llf - np.sum(log_y)

print(f"ARMA(2,2) log-lik: {arma_loglik:.1f} (p=5)")
print(f"SEIR model log-lik: {ll:.1f} (p=12)")
```

```
ARMA(2,2) log-lik: nan (p=5)
SEIR model log-lik: -3806.1 (p=12)
```

- Minimizing AIC, $2p - 2\ell$, is equivalent to maximizing $\ell - p$.
- The aim of mechanistic modeling is not to beat benchmarks, but falling far behind can diagnose problems.
- “Far” means many log units: differences of log-likelihoods are invariant to the scale of measurement.

Log-likelihood anomalies

The benchmark and model log-likelihoods can both be decomposed as a sum of conditional log-likelihoods,

$$\ell(\theta) = \sum_{n=1}^{N} \ell_n(\theta) \quad \text{where} \quad \ell_n(\theta) = \log f_{Y_n|Y_{1:n-1}}(y_n^* \,|\, y_{1:n-1}^*;\, \theta).$$

The **anomaly** for the model at time t_n is the difference between the model conditional log-likelihood and that of the benchmark.

Anomalies can be used similarly to regression residuals: they can indicate points where the model fails; patterns can reveal scope for model improvement.

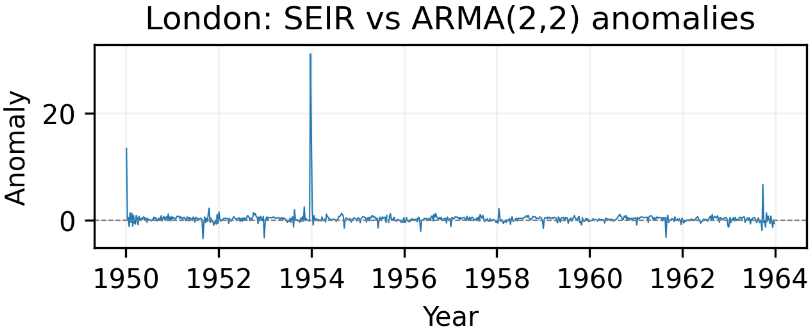


Figure 6: Conditional log-likelihood anomalies for London. Positive anomalies indicate the model outperforms ARMA; negative anomalies indicate the model struggles.

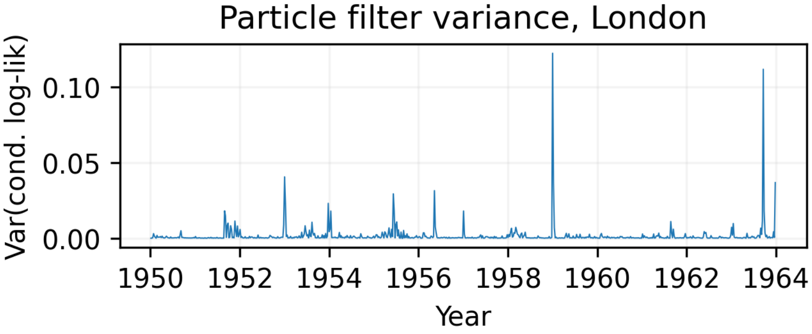


Figure 7: Variance of conditional log-likelihoods across replicates. Observations with high variance are numerically problematic.

Particle filter variance

Results from He et al. (2010)

The fitting procedure used is as follows:

- A large number of searches were started at points across the parameter space.
- Iterated filtering was used to maximize the likelihood.
- Point estimates of all parameters were obtained for 20 cities.
- Profile likelihoods were constructed to quantify uncertainty in London and Hastings.

The linked document `lesson5_profile.qmd` shows how a likelihood profile can be constructed using IF2 in `pypomp`.

Setting up IF2

The IF2 algorithm requires specifying random walk standard deviations for each parameter being estimated:

```python
from pypomp import RWSigma

rw_sd = RWSigma(
    sigmas={
        "R0": 0.02, "sigma": 0.02, "gamma": 0.02,
        "iota": 0.0, "rho": 0.0,
        "sigmaSE": 0.02, "psi": 0.02,
        "cohort": 0.02, "amplitude": 0.02,
        "S_0": 0.02, "E_0": 0.02, "I_0": 0.02, "R_0": 0.02
    },
    init_names=["S_0", "E_0", "I_0", "R_0"]
)

m1.mif(J=2000, M=50, a=0.95, rw_sd=rw_sd,
       key=jax.random.key(1234567), thresh=0)
```

```
             town    pop   R0    a   LP   IP  iota   rho  psi  sigSE
       Halesworth   2171 33.1 0.38  7.9  2.3  0.01 0.754 0.64  0.075
             Lees   4247 29.7 0.15  8.5  2.1  0.03 0.612 0.68  0.080
             Mold   6409 21.4 0.27  5.9  1.8  0.01 0.131 2.87  0.054
Dalton.in.Furness  10560 28.3 0.20  5.5  2.0  0.04 0.455 0.82  0.078
         Oswestry  10970 52.9 0.34 10.3  2.7  0.03 0.631 0.48  0.070
        Northwich  18330 30.1 0.42  8.5  3.0  0.06 0.795 0.40  0.086
        Bedwellty  28930 24.7 0.16  6.8  3.0  0.04 0.311 0.95  0.061
          Consett  39130 35.9 0.20  9.1  2.7  0.07 0.650 0.41  0.071
         Hastings  65690 34.2 0.30  7.0  5.4  0.19 0.695 0.40  0.096
          Cardiff 244600 34.4 0.22  9.9  3.1  0.14 0.602 0.27  0.054
         Bradford 294300 32.1 0.24  8.5  3.4  0.24 0.599 0.19  0.045
             Hull 302100 38.9 0.22  9.2  5.5  0.14 0.582 0.26  0.064
       Nottingham 307000 22.6 0.16  5.7  3.7  0.17 0.609 0.26  0.038
          Bristol 442600 26.8 0.20  6.2  4.9  0.44 0.626 0.20  0.039
            Leeds 509700 47.8 0.27  9.5 10.9  1.25 0.666 0.17  0.078
        Sheffield 515000 33.1 0.31  7.2  6.4  0.85 0.649 0.18  0.043
       Manchester 704500 32.9 0.29 11.1  6.9  0.59 0.550 0.16  0.055
        Liverpool 802300 48.1 0.30  7.9  9.8  0.26 0.494 0.14  0.053
```

```
Birmingham 1117900 43.4 0.43  8.5 11.6  0.34 0.544 0.18  0.061
    London 3389620 56.8 0.55 13.1 12.5  2.90 0.488 0.12  0.088
```

Some parameter estimates vary with city size

The Spearman rank correlations between each parameter and N_{1950} reveal which parameters vary systematically with city size.

```
Spearman r(parameter, N_1950):
  R0           : +0.46
  a            : +0.31
  LP           : +0.31
  IP           : +0.95
  iota         : +0.93
  rho          : -0.20
  psi          : -0.93
  sigSE        : -0.31
```

Imported infections

$$\text{force of infection} = \mu_{SE} = \frac{\beta(t)}{N(t)}\,(I+\iota)\,\zeta(t)$$

Profile likelihoods over the imported infections parameter ι (computed in `lesson5_profile.qmd`) show that:

- For London, ι is on the order of a few cases per week—the huge population sustains transmission and imported cases play a relatively small role.
- For Hastings, ι is much larger relative to population, consistent with small-town dynamics where local fadeouts require reintroduction.

Seasonality

Profile likelihoods over the amplitude of term-time seasonality show that:

- Both London and Hastings have high seasonality amplitude ($a > 0.4$).
- School terms drive measles transmission strongly.
- The confidence intervals are fairly tight, indicating this parameter is well-identified.

Cohort effect

Profile likelihoods over the cohort entry fraction show that:

- The cohort effect is moderately well-identified in London.
- In Hastings, the profile is flatter, reflecting weaker identifiability in small populations.

R_0 estimates inconsistent with literature

Recall that $\mathcal{R}_0$ is a measure of how communicable an infection is. Existing estimates ($\mathcal{R}_0 \approx 15$–20) come from serology surveys and feature-based methods.

- The MLE estimates of $\mathcal{R}_0$ from He et al. (2010) are much higher: $\mathcal{R}_0 \approx 40$–60.

- This reflects the fact that the susceptible fraction is much smaller than in a "virgin population" assumed by classical estimates.
- Profile likelihoods over $\mathcal{R}_0$ confirm this finding for both London and Hastings.

Extrademographic stochasticity

$$\mu_{SE} = \frac{\beta(t)}{N(t)}\,(I+\iota)\,\zeta(t)$$

- $\sigma_{SE} > 0$ is strongly supported by the data.
- Without extra-demographic stochasticity, the model fits poorly.
- Profile likelihoods over σ_{SE} also reveal trade-offs with the infectious period (IP) and latent period (LP): as σ_{SE} increases, IP tends to decrease.

Reporting rate

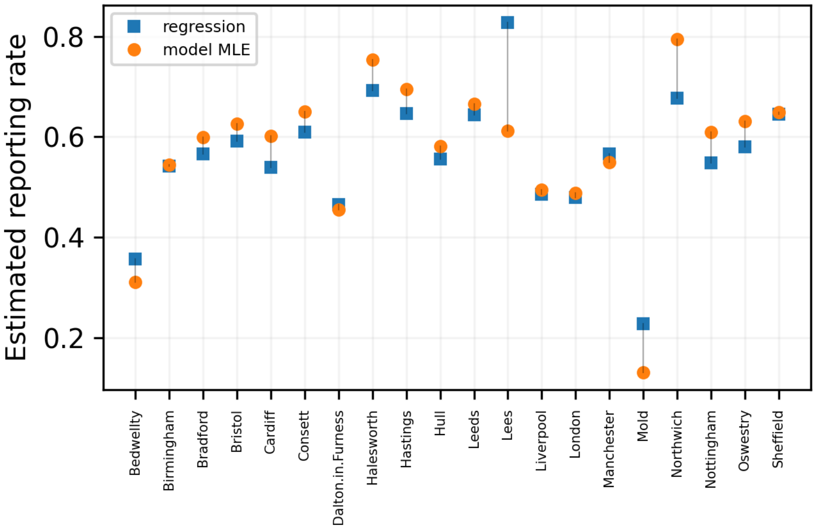


Figure 8: Reporting rate: regression estimate (cases/births slope) versus model MLE.

Predicted vs observed critical community size

Questions

- What does it mean that parameter estimates from the fitting disagree with estimates from other data?

- How can one interpret the correlation between infectious period and city size in the parameter estimates?

- How do we interpret the need for extrademographic stochasticity in this model?

Exercise 5.1. Reformulate the model

- Modify the He et al. (2010) model to remove the cohort effect. Run simulations and compute likelihoods to convince yourself that the resulting codes agree with the original ones for `cohort = 0`.

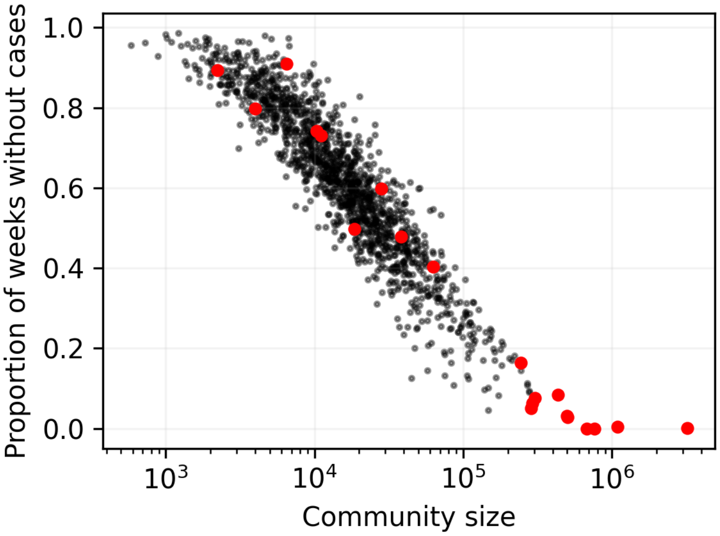


Figure 9: Fraction of weeks with zero cases versus population size. The critical community size is approximately 300,000.

- Now modify the transmission seasonality to use a sinusoidal form. How many parameters must you use? Fixing the other parameters at their MLE values, compute and visualize a profile likelihood over these parameters.

Exercise 5.2. Extrademographic stochasticity

Set the extrademographic stochasticity parameter $\sigma_{SE} = 0$, set $\alpha = 1$ (already the case in `model_001b`), and fix ρ and ι at their MLE values, then maximize the likelihood over the remaining parameters.

- How do your results compare with those at the MLE? Compare likelihoods but also use simulations to diagnose differences between the models.

References

Bhadra, A., Ionides, E. L., Laneri, K., Pascual, M., Bouma, M., and Dhiman, R. (2011). Malaria in northwest India: data analysis via partially observed stochastic differential equation models driven by Lévy noise. *J Am Stat Assoc*, 106(494):440–451.

He, D., Ionides, E. L., and King, A. A. (2010). Plug-and-play inference for disease dynamics: measles in large and small populations as a case study. *J R Soc Interface*, 7:271–283.

Rohani, P. and King, A. A. (2010). Never mind the length, feel the quality: the impact of long-term epidemiological data sets on theory, application and policy. *Trends Ecol Evol*, 25(10):611–618.