

Lesson 5.
Case study:
Measles in large and small towns

Aaron A. King and Edward L. Ionides

Outline

- 1 Introduction
- 2 Model and implementation
 - Overview
 - Data sets
 - Modeling
 - Model implementation in **pomp**
- 3 Estimation
 - He *et al.* (2010)
 - Simulations
 - Parameter estimation
- 4 Model diagnostics
- 5 Findings
 - Notable findings
 - Problematic results
- 6 Exercises

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 **pomp**.
- 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, 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.

Challenges in inference from disease dynamics II

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

Outline

- 1 Introduction
- 2 Model and implementation
 - Overview
 - Data sets
 - Modeling
 - Model implementation in **pomp**
- 3 Estimation
 - He *et al.* (2010)
 - Simulations
 - Parameter estimation
- 4 Model diagnostics
- 5 Findings
 - Notable findings
 - Problematic results
- 6 Exercises

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.

Measles in England and Wales II

- 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?

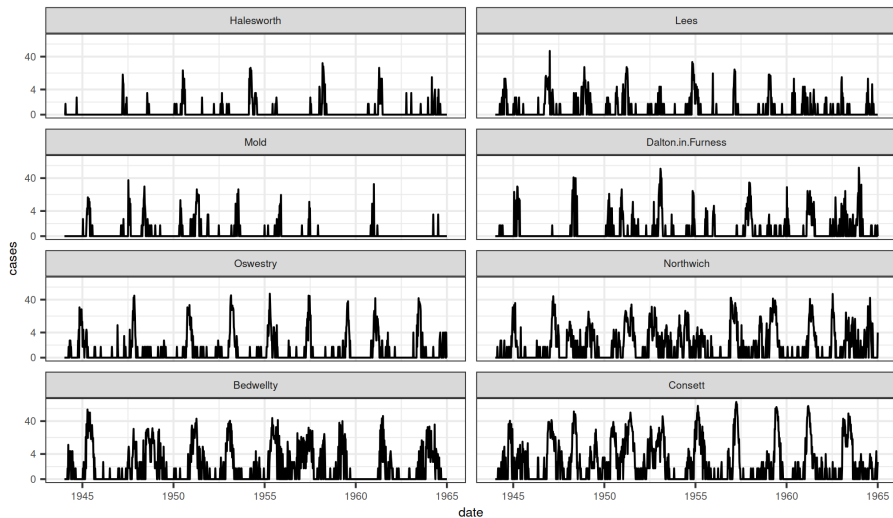
Data sets

- He, Ionides, & King, *J. R. Soc. Interface* (2010)
- Twenty towns, including
 - 10 largest
 - 10 smaller, chosen at random
- Population sizes: 2k–3.4M
- Weekly case reports, 1950–1963
- Annual birth records and population sizes, 1944–1963

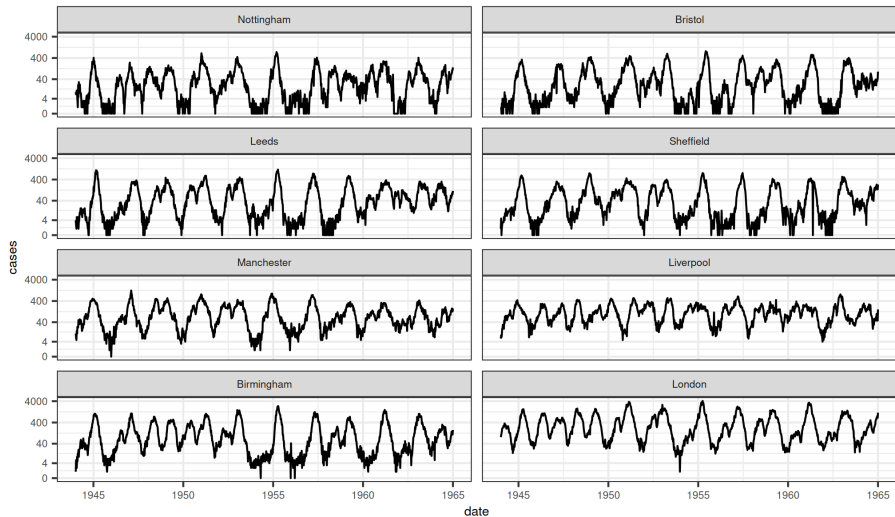
Map of cities in the analysis



City case counts I: smallest 8 cities



City case counts II: largest 8 cities



Continuous-time Markov process model



Continuous-time Markov process model

- Covariates:
 - $B(t)$ = birth rate, from data
 - $N(t)$ = population size, from data
- Entry into susceptible class:

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

- c = cohort effect
 - τ = school-entry delay
 - $\lfloor t \rfloor$ = most recent 1 September before t
- Force of infection:

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

Continuous-time Markov process model II

- 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
- $p = 0.7589$ is the fraction of the year children are in school.
- The factor $(1 - p)/p$ ensures that the average transmission rate is β_0 .
- Overdispersed binomial measurement model:
 $\text{cases}_t \mid \Delta N_{IR} = z_t \sim \text{Normal}(\rho z_t, \rho(1 - \rho) z_t + (\psi \rho z_t)^2)$

Implementation in **pomp**

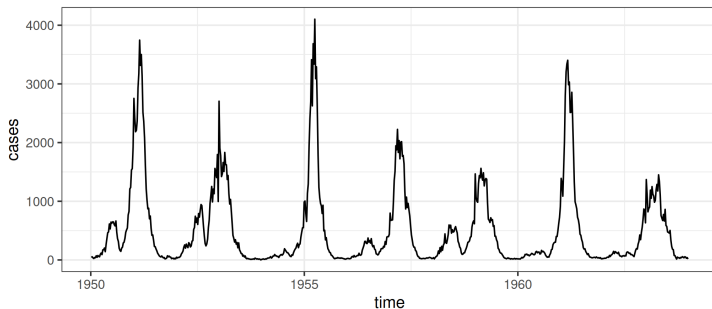
- Codes that implement the model in **pomp** are to be found on [the course website](#).
- In those codes, we first load the needed packages and set the random seed, to allow reproducibility.
- Note that the codes make heavy use of the **ggplot2** plotting package and **tidyverse** methods.
- We also use the convenient R pipe syntax, `|>`.

Data and covariates

- The data are measles reports from 20 cities in England and Wales.
- We also have information on the population sizes and birth-rates in these cities; we'll treat these variables as covariates.
- The codes illustrate the pre-processing of the measles and demography data using London as an example.

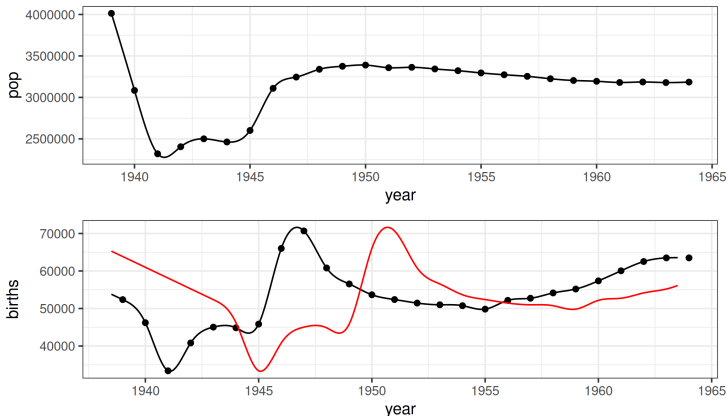
Data and covariates II

Measles case reports from London:



Data and covariates III

We smooth the covariates and delay the entry of newborns into the susceptible pool.



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 (`cohort`) of the cohort enter the susceptible pool all at once.
- ③ Seasonality in the transmission rate: during school terms, the transmission rate is higher than it is during 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

Let's walk through the rprocess C snippet.

```
double beta, br, seas, foi, dw, births;
double rate[6], trans[6];

// cohort effect
if (fabs(t-floor(t)-251.0/365.0) < 0.5*dt)
    br = cohort*birthrate/dt + (1-cohort)*birthrate;
else
    br = (1.0-cohort)*birthrate;
```

Implementation of the process model II

```
// term-time seasonality
t = (t-floor(t))*365.25;
if ((t>=7 && t<=100) ||
    (t>=115 && t<=199) ||
    (t>=252 && t<=300) ||
    (t>=308 && t<=356))
  seas = 1.0+amplitude*0.2411/0.7589;
else
  seas = 1.0-amplitude;
```

Implementation of the process model III

```
// transmission rate
beta = R0*(gamma+mu)*seas;

// expected force of infection
foi = beta*pow(I+iota,alpha)/pop;

// white noise (extrademographic stochasticity)
dw = rgammawn(sigmaSE,dt);
```

Implementation of the process model IV

```
rate[0] = foi*dw/dt; // stochastic force of infection
rate[1] = mu;        // natural S death
rate[2] = sigma;     // rate of ending of latent stage
rate[3] = mu;        // natural E death
rate[4] = gamma;     // recovery
rate[5] = mu;        // natural I death
```


Implementation of the process model V

```
// Poisson births
births = rpois(br*dt);

// transitions between classes
reulermultinom(2, S, &rate[0], dt, &trans[0]);
reulermultinom(2, E, &rate[2], dt, &trans[2]);
reulermultinom(2, I, &rate[4], dt, &trans[4]);
```

Implementation of the process model VI

```
S += births - trans[0] - trans[1];  
E += trans[0] - trans[2] - trans[3];  
I += trans[2] - trans[4] - trans[5];  
R = pop - S - E - I;  
W += (dw - dt)/sigmaSE; // standardized i.i.d. white noise  
C += trans[4];           // true incidence
```

Process model observations

- In the above, C represents the true incidence, i.e., the number of new infections occurring over an interval.
- Since recognized measles infections are quarantined, we argue that most infection occurs before case recognition so that true incidence is a measure of the number of individuals progressing from the I to the R compartment in a given interval.

State initializations

We complete the process model definition by specifying the distribution of initial unobserved states. The following codes assume that the fraction of the population in each of the four compartments is known.

```
double m = pop/(S_0+E_0+I_0+R_0);  
S = nearbyint(m*S_0);  
E = nearbyint(m*E_0);  
I = nearbyint(m*I_0);  
R = nearbyint(m*R_0);  
W = 0;  
C = 0;
```

Measurement model

- We'll model both under-reporting and measurement error.
- We want $\mathbb{E}[\text{cases}|C] = \rho C$, where C is the true incidence and $0 < \rho < 1$ is the reporting efficiency.
- We'll also assume that $\text{Var}[\text{cases}|C] = \rho(1 - \rho)C + (\psi \rho C)^2$, where ψ quantifies overdispersion.
- Note that when $\psi = 0$, the variance-mean relation is that of the binomial distribution. To be specific, we'll choose $\text{cases} | C \sim f(\cdot | \rho, \psi, C)$, where

$$f(c | \rho, \psi, C) = \Phi(c + \frac{1}{2}, \rho C, \rho(1 - \rho)C + (\psi \rho C)^2) - \Phi(c - \frac{1}{2}, \rho C, \rho(1 - \rho)C + (\psi \rho C)^2).$$

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

Measurement model II

The following computes $\mathbb{P}[\text{cases}|C]$.

```
double m = rho*C;
double v = m*(1.0-rho+psi*psi*m);
double tol = 0.0;
if (cases > 0.0) {
    lik = pnorm(cases+0.5,m,sqrt(v)+tol,1,0)
          - pnorm(cases-0.5,m,sqrt(v)+tol,1,0) + tol;
} else {
    lik = pnorm(cases+0.5,m,sqrt(v)+tol,1,0) + tol;
}
if (give_log) lik = log(lik);
```

Measurement model III

The following codes simulate cases| C .

```
double m = rho*C;  
double v = m*(1.0-rho+psi*psi*m);  
double tol = 0.0;  
cases = rnorm(m,sqrt(v)+tol);  
if (cases > 0.0) {  
    cases = nearbyint(cases);  
} else {  
    cases = 0.0;  
}
```

Constructing the pomp object

```
dat |>
  pomp(t0=with(dat,2*time[1]-time[2]),
       time="time",
       rprocess=euler(rproc,delta.t=1/365.25),
       rinit=rinit,
       dmeasure=dmeas,
       rmeasure=rmeas,
       covar=covariate_table(covar,times="time"),
       accumvars=c("C","W"),
       statenames=c("S","E","I","R","C","W"),
       paramnames=c("R0","mu","sigma","gamma","alpha","iota",
                    "rho","sigmaSE","psi","cohort","amplitude",
                    "S_0","E_0","I_0","R_0")
  ) -> m1
```


Outline

- 1 Introduction
- 2 Model and implementation
 - Overview
 - Data sets
 - Modeling
 - Model implementation in **pomp**
- 3 Estimation
 - He *et al.* (2010)
 - Simulations
 - Parameter estimation
- 4 Model diagnostics
- 5 Findings
 - Notable findings
 - Problematic results
- 6 Exercises

Estimates from He *et al.* (2010)

He *et al.* (2010) estimated the parameters of this model. The full set of estimates is included in the R code accompanying this document, where they are read into a data frame called `mles`.

We verify that we get the same likelihood as He *et al.* (2010).

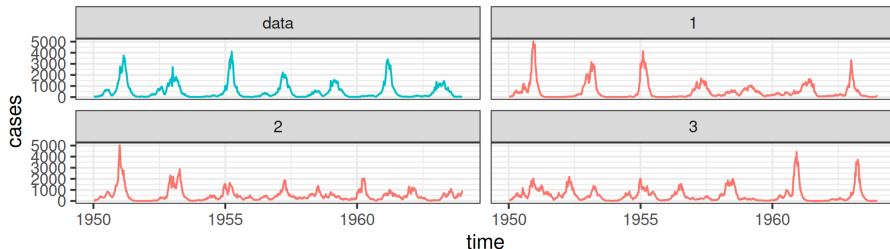
```
library(doFuture)
plan(multicore)
foreach(i=1:4, .combine=c,
        .options.future=list(seed=998468235L)
) %dopar% {
  pfilter(m1,Np=10000,params=theta)
} -> pfs
```

```
logmeanexp(logLik(pfs),se=TRUE)
```

est	se
-3801.6339190	0.2186696

Simulations at the MLE

```
m1 |>  
  simulate(params=theta,nsim=3,format="d",include.data=TRUE) |>  
  ggplot(aes(x=time,y=cases,group=.id,color=(.id=="data")))+  
  guides(color="none")+  
  geom_line()+facet_wrap(~.id,ncol=2)
```



Parameter transformations

- The parameters are constrained to be positive, and some of them are constrained to lie between 0 and 1.
- We can turn the likelihood maximization problem into an unconstrained maximization problem by transforming the parameters.
- Specifically, to enforce positivity, we log transform, to constrain parameters to $(0, 1)$, we logit transform, and to confine parameters to the unit simplex, we use the log-barycentric transformation.

```
pt <- parameter_trans(  
  log=c("sigma", "gamma", "sigmaSE", "psi", "R0"),  
  logit=c("cohort", "amplitude"),  
  barycentric=c("S_0", "E_0", "I_0", "R_0")  
)
```

Outline

- 1 Introduction
- 2 Model and implementation
 - Overview
 - Data sets
 - Modeling
 - Model implementation in **pomp**
- 3 Estimation
 - He *et al.* (2010)
 - Simulations
 - Parameter estimation
- 4 Model diagnostics**
- 5 Findings
 - Notable findings
 - Problematic results
- 6 Exercises

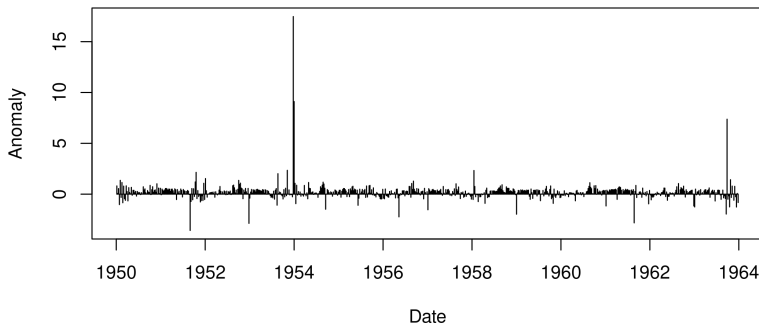
ARMA benchmark

- Linear, Gaussian auto-regressive moving-average (ARMA) models provide a flexible non-mechanistic benchmark comparison.
- We fit an ARMA(2,2) model to $\log(y_n^* + 1)$ and correct the likelihood back to the untransformed data (see Lesson 6 for more details). This has $p = 5$ parameters and a log-likelihood of $\ell = -3958.3$ for London.
- The SEIR model has $\ell = -3804.9$ with $p = 12$.
- Minimizing the 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; ratios of log-likelihoods are not.

Log-likelihood anomalies

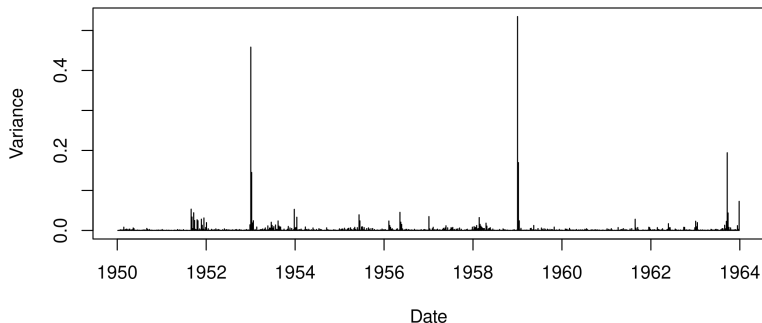
- The benchmark and model log-likelihoods can both be decomposed as a sum of conditional log-likelihoods, $\ell(\theta) = \sum_{n=1}^N \ell_n(\theta)$ where $\ell_n(\theta) = \log f_{Y_n|Y_{1:n-1}}(y_n^*|y_{1:n-1}^*; \theta)$.
- The **anomaly** for the model at time t_n is the difference between the 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.
- Conditional log-likelihoods are not scale-invariant, but anomalies are.

Anomaly plot for London



- Here, the main anomalies are positive: the log-scale ARMA model is not good at explaining very low counts.
- Negative anomalies result if cases fail to drop when susceptibles should be depleted. These are not big anomalies.
- One major outlier was previously identified and “cleaned.”

Particle filter variance for London



- The variance of the `pfilter` log-likelihood estimate is approximately the sum of the variances of the conditional log-likelihoods.
- Observations with high variance are numerically problematic.
- Here, none are larger than 1. We have 2000 particles here, which seems just enough, though more may be preferable.

Outline

- 1 Introduction
- 2 Model and implementation
 - Overview
 - Data sets
 - Modeling
 - Model implementation in **pomp**
- 3 Estimation
 - He *et al.* (2010)
 - Simulations
 - Parameter estimation
- 4 Model diagnostics
- 5 Findings
 - Notable findings
 - Problematic results
- 6 Exercises

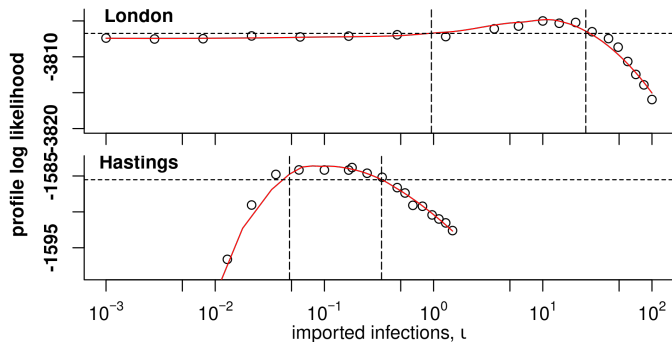
Results from He *et al.* (2010)

The [linked document](#) shows how a likelihood profile can be constructed using IF2. The fitting procedure used is as follows:

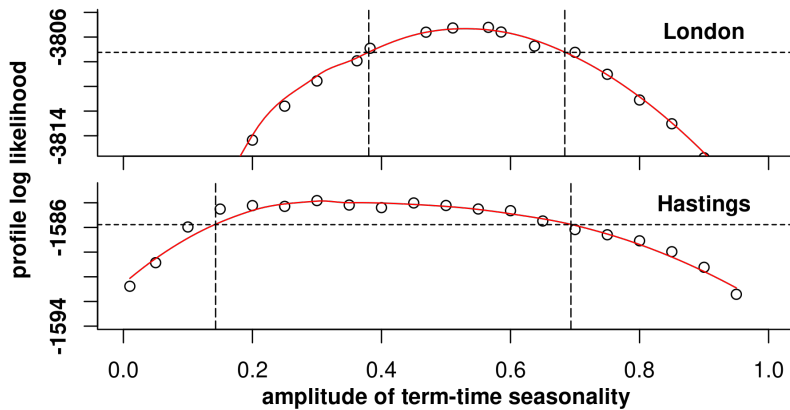
- A large number of searches were started at points across the parameter space.
- Iterated filtering was used to maximize the likelihood.
- We obtained point estimates of all parameters for 20 cities.
- We constructed profile likelihoods to quantify uncertainty in London and Hastings.

Imported infections

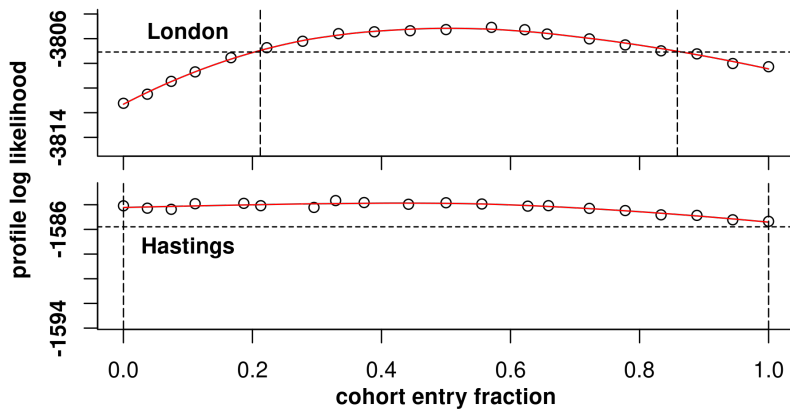
$$\text{force of infection} = \mu_{SE} = \frac{\beta(t)}{N(t)} (I + \iota)^\alpha \zeta(t)$$



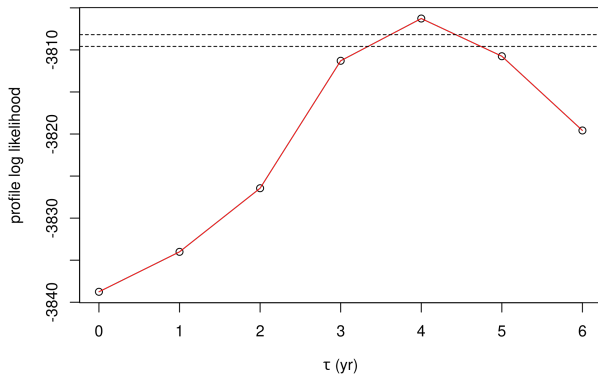
Seasonality



Cohort effect

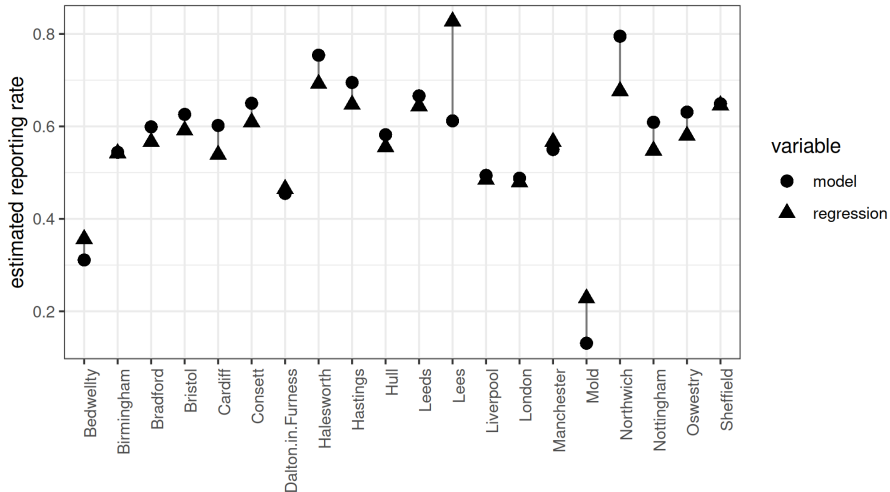


Birth delay

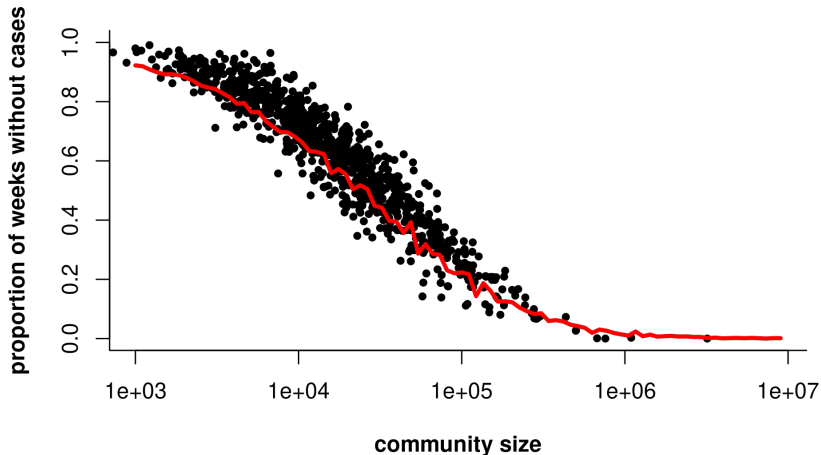


Profile likelihood for birth-cohort delay, showing 95% and 99% critical values of the log likelihood.

Reporting rate

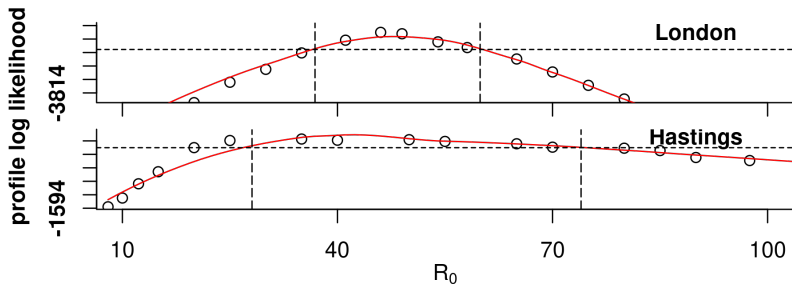


Predicted vs observed critical community size



R_0 estimates inconsistent with literature

- Recall that R_0 : a measure of how communicable an infection is.
- Existing estimates of R_0 (c. 15–20) come from two sources: serology surveys, and models fit to data using feature-based methods.



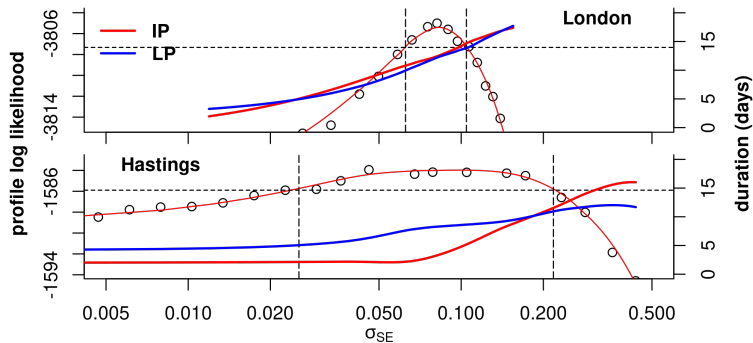
Parameter estimates

	N_{1950}	R_0	IP	LP	α	a	ι	ψ	ρ	σ_{SE}
Halesworth	2200	33.00	2.30	7.90	0.95	0.38	0.0091	0.64	0.75	0.075
Lees	4200	30.00	2.10	8.50	0.97	0.15	0.0310	0.68	0.61	0.080
Mold	6400	21.00	1.80	5.90	1.00	0.27	0.0140	2.90	0.13	0.054
Dalton in Furness	11000	28.00	2.00	5.50	0.99	0.20	0.0390	0.82	0.46	0.078
Oswestry	11000	53.00	2.70	10.00	1.00	0.34	0.0300	0.48	0.63	0.070
Northwich	18000	30.00	3.00	8.50	0.95	0.42	0.0600	0.40	0.80	0.086
Bedwellty	29000	25.00	3.00	6.80	0.94	0.16	0.0400	0.95	0.31	0.061
Consett	39000	36.00	2.70	9.10	1.00	0.20	0.0730	0.41	0.65	0.071
Hastings	66000	34.00	5.40	7.00	1.00	0.30	0.1900	0.40	0.70	0.096
Cardiff	240000	34.00	3.10	9.90	1.00	0.22	0.1400	0.27	0.60	0.054
Bradford	290000	32.00	3.40	8.50	0.99	0.24	0.2400	0.19	0.60	0.045
Hull	300000	39.00	5.50	9.20	0.97	0.22	0.1400	0.26	0.58	0.064
Nottingham	310000	23.00	3.70	5.70	0.98	0.16	0.1700	0.26	0.61	0.038
Bristol	440000	27.00	4.90	6.20	1.00	0.20	0.4400	0.20	0.63	0.039
Leeds	510000	48.00	11.00	9.50	1.00	0.27	1.2000	0.17	0.67	0.078
Sheffield	520000	33.00	6.40	7.20	1.00	0.31	0.8500	0.18	0.65	0.043
Manchester	700000	33.00	6.90	11.00	0.96	0.29	0.5900	0.16	0.55	0.055
Liverpool	800000	48.00	9.80	7.90	0.98	0.30	0.2600	0.14	0.49	0.053
Birmingham	1100000	43.00	12.00	8.50	1.00	0.43	0.3400	0.18	0.54	0.061
London	3400000	57.00	13.00	13.00	0.98	0.55	2.9000	0.12	0.49	0.088
r	1	0.46	0.95	0.32	0.11	0.30	0.9300	-0.93	-0.20	-0.330

$r = \text{cor}_S(\cdot, N_{1950})$ (Spearman rank correlation)

Extrademographic stochasticity

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

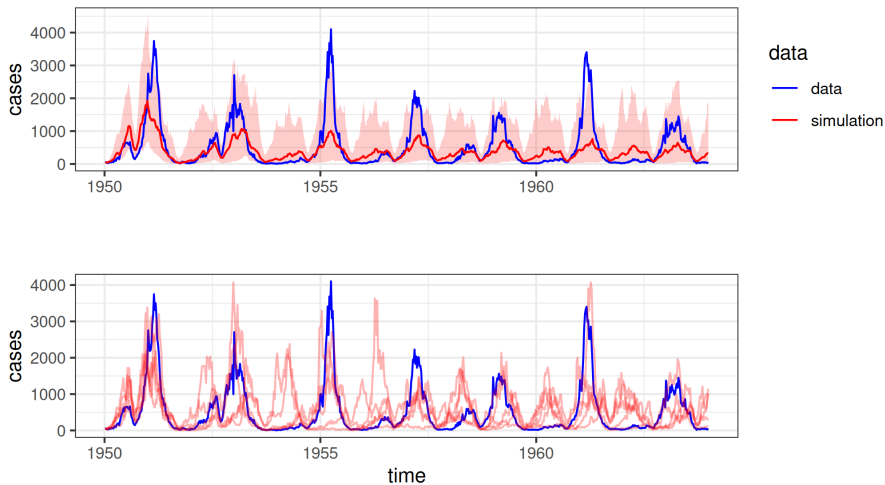


Questions

- 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?

Simulations at the MLE

London



Outline

- 1 Introduction
- 2 Model and implementation
 - Overview
 - Data sets
 - Modeling
 - Model implementation in **pomp**
- 3 Estimation
 - He *et al.* (2010)
 - Simulations
 - Parameter estimation
- 4 Model diagnostics
- 5 Findings
 - Notable findings
 - Problematic results
- 6 Exercises

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'.
- 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 $\alpha = 1$, 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 EL, Laneri K, Pascual M, Bouma M, 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. doi: [10.1198/jasa.2011.ap10323](https://doi.org/10.1198/jasa.2011.ap10323).
- He D, Ionides EL, King AA (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. doi: [10.1098/rsif.2009.0151](https://doi.org/10.1098/rsif.2009.0151).
- Rohani P, King AA (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. doi: [10.1016/j.tree.2010.07.010](https://doi.org/10.1016/j.tree.2010.07.010).

License, acknowledgments, and links

- This lesson is prepared for the [Simulation-based Inference for Epidemiological Dynamics](#) module at the Summer Institute in Statistics and Modeling in Infectious Diseases, [SISMID](#).
- The materials build on [previous versions of this course and related courses](#).
- Licensed under the [Creative Commons Attribution-NonCommercial license](#). Please share and remix non-commercially, mentioning its origin. 
- Produced with R version 4.3.2 and **pomp** version 5.6.
- Compiled on February 19, 2024.

[Back to Lesson](#)

[R codes for this lesson](#)