Lesson 2:
Simulation of stochastic dynamic models

Aaron A. King, Edward L. Ionides, and Kidus Asfaw

Outline

1. Compartment models
   - Example: the SIR model
   - Notation
   - A deterministic interpretation
   - A stochastic interpretation
2. Euler's method
   - Numerical solution of deterministic dynamics
   - Numerical solution of stochastic dynamics
3. Compartment models in **pomp**
   - A basic pomp model for measles
   - C snippets
   - Choosing parameters
4. Exercises

Objectives

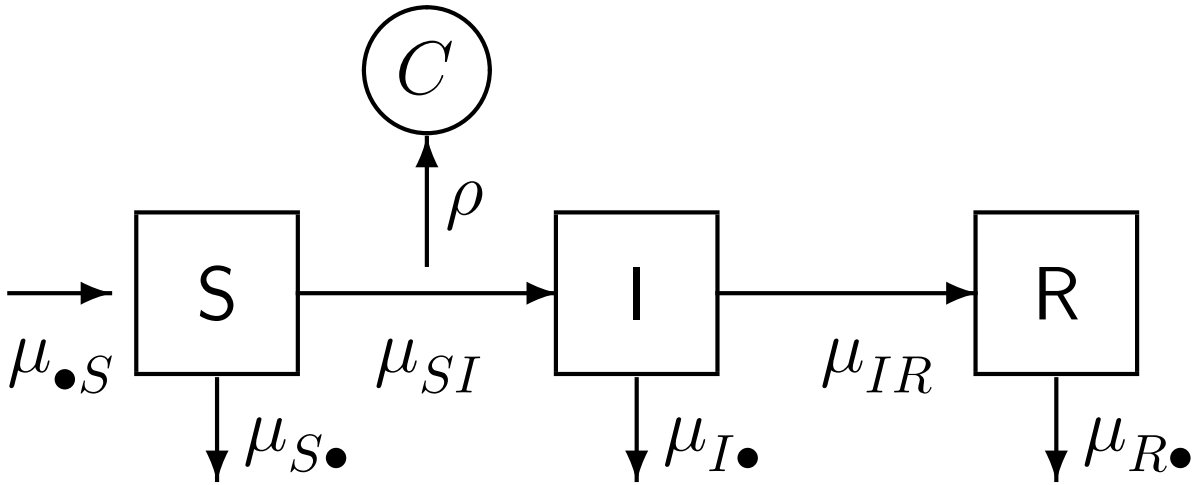
This tutorial develops some classes of dynamic models relevant to biological systems, especially for epidemiology.

1. Dynamic systems can often be represented in terms of **flows** between **compartments**.
2. We develop the concept of a **compartmental model** for which we specify **rates** for the flows between compartments.
3. We show how deterministic and stochastic versions of a compartmental model are derived and related.
4. We introduce Euler's method to simulate from dynamic models.
5. We specify deterministic and stochastic compartmental models in **pomp** using Euler method simulation.

A basic compartment model: The SIR model

- We develop deterministic and stochastic representations of a susceptible-infected-recovered (SIR) system, a fundamental class of models for disease transmission dynamics.
- We set up notation applicable to general compartment models (Bretó *et al.*, 2009).

A basic compartment model: The SIR model II



S : susceptible
I : infected and infectious
R : recovered and/or removed
C : reported cases

A basic compartment model: The SIR model III

- We suppose that each arrow has an associated rate, so here there is a rate $\mu_{SI}(t)$ at which individuals in S transition to I, and μ_{IR} at which individuals in I transition to R.
- To account for demography (births/deaths/migration) we allow the possibility of a source and sink compartment, which is not usually represented on the flow diagram. We write $\mu_{\bullet S}$ for a rate of births into S, and denote mortality rates by $\mu_{S\bullet}$, $\mu_{I\bullet}$, $\mu_{R\bullet}$.

A basic compartment model: The SIR model IV

- The rates may be either constant or varying. In particular, for a simple SIR model, the recovery rate μ_{IR} is a constant but the infection rate has the time-varying form

$$\mu_{SI}(t) = \beta\, I(t),$$

with β being the **contact rate**. For the simplest SIR model, ignoring demography, we set

$$\mu_{\bullet S} = \mu_{S\bullet} = \mu_{I\bullet} = \mu_{R\bullet} = 0.$$

General notation for compartment models

- To develop a systemtic notation, it turns out to be convenient to keep track of the flows between compartments as well as the number of individuals in each compartment. Let

$$N_{SI}(t)$$

count the number of individuals who have transitioned from S to I by time t. We say that $N_{SI}(t)$ is a **counting process**. A similarly constructed process

$$N_{IR}(t)$$

General notation for compartment models II

counts individuals transitioning from I to R. To include demography, we could keep track of birth and death events by the counting processes $N_{\bullet S}(t)$, $N_{S\bullet}(t)$, $N_{I\bullet}(t)$, $N_{R\bullet}(t)$.

- For discrete population compartment models, the flow counting processes are non-decreasing and integer valued.
- For continuous population compartment models, the flow counting processes are non-decreasing and real valued.

Compartment processes from counting processes

- The numbers of people in each compartment can be computed via these counting processes. Ignoring demography, we have:

$$\begin{aligned} S(t) &= S(0) - N_{SI}(t) \\ I(t) &= I(0) + N_{SI}(t) \quad -N_{IR}(t) \\ R(t) &= R(0) \qquad\qquad\quad +N_{IR}(t) \end{aligned}$$

- These equations represent *conservation of individuals* or *what goes in must come out*.

Ordinary differential equation interpretation

Together with initial conditions specifying $S(0)$, $I(0)$ and $R(0)$, we just need to write down ordinary differential equations (ODEs) for the flow counting processes. These are:

$$\frac{\mathrm{d}N_{SI}}{\mathrm{d}t} = \mu_{SI}(t)\, S(t)$$
$$\frac{\mathrm{d}N_{IR}}{\mathrm{d}t} = \mu_{IR}\, I(t)$$

Continuous-time Markov chain interpretation

- Continuous-time Markov chains are the basic tool for building discrete population epidemic models.
- The Markov property lets us specify a model by the transition probabilities on small intervals (together with the initial conditions). For the SIR model, we have

$$\begin{aligned}
\mathbb{P}\big[N_{SI}(t+\delta) = N_{SI}(t) + 1\big] &= \mu_{SI}(t)\, S(t)\delta + o(\delta) \\
\mathbb{P}\big[N_{SI}(t+\delta) = N_{SI}(t)\big] &= 1 - \mu_{SI}(t)\, S(t)\delta + o(\delta) \\
\mathbb{P}\big[N_{IR}(t+\delta) = N_{IR}(t) + 1\big] &= \mu_{IR}\, I(t)\delta + o(\delta) \\
\mathbb{P}\big[N_{IR}(t+\delta) = N_{IR}(t)\big] &= 1 - \mu_{IR}(t)\, I(t)\delta + o(\delta)
\end{aligned}$$

- Here, we are using **little o notation** We write $h(\delta) = o(\delta)$ to mean $\lim_{\delta \to 0} \frac{h(\delta)}{\delta} = 0$.

Exercise 2.1

What is the link between little o notation and the derivative? Explain why

$$f(x+\delta) = f(x) + \delta g(x) + o(\delta)$$

is the same statement as

$$\frac{df}{dx} = g(x).$$

What considerations might help you choose which of these notations to use?

Worked solution to the Exercise

Simple counting processes

- A **simple counting process** is one which cannot count more than one event at a time.
- Technically, the SIR Markov chain model we have written is simple.
- One may want to model the extra randomness resulting from multiple simultaneous events: someone sneezing in a bus; large gatherings at football matches; etc. This extra randomness may even be critical to match the variability in data.
- Later in the course, we may see situations where this extra randomness plays an important role. Setting up the model using counting processes, as we have done here, turns out to be useful for this.

Euler's method for ordinary differential equations

- Euler (1707–1783) wanted a numeric solution of an ordinary differential equation (ODE) $dx/dt = h(x)$ with an initial condition $x(0)$.
- He supposed this ODE has some true solution $x(t)$ which could not be worked out analytically. He wanted to approximate $x(t)$ numerically.
- He initialized the numerical solution at the known starting value,

$$\tilde{x}(0) = x(0).$$

Euler's method for ordinary differential equations II

- For $k = 1, 2, \ldots,$ he supposed that the gradient dx/dt is approximately constant over the small time interval $k\delta \leq t \leq (k+1)\delta$. Therefore, he defined

$$\tilde{x}\big((k+1)\delta\big) = \tilde{x}(k\delta) + \delta\, h\big(\tilde{x}(k\delta)\big).$$

- This only defines $\tilde{x}(t)$ when t is a multiple of δ, but suppose $\tilde{x}(t)$ is constant between these discrete times.
- We now have a numerical scheme, stepping forwards in time increments of size δ, that can be readily evaluated by computer.

Euler's method versus other numerical methods

- Mathematical analysis of Euler's method says that, as long as the function $h(x)$ is not too exotic, then $x(t)$ is well approximated by $\tilde{x}(t)$ when the discretization time-step, δ, is sufficiently small.
- Euler's method is not the only numerical scheme to solve ODEs. More advanced schemes have better convergence properties, meaning that the numerical approximation is closer to $x(t)$. However, there are 3 reasons we choose to lean heavily on Euler's method:
  1. Euler's method is the simplest (cf. the KISS principle).
  2. Euler's method extends naturally to stochastic models, both continuous-time Markov chains models and stochastic differential equation (SDE) models.
  3. Close approximation of the numerical solutions to a continuous-time model is less important than it may at first appear, a topic to be discussed.

Continuous-time models and discretized approximations

- In some physical and engineering situations, a system follows an ODE model closely. For example, Newton's laws provide a very good approximation to the motions of celestial bodies.
- In many biological situations, ODE models only become close mathematical approximations to reality at reasonably large scale. On small temporal scales, models cannot usually capture the full scope of biological variation and biological complexity.
- If we are going to expect substantial error in using $x(t)$ to model a biological system, maybe the numerical solution $\tilde{x}(t)$ represents the system being modeled as well as $x(t)$ does.

Continuous-time models and discretized approximations II

- If our model fitting, model investigation, and final conclusions are all based on our numerical solution $\tilde{x}(t)$ (i.e., we are sticking entirely to simulation-based methods) then we are most immediately concerned with how well $\tilde{x}(t)$ describes the system of interest. $\tilde{x}(t)$ becomes more important than the original model, $x(t)$.

Numerical solutions as scientific models

- It is important that a scientist fully describe the numerical model $\tilde{x}(t)$. Arguably, the main purpose of the original model $x(t)$ is to give a succinct description of how $\tilde{x}(t)$ was constructed.
- All numerical methods are, ultimately, discretizations. Epidemiologically, setting δ to be a day, or an hour, can be quite different from setting δ to be two weeks or a month. For continuous-time modeling, we still require that δ is small compared to the timescale of the process being modeled, so the choice of δ should not play an explicit role in the interpretation of the model.
- Putting more emphasis on the scientific role of the numerical solution itself reminds you that the numerical solution has to do more than approximate a target model in some asymptotic sense: the numerical solution should be a sensible model in its own right.

Euler's method for a discrete SIR model

- Recall the simple continuous-time Markov chain interpretation of the SIR model without demography:

$$\begin{aligned}\mathbb{P}\big[N_{SI}(t+\delta) = N_{SI}(t) + 1\big] &= \mu_{SI}(t)\, S(t)\delta + o(\delta),\\ \mathbb{P}\big[N_{IR}(t+\delta) = N_{IR}(t) + 1\big] &= \mu_{IR}\, I(t)\delta + o(\delta).\end{aligned}$$

- We want a numerical solution with state variables $\tilde{S}(k\delta)$, $\tilde{I}(k\delta)$, $\tilde{R}(k\delta)$.

Euler's method for a discrete SIR model II

- The counting processes for the flows between compartments are $\tilde{N}_{SI}(t)$ and $\tilde{N}_{IR}(t)$. The counting processes are related to the numbers of individuals in the compartments by the same flow equations we had before:

$$\begin{aligned} \tilde{S}(k\delta) &= S(0) - \tilde{N}_{SI}(k\delta) \\ \tilde{I}(k\delta) &= I(0) + \tilde{N}_{SI}(k\delta) - \tilde{N}_{IR}(k\delta) \\ \tilde{R}(k\delta) &= R(0) + \tilde{N}_{IR}(k\delta) \end{aligned}$$

- We focus on a numerical solution to $N_{SI}(t)$, since the same methods can also be applied to $N_{IR}(t)$.

Three different stochastic Euler solutions

(1) A Poisson approximation.

$$\tilde{N}_{SI}(t+\delta) = \tilde{N}_{SI}(t) + \text{Poisson}\big[\mu_{SI}\big(\tilde{I}(t)\big)\,\tilde{S}(t)\,\delta\big],$$

where $\text{Poisson}(\mu)$ is a Poisson random variable with mean μ and

$$\mu_{SI}\big(\tilde{I}(t)\big) = \beta\,\tilde{I}(t).$$

(2) A binomial approximation,

$$\tilde{N}_{SI}(t+\delta) = \tilde{N}_{SI}(t) + \text{Binomial}\big[\tilde{S}(t), \mu_{SI}\big(\tilde{I}(t)\big)\,\delta\big],$$

where $\text{Binomial}(n,p)$ is a binomial random variable with mean np and variance $np(1-p)$. Here, $p = \mu_{SI}\big(\tilde{I}(t)\big)\,\delta$.

(3) A binomial approximation with exponential transition probabilities.

$$\tilde{N}_{SI}(t+\delta) = \tilde{N}_{SI}(t) + \text{Binomial}\big[\tilde{S}(t), 1 - \exp\big\{-\mu_{SI}\big(\tilde{I}(t)\big)\delta\big\}\big].$$

Analytically, it is usually easiest to reason using (1) or (2). Practically, it is usually preferable to work with (3).

Compartment models as stochastic differential equations

- The Euler method extends naturally to stochastic differential equations (SDEs).
- A natural way to add stochastic variation to an ODE $dx/dt = h(x)$ is

$$dX/dt = h(X) + \sigma\, dB/dt$$

where $\{B(t)\}$ is Brownian motion and so dB/dt is Brownian noise.

- An Euler approximation $\tilde{X}(t)$ is

$$\tilde{X}\big((k+1)\delta\big) = \tilde{X}(k\delta) + \delta\, h\big(\tilde{X}(k\delta)\big) + \sigma\sqrt{\delta}\, Z_k$$

where $Z_1, Z_2, \ldots$ is a sequence of independent standard normal random variables, i.e., $Z_k \sim N[0,1]$.

Compartment models as stochastic differential equations II

- Although SDEs are often considered an advanced topic in probability, the Euler approximation doesn't demand much more than familiarity with the normal distribution.

Exercise 2.2. Euler's method vs Gillspie's algorithm

A widely used, exact simulation method for continuous time Markov chains is Gillspie's algorithm. We do not put much emphasis on Gillespie's algorithm here. Why? When would you prefer an implementation of Gillespie's algorithm to an Euler solution?

Worked solution to the Exercise

Numerically, Gillespie's algorithm is often approximated using so-called tau-leaping methods. These are closely related to Euler's approach. In this context, the Euler method has sometimes been called a tau-leaping.

The Consett measles outbreak

As an example that we can probe in some depth, let's look at outbreak of measles that occurred in the small town of Consett in England in 1948. The town had population of 38820, with 737 births over the course of the year.

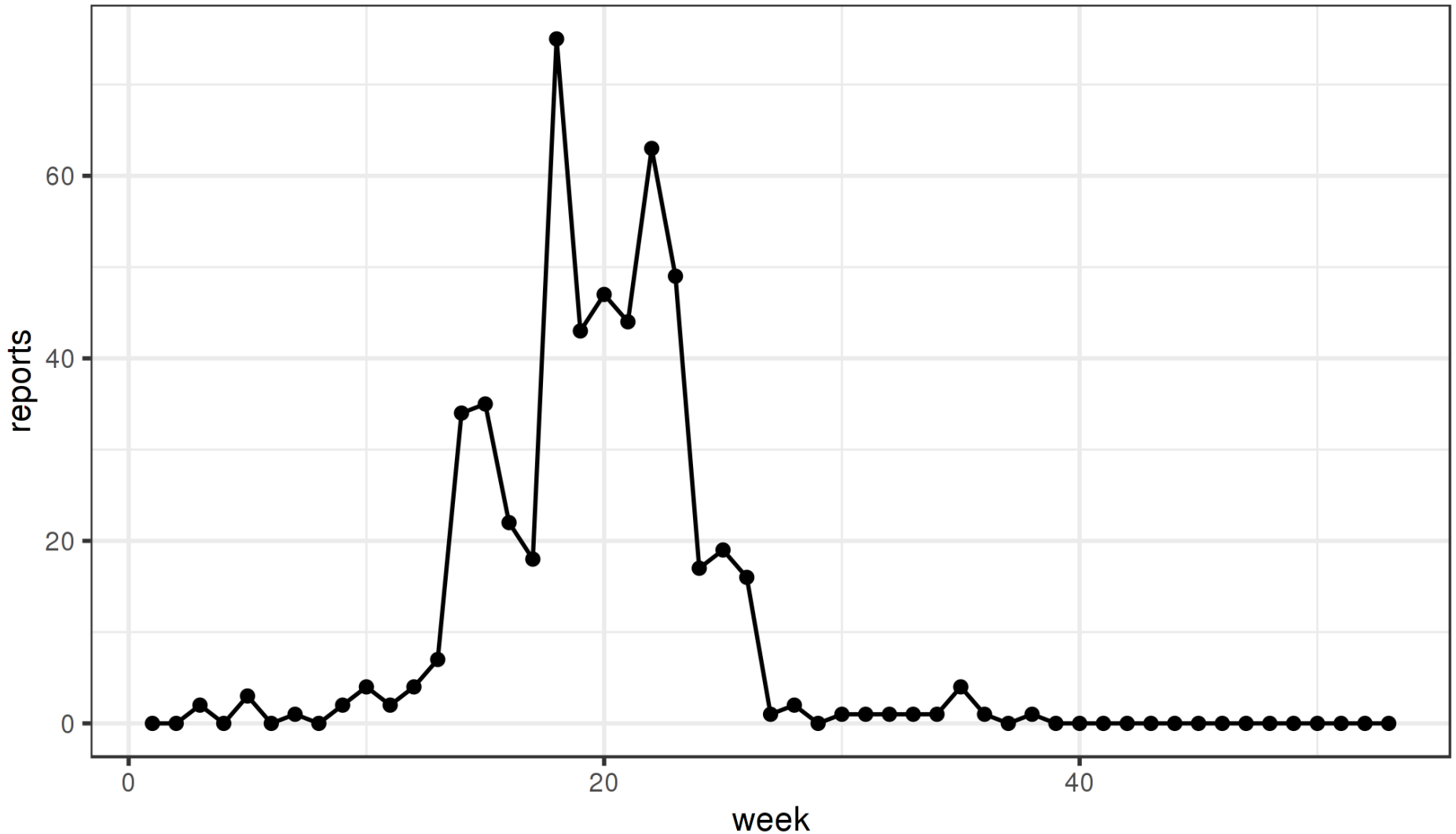
The Consett measles outbreak II

We download the data and examine them:

```
library(tidyverse)
read_csv(paste0("https://kingaa.github.io/sbied/stochsim/",
                "Measles_Consett_1948.csv")) %>%
  select(week,reports=cases) -> meas
meas %>% as.data.frame() %>% head()
```

```
 week reports
    1       0
    2       0
    3       2
    4       0
    5       3
    6       0
```

The Consett measles outbreak III



A simple POMP model for measles

- These are incidence data: The `reports` variable counts the number of reports of new measles cases each week.
- Let us model the outbreak using the simple SIR model.
- Our tasks will be, first, to estimate the parameters of the SIR and, second, to decide whether or not the SIR model is an adequate description of these data.
- The rate at which individuals move from S to I is the **force of infection**, $\mu_{SI} = \beta\, I/N$, while that at which individuals move into the R class is μ_{IR}.

Framing the SIR as a POMP model

- The unobserved state variables, in this case, are the numbers of individuals, $S(t)$, $I(t)$, $R(t)$ in the S, I, and R compartments, respectively.
- It's reasonable in this case to view the population size $N = S(t) + I(t) + R(t)$, as fixed at the known population size of 38,000.
- The numbers that actually move from one compartment to another over any particular time interval are modeled as stochastic processes.
- In this case, we'll assume that the stochasticity is purely demographic, i.e., that each individual in a compartment at any given time faces the same risk of exiting the compartment.
- **Demographic stochasticity** is the unavoidable randomness that arises from chance events occuring in a discrete and finite population.

Implementing the SIR model in **pomp**

- To implement the model in **pomp**, the first thing we need is a stochastic simulator for the unobserved state process.
- We follow method 3 above, modeling the number, ΔN_{SI}, moving from S to I over interval Δt as

$$\Delta N_{SI} \sim \text{Binomial}\left(S, 1 - e^{-\beta\, I/N \Delta t}\right),$$

and the number moving from I to R as

$$\Delta N_{IR} \sim \text{Binomial}\left(I, 1 - e^{-\mu_{IR} \Delta t}\right).$$

Implementing the SIR model in **pomp** II

```
sir_step <- function (S, I, R, N, Beta, mu_IR, delta.t, ...) {
  dN_SI <- rbinom(n=1,size=S,prob=1-exp(-Beta*I/N*delta.t))
  dN_IR <- rbinom(n=1,size=I,prob=1-exp(-mu_IR*delta.t))
  S <- S - dN_SI
  I <- I + dN_SI - dN_IR
  R <- R + dN_IR
  c(S = S, I = I, R = R)
}
```

- At day zero, we'll assume that $I = 1$ but we don't know how many people are susceptible, so we'll treat this fraction, η, as a parameter to be estimated.

```
sir_rinit <- function(N, eta, ...) {
  c(S = round(N*eta), I = 1, R = round(N*(1-eta)))
}
```

Implementing the SIR model in **pomp** III

- We fold these basic model components, with the data, into a `pomp` object thus:

```
library(pomp)
meas %>%
  pomp(times="week",t0=0,
       rprocess=euler(sir_step,delta.t=1/7),
       rinit=sir_rinit
       ) -> measSIR
```

- Now assume the case reports result from a process by which new infections are diagnosed and reported with probability ρ, which we can think of as the probability that a child's parents take the child to the doctor, who recognizes measles and reports it to the authorities.

Implementing the SIR model in **pomp** IV

- Measles symptoms tend to be quite recognizable, and children with measles tend to be confined to bed. Therefore diagnosed cases have, presumably, a much lower transmission rate. Accordingly, let's treat each week's `reports` as being related to the number of individuals who have moved from I to R over the course of that week.
- We need a variable to track these daily counts. We modify our rprocess function above, adding a variable H to tally the true incidence.

Implementing the SIR model in **pomp** V

```
sir_step <- function (S, I, R, H, N, Beta, mu_IR, delta.t, ...)
  dN_SI <- rbinom(n=1,size=S,prob=1-exp(-Beta*I/N*delta.t))
  dN_IR <- rbinom(n=1,size=I,prob=1-exp(-mu_IR*delta.t))
  S <- S - dN_SI
  I <- I + dN_SI - dN_IR
  R <- R + dN_IR
  H <- H + dN_IR;
  c(S = S, I = I, R = R, H = H)
}

sir_rinit <- function (N, eta, ...) {
  c(S = round(N*eta), I = 1, R = round(N*(1-eta)), H = 0)
}
```

Implementing the SIR model in **pomp** VI

- In **pomp** terminology, H is an **accumulator variable**. Since we want H to tally only the incidence over the week, we'll need to reset it to zero at the beginning of each week. We accomplish this using the `accumvars` argument to `pomp`:

```
measSIR %>%
  pomp(
    rprocess=euler(sir_step,delta.t=1/7),
    rinit=sir_rinit,accumvars="H"
  ) -> measSIR
```

- Now, we'll model the data as a binomial process,

$$\text{reports}_t \sim \text{Binomial}\left(H(t), \rho\right).$$

Implementing the SIR model in **pomp** VII

- Now, to include the observations in the model, we must write either a `dmeasure` or an `rmeasure` component, or both:

```
sir_dmeas <- function (reports, H, rho, log, ...) {
  dbinom(x=reports, size=H, prob=rho, log=log)
}

sir_rmeas <- function (H, rho, ...) {
  c(reports=rbinom(n=1, size=H, prob=rho))
}
```

Implementing the SIR model in **pomp** VIII

- We then put these into our pomp object:

```
measSIR %>%
  pomp(
    rmeasure=sir_rmeas,
    dmeasure=sir_dmeas
  ) -> measSIR
```

Specifying model components using C snippets

- Although we can always specify basic model components using R functions, as above, we'll typically want the computational speed-up that we can obtain only by using compiled native code.
- **pomp** provides a facility for doing so with ease, using **C snippets**.
- C snippets are small pieces of C code used to specify basic model components.
- For example, a C snippet encoding the rprocess for an `sir` model is as follows.

Specifying model components using C snippets II

```
sir_step <- Csnippet("
  double dN_SI = rbinom(S,1-exp(-Beta*I/N*dt));
  double dN_IR = rbinom(I,1-exp(-mu_IR*dt));
  S -= dN_SI;
  I += dN_SI - dN_IR;
  R += dN_IR;
  H += dN_IR;
  ")
```

Specifying model components using C snippets III

- C snippets for the initializer and measurement model are:

```
sir_rinit <- Csnippet("
  S = nearbyint(eta*N);
  I = 1;
  R = nearbyint((1-eta)*N);
  H = 0;
  ")

sir_dmeas <- Csnippet("
  lik = dbinom(reports,H,rho,give_log);
  ")

sir_rmeas <- Csnippet("
  reports = rbinom(H,rho);
  ")
```

Specifying model components using C snippets IV

- A call to pomp replaces the basic model components with these, much faster, implementations:

```
measSIR %>%
  pomp(rprocess=euler(sir_step,delta.t=1/7),
       rinit=sir_rinit,
       rmeasure=sir_rmeas,
       dmeasure=sir_dmeas,
       accumvars="H",
       statenames=c("S","I","R","H"),
       paramnames=c("Beta","mu_IR","N","eta","rho")
       ) -> measSIR
```

- Note that, when using C snippets, one has to tell **pomp** which of the variables referenced in the C snippets are state variables and which are parameters. This is accomplished using the `statenames` and `paramnames` arguments.

Guessing plausible parameter values

- To check the code is working properly, we simulate. This requires us to assign parameters. A little thought will get us some ballpark estimates.
- Recall that $\mathfrak{R}_0$ is the expected number of secondary infections resulting from one primary infection introduced into a fully susceptible population. For an SIR infection, one has that $\mathfrak{R}_0 \approx \frac{L}{A}$, where L is the lifespan of a host and A is the mean age of infection. Analysis of age-stratified serology data establish that the mean age of infection for measles during this period was around 4–5yr (Anderson and May, 1991). Assuming a lifespan of 60–70yr, we have $\mathfrak{R}_0 \approx 15$.

Guessing plausible parameter values II

- The basic theory of SIR epidemics gives the final-size equation,

$$\mathfrak{R}_0 = -\frac{\log(1-f)}{f},$$

  where f is the final size of the epidemic—the fraction of those susceptible at the beginning of the outbreak who ultimately become infected. For $\mathfrak{R}_0 > 5$, this equation predicts that $f > 0.99$.
- In the data, it looks like there were a total of 521 infections. Assuming 50% reporting, we have that $S_0 \approx 1042$, so that $\eta = \frac{S_0}{N} \approx 0.027$.
- If the infectious period is roughly 2 weeks, then $1/\mu_{IR} \approx 2$ wk and $\beta = \mu_{IR}\,\mathfrak{R}_0 \approx 7.5$ wk^{-1}.

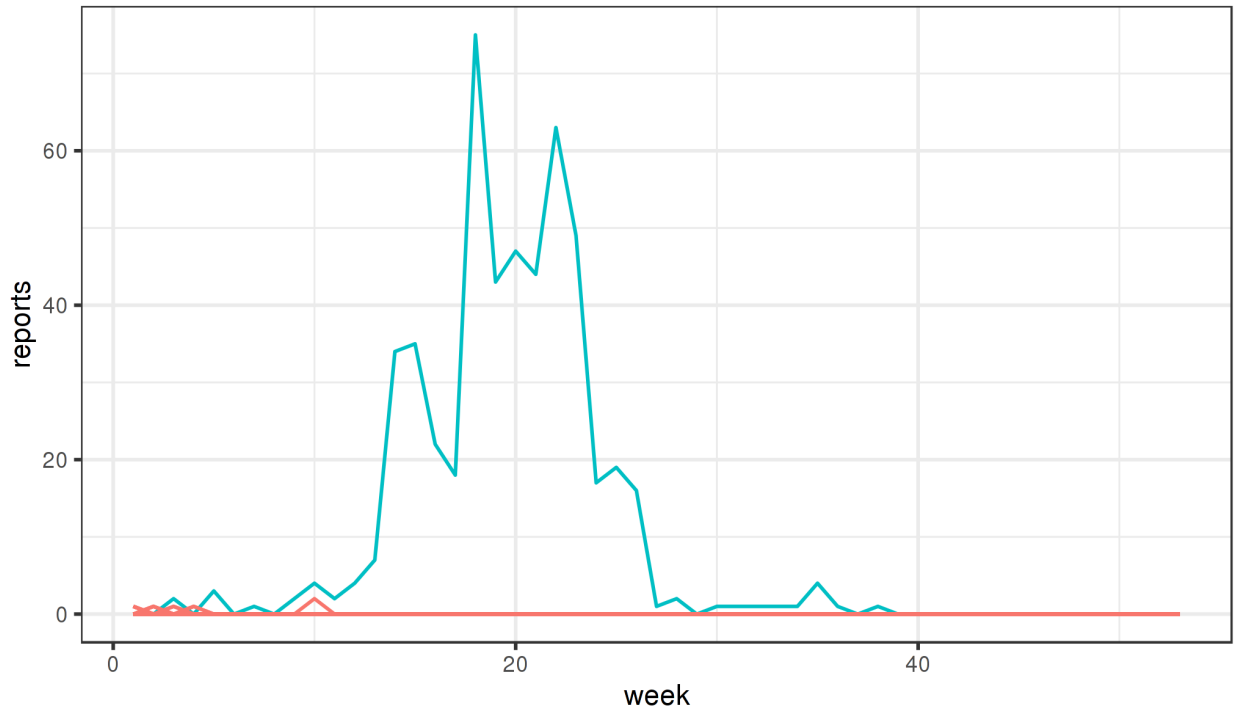
Guessing plausible parameter values III

- Let's simulate the model at these parameters.

```
measSIR %>%
  simulate(
    params=c(Beta=7.5,mu_IR=0.5,rho=0.5,eta=0.03,N=38000),
    nsim=20,format="data.frame",include.data=TRUE
  ) -> sims

sims %>%
  ggplot(aes(x=week,y=reports,group=.id,color=.id=="data"))+
  geom_line()+
  guides(color=FALSE)
```

Guessing plausible parameter values IV



This leaves something to be desired. In the exercises, you'll see if this model can do better.

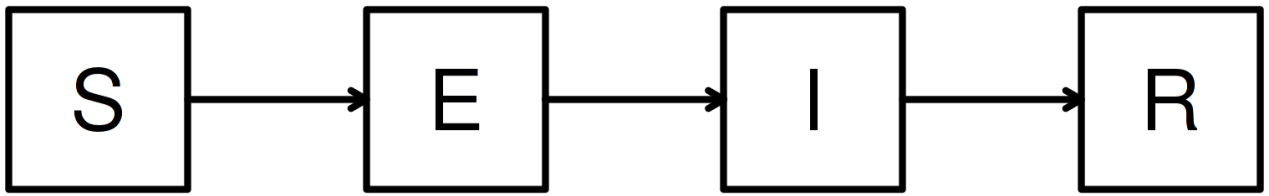
Exercise 2.3. Explore the SIR model.

Fiddle with the parameters to see if you can't find a model for which the data are a more plausible realization.

Worked solution to the Exercise

Exercise 2.4. The SEIR model

Below is a diagram of the so-called SEIR model. This differs from the SIR model in that infected individuals must pass a period of latency before becoming infectious.



Modify the codes above to construct a `pomp` object containing the flu data and an SEIR model. Perform simulations as above and adjust parameters to get a sense of whether improvement is possible by including a latent period.

Worked solution to the Exercise

References

Anderson RM, May RM (1991). *Infectious Diseases of Humans*. Oxford Univesity Press, Oxford.

Bretó C, He D, Ionides EL, King AA (2009). "Time series analysis via mechanistic models." *Annals of Applied Statistics*, **3**(1), 319–348. doi: 10.1214/08-AOAS201.

License, acknowledgments, and links

- This lesson is prepared for the Simulation-based Inference for Epidemiological Dynamics module at the 2020 Summer Institute in Statistics and Modeling in Infectious Diseases, SISMID 2020.
- 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 4.0.2 and **pomp** version 3.1.0.1.
- Compiled on July 6, 2020.

Back to course homepage
R codes for this lesson