

# **Stochastic and non-ODE epidemiological models**

## **Populate Summer School – Course 03**

**Julien Arino**

**Department of Mathematics @ University of Manitoba**  
**Maud Menten Institute @ PIMS**

**[julien.arino@umanitoba.ca](mailto:julien.arino@umanitoba.ca)**

**20 June 2025**

The University of Manitoba campuses are located on original lands of Anishinaabeg, Ininew, Anisininew, Dakota and Dene peoples, and on the National Homeland of the Red River Métis.

We respect the Treaties that were made on these territories, we acknowledge the harms and mistakes of the past, and we dedicate ourselves to move forward in partnership with Indigenous communities in a spirit of Reconciliation and collaboration.

# Outline

Why incorporate stochasticity?

Stochasticity in deterministic models

Continuous time Markov chains

Branching process approximations of CTMC



**Why incorporate stochasticity?**

**Stochasticity in deterministic models**

**Continuous time Markov chains**

**Branching process approximations of CTMC**

## At the beginning of the COVID-19 crisis

- ▶ I was working under contract with the Public Health Agency of Canada on *COVID-19 importation risk assessment*
- ▶ Produced daily report with list of countries most likely to next report cases of COVID-19
- ▶ Used ensemble runs of a fitted global deterministic metapopulation model



- ▶ Very very long days (18-20 hours, 7 days a week)
  - ▶ including a lot of time waiting for the “cluster” to finish
- ⇒ PHAC gave me money for a cluster (yay Threadrippers!!!)
- ⇒ Also thought about whether my model was really adequate as our focus switched from thinking about movement on a planetary scale to movement within Canadian provinces

## What is wrong with deterministic models?

- ▶ I pointed out yesterday that SARS-CoV-2 is one *single* realisation of a stochastic process
- ▶ Deterministic models “operate on averages” over a large ( $\rightarrow \infty$ ) number of realisations
- ▶ If we want to get a better sense of what could happen, not only on average, then we need to see what can indeed happen

## My new focus – Introductions

- ▶ I started thinking in particular about **introductions** (or importations) of pathogens into new populations
- ▶ Indeed, introductions are an obligatory step in spatial spread

## First piece of evidence

In real life, introductions of pathogens does not always follow the pattern

$$\{\mathcal{R}_0 < 1 \implies \text{DFE} \quad \mathcal{R}_0 > 1 \implies \text{epidemic or } \rightarrow \text{EEP}\}$$

## Short Communication

# SARS-CoV-2 in Nursing Homes: Analysis of Routine Surveillance Data in Four European Countries

**Tristan Delory<sup>1,2\*</sup>, Julien Arino<sup>3</sup>, Paul-Emile Hay<sup>4</sup>, Vincent Klotz<sup>4</sup>, Pierre-Yves Boëlle<sup>1</sup>**

<sup>1</sup> Sorbonne Université, INSERM, Institut Pierre Louis d’Épidémiologie et de Santé Publique, IPLESP, F-75012, Paris, France. <sup>2</sup>Centre Hospitalier Annecy Genevois, France. <sup>3</sup>Department of Mathematics, University of Manitoba, Winnipeg, Manitoba, Canada. <sup>4</sup>Groupe Colisee.

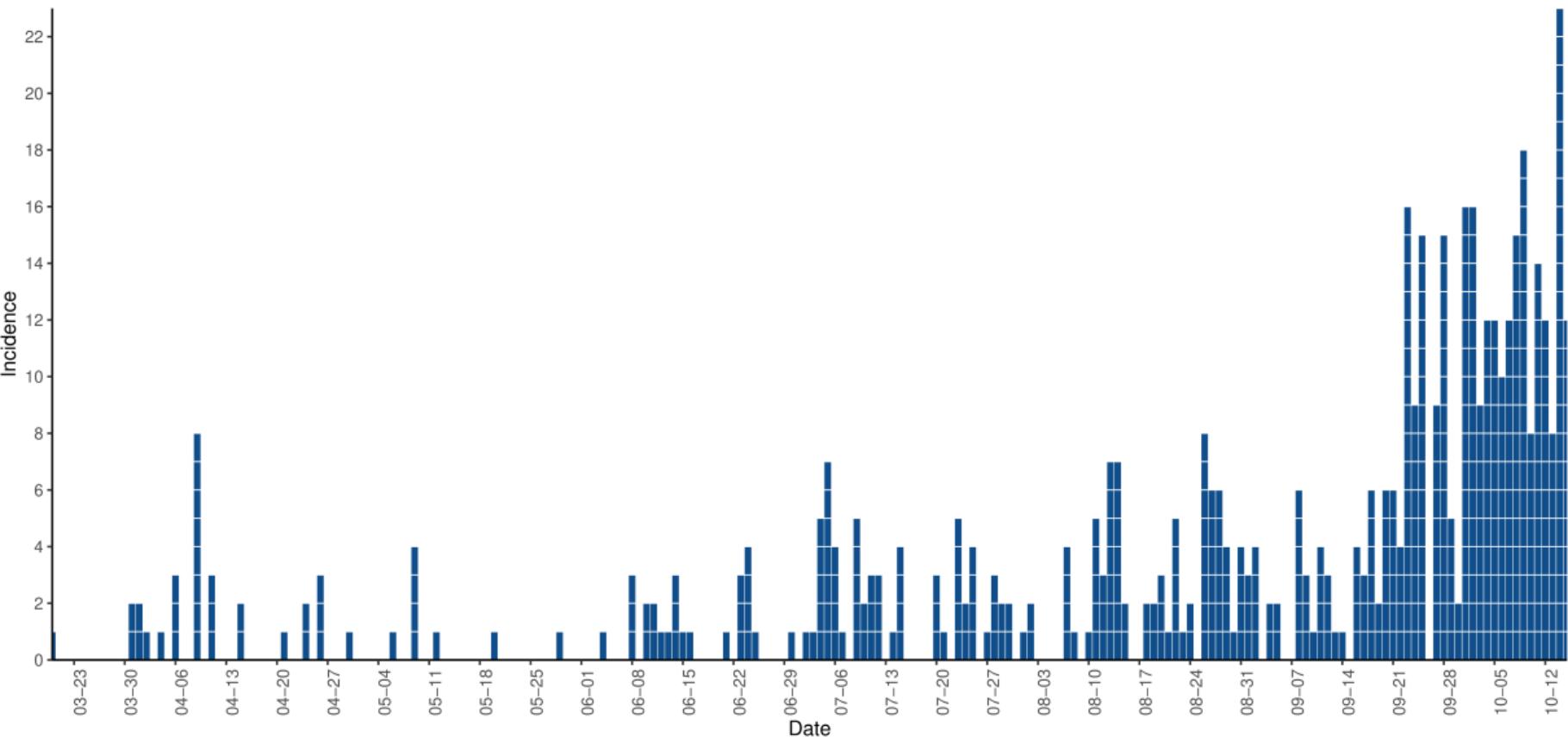
**Table 1.** Effect of vaccination scaling-up on the probability of successful viral introduction.

Period	Failed N = 136	Successful N = 366	aOR*	95%CI	P-value
<b>Before vaccination</b>	94 (69.1%)	311 (85.0%)	Ref		
<b>January 15 to January 31</b>	12 (8.8%)	37 (10.1%)	0.89	0.42 – 1.92	0.770
<b>February 01 to February 15</b>	17 (12.5%)	14 (3.8%)	0.23	0.10 – 0.52	<0.001
<b>February 16 to February 28</b>	13 (9.6%)	4 (1.1%)	0.08	0.02 - 0.29	<0.001

\* Adjusted on study period, country, staffing ratio, cumulative attack rate at onset of introduction, and number of PCR per 1000-residents or 1000-staff members, at onset of introduction, and nursing home maximal capacity.

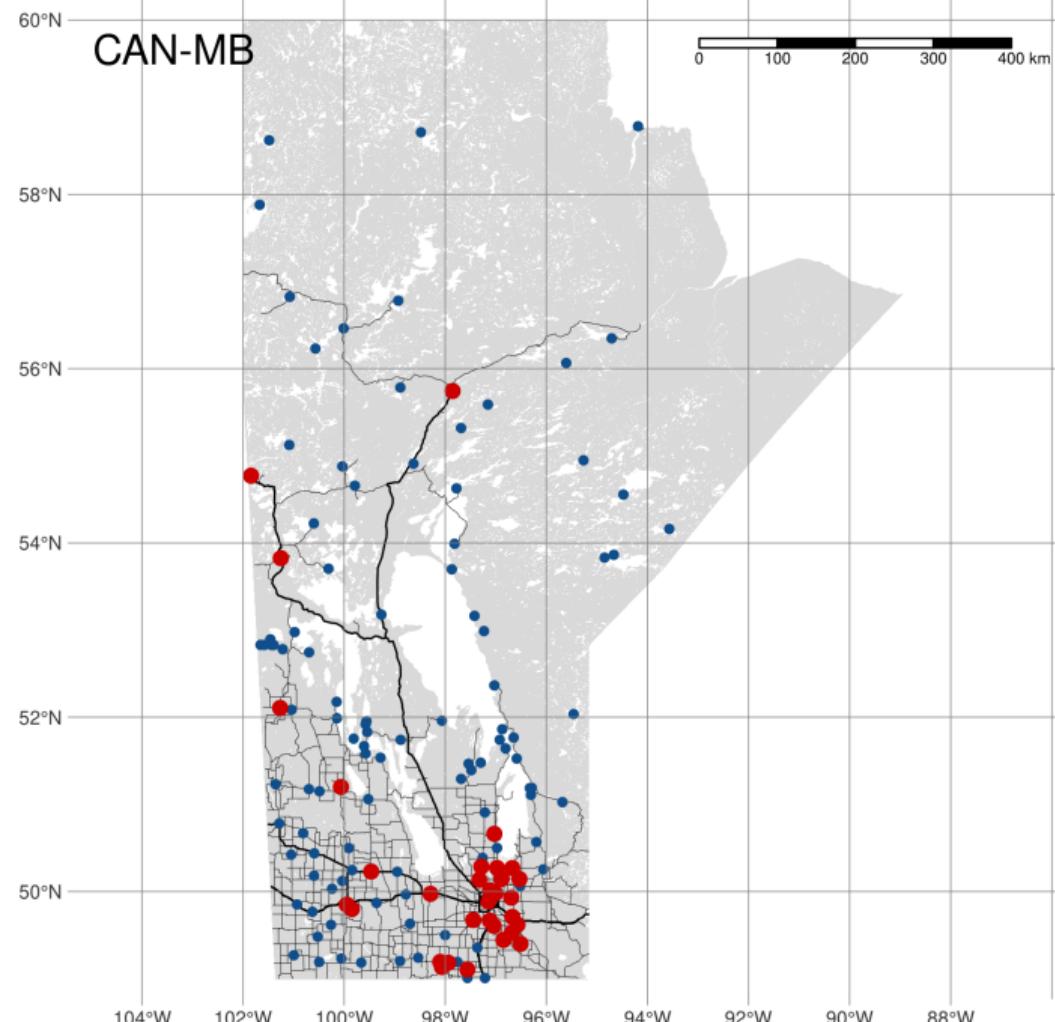
## Second piece of evidence

The start of an outbreak can be extremely slow, with very few cases for quite a while



## Why this is relevant

Far from the only reason, but as an example: Canada has remote/isolated communities that are vulnerable to introductions of pathogens



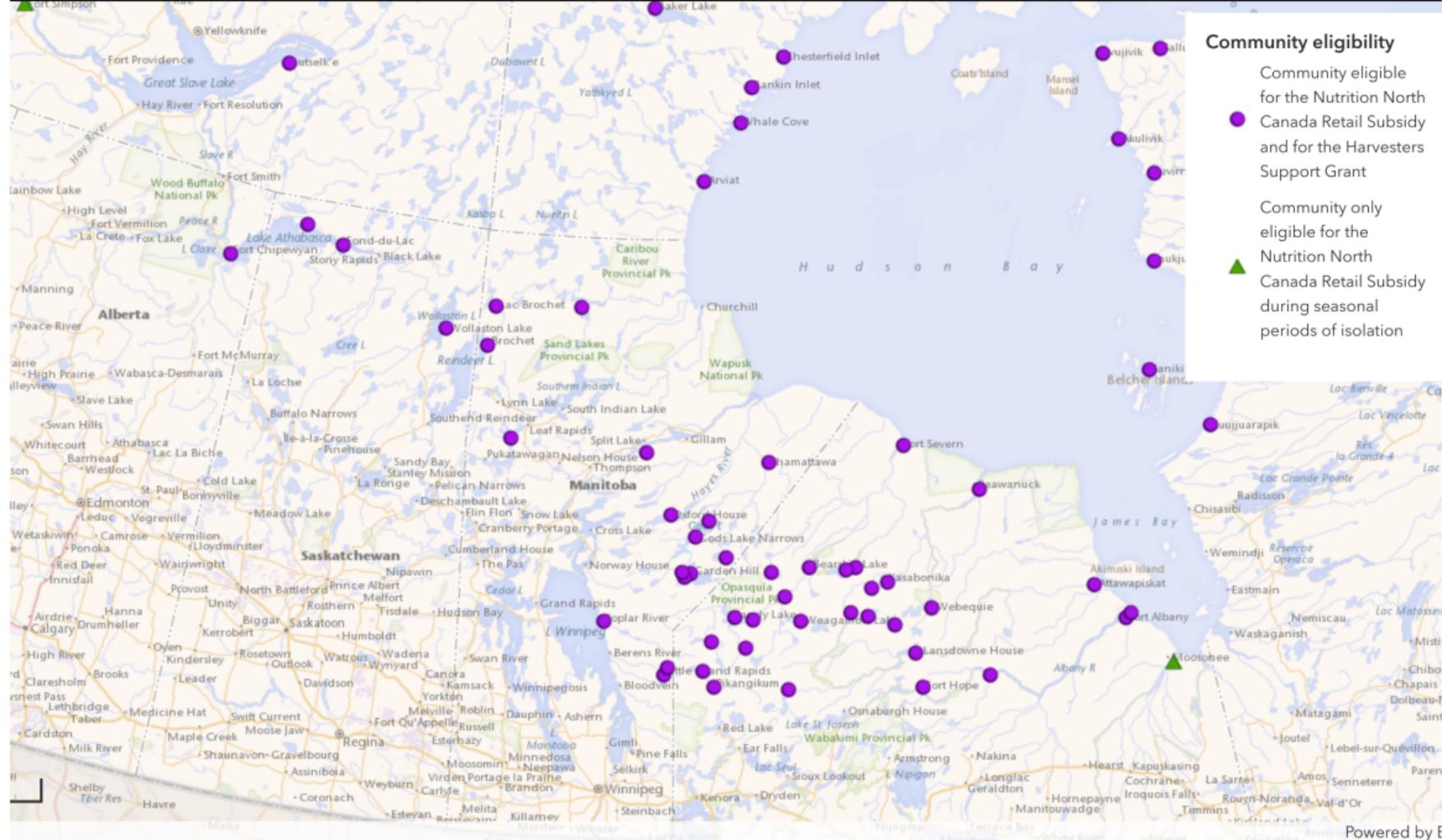
## Community eligibility

Community eligible  
for the Nutrition North

● Canada Retail Subsidy  
and for the Harvesters  
Support Grant

Community only  
eligible for the  
Nutrition North

▲ Canada Retail Subsidy  
during seasonal  
periods of isolation



# Northern Manitoba chiefs call for immediate federal action on health-care crisis

Recent deaths linked to inadequate medical care include mother of 5 from Manto Sipi Cree Nation, chief says

CBC News · Posted: Apr 03, 2023 3:20 PM CDT | Last Updated: April 3, 2023



## 'A lengthy process to get help here'

Wasagamack is one of four First Nations communities that make up Island Lake, an area in northeastern Manitoba dotted with hundreds of small islands.

Island Lake has a population of at least 15,000, according to Scott Harper, the grand chief of Anisininew Okimawin, which represents the four communities.

Despite having a population roughly the size of Thompson, and having diabetes and hospitalization rates well above provincial averages, Island Lake has no hospital of its own. The region is accessible only by air, boat and an unreliable winter road.

The nursing station in Wasagamack First Nation, which has about 2,300 people, according to federal government data, typically operates short-staffed, with only two or three of five registered nurses working on any given rotation and a fly-in doctor who comes weekly.

## For First Nation and Métis Communities

**Remote** describes a **geographical area** where a community is **located over 350 km** from the **nearest service centre having year-round access** by land and/or water routes normally used in all weather conditions

**Isolated** means a **geographical area** that has **scheduled flights** and good telephone service, but is **without year-round access** by land and/or water normally used in all weather conditions

**Remote-Isolated** means a **geographic area** that has **neither scheduled flights nor year-round access** by land and/or water routes normally that can be used in all weather conditions, irrespective of the level of telephone and radio service available

## For Inuit communities

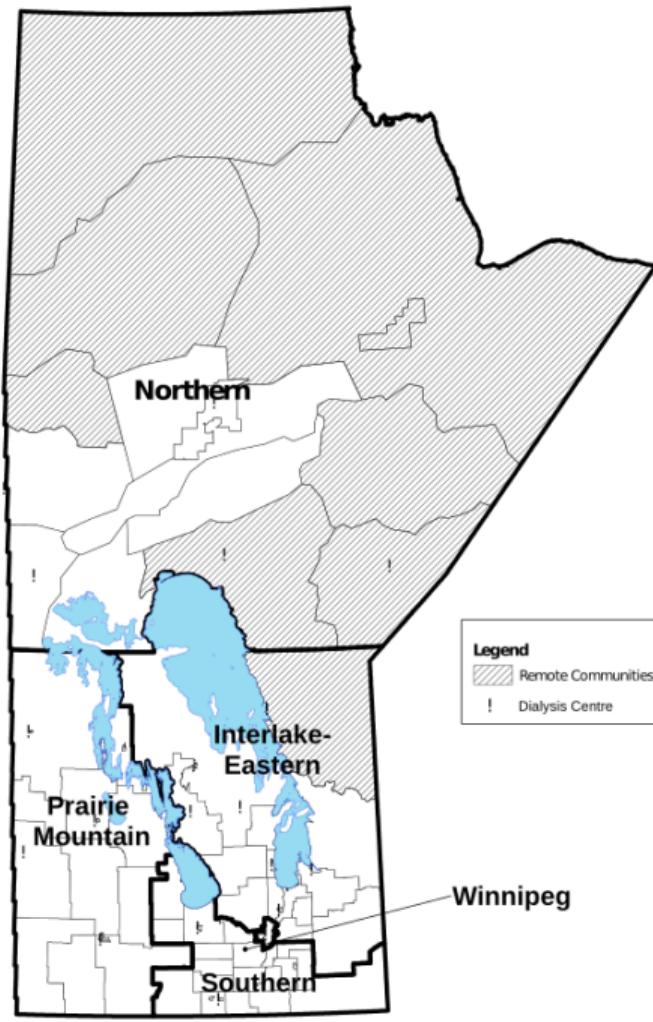
Inuit Communities to be referred to as **Inuit Nunangat**, not remote and isolated communities to respect the unique language and culture of Inuit regions, as well as the common challenges in social determinants of health, access to care, and infrastructure found across all Inuit communities

## MB remote communities

**Remote communities** are communities in Manitoba that **do not have permanent road access** (i.e., no all-weather road), are **more than a four-hour drive from a major rural hospital (and a dialysis unit)**, or **have rail or fly-in access only**. This includes Norway House, Lynn Lake, Leaf Rapids, Gillam, and Cross Lake. If most communities in a health district are designated as "remote", the entire district is designated as "remote". In Manitoba, remote districts include:

- ▶ Northern Health Region: NO23, NO13, NO25, NO16, NO22, NO26, NO28, NO31, and
- ▶ Interlake-Eastern Health Region: IE61.

Chartier M, Dart A, Tangri N, Komenda P, Walld R, Bogdanovic B, Burchill C, Koseva I, McGowan K, Rajotte L. Care of Manitobans Living with Chronic Kidney Disease. Winnipeg, MB. Manitoba Centre for Health Policy, December 2015



## Travel to/from remote or isolated communities

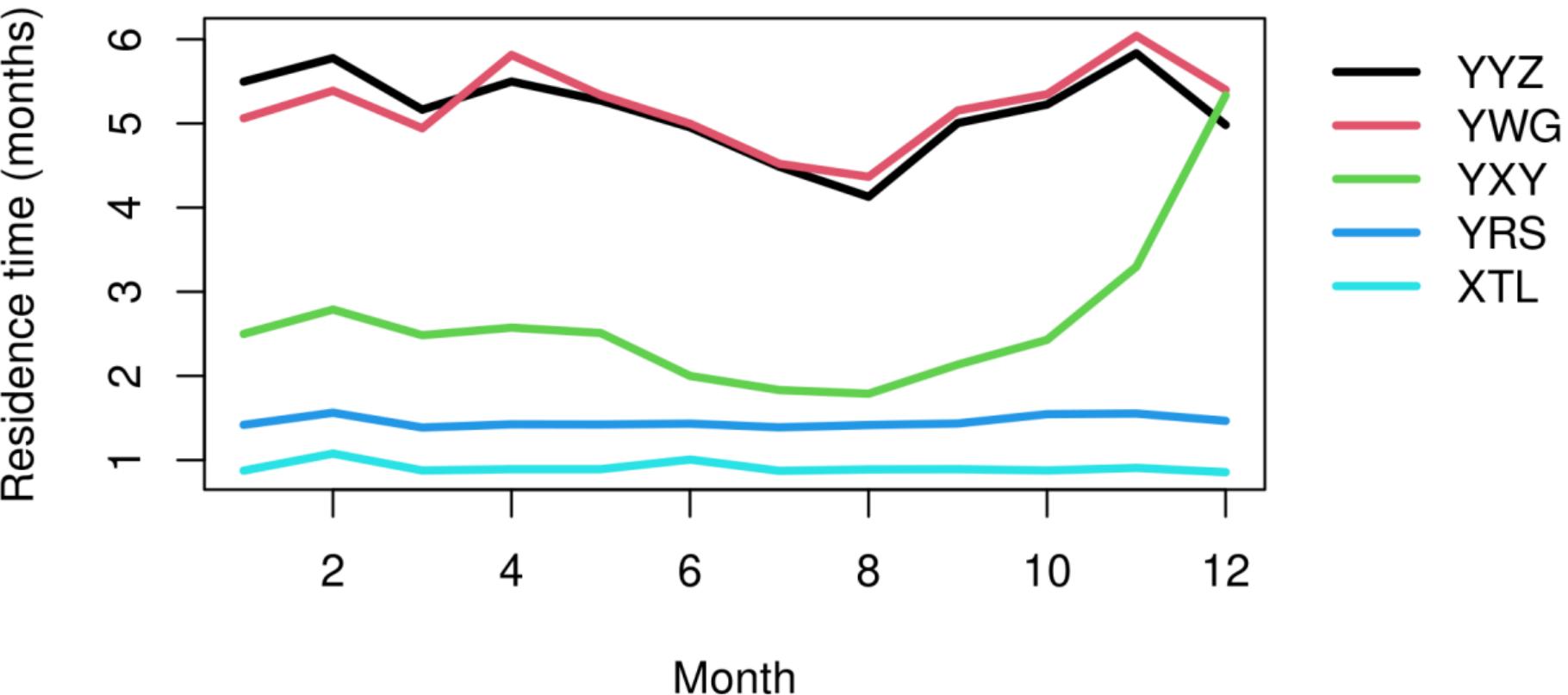
How do you think this compares to travel in non-remote/isolated communities ?

Residence time (the lake ecology version): theoretic time an average water or comparable molecule spends in a lake, considering inflow into and outflow from the lake

Think of residence times in these communities: what is the average time a person spends in a remote or isolated community before leaving it?

The **residence time in a location** is the total number of trips inbound into and outbound from location over a duration of time (1 month here) divided by the normal population in the location

# Residence times in months



## The paradox of travel to/from remote/isolated communities

Travel volumes small but movement rates high

ICs are highly connected to the urban centre(s) they are subordinated to

Further reinforced in Winnipeg by urban indigenous population (102,075 or 12.45% of metro population), meaning many family connections exist



Why incorporate stochasticity?

Stochasticity in deterministic models

Continuous time Markov chains

Branching process approximations of CTMC

## Stochasticity in deterministic models

Distributions of times to events

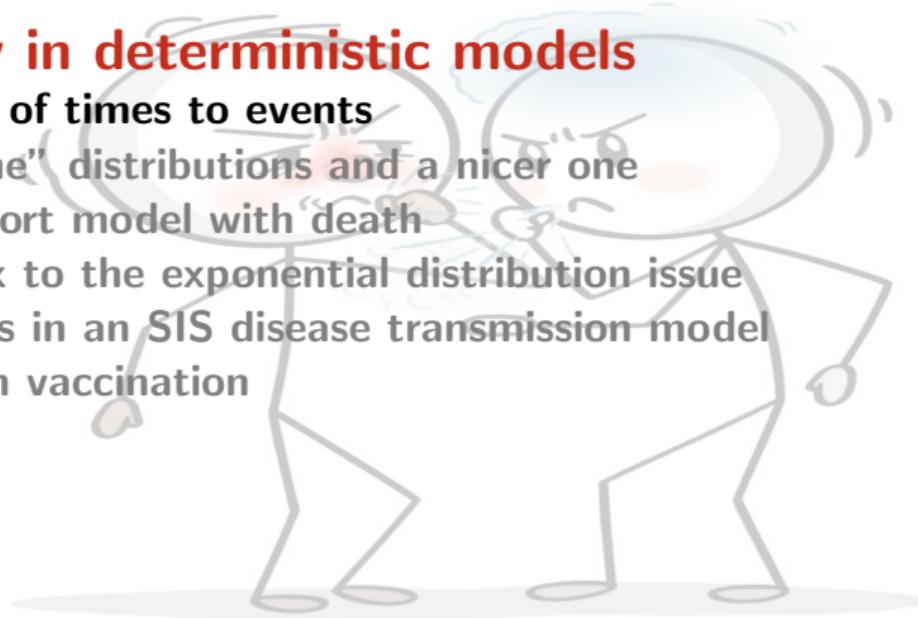
Two “extreme” distributions and a nicer one

A simple cohort model with death

A possible fix to the exponential distribution issue

Sojourn times in an SIS disease transmission model

A model with vaccination



See in particular the work of Horst Thieme

If one considers time of sojourn in compartments from a more detailed perspective, one obtains integro-differential models

We use here continuous random variables. See chapters 12 and 13 in Thieme's book for arbitrary distributions

## Time to events

We suppose that a system can be in two states,  $A$  and  $B$

- ▶ At time  $t = 0$ , the system is in state  $A$
- ▶ An event happens at some time  $t = \tau$ , which triggers the switch from state  $A$  to state  $B$

Let us call  $T$  the random variable

*“time spent in state A before switching into state B”*

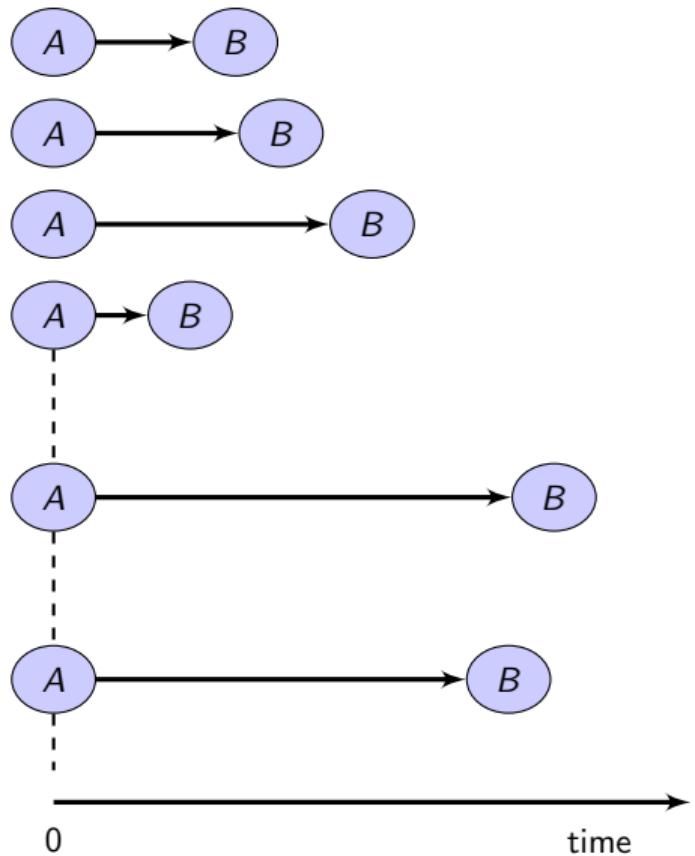
The states can be anything:

- ▶  $A$ : working,  $B$ : broken
- ▶  $A$ : infected,  $B$ : recovered
- ▶  $A$ : alive,  $B$ : dead
- ▶ ...

We take a collection of objects or individuals that are in state  $A$  and want some law for the **distribution** of the times spent in  $A$ , i.e., a law for  $T$

For example, we make light bulbs and would like to tell our customers that on average, our light bulbs last 200 years...

We conduct an **infinite** number of experiments, and observe the time that it takes, in every experiment, to switch from  $A$  to  $B$



## A distribution of probability is a model

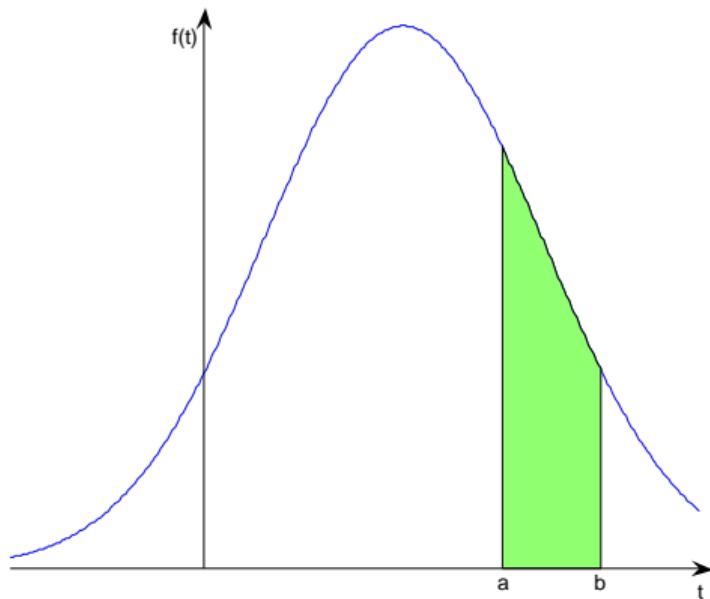
From the sequence of experiments, we deduce a model, which in this context is called a **probability distribution**

We assume that  $T$  is a **continuous** random variable

## Probability density function

Since  $T$  is continuous, it has a continuous **probability density function**  $f$

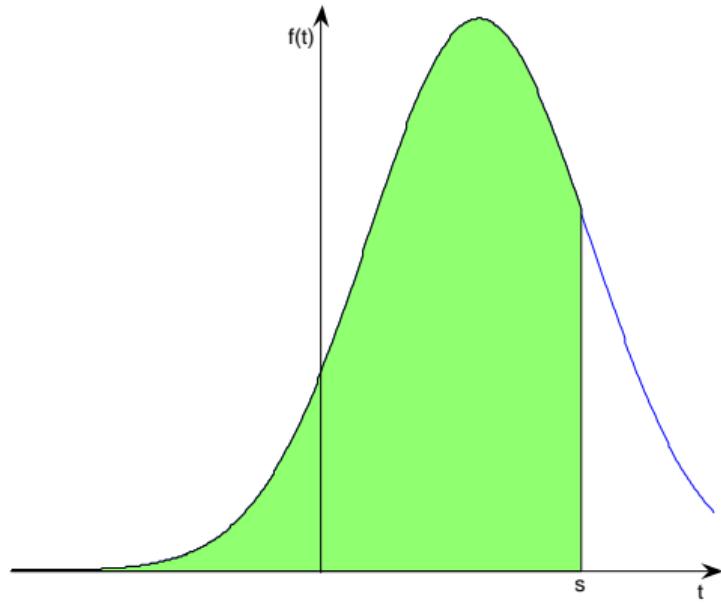
- ▶  $f \geq 0$
- ▶  $\int_{-\infty}^{+\infty} f(s)ds = 1$
- ▶  $\mathbb{P}(a \leq T \leq b) = \int_a^b f(t)dt$



## Cumulative distribution function

The cumulative distribution function (c.d.f.) is a function  $F(t)$  that characterizes the distribution of  $T$ , and defined by

$$F(s) = \mathbb{P}(T \leq s) = \int_{-\infty}^s f(x)dx$$



## Survival function

Another characterization of the distribution of the random variable  $T$  is through the **survival** (or **sojourn**) function

The survival function of state  $A$  is given by

$$S(t) = 1 - F(t) = \mathbb{P}(T > t) \quad (1)$$

This gives a description of the **sojourn time** of a system in a particular state (the time spent in the state)

$S$  is a nonincreasing function (since  $S = 1 - F$  with  $F$  a c.d.f.), and  $S(0) = 1$  (since  $T$  is a nonnegative random variable)

The **average sojourn time**  $\tau$  in state  $A$  is given by

$$\tau = E(T) = \int_0^\infty t f(t) dt$$

Since  $\lim_{t \rightarrow \infty} t S(t) = 0$ , it follows that

$$\tau = \int_0^\infty S(t) dt$$

**Expected future lifetime:**

$$\frac{1}{S(t_0)} \int_0^\infty t f(t + t_0) dt$$

$$\begin{aligned} S(t) - S(a) &= \mathbb{P}\{\text{survive during } (a, t) \text{ having survived until } a\} \\ &= \exp\left(-\int_a^t h(u) du\right) \end{aligned}$$

## Hazard rate

The **hazard rate** (or **failure rate**) is

$$\begin{aligned} h(t) &= \lim_{\Delta t \rightarrow 0} \frac{\mathcal{S}(t) - \mathcal{S}(t + \Delta t)}{\Delta t} \\ &= \lim_{\Delta t \rightarrow 0} \frac{\mathbb{P} T < t + \Delta t | T \geq t}{\Delta t} \\ &= \frac{f(t)}{\mathcal{S}(t)} \end{aligned}$$

It gives probability of failure between  $t$  and  $\Delta t$ , given survival to  $t$ .

We have

$$h(t) = -\frac{d}{dt} \ln \mathcal{S}(t)$$

## Competing risks

Suppose now that the system starts in state  $A$  at time  $t = 0$  and that depending on which of the two events  $\mathcal{E}_1$  or  $\mathcal{E}_2$  takes place first, it switches to state  $B_1$  or  $B_2$ , respectively

Consider the random variables  $T_A$ , *time spent in state A* (or sojourn time in  $A$ ),  $T_{AB_1}$ , *time before switch to  $B_1$*  and  $T_{AB_2}$ , *time before switch to  $B_2$*

If we consider state  $A$ , we cannot observe the variables  $T_{AB_1}$  or  $T_{AB_2}$ . What is observable is the sojourn time in  $A$

$$T_A^* = \min(T_{AB_1}, T_{AB_2})$$

(where \* indicates that a quantity is observable)

## Failure rate by type of event

We have two (or more) types of events whose individual failure rates have to be accounted for

$$h_j(t) = \lim_{\Delta t \rightarrow 0} \frac{\mathbb{P}(T < t + \Delta t, S = S_j | T \geq t)}{\Delta t}$$

where  $\mathbb{P}(T < t + \Delta t, S = S_j | T \geq t)$  is the probability of failure due to cause  $S_j$  ( $j = 1, 2$  ici), i.e.,  $S$  is a discrete r.v. representing the event that is taking place

By the law of total probability, since only one of the event can take place, if there are  $n$  risks, then

$$h(t) = \sum_{i=1}^n h_j(t)$$

or, identically,

$$\mathcal{S}(t) = \exp \left( - \int_0^t \sum_{j=1}^n h_j(s) \, ds \right)$$

As a consequence, suppose a process is subject to two competing exponential risks with respective distributions with parameters  $\theta_1$  and  $\theta_2$

Then the mean sojourn time in the initial state before being affected by one of the two risks is

$$\frac{1}{\theta_1 + \theta_2}$$

## Stochasticity in deterministic models

Distributions of times to events

Two “extreme” distributions and a nicer one

A simple cohort model with death

A possible fix to the exponential distribution issue

Sojourn times in an SIS disease transmission model

A model with vaccination

## The exponential distribution

The random variable  $T$  has an **exponential** distribution if its probability density function takes the form

$$f(t) = \begin{cases} 0 & \text{if } t < 0, \\ \theta e^{-\theta t} & \text{if } t \geq 0, \end{cases} \quad (2)$$

with  $\theta > 0$ . Then the survival function for state  $A$  is of the form  $S(t) = e^{-\theta t}$ , for  $t \geq 0$ , and the average sojourn time in state  $A$  is

$$\tau = \int_0^\infty e^{-\theta t} dt = \frac{1}{\theta}$$

## Particularities of the exponential distribution

The standard deviation of an exponential distribution is also  $1/\theta$ . When estimating  $\theta$ , it is impossible to distinguish the mean and the standard deviation

The exponential distribution is **memoryless**: its conditional probability obeys

$$P(T > s + t \mid T > s) = P(T > t), \quad \forall s, t \geq 0$$

The exponential and geometric distributions are the only memoryless probability distributions

The exponential distribution has a constant hazard function

## The Dirac delta distribution

If for some constant  $\omega > 0$ ,

$$S(t) = \begin{cases} 1, & 0 \leq t \leq \omega \\ 0, & \omega < t \end{cases}$$

meaning that  $T$  has a Dirac delta distribution  $\delta_\omega(t)$ , then the average sojourn time is

$$\tau = \int_0^\omega dt = \omega$$

with standard deviation  $\sigma = 0$

## The Gamma distribution

R.v.  $X$  is **Gamma** distributed ( $X \sim \Gamma(k, \theta)$ ) with **shape parameter**  $k$  and **scale parameter**  $\theta$  (or **rate**  $\beta = 1/\theta$ ) (all positive) if its probability density function takes the form

$$f(x; k, \theta) = \frac{x^{k-1} e^{-\frac{x}{\theta}}}{\Gamma(k)\theta^k} \quad (3)$$

where  $x > 0$  and  $\Gamma$  is the Euler Gamma function, defined for all  $z \in \mathbb{C}$  s.t.  $\operatorname{Re}(z) > 0$  by

$$\Gamma : z \mapsto \int_0^{+\infty} t^{z-1} e^{-t} dt$$

## Properties of the Gamma distribution

Mean  $k\theta$ , variance  $k\theta^2$

Survival function

$$S(t) = 1 - \frac{1}{\Gamma(k)} \gamma\left(k, \frac{t}{\theta}\right) = 1 - \frac{1}{\Gamma(k)} \gamma(k, \beta t)$$

where

$$\gamma(a, x) = \int_0^x t^{a-1} e^{-t} dt$$

is an incomplete Gamma function

## Stochasticity in deterministic models

Distributions of times to events

Two “extreme” distributions and a nicer one

A simple cohort model with death

A possible fix to the exponential distribution issue

Sojourn times in an SIS disease transmission model

A model with vaccination

## A model for a cohort with one cause of death

Consider a **cohort** of individuals born at the same time, e.g., the same year

- ▶ At time  $t = 0$ , there are initially  $N_0 > 0$  individuals
- ▶ All causes of death are compounded together
- ▶ The time until death, for a given individual, is a random variable  $T$ , with continuous probability density distribution  $f(t)$  and survival function  $\mathcal{S}(t)$

$N(t)$  the cohort population at time  $t \geq 0$

$$N(t) = N_0 \mathcal{S}(t) \tag{4}$$

$\mathcal{S}(t)$  proportion of initial population still alive at time  $t$ , so  $N_0 \mathcal{S}(t)$  number in the cohort still alive at time  $t$

## Case where $T$ is exponentially distributed

Suppose that  $T$  has an exponential distribution with mean  $1/d$  (or parameter  $d$ ),  $f(t) = de^{-dt}$ . Then the survival function is  $S(t) = e^{-dt}$ , and (4) takes the form

$$N(t) = N_0 e^{-dt} \tag{5}$$

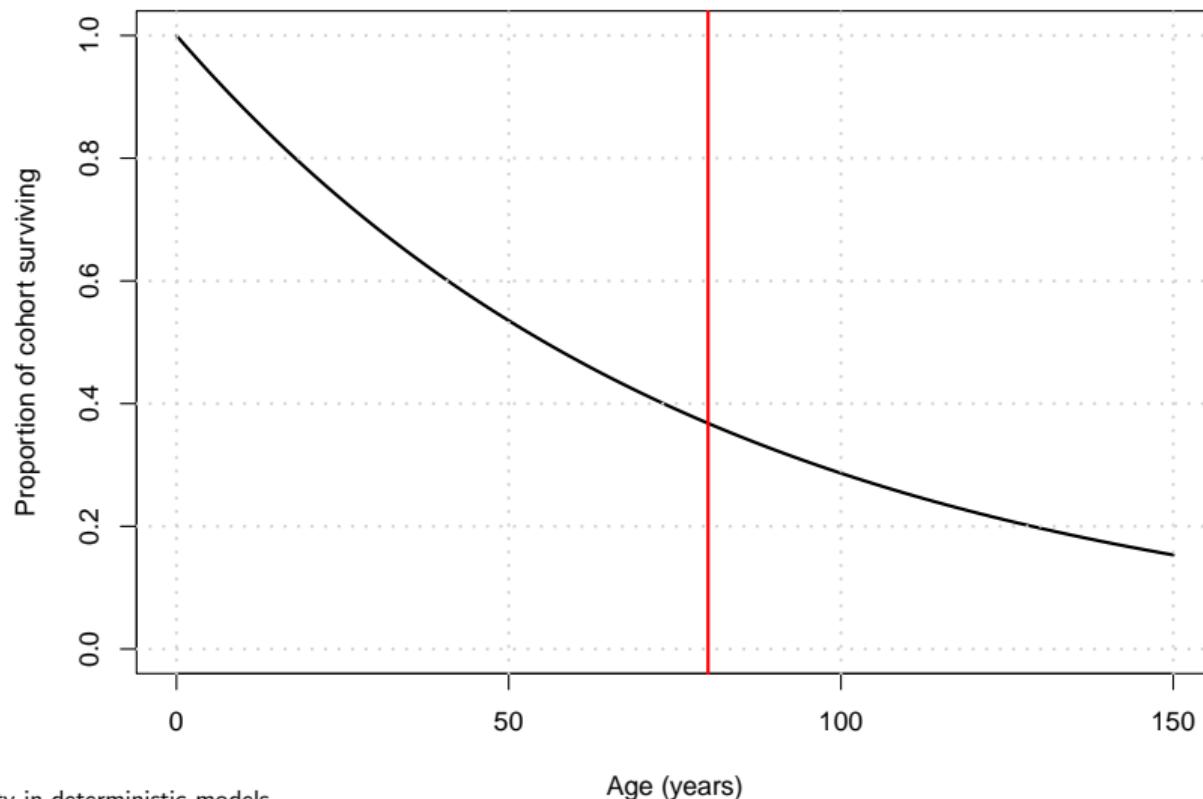
Now note that

$$\begin{aligned}\frac{d}{dt} N(t) &= -dN_0 e^{-dt} \\ &= -dN(t)\end{aligned}$$

with  $N(0) = N_0$ .

⇒ The ODE  $N' = -dN$  makes the assumption that the life expectancy at birth is exponentially distributed

Survival function,  $\mathcal{S}(t) = \mathbb{P}(T > t)$ , for an exponential distribution with mean 80 years



## Case where $T$ has a Dirac delta distribution

Suppose that  $T$  has a Dirac delta distribution at  $t = \omega$ , giving the survival function

$$S(t) = \begin{cases} 1, & 0 \leq t \leq \omega \\ 0, & t > \omega \end{cases}$$

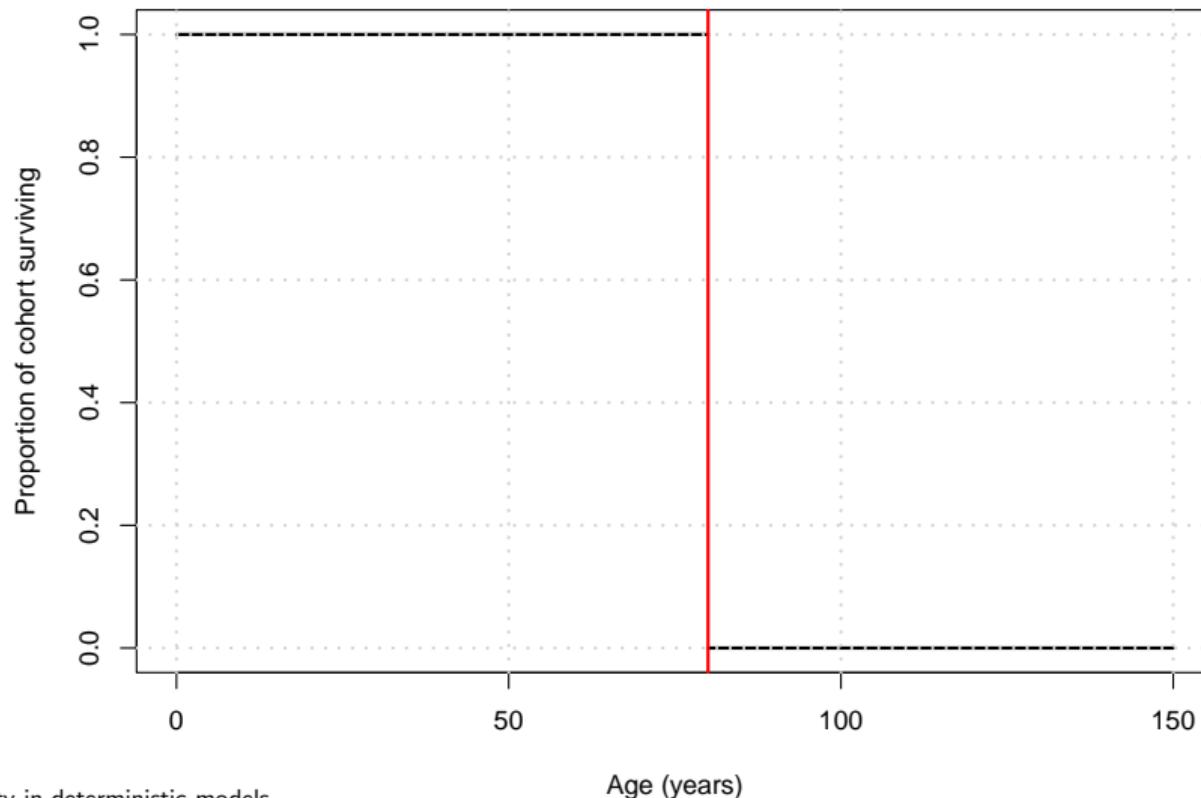
Then (4) takes the form

$$N(t) = \begin{cases} N_0, & 0 \leq t \leq \omega \\ 0, & t > \omega \end{cases} \quad (6)$$

All individuals survive until time  $\omega$ , then they all die at time  $\omega$

Here,  $N' = 0$  everywhere except at  $t = \omega$ , where it is undefined

Survival function,  $\mathcal{S}(t) = \mathbb{P}(T > t)$ , for a Dirac distribution with mean 80 years



## Stochasticity in deterministic models

Distributions of times to events

Two “extreme” distributions and a nicer one

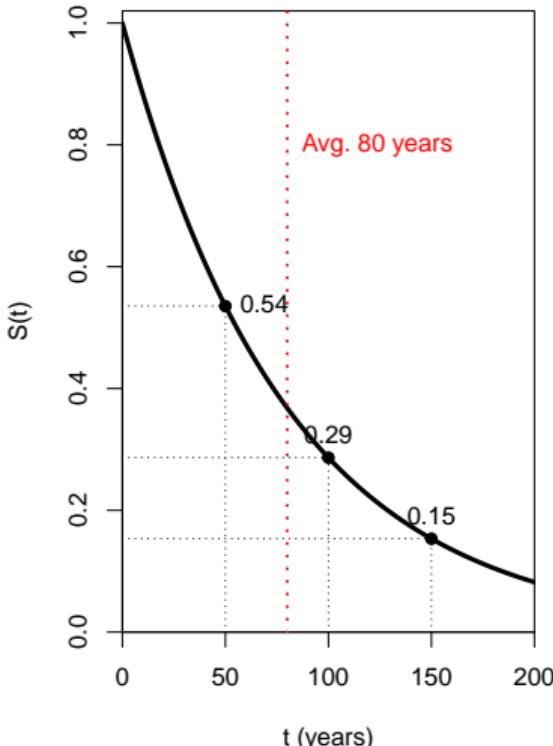
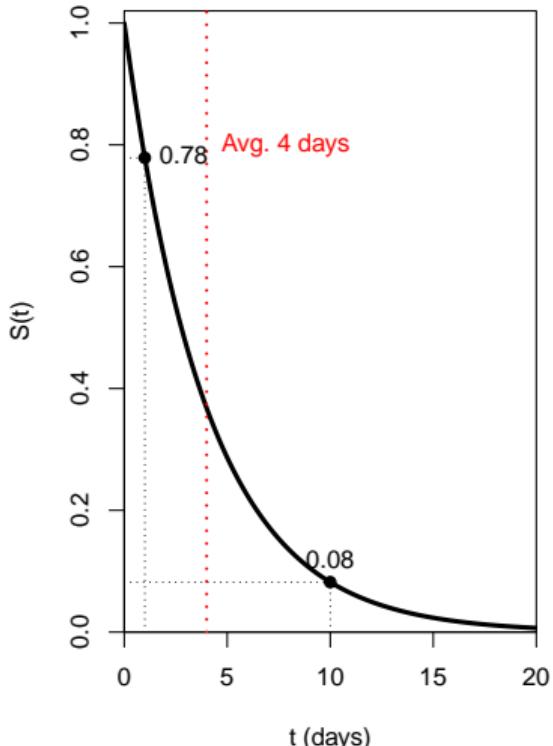
A simple cohort model with death

**A possible fix to the exponential distribution issue**

Sojourn times in an SIS disease transmission model

A model with vaccination

## Survival for the exponential distribution



## Issues with the exponential distribution

- ▶ Survival drops quickly
- ▶ Survival continues way beyond the mean

Acceptable if what matters is the average duration of sojourn in a compartment (e.g., long term dynamics)

More iffy if one is interested in short-term dynamics

- ▶ Exponential distribution with parameter  $\theta$  has same mean and standard deviation  $1/\theta$ , i.e., a single parameter controls mean and dispersion about the mean

## Exponential distributions are “bad” but also cool

$X_1$  and  $X_2$  2 i.i.d. (independent and identically distributed) r.v. with parameters  $\theta_1$  and  $\theta_2$ . Then the probability density function of the r.v.  $Z = X_1 + X_2$  is given by the convolution

$$\begin{aligned} f_Z(z) &= \int_{-\infty}^{\infty} f_{X_1}(x_1) f_{X_2}(z - x_1) dx_1 \\ &= \int_0^z \theta_1 e^{-\theta_1 x_1} \theta_2 e^{-\theta_2(z-x_1)} dx_1 \\ &= \theta_1 \theta_2 e^{-\theta_2 z} \int_0^z e^{(\theta_2 - \theta_1)x_1} dx_1 \\ &= \begin{cases} \frac{\theta_1 \theta_2}{\theta_2 - \theta_1} (e^{-\theta_1 z} - e^{-\theta_2 z}) & \text{if } \theta_1 \neq \theta_2 \\ \theta^2 z e^{-\theta z} & \text{if } \theta_1 = \theta_2 =: \theta \end{cases} \end{aligned} \tag{7}$$

## The tool we use

### Theorem 1

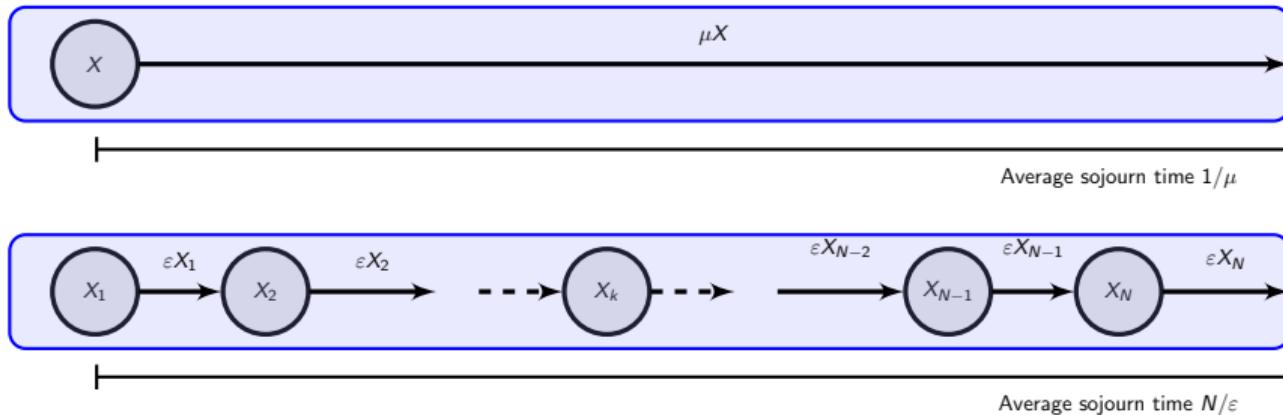
Let  $X_i$  be independent exponentially distributed random variables with parameter  $\xi$  and  $Y = \sum_{i=1}^n X_i$

Then the random variable  $Y \rightsquigarrow E(n, \xi)$ , an Erlang distribution with shape parameter  $n$  and scale parameter  $\xi$

(Erlang distribution: Gamma distribution with integer shape parameter)

## Consequences for compartmental models

If  $n$  compartments are traversed successively by individuals, with each compartment having an outflow rate of  $1/\xi$  (or a mean sojourn time of  $\xi$ ), then the time of sojourn from entry into the first compartment to exit from the last is Erlang distributed with mean  $E(Y) = n\xi$  and variance  $\text{Var}(Y) = n\xi^2$



I have a Shiny app for this :)

## Example: EVD incubation periods

Consider the incubation period for Ebola Virus Disease. During the 2014 EVD crisis in Western Africa, the WHO Ebola Response Team estimated incubation periods in a 2015 paper

Table S2 in the Supplementary Information in that paper gives the best fit for the distribution of incubation periods for EVD as a Gamma distribution with mean 10.3 days and standard deviation 8.2, i.e.,  $n\varepsilon = 10.3$  and  $\varepsilon\sqrt{n} = 8.2$

From this,  $\varepsilon = 8.2^2/10.3 \simeq 6.53$  and  $n = 10.3^2/8.2^2 \simeq 1.57$ . However, that is a Gamma distribution

## Switching to a compartmental model approach

To use multiple compartments to better fit residence times, we need to find the closest possible Erlang distribution to this Gamma distribution

⇒ compute RSS errors between data points generated from the given Gamma distribution and an Erlang

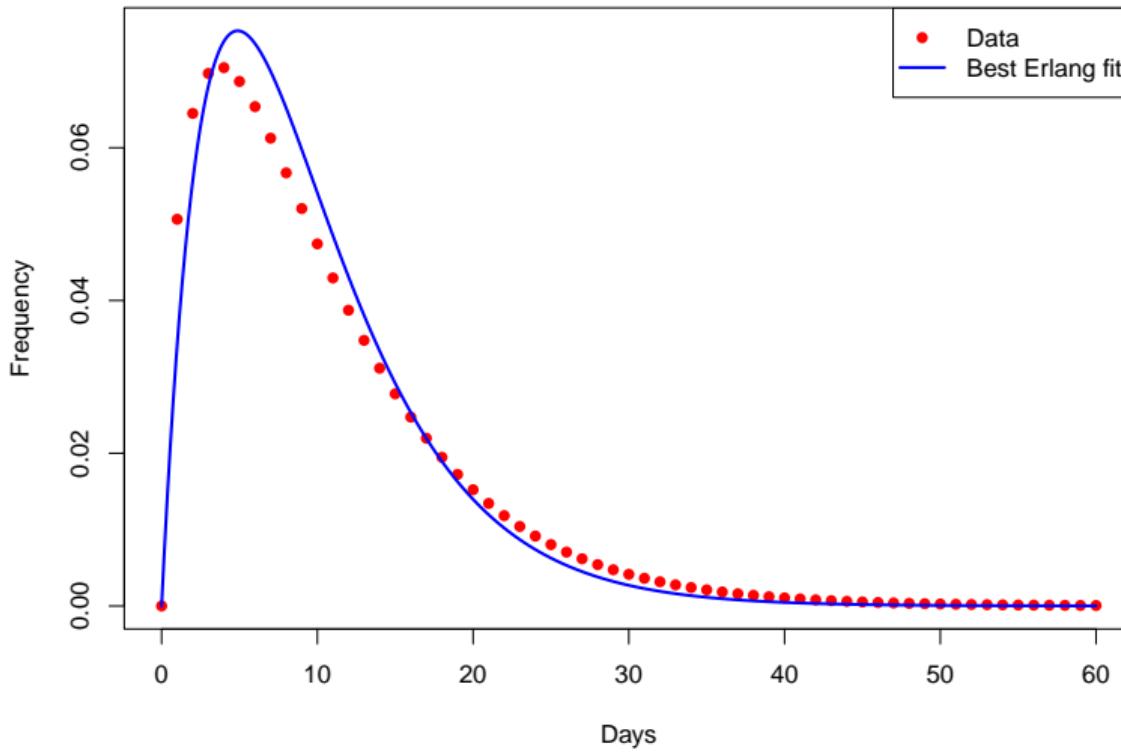
```
error_Gamma <- function(theta,shape,t,d) {  
  test_points <- dgamma(t, shape = shape, scale = theta)  
  ls_error <- sum((d-test_points)^2)  
  return(ls_error)  
}
```

```
optimize_gamma <- function(t,d) {
  max_shape <- 10
  error_vector <- mat.or.vec(max_shape,1)
  scale_vector <- mat.or.vec(max_shape,1)
  for (i in 1:max_shape) {
    result_optim <- try(optim(par = 3,
                                fn = error_Gamma,
                                lower = 0,
                                method = "L-BFGS-B",
                                shape = i,
                                t = t,
                                d = d),
                           TRUE)
    if (!inherits(result_optim,"try-error")) {
      error_vector[i] <- result_optim$value
      scale_vector[i] <- result_optim$par
```

```
    } else {
      error_vector[i] <- NaN
      scale_vector[i] <- NaN
    }
  }
result_optim <- data.frame(seq(1,max_shape),
                           scale_vector,
                           error_vector)
colnames(result_optim) <- c("shape","scale","error")
result_optim <- result_optim[complete.cases(result_optim),]
return(result_optim)
}
```

```
time_points <- seq(0,60)
data_points <- dgamma(time_points, shape = 1.57,
                      scale = 6.53)
# Run the minimization
optim_fits <- optimize_gamma(time_points,data_points)
# Which is the best Erlang to fit the data
idx_best <- which.min(optim_fits$error)
```

Now plot the result as well as the original curve (code chunk not shown)



## Stochasticity in deterministic models

Distributions of times to events

Two “extreme” distributions and a nicer one

A simple cohort model with death

A possible fix to the exponential distribution issue

**Sojourn times in an SIS disease transmission model**

A model with vaccination

# An SIS model

## Hypotheses

- ▶ Individuals typically recover from the disease
- ▶ The disease does not confer immunity
- ▶ There is no birth or death (from the disease or natural)  
⇒ Constant total population  $N \equiv N(t) = S(t) + I(t)$
- ▶ Infection is of **standard incidence** type

## Recovery

- ▶ Traditional models suppose that recovery occurs with rate constant  $\gamma$
- ▶ Here, of the individuals that become infective at time  $t_0$ , a fraction  $\mathcal{S}(t - t_0)$  remain infective at time  $t \geq t_0$
- ▶  $\Rightarrow$  For  $t \geq 0$ ,  $\mathcal{S}(t)$  is a survival function. As such, it verifies  $\mathcal{S}(0) = 1$  and  $\mathcal{S}$  is nonnegative and nonincreasing

## Model for infectious individuals

Since  $N$  is constant,  $S(t) = N - I(t)$  and we need only consider the following equation (where  $S$  is used for clarity)

$$I(t) = I_0(t) + \int_0^t \beta \frac{S(u)I(u)}{N} S(t-u) du \quad (8)$$

- ▶  $I_0(t)$  number of individuals who were infective at time  $t = 0$  and still are at time  $t$ 
  - ▶  $I_0(t)$  is nonnegative, nonincreasing, and such that  $\lim_{t \rightarrow \infty} I_0(t) = 0$
- ▶  $S(t - u)$  proportion of individuals who became infective at time  $u$  and who still are at time  $t$

## Expression under the integral

Integral equation for the number of infective individuals:

$$I(t) = I_0(t) + \int_0^t \beta \frac{(N - I(u))I(u)}{N} S(t - u) du \quad (8)$$

The term

$$\beta \frac{(N - I(u))I(u)}{N} S(t - u)$$

- ▶  $\beta(N - I(u))I(u)/N$  is the rate at which new infectives are created, at time  $u$
- ▶ multiplying by  $S(t - u)$  gives the proportion of those who became infectives at time  $u$  and who still are at time  $t$

Summing over  $[0, t]$  gives the number of infective individuals at time  $t$

## Case of an exponentially distributed time to recovery

Suppose  $\mathcal{S}(t)$  such that sojourn time in the infective state has exponential distribution with mean  $1/\gamma$ , i.e.,  $\mathcal{S}(t) = e^{-\gamma t}$

Initial condition function  $I_0(t)$  takes the form

$$I_0(t) = I_0(0)e^{-\gamma t}$$

with  $I_0(0)$  the number of infective individuals at time  $t = 0$ . Obtained by considering the cohort of initially infectious individuals, giving a model such as (4)

Equation (8) becomes

$$I(t) = I_0(0)e^{-\gamma t} + \int_0^t \beta \frac{(N - I(u))I(u)}{N} e^{-\gamma(t-u)} du \quad (9)$$

Taking the time derivative of (9) yields

$$\begin{aligned}I'(t) &= -\gamma I_0(0)e^{-\gamma t} - \gamma \int_0^t \beta \frac{(N - I(u))I(u)}{N} e^{-\gamma(t-u)} du \\&\quad + \beta \frac{(N - I(t))I(t)}{N} \\&= -\gamma \left( I_0(0)e^{-\gamma t} + \int_0^t \beta \frac{(N - I(u))I(u)}{N} e^{-\gamma(t-u)} du \right) \\&\quad + \beta \frac{(N - I(t))I(t)}{N} \\&= \beta \frac{(N - I(t))I(t)}{N} - \gamma I(t)\end{aligned}$$

This is the classical logistic type ordinary differential equation (ODE) for  $I$  in an SIS model without vital dynamics (no birth or death)

## Case of a step function survival function

Consider case where the time spent infected has survival function

$$\mathcal{S}(t) = \begin{cases} 1, & 0 \leq t \leq \omega, \\ 0, & t > \omega. \end{cases}$$

i.e., the sojourn time in the infective state is a constant  $\omega > 0$

In this case (8) becomes

$$I(t) = I_0(t) + \int_{t-\omega}^t \beta \frac{(N - I(u))I(u)}{N} du. \quad (10)$$

Here, it is more difficult to obtain an expression for  $I_0(t)$ . It is however assumed that  $I_0(t)$  vanishes for  $t > \omega$

When differentiated, (10) gives, for  $t \geq \omega$ ,

$$I'(t) = I'_0(t) + \beta \frac{(N - I(t))I(t)}{N} - \beta \frac{(N - I(t - \omega))I(t - \omega)}{N}.$$

Since  $I_0(t)$  vanishes for  $t > \omega$ , this gives the delay differential equation (DDE)

$$I'(t) = \beta \frac{(N - I(t))I(t)}{N} - \beta \frac{(N - I(t - \omega))I(t - \omega)}{N}.$$

## Stochasticity in deterministic models

Distributions of times to events

Two “extreme” distributions and a nicer one

A simple cohort model with death

A possible fix to the exponential distribution issue

Sojourn times in an SIS disease transmission model

**A model with vaccination**

## AN EPIDEMIOLOGY MODEL THAT INCLUDES A LEAKY VACCINE WITH A GENERAL WANING FUNCTION

JULIEN ARINO

Department of Mathematics and Statistics  
University of Victoria, Victoria B.C., Canada V8W 3P4

K.L. COOKE

Department of Mathematics  
Pomona College, Claremont, CA 91711-6348 USA

P. VAN DEN DRIESSCHE

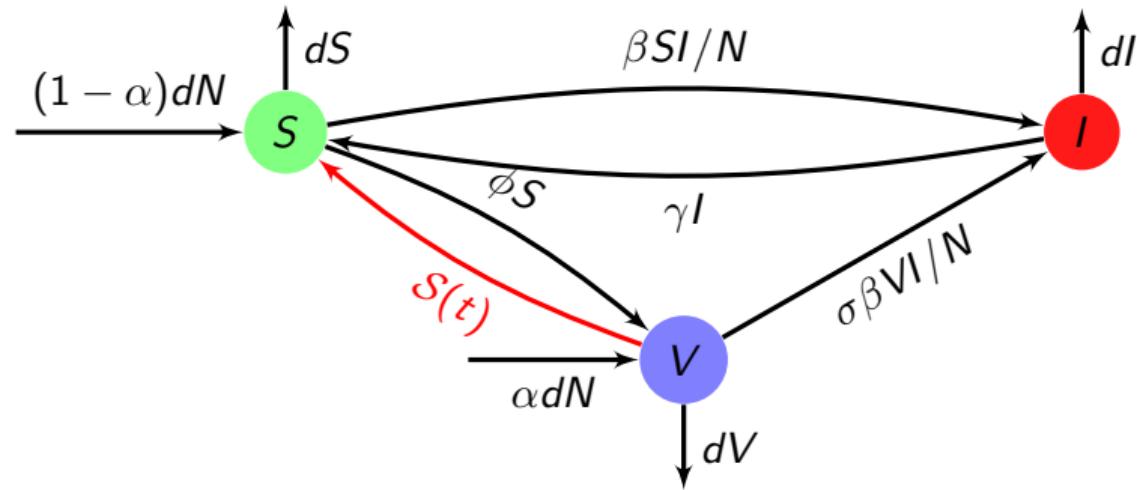
Department of Mathematics and Statistics  
University of Victoria, Victoria B.C., Canada V8W 3P4

J. VELASCO-HERNÁNDEZ

Programa de Matemáticas Aplicadas y Computación  
Instituto Mexicano del Petróleo, Eje Central Lázaro Cárdenas 152  
San Bartolo Atepehuacan, D.F. 07730, Mexico

(Communicated by Linda Allen)

## Model structure



## Assumptions on $\mathcal{S}$

$\mathcal{S}(t)$  is a nonnegative and nonincreasing function with  $\mathcal{S}(0^+) = 1$ , and such that  $\int_0^\infty \mathcal{S}(u)du$  is positive and finite

So  $\mathcal{S}(t)$  is a survival function

## The SIS model with vaccination

$$\frac{dI(t)}{dt} = \beta(S(t) + \sigma V(t))I(t) - (d + \gamma)I(t) \quad (11a)$$

$$V(t) = V_0(t) + \int_0^t (\phi S(u) + \alpha d)S(t-u)e^{-d(t-u)}e^{-\sigma\beta\int_u^t I(x)dx}du \quad (11b)$$

- ▶  $\alpha d$  proportion of vaccinated newborns
- ▶  $\phi S(u)$  proportion of vaccinated susceptibles
- ▶  $S(t-u)$  fraction of the proportion vaccinated still in the  $V$  class  $t-u$  time units after going in
- ▶  $e^{-d(t-u)}$  fraction of the proportion vaccinated not dead due to natural causes
- ▶  $e^{-\sigma\beta\int_u^t I(x)dx}$  fraction of the proportion vaccinated not gone to the infective class

## Reduction of the system using specific $\mathcal{S}(t)$ functions

- ▶ The distribution of waning times being exponential leads to an ODE system
- ▶  $\mathcal{S}(t)$  originating in a Dirac distribution leads to a discrete DDE model

## Be aware (beware?) of what's under the hood

ODEs and discrete-delay DDEs are fine, but they hide some pretty strong assumptions

If you are using an ODE for an *endemic* model, fine: you are typically working with  $t \rightarrow \infty$  and the ODE “uses” the mean of your parameter

If, on the other hand, you are using an ODE to assess behaviour for times that are not too different from the mean of your parameters, then beware

Discrete delays make the assumption that every single individual in your population has the same sojourn time if whatever compartment is being applied delay to... You may want to think about distributed delays or other mechanisms to compensate for this

You may want to use a different hammer



**Why incorporate stochasticity?**

**Stochasticity in deterministic models**

**Continuous time Markov chains**

**Branching process approximations of CTMC**

# Continuous time Markov chains

Continuous time Markov chains

ODE and CTMC

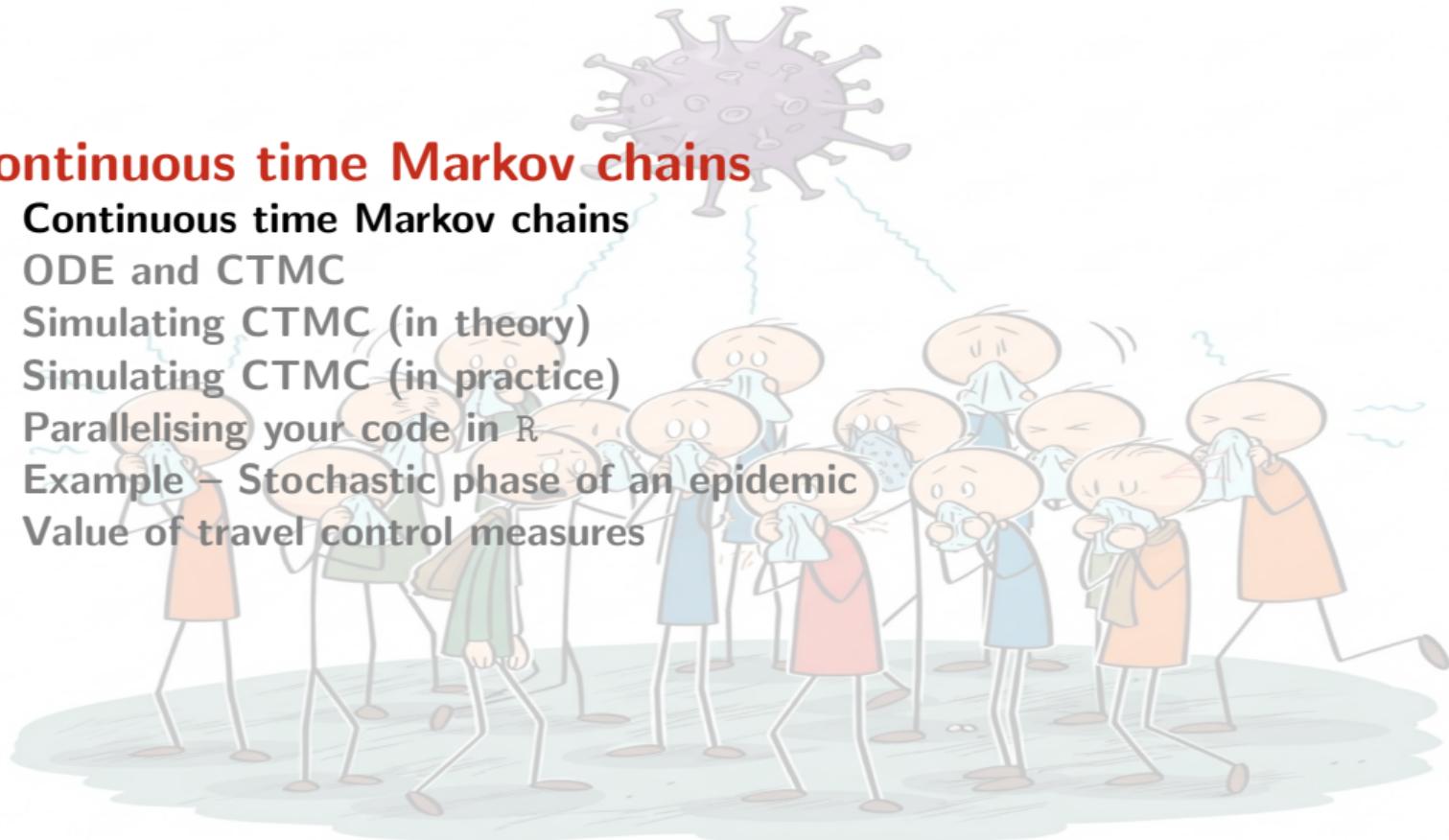
Simulating CTMC (in theory)

Simulating CTMC (in practice)

Parallelising your code in R

Example – Stochastic phase of an epidemic

Value of travel control measures



## From discrete to continuous time

### Discrete-Time Markov Chains (DTMCs)

A system transitions between states at fixed, discrete time steps ( $n = 0, 1, 2, \dots$ )

- ▶ The future depends only on the present state (Markov Property)
- ▶ Governed by a **transition probability matrix**  $P$ , where  $P_{ij}$  is the probability of moving from state  $i$  to  $j$  in one step

### Continuous-Time Markov Chains (CTMCs)

A system can transition between states at **any point in time**

- ▶ Time spent in a state is a **continuous random variable**
- ▶ The “holding time” in any state  $i$  follows an **exponential distribution** parameterised by an *exit rate*  $q_i$
- ▶ This is a direct consequence of the Markov Property being applied to continuous time (exponential is only continuous distribution that is “memoryless”)

## Transition rates

Dynamics of a CTMC defined by **transition rates**, not probabilities

### Definition 2 (Transition Rates)

For two states  $i \neq j$ , the rate  $q_{ij} \geq 0$  is the instantaneous rate of transition from state  $i$  to state  $j$

- ▶ For a small time interval  $\Delta t$ , the probability of transitioning from  $i$  to  $j$  is approximately  $q_{ij}\Delta t$
- ▶ Total **exit rate** from state  $i$  is  $q_i = \sum_{j \neq i} q_{ij}$
- ▶ Time spent in state  $i$  is an exponential random variable  $T_i \sim E(q_i)$

# The generator matrix

## Generator matrix (Q-Matrix)

Collect all transition rates into a single matrix  $Q$

- ▶ Off-diagonal:  $Q_{ij} = q_{ij}$  for  $i \neq j$  (The rate of going from  $i$  to  $j$ )
- ▶ Diagonal:  $Q_{ii} = -q_i = -\sum_{j \neq i} q_{ij}$ . (The negative of the total exit rate from  $i$ )

A key property is that all rows of  $Q$  sum to zero:  $\sum_j Q_{ij} = 0$ .

# Kolmogorov equations

Let  $P(t)$  be the matrix where  $P_{ij}(t) = \mathbb{P}(X(t) = j | X(0) = i)$ . How does  $P(t)$  evolve over time?

## Kolmogorov forward equations

Describes rate of change of probability of ending up in a target state  $j$

$$\frac{d}{dt} P(t) = P(t)Q$$

In element form:

$$P'_{ij}(t) = \sum_k P_{ik}(t)Q_{kj}$$

## Solution to the KFE

The solution is the **matrix exponential**

$$P(t) = e^{tQ} = \sum_{k=0}^{\infty} \frac{(tQ)^k}{k!}$$

The generator matrix  $Q$  “generates” the process’s evolution

# Continuous time Markov chains

Continuous time Markov chains

ODE and CTMC

Simulating CTMC (in theory)

Simulating CTMC (in practice)

Parallelising your code in R

Example – Stochastic phase of an epidemic

Value of travel control measures

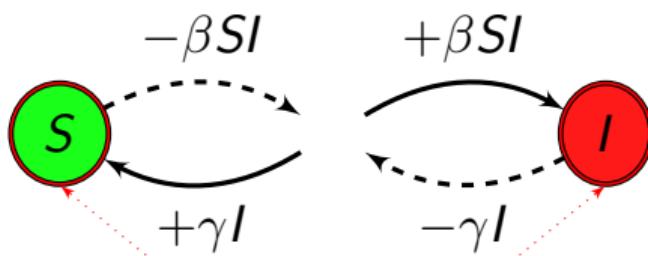
## Converting your compartmental ODE model to CTMC

Easy as  $\pi$  :)

- ▶ Compartmental ODE model focuses on flows into and out of compartments
- ▶ ODE model has as many equations as there are compartments
- ▶ Compartmental CTMC model focuses on transitions
- ▶ CTMC model has as many transitions as there are arrows between (or into or out of) compartments

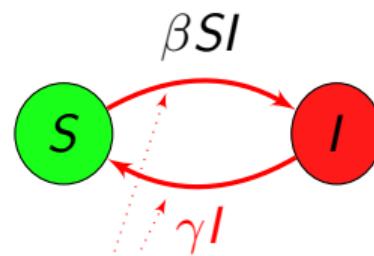
## ODE to CTMC : focus on different components

ODE



focus

CTMC



focus

## SIS without demography

Transition	Effect	Weight	Probability
$S \rightarrow S - 1, I \rightarrow I + 1$	new infection	$\beta SI$	$\frac{\beta SI}{\beta SI + \gamma I}$
$S \rightarrow S + 1, I \rightarrow I - 1$	recovery of an infectious	$\gamma I$	$\frac{\gamma I}{\beta SI + \gamma I}$

States are  $S, I$

## SIS with demography

Transition	Effect	Weight	Probability
$S \rightarrow S + 1$	birth of a susceptible	$b$	$\frac{b}{b+d(S+I)+\beta SI+\gamma I}$
$S \rightarrow S - 1$	death of a susceptible	$dS$	$\frac{dS}{b+d(S+I)+\beta SI+\gamma I}$
$S \rightarrow S - 1, I \rightarrow I + 1$	new infection	$\beta SI$	$\frac{\beta SI}{b+d(S+I)+\beta SI+\gamma I}$
$I \rightarrow I - 1$	death of an infectious	$dI$	$\frac{dI}{b+d(S+I)+\beta SI+\gamma I}$
$S \rightarrow S + 1, I \rightarrow I - 1$	recovery of an infectious	$\gamma I$	$\frac{\gamma I}{b+d(S+I)+\beta SI+\gamma I}$

States are  $S, I$

## Kermack & McKendrick model

Transition	Effect	Weight	Probability
$S \rightarrow S - 1, I \rightarrow I + 1$	new infection	$\beta SI$	$\frac{\beta SI}{\beta SI + \gamma I}$
$I \rightarrow I - 1, R \rightarrow R + 1$	recovery of an infectious	$\gamma I$	$\frac{\gamma I}{\beta SI + \gamma I}$

States are  $S, I, R$

# Continuous time Markov chains

Continuous time Markov chains

ODE and CTMC

**Simulating CTMC (in theory)**

Simulating CTMC (in practice)

Parallelising your code in R

Example – Stochastic phase of an epidemic

Value of travel control measures

## Gillespie's algorithm

- ▶ A.k.a. the stochastic simulation algorithm (SSA)
- ▶ Derived in 1976 by Daniel Gillespie
- ▶ Generates possible solutions for CTMC
- ▶ Extremely simple, so worth learning how to implement; there are however packages that you can use (see later)

## Gillespie's algorithm

Suppose system has state  $\mathbf{x}(t)$  with initial condition  $\mathbf{x}(t_0) = \mathbf{x}_0$  and *propensity functions*  $a_i$  of elementary reactions

set  $t \leftarrow t_0$  and  $\mathbf{x}(t) \leftarrow \mathbf{x}_0$

while  $t \leq t_f$

- $\xi_t \leftarrow \sum_j a_j(\mathbf{x}(t))$
- Draw  $\tau_t$  from  $T \sim \mathcal{E}(\xi_t)$
- Draw  $\zeta_t$  from  $\mathcal{U}([0, 1])$
- Find  $r$ , smallest integer s.t.  $\sum_{k=1}^j a_k(\mathbf{x}(t)) > \zeta_t \sum_j a_j(\mathbf{x}(t)) = \zeta_t \xi_t$
- Effect the next reaction (the one indexed  $r$ )
- $t \leftarrow t + \tau_t$

## Drawing at random from an exponential distribution

If you do not have an exponential distribution random number generator.. We want  $\tau_t$  from  $T \sim \mathcal{E}(\xi_t)$ , i.e.,  $T$  has probability density function

$$f(x, \xi_t) = \xi_t e^{-\xi_t x} \mathbf{1}_{x \geq 0}$$

Use cumulative distribution function  $F(x, \xi_t) = \int_{-\infty}^x f(s, \xi_t) ds$

$$F(x, \xi_t) = (1 - e^{-\xi_t x}) \mathbf{1}_{x \geq 0}$$

which has values in  $[0, 1]$ . So draw  $\zeta$  from  $\mathcal{U}([0, 1])$  and solve  $F(x, \xi_t) = \zeta$  for  $x$

$$\begin{aligned} F(x, \xi_t) = \zeta &\Leftrightarrow 1 - e^{-\xi_t x} = \zeta \\ &\Leftrightarrow e^{-\xi_t x} = 1 - \zeta \\ &\Leftrightarrow \xi_t x = -\ln(1 - \zeta) \\ &\Leftrightarrow x = \boxed{\frac{-\ln(1 - \zeta)}{\xi_t}} \end{aligned}$$

## Gillespie's algorithm (SIS model with only 1 eq.)

set  $t \leftarrow t_0$  and  $I(t) \leftarrow I(t_0)$

while  $t \leq t_f$

- $\xi_t \leftarrow \beta(P^* - i)i + \gamma i$
- Draw  $\tau_t$  from  $T \sim \mathcal{E}(\xi_t)$
- $v \leftarrow [\beta(P^* - i)i, \xi_t] / \xi_t$
- Draw  $\zeta_t$  from  $\mathcal{U}(0, 1)$
- Find  $pos$  such that  $v_{pos-1} \leq \zeta_t \leq v_{pos}$
- switch  $pos$ 
  - 1: New infection,  $I(t + \tau_t) = I(t) + 1$
  - 2: End of infectious period,  $I(t + \tau_t) = I(t) - 1$
- $t \leftarrow t + \tau_t$

## Sometimes Gillespie goes bad

- ▶ Recall that the inter-event time is exponentially distributed
- ▶ Critical step of the Gillespie algorithm:
  - ▶  $\xi_t \leftarrow$  weight of all possible events (*propensity*)
  - ▶ Draw  $\tau_t$  from  $T \sim \mathcal{E}(\xi_t)$
- ▶ So the inter-event time  $\tau_t \rightarrow 0$  if  $\xi_t$  becomes very large for some  $t$
- ▶ This can cause the simulation to grind to a halt

## Example: a birth and death process

- ▶ Individuals born at *per capita* rate  $b$
- ▶ Individuals die at *per capita* rate  $d$
- ▶ Let's implement this using classic Gillespie

(See `simulate_birth_death_CTMC.R` on course GitHub repo)

## Gillespie's algorithm (birth-death model)

set  $t \leftarrow t_0$  and  $N(t) \leftarrow N(t_0)$

while  $t \leq t_f$

- $\xi_t \leftarrow (b + d)N(t)$
- Draw  $\tau_t$  from  $T \sim \mathcal{E}(\xi_t)$
- $v \leftarrow [bN(t), \xi_t] / \xi_t$
- Draw  $\zeta_t$  from  $\mathcal{U}(0, 1)$
- Find  $pos$  such that  $v_{pos-1} \leq \zeta_t \leq v_{pos}$
- switch  $pos$ 
  - 1: Birth,  $N(t + \tau_t) = N(t) + 1$
  - 2: Death,  $N(t + \tau_t) = N(t) - 1$
- $t \leftarrow t + \tau_t$

```
birth_death_CTMC = function(b = 0.01, d = 0.01) {
  t_0 = 0      # Initial time
  N_0 = 100    # Initial population

  # Vectors to store time and state. Initialise with initial condition.
  t = t_0
  N = N_0

  t_f = 1000  # Final time

  # Track the current time and state (could just check last entry in t
  # and N, but will take more operations)
  t_curr = t_0
  N_curr = N_0
  while (t_curr<=t_f) {
    xi_t = (b+d)*N_curr
```

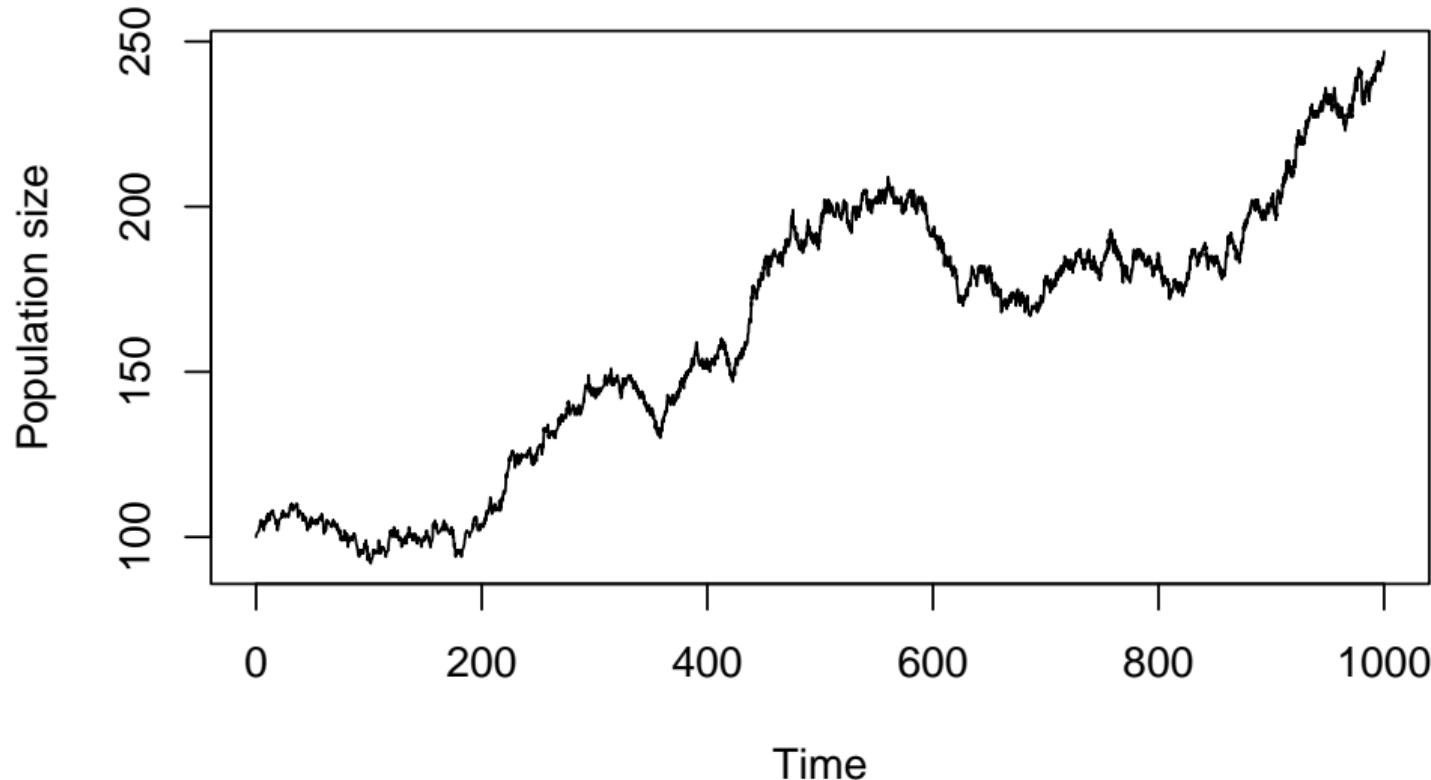
```

if (N_curr == 0) {
  break # Avoid error with rexp when xi_t = 0
}
tau_t = rexp(1, rate = xi_t)
t_curr = t_curr+tau_t
v = c(b*N_curr, xi_t)/xi_t
zeta_t = runif(n = 1)
pos = findInterval(zeta_t, v)+1
switch(pos,
       { N_curr = N_curr+1}, # Birth
       { N_curr = N_curr-1}) # Death
N = c(N, N_curr)
t = c(t, t_curr)
}
plot(t, N, type = "l",
      xlab = "Time", ylab = "Population size",
      main = paste("Birth-death CTMC with b =", b, "and d =", d))

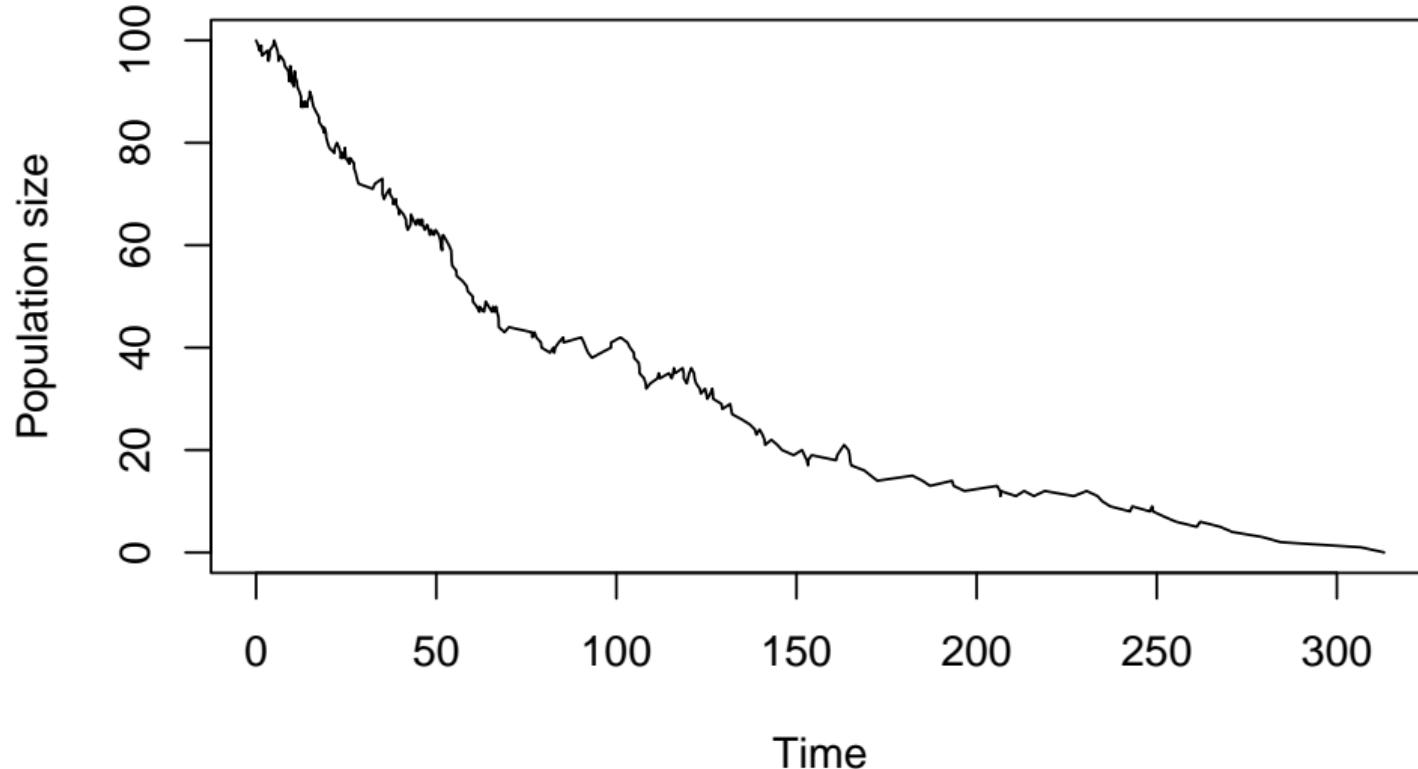
```

}

## Birth-death CTMC with $b = 0.01$ and $d = 0.01$



## Birth-death CTMC with $b = 0.01$ and $d = 0.02$

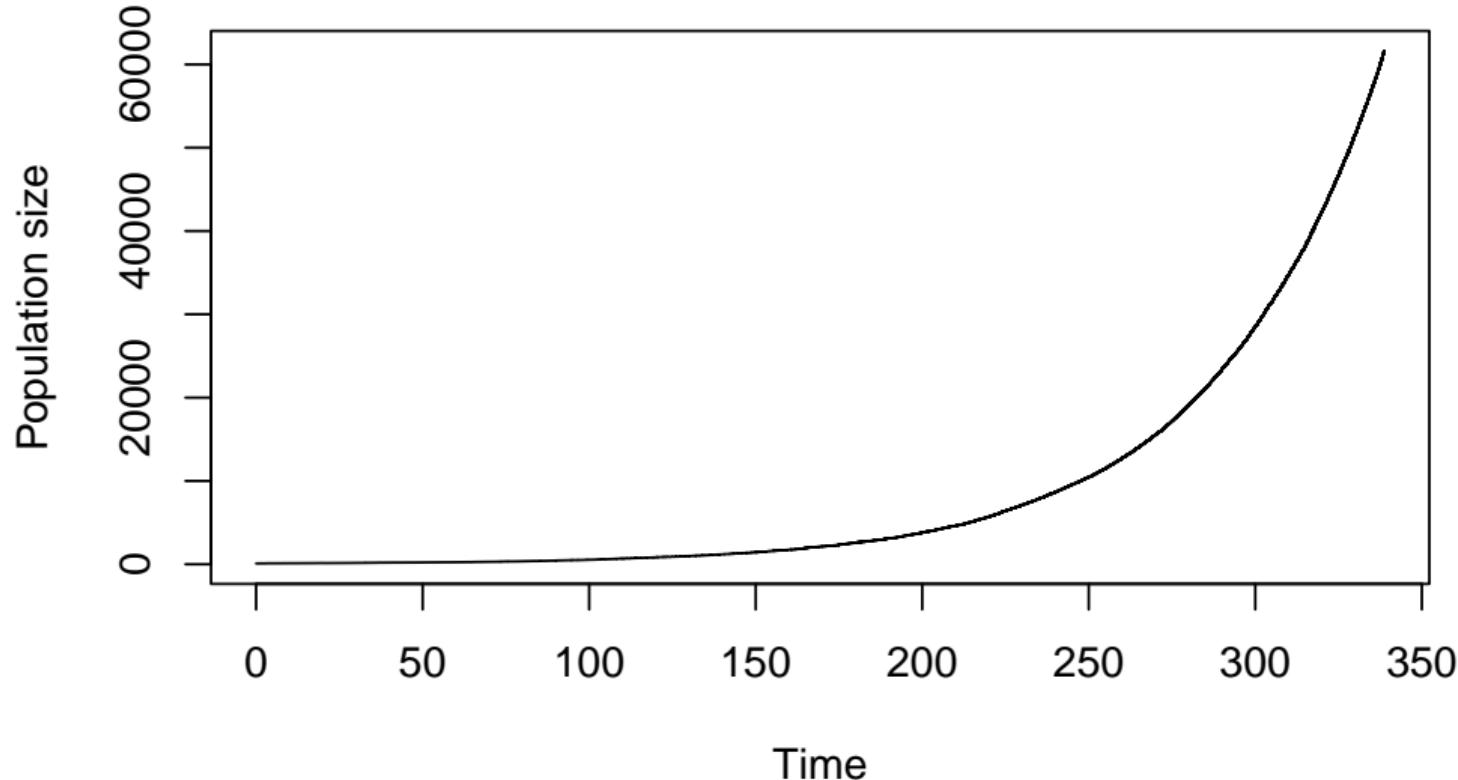


$$b = 0.03 \text{ & } d = 0.01\dots$$

We want to run the function with these parameter values but I know in advance this will not work well, so let's tweak the function a bit. We add a test:

```
if (t[length(t)]-t[(length(t)-1)] < 1e-8) {  
    # If the time step is too small, stop the simulation  
    message("Stopping simulation because time step is too small")  
    break  
}
```

## Birth-death CTMC with $b = 0.03$ and $d = 0.01$



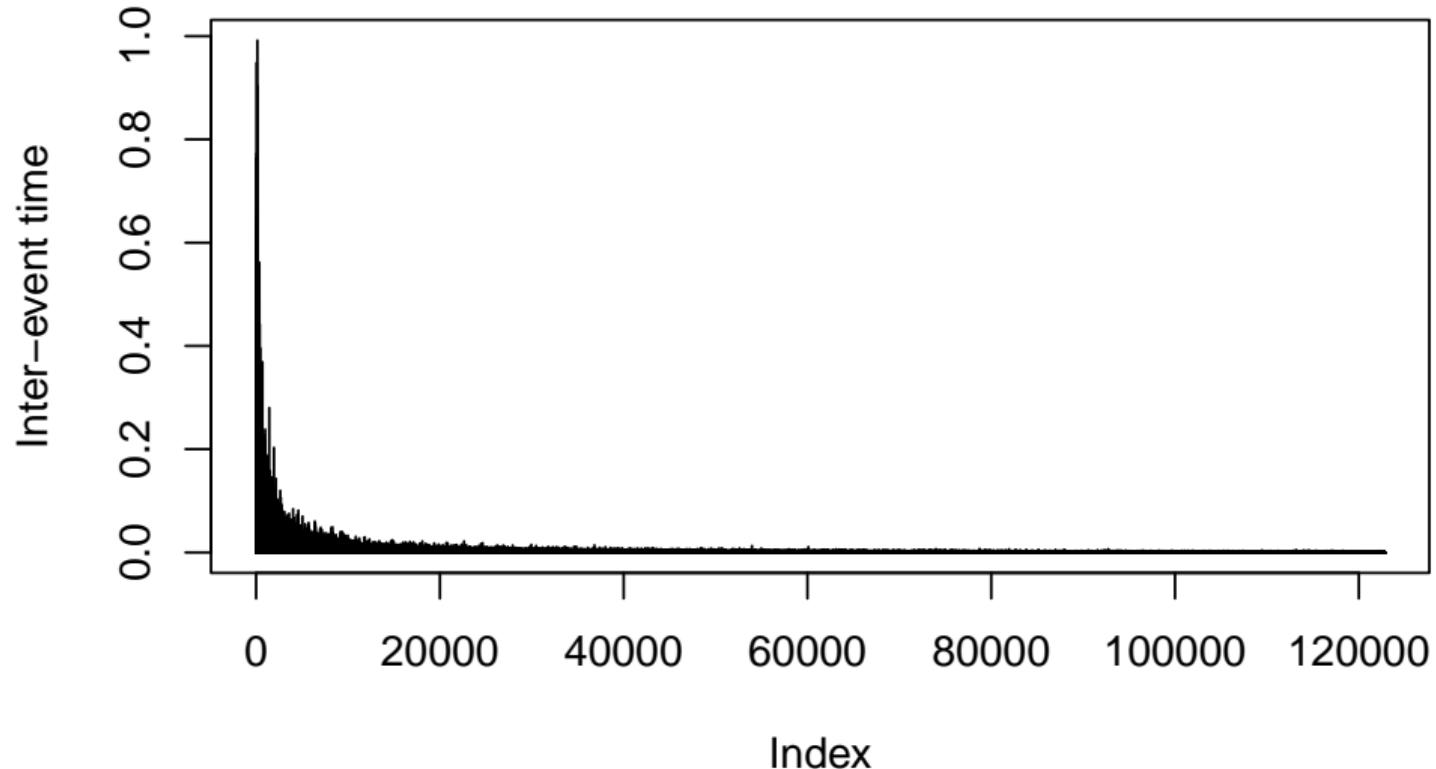
## Last one did not go well

- ▶ Wanted 1000 time units (days?)
- ▶ Interrupted at 338.7126862 because of the test  
(Slide with  $b < d$ : sim stopped because the population went extinct, I did not stop it!)
- ▶ At stop time
  - ▶  $N = 6.1564 \times 10^4$
  - ▶  $|N| = 122767$  (and  $|t|$  as well, of course!)
  - ▶ time was moving slowly

```
tail(diff(results$t))
```

```
## [1] 1.032524e-04 1.501491e-04 1.283565e-03 7.772778e-04 2.509549e-04
## [6] 6.210030e-09
```

# Inter-event time for birth-death CTMC with $b=0.03$ and $d=0.01$



# Continuous time Markov chains

Continuous time Markov chains

ODE and CTMC

Simulating CTMC (in theory)

**Simulating CTMC (in practice)**

Parallelising your code in R

Example – Stochastic phase of an epidemic

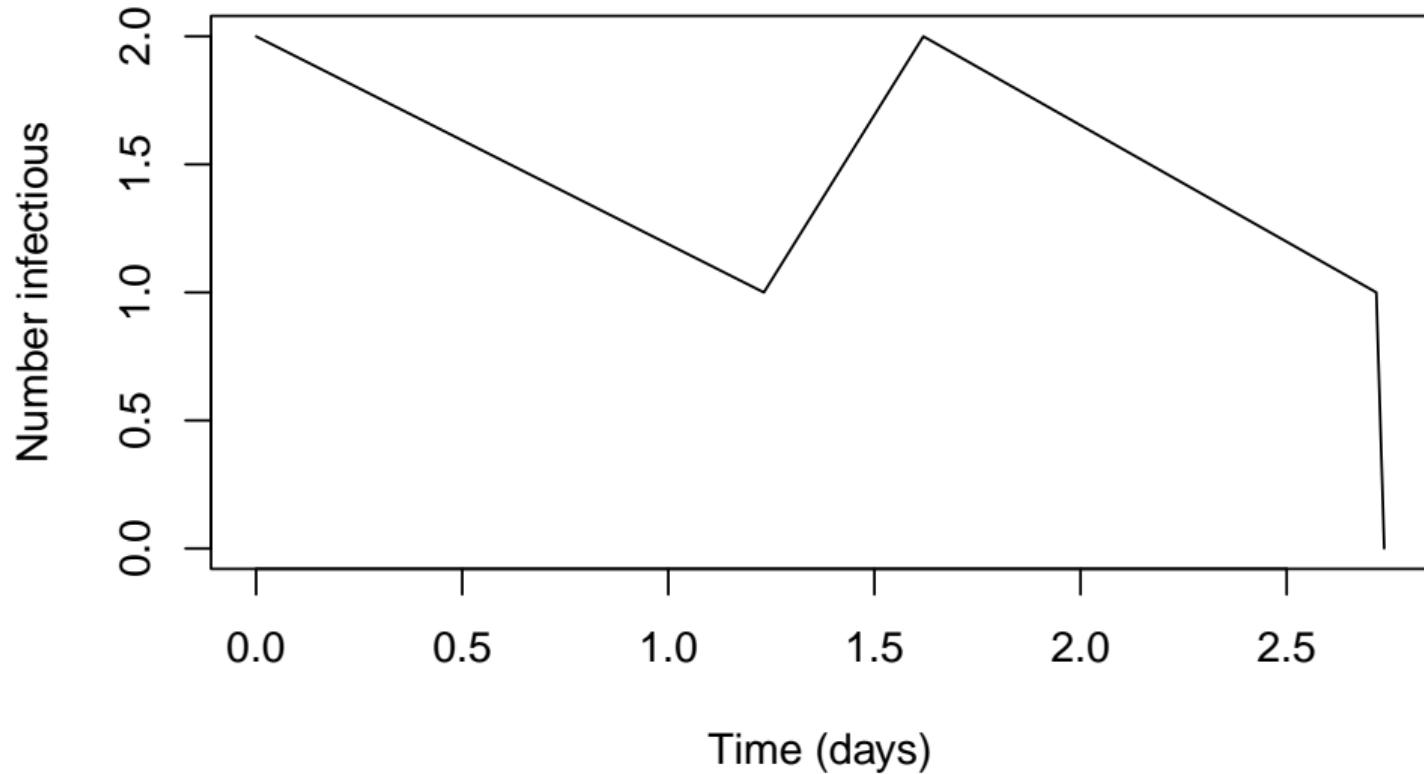
Value of travel control measures

## Tau-leaping (and packages) to the rescue!

- ▶ *Approximation* method (compared to classic Gillespie, which is exact)
- ▶ Roughly: consider "groups" of events instead of individual events
- ▶ Good news: GillespieSSA2 and adaptivetau, two standard packages for SSA in R, implement tau leaping

```
library(GillespieSSA2)
Pop <- 1000
I_0 <- 2
IC <- c(S = (Pop-I_0), I = I_0)
gamma = 1/3
#  $R_0 = \beta/\gamma * S_0$ , so  $\beta = R_0 * \gamma / S_0$ 
beta = as.numeric(1.5*gamma/IC["S"])
params <- c(gamma = gamma, beta = beta)
t_f = 100
reactions <- list(
  reaction("beta*S*I", c(S=-1,I=+1), "new_infection"),
  reaction("gamma*I", c(S=+1,I=-1), "recovery")
)
set.seed(NULL)
```

```
sol <- ssa(  
    initial_state = IC,  
    reactions = reactions,  
    params = params,  
    method = ssa_exact(),  
    final_time = t_f,  
)  
plot(sol$time, sol$state[, "I"], type = "l",  
     xlab = "Time (days)", ylab = "Number infectious")
```



# Continuous time Markov chains

Continuous time Markov chains

ODE and CTMC

Simulating CTMC (in theory)

Simulating CTMC (in practice)

Parallelising your code in R

Example – Stochastic phase of an epidemic

Value of travel control measures

## Parallelisation

To see multiple realisations: good idea to parallelise, then interpolate results. Write a function, e.g., `run_one_sim` that .. runs one simulation

On the GitHub repo for the course, see

- ▶ `SIS-CTMC-parallel.R`
- ▶ `SIS-CTMC-parallel-multiple-R0.R`

```
library(parallel)
run_one_sim = function(params) {
  IC <- c(S = (params$Pop - params$I_0), I = params$I_0)
  params_local <- c(gamma = params$gamma, beta = params$beta)
  reactions <- list(
    # propensity function effects name for reaction
    reaction("beta*S*I", c(S=-1, I=+1), "new_infection"),
    reaction("gamma*I", c(S=+1, I=-1), "recovery")
  )
  set.seed(NULL)
  sol <- ssa(
    initial_state = IC,
    reactions = reactions,
    params = params_local,
    method = ssa_exact(),
    final_time = params$t_f,
```

```
    log_firings = TRUE      # This way we keep track of events
)
# Interpolate result (just I will do)
wanted_t = seq(from = 0, to = params$t_f, by = 0.01)
sol$interp_I = approx(x = sol$time, y = sol$state[, "I"], xout = wanted_t)
names(sol$interp_I) = c("time", "I")
# Return result
return(sol)
}
nb_cores <- detectCores()
if (nb_cores > 124) {
  nb_cores = 124
}
cl <- makeCluster(nb_cores)
clusterEvalQ(cl, {
  library(GillespieSSA2)
})
```

```
## [[1]]
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"
## [5] "utils"          "datasets"        "methods"       "base"
##
## [[2]]
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"
## [5] "utils"          "datasets"        "methods"       "base"
##
## [[3]]
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"
## [5] "utils"          "datasets"        "methods"       "base"
##
## [[4]]
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"
## [5] "utils"          "datasets"        "methods"       "base"
##
## [[5]]
```

```
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"
## [5] "utils"          "datasets"        "methods"       "base"
##
## [[6]]
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"
## [5] "utils"          "datasets"        "methods"       "base"
##
## [[7]]
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"
## [5] "utils"          "datasets"        "methods"       "base"
##
## [[8]]
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"
## [5] "utils"          "datasets"        "methods"       "base"
##
## [[9]]
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"
```

```
## [5] "utils"           "datasets"        "methods"        "base"
##
## [[10]]
## [1] "GillespieSSA2" "stats"          "graphics"       "grDevices"
## [5] "utils"           "datasets"        "methods"        "base"
##
## [[11]]
## [1] "GillespieSSA2" "stats"          "graphics"       "grDevices"
## [5] "utils"           "datasets"        "methods"        "base"
##
## [[12]]
## [1] "GillespieSSA2" "stats"          "graphics"       "grDevices"
## [5] "utils"           "datasets"        "methods"        "base"
##
## [[13]]
## [1] "GillespieSSA2" "stats"          "graphics"       "grDevices"
## [5] "utils"           "datasets"        "methods"        "base"
```

```
##  
## [[14]]  
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"  
## [5] "utils"          "datasets"        "methods"       "base"  
##  
## [[15]]  
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"  
## [5] "utils"          "datasets"        "methods"       "base"  
##  
## [[16]]  
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"  
## [5] "utils"          "datasets"        "methods"       "base"  
##  
## [[17]]  
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"  
## [5] "utils"          "datasets"        "methods"       "base"  
##
```

```
## [[18]]
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"
## [5] "utils"          "datasets"        "methods"       "base"
##
## [[19]]
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"
## [5] "utils"          "datasets"        "methods"       "base"
##
## [[20]]
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"
## [5] "utils"          "datasets"        "methods"       "base"
##
## [[21]]
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"
## [5] "utils"          "datasets"        "methods"       "base"
##
## [[22]]
```

```
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"
## [5] "utils"          "datasets"        "methods"       "base"
##
## [[23]]
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"
## [5] "utils"          "datasets"        "methods"       "base"
##
## [[24]]
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"
## [5] "utils"          "datasets"        "methods"       "base"
##
## [[25]]
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"
## [5] "utils"          "datasets"        "methods"       "base"
##
## [[26]]
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"
```

```
## [5] "utils"           "datasets"        "methods"        "base"
##
## [[27]]
## [1] "GillespieSSA2" "stats"          "graphics"       "grDevices"
## [5] "utils"           "datasets"        "methods"        "base"
##
## [[28]]
## [1] "GillespieSSA2" "stats"          "graphics"       "grDevices"
## [5] "utils"           "datasets"        "methods"        "base"
##
## [[29]]
## [1] "GillespieSSA2" "stats"          "graphics"       "grDevices"
## [5] "utils"           "datasets"        "methods"        "base"
##
## [[30]]
## [1] "GillespieSSA2" "stats"          "graphics"       "grDevices"
## [5] "utils"           "datasets"        "methods"        "base"
```

```
##  
## [[31]]  
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"  
## [5] "utils"          "datasets"        "methods"       "base"  
##  
## [[32]]  
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"  
## [5] "utils"          "datasets"        "methods"       "base"  
##  
## [[33]]  
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"  
## [5] "utils"          "datasets"        "methods"       "base"  
##  
## [[34]]  
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"  
## [5] "utils"          "datasets"        "methods"       "base"  
##
```

```
## [[35]]
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"
## [5] "utils"          "datasets"        "methods"       "base"
##
## [[36]]
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"
## [5] "utils"          "datasets"        "methods"       "base"
##
## [[37]]
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"
## [5] "utils"          "datasets"        "methods"       "base"
##
## [[38]]
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"
## [5] "utils"          "datasets"        "methods"       "base"
##
## [[39]]
```

```
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"
## [5] "utils"          "datasets"        "methods"       "base"
##
## [[40]]
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"
## [5] "utils"          "datasets"        "methods"       "base"
##
## [[41]]
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"
## [5] "utils"          "datasets"        "methods"       "base"
##
## [[42]]
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"
## [5] "utils"          "datasets"        "methods"       "base"
##
## [[43]]
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"
```

```
## [5] "utils"           "datasets"        "methods"        "base"
##
## [[44]]
## [1] "GillespieSSA2" "stats"          "graphics"       "grDevices"
## [5] "utils"           "datasets"        "methods"        "base"
##
## [[45]]
## [1] "GillespieSSA2" "stats"          "graphics"       "grDevices"
## [5] "utils"           "datasets"        "methods"        "base"
##
## [[46]]
## [1] "GillespieSSA2" "stats"          "graphics"       "grDevices"
## [5] "utils"           "datasets"        "methods"        "base"
##
## [[47]]
## [1] "GillespieSSA2" "stats"          "graphics"       "grDevices"
## [5] "utils"           "datasets"        "methods"        "base"
```

```
##  
## [[48]]  
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"  
## [5] "utils"          "datasets"        "methods"       "base"  
##  
## [[49]]  
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"  
## [5] "utils"          "datasets"        "methods"       "base"  
##  
## [[50]]  
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"  
## [5] "utils"          "datasets"        "methods"       "base"  
##  
## [[51]]  
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"  
## [5] "utils"          "datasets"        "methods"       "base"  
##
```

```
## [[52]]
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"
## [5] "utils"          "datasets"        "methods"       "base"
##
## [[53]]
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"
## [5] "utils"          "datasets"        "methods"       "base"
##
## [[54]]
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"
## [5] "utils"          "datasets"        "methods"       "base"
##
## [[55]]
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"
## [5] "utils"          "datasets"        "methods"       "base"
##
## [[56]]
```

```
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"
## [5] "utils"          "datasets"        "methods"       "base"
##
## [[57]]
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"
## [5] "utils"          "datasets"        "methods"       "base"
##
## [[58]]
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"
## [5] "utils"          "datasets"        "methods"       "base"
##
## [[59]]
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"
## [5] "utils"          "datasets"        "methods"       "base"
##
## [[60]]
## [1] "GillespieSSA2" "stats"           "graphics"      "grDevices"
```

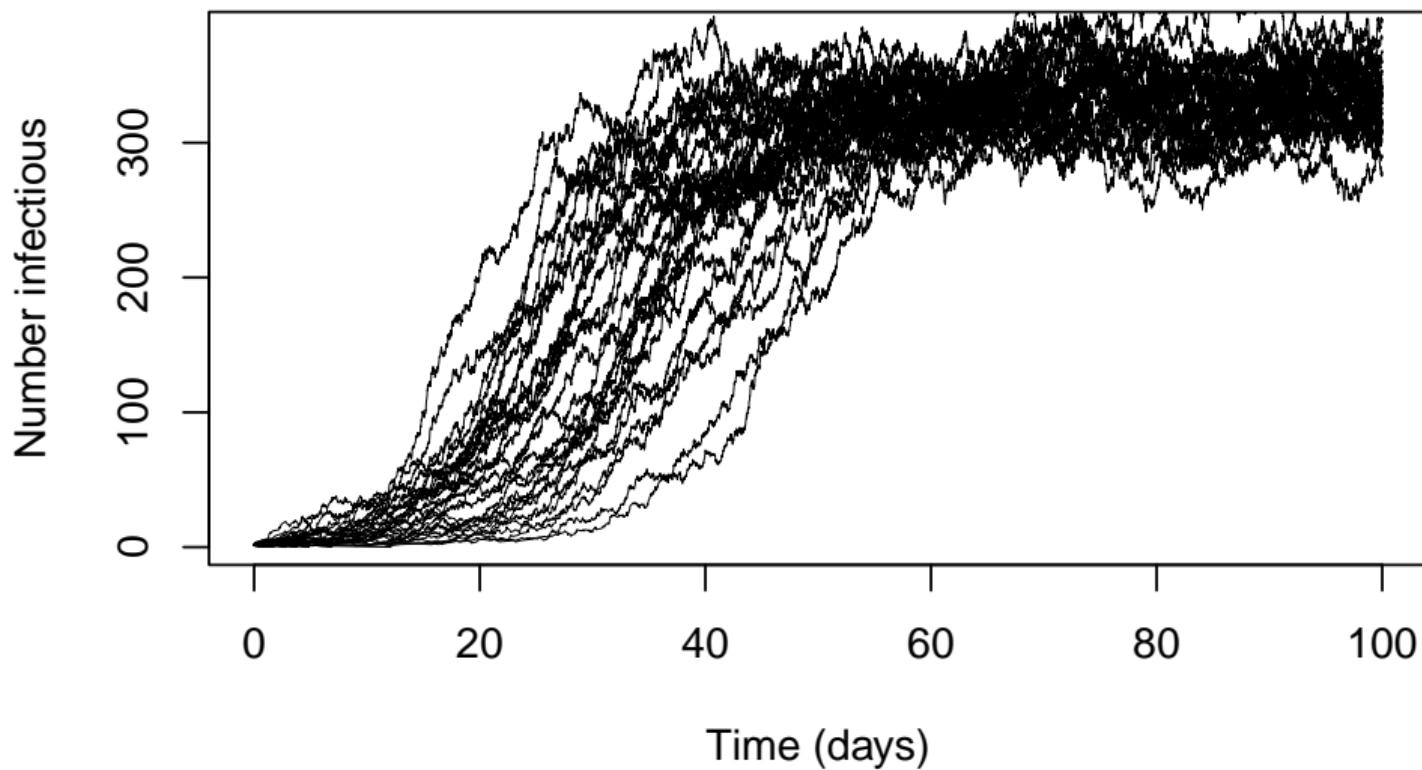
```
## [5] "utils"           "datasets"        "methods"        "base"
##
## [[61]]
## [1] "GillespieSSA2" "stats"          "graphics"       "grDevices"
## [5] "utils"           "datasets"        "methods"        "base"
##
## [[62]]
## [1] "GillespieSSA2" "stats"          "graphics"       "grDevices"
## [5] "utils"           "datasets"        "methods"        "base"
##
## [[63]]
## [1] "GillespieSSA2" "stats"          "graphics"       "grDevices"
## [5] "utils"           "datasets"        "methods"        "base"
##
## [[64]]
## [1] "GillespieSSA2" "stats"          "graphics"       "grDevices"
## [5] "utils"           "datasets"        "methods"        "base"
```

```
gamma = 1/3
# R0=beta/gamma*S0, so beta=R0*gamma/S0
beta = as.numeric(1.5*gamma/IC["S"])
t_f = 100
params <- list(gamma = gamma, beta = beta,
                 Pop = 1000, I_0 = 2,
                 t_f = 100, nb_sims = 50)

clusterExport(cl,
              c("params",
                "run_one_sim"),
              envir = .GlobalEnv)
SIMS = parLapply(cl = cl,
                  X = 1:params$nb_sims,
                  fun = function(x) run_one_sim(params))
stopCluster(cl)
plot(SIMS[[1]]$interp_I$time,
```

```
SIMS[[1]]$interp_I$I,
type = "l", lwd = 0.5,
xlab = "Time (days)",
ylab = "Number infectious",
main = paste("CTMC with R0 =", params$R0))
for (i in 2:length(SIMS)) {
  lines(SIMS[[i]]$interp_I$time,
        SIMS[[i]]$interp_I$I,
        type = "l", lwd = 0.5)
}
```

**CTMC with  $R_0 =$**



## Benefit of parallelisation

Run the parallel code for 100 sims between ‘tictoc::tic()’ and ‘tictoc::toc()’, giving ‘66.958 sec elapsed’, then the sequential version

```
tictoc::tic()  
SIMS = lapply(X = 1:params$number_sims,  
              FUN = function(x) run_one_sim(params))  
tictoc::toc()
```

which gives ‘318.141 sec elapsed’ on a 6C/12T Intel(R) Core(TM) i9-8950HK CPU @ 2.90GHz ( $4.75\times$  faster) or ‘12.067 sec elapsed’ versus ‘258.985 sec elapsed’ on a 32C/64T AMD Ryzen Threadripper 3970X 32-Core Processor ( $21.46\times$  faster !)

# Continuous time Markov chains

Continuous time Markov chains

ODE and CTMC

Simulating CTMC (in theory)

Simulating CTMC (in practice)

Parallelising your code in R

**Example – Stochastic phase of an epidemic**

Value of travel control measures

ORIGINAL ARTICLE

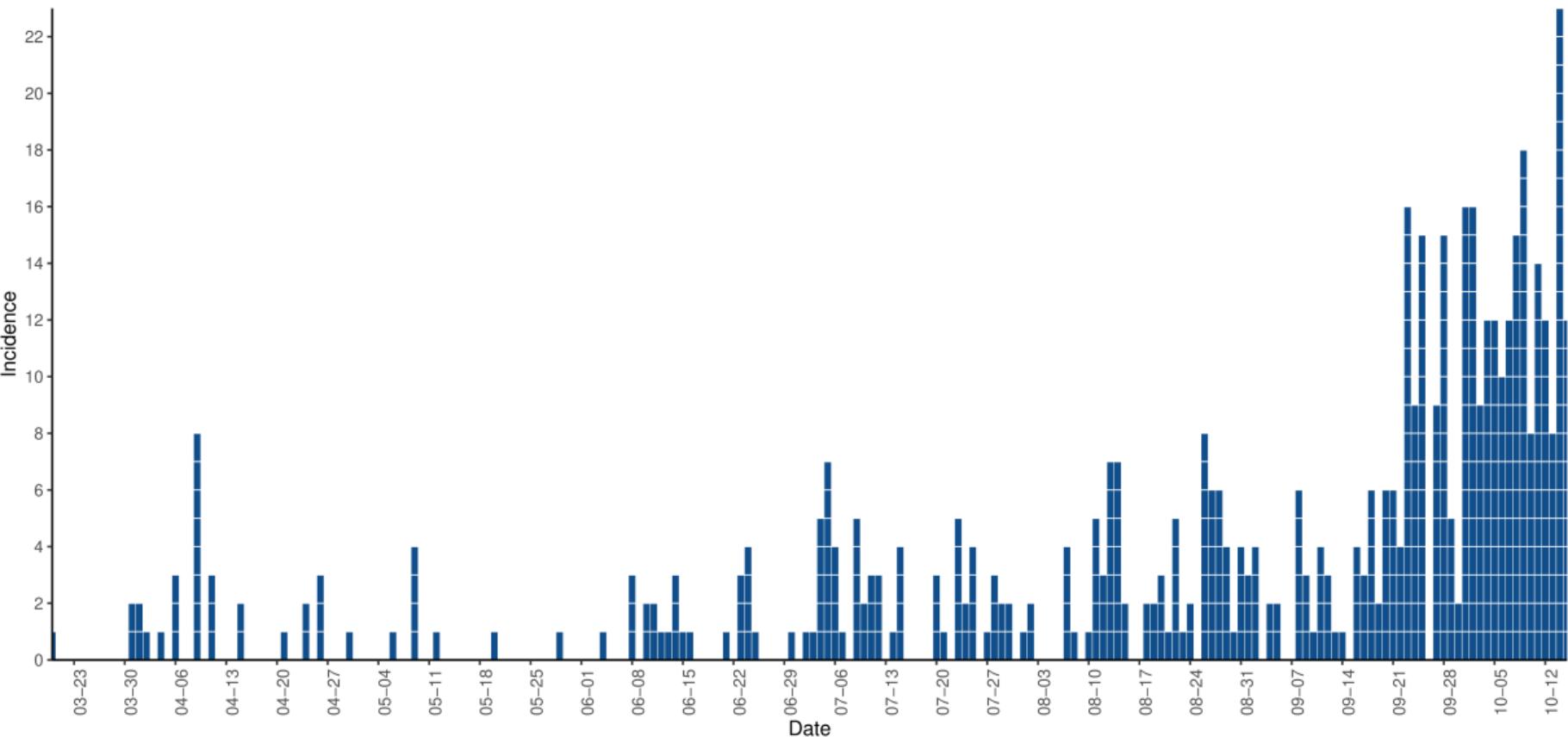


# Effect of Movement on the Early Phase of an Epidemic

Julien Arino<sup>1</sup> · Evan Milliken<sup>2</sup> 

Received: 6 July 2021 / Accepted: 29 August 2022 / Published online: 23 September 2022

© The Author(s), under exclusive licence to Society for Mathematical Biology 2022



## Investigating outbreak types using a simple CTMC SIS

$$\mathbf{X}(t) = (S^A(t), I^A(t))$$

CTMC  $\mathbf{X}(t)$  characterized by transitions

Description	Transition	Rate
Infection	$(S^A, I^A) \rightarrow (S^A - 1, I^A + 1)$	$\beta^A S^A I^A$
Recovery	$(S^A, I^A) \rightarrow (S^A + 1, I^A - 1)$	$\gamma^A I^A$

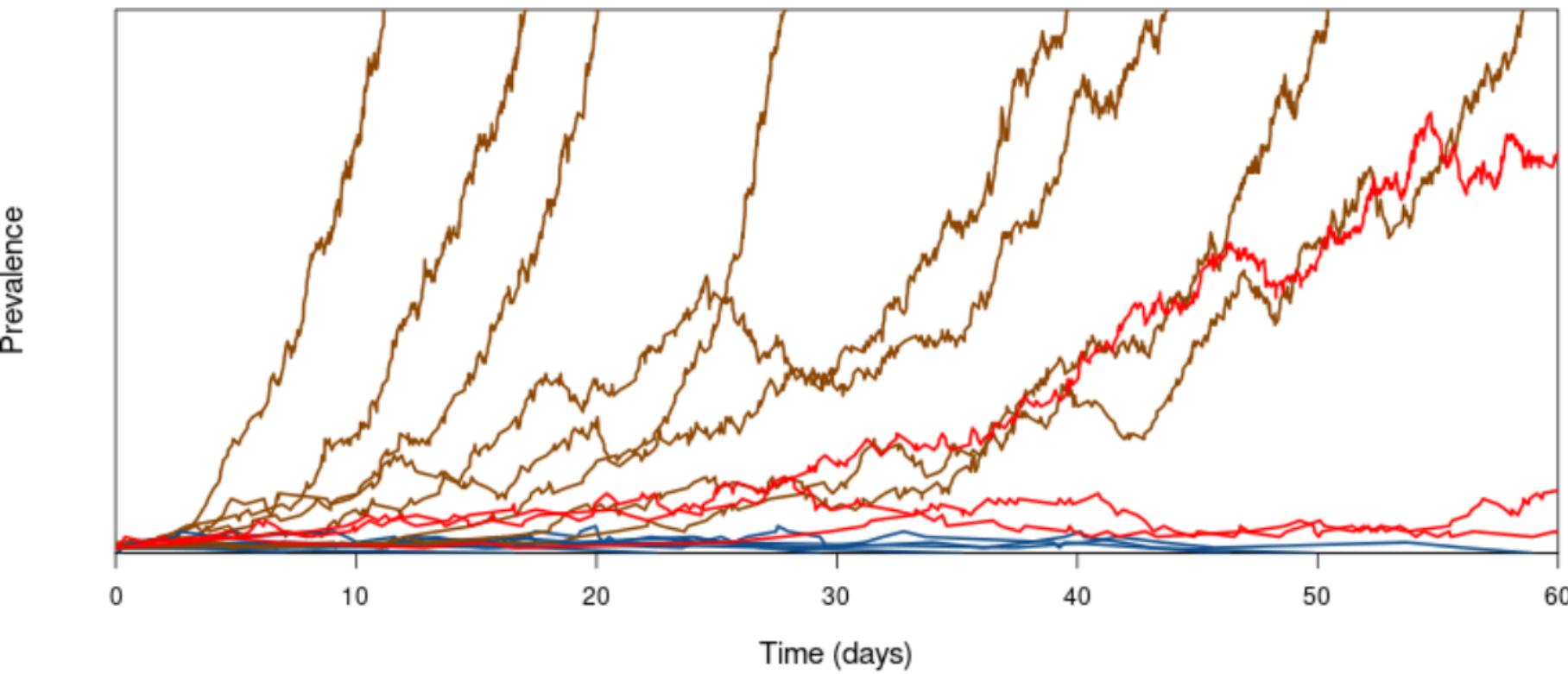
## Investigating outbreak types using a simple CTMC SIS *with a twist*

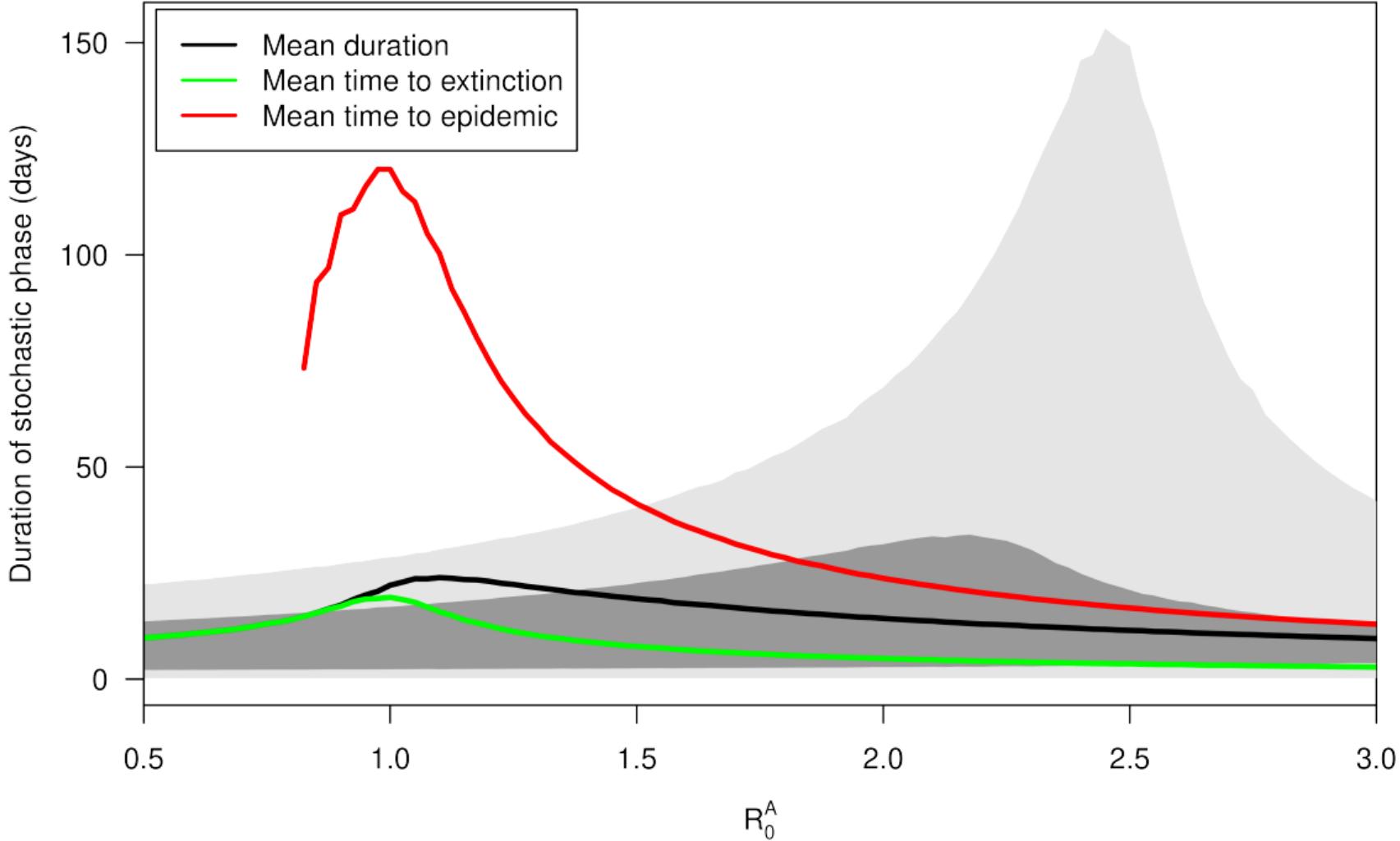
Regular chain of this type has  $I = 0$  as sole absorbing state

We add another absorbing state: if  $I = \hat{I}$ , then the chain has \*left\* the stochastic phase and is in a quasi-deterministic phase with exponential growth

Doing this, time to absorption measures become usable additionally to first passage time ones

And the question becomes: how long does the chain “linger on” (“stutter”) before it is absorbed? We define the inter-absorption trajectory as the stochastic phase





## Problem of the value of the upper bound $\hat{I}$

- ▶ Choose  $\hat{I}$  too small and the stochastic phase will not last long
- ▶ Choose  $\hat{I}$  too large and absorption will only be at the DFE
- ▶ So, how does one choose  $\hat{I}$ ?
  - ▶ A formula of Whittle (1955)
  - ▶ Multitype branching process (MTBP)

# Continuous time Markov chains

Continuous time Markov chains

ODE and CTMC

Simulating CTMC (in theory)

Simulating CTMC (in practice)

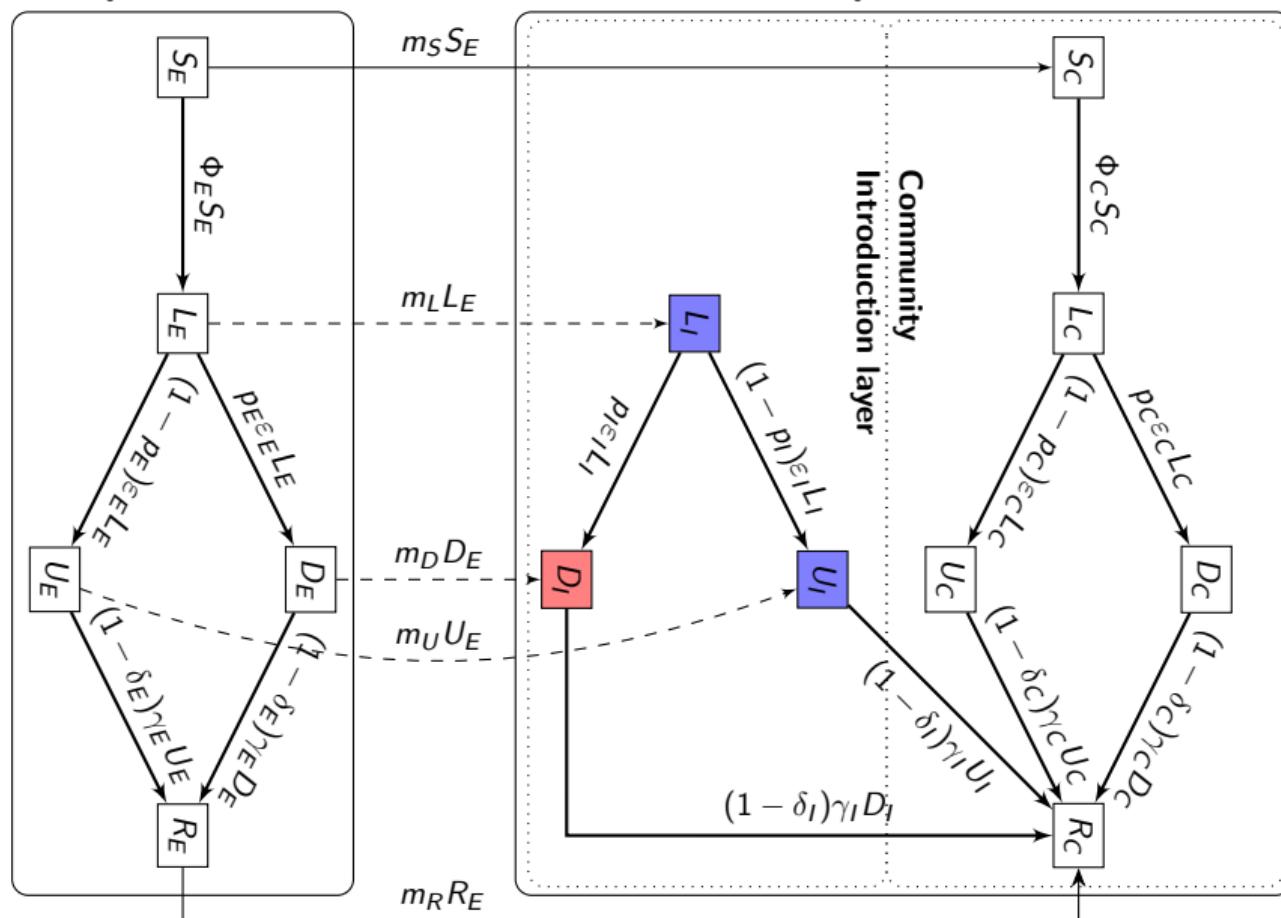
Parallelising your code in R

Example – Stochastic phase of an epidemic

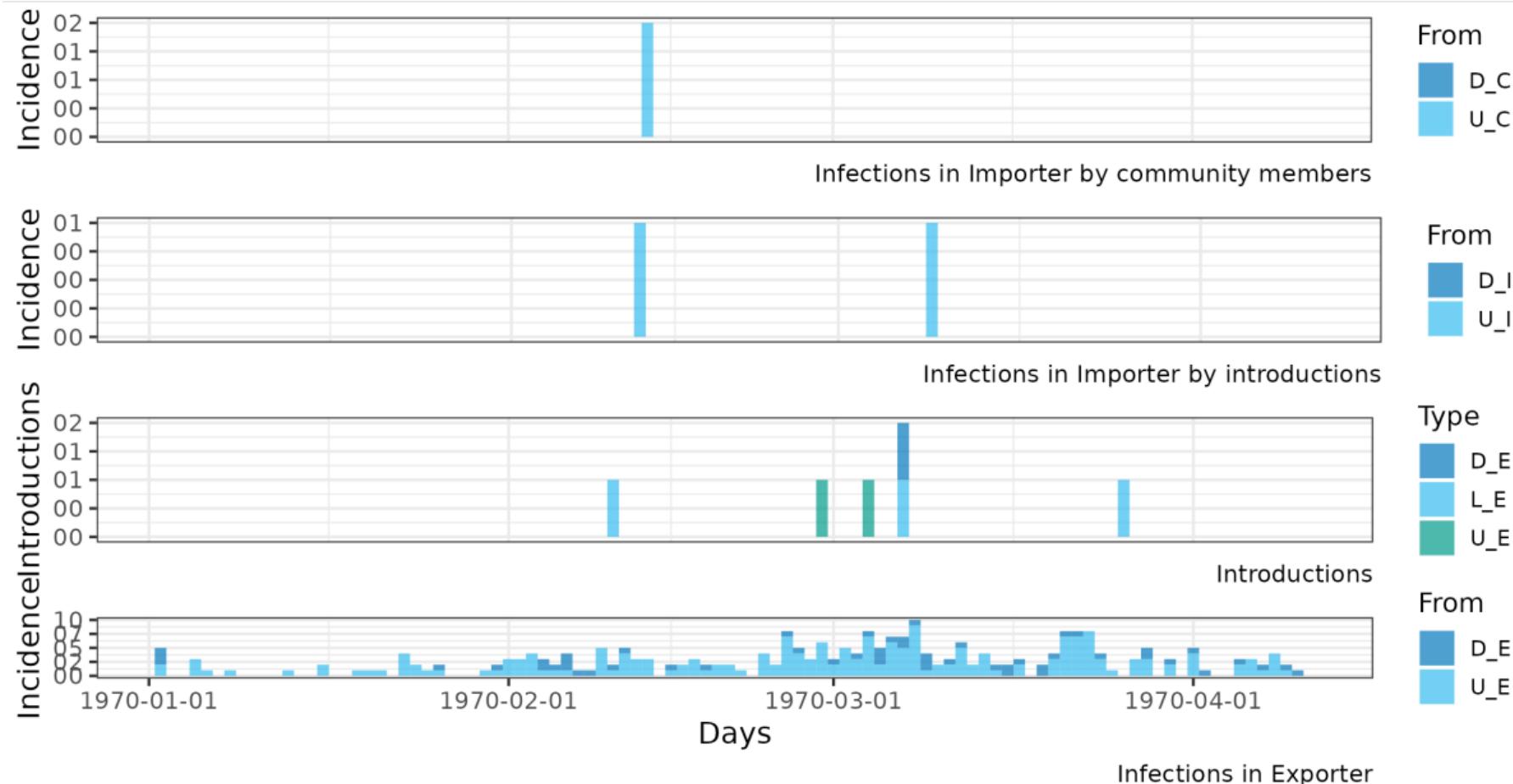
**Value of travel control measures**

# Exporter

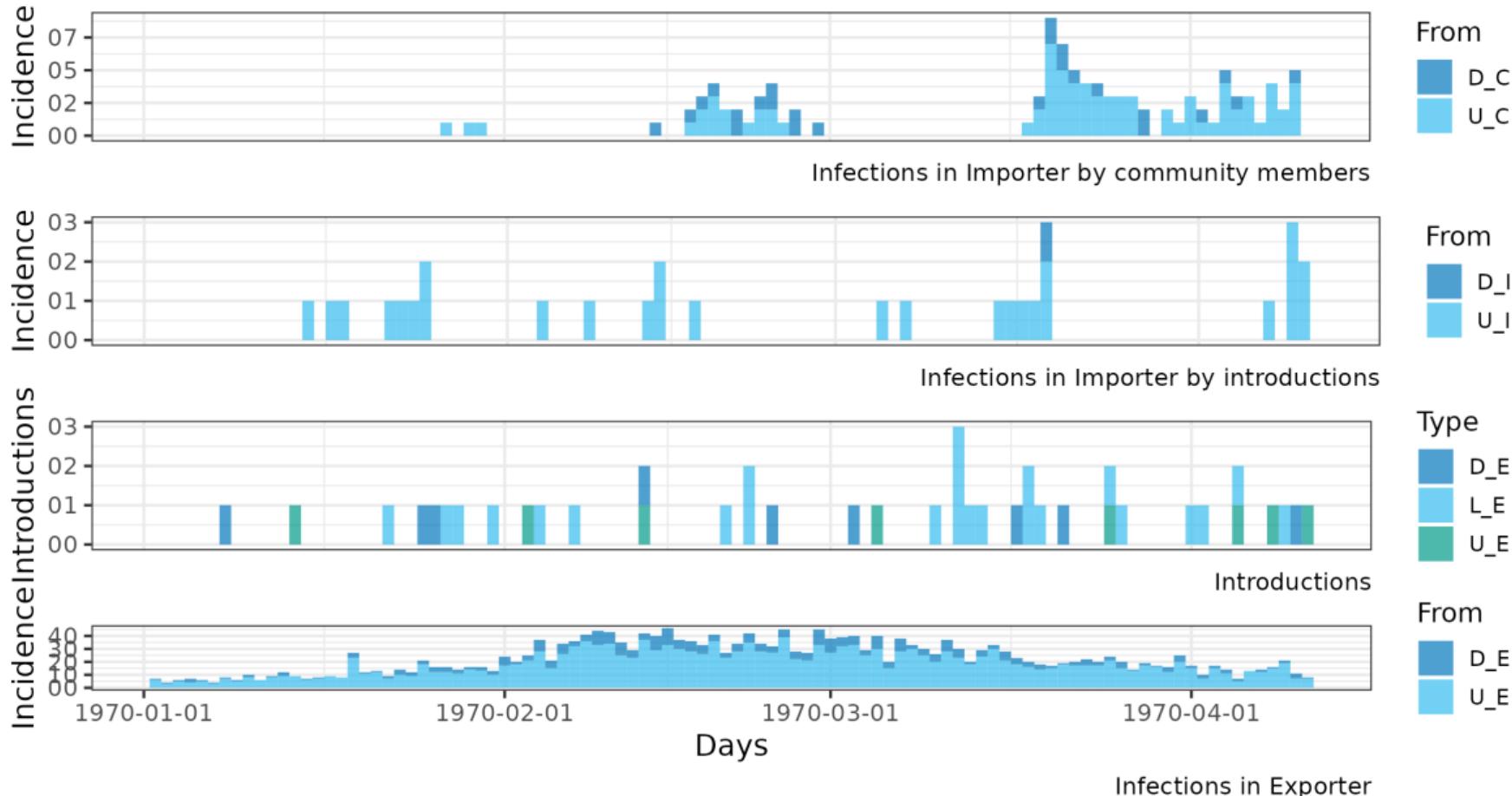
# Importer



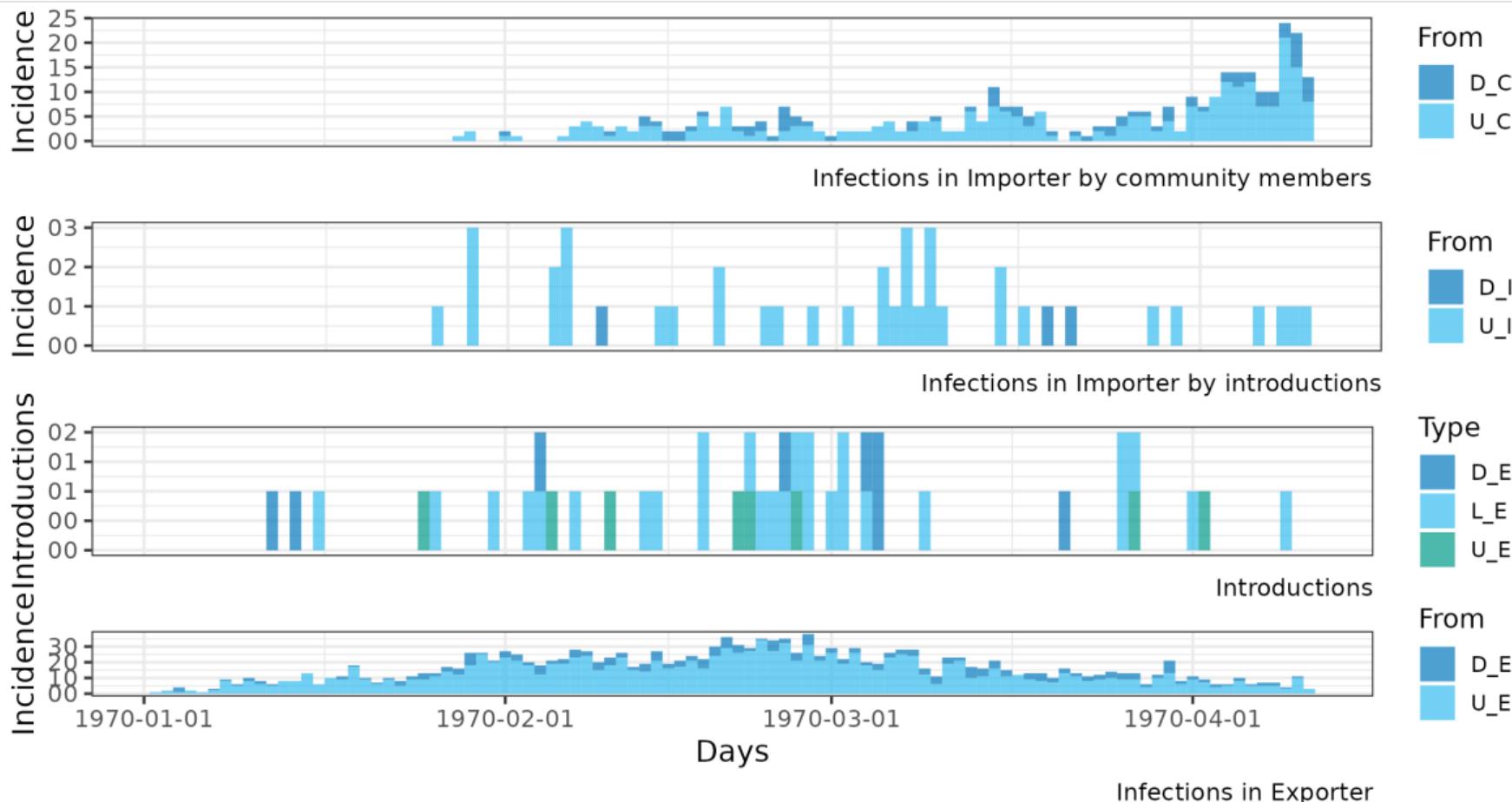
$$R_0^E = 1.5, R_0^C = 0.8, \text{pop}_E = 10000, \text{pop}_I = 10000$$



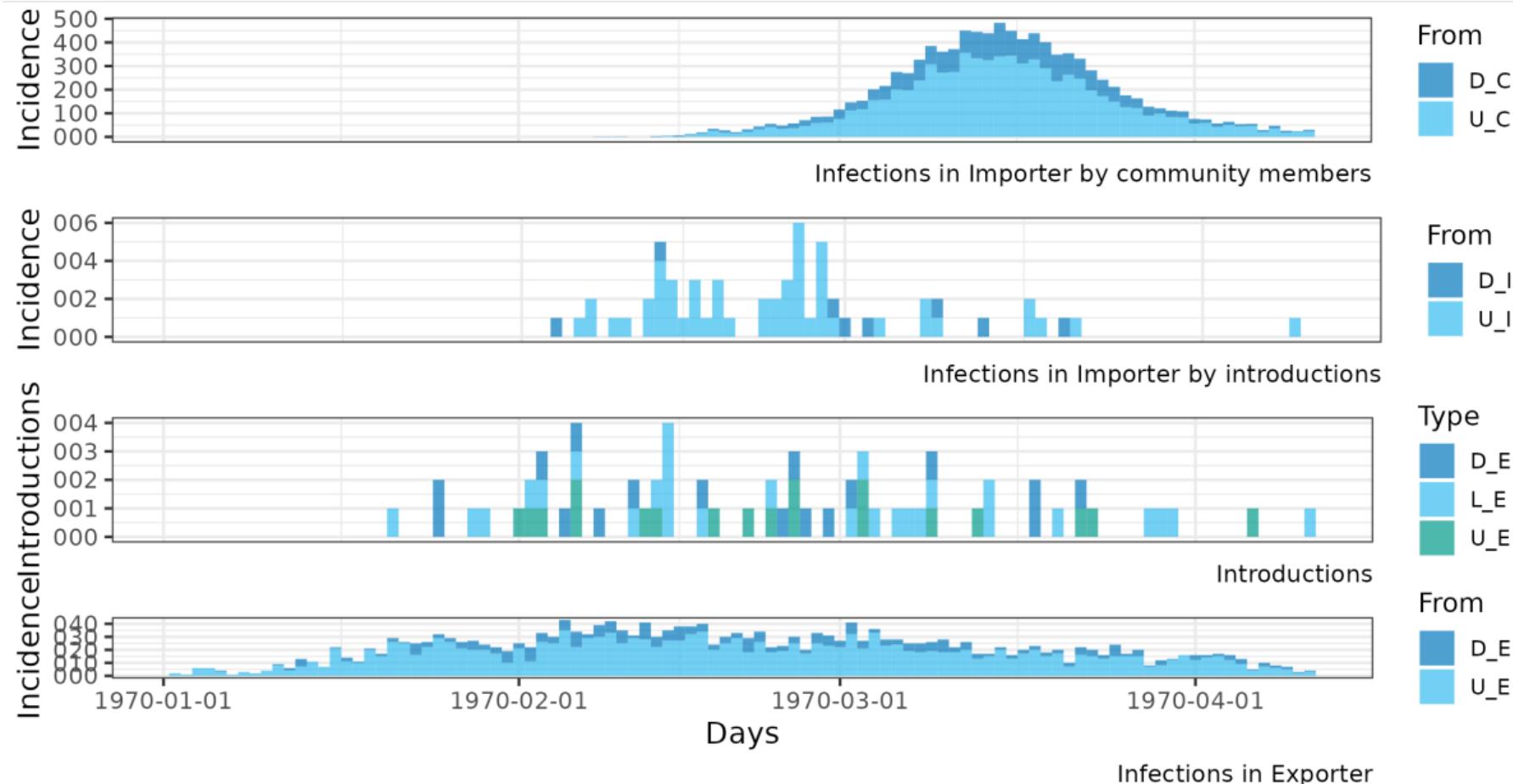
$$R_0^E = 1.5, R_0^C = 0.8, \text{pop}_E = 10000, \text{pop}_I = 10000$$

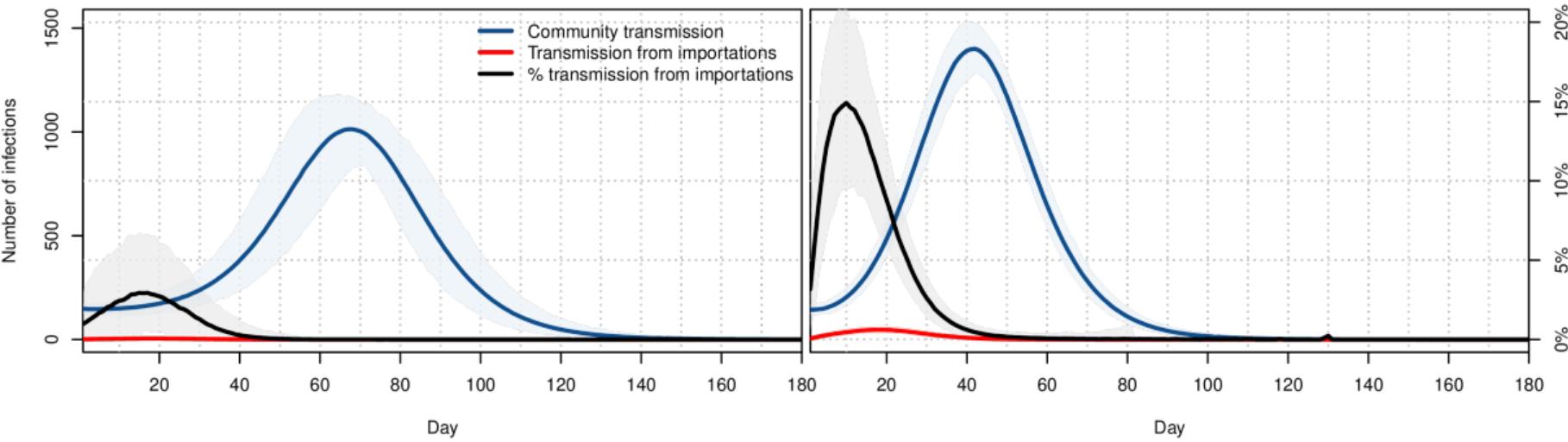


$$R_0^E = 1.5, R_0^C = 0.8, \text{pop}_E = 10000, \text{pop}_I = 10000$$



$$R_0^E = 1.5, R_0^C = 1.5, \text{pop}_E = 10000, \text{pop}_I = 10000$$





## One last thought for the road

V. Chetail. Crisis without borders: What does international law say about border closure in the context of Covid-19? *Frontiers in Political Science*, 2 (12) (2020)

*[...] a powerful expression of state's sovereignty, immigration control provides a typical avenue for governments to reassure their citizens and bolster a national sense of belonging, while providing an ideal scapegoat for their own failure or negligence.*



Why incorporate stochasticity?

Stochasticity in deterministic models

Continuous time Markov chains

Branching process approximations of CTMC

# What is a Branching Process?

## The Core Idea

A branching process is a mathematical model for a population where individuals produce a random number of offspring and then die.

- ▶ Think of bacteria splitting, a virus spreading, or even the survival of family surnames.
- ▶ We start with an initial population,  $Z_0$ .
- ▶ Each individual in generation  $n$  produces a number of offspring for generation  $n + 1$ .
- ▶ This "number of offspring" is a random variable. All individuals produce offspring according to the same probability distribution, independently of each other.

## The Galton-Watson Process

Let  $Z_n$  be the size of the population in generation  $n$ . We typically start with  $Z_0 = 1$ . The population evolves according to the rule:

$$Z_{n+1} = \sum_{i=1}^{Z_n} X_{n,i}$$

- ▶ The term  $X_{n,i}$  represents the number of offspring produced by the  $i$ -th individual in generation  $n$ .
- ▶ The variables  $\{X_{n,i}\}$  are assumed to be **independent and identically distributed (i.i.d.)** integer-valued random variables.
- ▶ We call their common distribution  $\{p_k\}_{k=0}^{\infty}$  the **offspring distribution**, where  $p_k = P(X = k)$ .

### The Fundamental Questions

1. What is the long-term expected size of the population?
2. What is the probability that the population eventually dies out?

## Mean Offspring

The fate of the population hinges on a single parameter: the mean of the offspring distribution

$$\mu = E[X] = \sum_{k=0}^{\infty} k \cdot p_k$$

## Expected Population Size

Using the law of total expectation, we find the expected size of the next generation:

$$E[Z_{n+1}|Z_n] = E\left[\sum_{i=1}^{Z_n} X_{n,i} \middle| Z_n\right] = Z_n E[X] = Z_n \mu$$

Taking the expectation again, we get a simple recurrence:

$$E[Z_{n+1}] = \mu E[Z_n]$$

# The Three Regimes of Population Growth

The behavior of  $E[Z_n] = Z_0\mu^n$  suggests three distinct cases:

## Subcritical ( $\mu < 1$ )

$E[Z_n] \rightarrow 0$ . The population is expected to shrink. It goes extinct with probability 1.

## Critical ( $\mu = 1$ )

$E[Z_n] = Z_0$ . The population is expected to remain stable. Surprisingly, it still goes extinct with probability 1.

## Supercritical ( $\mu > 1$ )

$E[Z_n] \rightarrow \infty$ . The population is expected to grow exponentially. It has a non-zero probability of surviving forever.

## Tool: The Probability Generating Function

To find the extinction probability, we need a powerful tool: the **probability generating function (PGF)** of the offspring distribution  $X$ .

$$G(s) = E[s^X] = \sum_{k=0}^{\infty} p_k s^k \quad \text{for } |s| \leq 1$$

### Key Properties

- ▶  $G(1) = \sum p_k = 1$
- ▶ The mean can be found from the derivative:  $G'(1) = \sum kp_k = \mu$ .
- ▶ The PGF of  $Z_n$  is the  $n$ -th iterate of  $G(s)$  with itself. If  $G_n(s)$  is the PGF of  $Z_n$ , then  $G_{n+1}(s) = G(G_n(s))$ .

## The Extinction Probability Equation

Let  $\pi_0$  be the probability of eventual extinction, starting with  $Z_0 = 1$ .

$$\pi_0 = P(\text{population dies out}) = \lim_{n \rightarrow \infty} P(Z_n = 0)$$

Since  $P(Z_n = 0) = G_n(0)$ , and  $G_{n+1}(0) = G(G_n(0))$ , in the limit the extinction probability  $\pi_0$  must satisfy the equation:

$$\pi_0 = G(\pi_0)$$

### Theorem 3

*The extinction probability  $\pi_0$  is the **smallest non-negative solution** to the equation  $s = G(s)$ .*

- ▶ If  $\mu \leq 1$ , the only solution in  $[0, 1]$  is  $s = 1$ . So  $\pi_0 = 1$ .
- ▶ If  $\mu > 1$ , there is a unique solution in  $[0, 1)$ , which is the extinction probability  $\pi_0 < 1$ .

# From Discrete to Continuous Time

## Limitation of Galton-Watson

Generations don't happen in synchronized steps in the real world. Individuals give birth and die at random times.

This leads us to **Continuous-Time Markov Chains (CTMCs)**.

- ▶ The state of the system is the population size,  $k \in \{0, 1, 2, \dots\}$ .
- ▶ Instead of generations, we have transition rates:
  - ▶  $\lambda_k$ : rate of birth when population is size  $k$  (moves to  $k + 1$ ).
  - ▶  $\delta_k$ : rate of death when population is size  $k$  (moves to  $k - 1$ ).
- ▶ Often, we assume these rates are linear:  $\lambda_k = k\lambda$  and  $\delta_k = k\delta$ . This means individuals act independently.

# Branching Process Approximation of a CTMC

## The Key Insight

At the beginning of an outbreak (or for a very large population), the dynamics caused by a single individual are largely independent of others.

This allows us to approximate the start of a CTMC population process with a branching process.

### Example: A Simple Epidemic (SIR Model)

- ▶  $S$ : Susceptible,  $I$ : Infected,  $R$ : Recovered.
- ▶ An infected person meets others at a certain rate. If they meet a susceptible, a new infection may occur (an "offspring").
- ▶ The infected person recovers (or dies) at another rate, ending their infectious period.
- ▶ **Question:** How many new infections does a single infected person cause on average?

## Case Study: The Basic Reproduction Number $\mathcal{R}_0$

Consider a single infected individual in a large population of susceptibles.

- ▶ Let  $\beta$  be the infection rate (rate of producing "offspring").
- ▶ Let  $\gamma$  be the recovery rate (rate of "dying").

The individual's infectious lifetime is an exponential random variable with mean  $1/\gamma$ .  
The average number of secondary infections they cause is:

$$\mathcal{R}_0 = (\text{rate of infection}) \times (\text{average infectious period}) = \beta \times \frac{1}{\gamma} = \frac{\beta}{\gamma}$$

### The Connection

$\mathcal{R}_0$  is precisely the **mean offspring number**  $\mu$  for the embedded branching process that approximates the start of the epidemic.

## Applying Branching Theory to Epidemics

The fate of the epidemic's initial phase is determined by  $\mathcal{R}_0$ :

- ▶ If  $\mathcal{R}_0 \leq 1$  ( $\mu \leq 1$ ): The number of infected individuals is a subcritical or critical process. The epidemic will die out with probability 1.
- ▶ If  $\mathcal{R}_0 > 1$  ( $\mu > 1$ ): The process is supercritical. There is a positive probability that the epidemic takes off and causes a major outbreak.

We can even calculate the probability of a major outbreak! It is  $1 - \pi_0$ , where  $\pi_0$  is the extinction probability.

For this simple birth-death infection process, the PGF is  $G(s) = \frac{\gamma}{\beta+\gamma} + \frac{\beta}{\beta+\gamma}s$ . Solving  $s = G(s)$  gives the extinction probability:

$$\pi_0 = \frac{\gamma}{\beta} = \frac{1}{\mathcal{R}_0}$$

The probability of a major outbreak is  $1 - 1/\mathcal{R}_0$ .

## Summary

- ▶ **Branching Processes** model populations with i.i.d. offspring generation.
- ▶ The fate of the population is determined by the **mean offspring number**  $\mu$ . Extinction is certain if  $\mu \leq 1$ .
- ▶ The **extinction probability**  $\pi_0$  can be calculated as the smallest non-negative fixed point of the probability generating function  $G(s)$ .
- ▶ The initial stages of many large-scale **Continuous-Time Markov Chains** can be approximated by a branching process.
- ▶ This allows us to apply the theory to real-world problems, like calculating an epidemic's **basic reproduction number**  $\mathcal{R}_0$  and its probability of causing a major outbreak.