

Stochastic models

Lecture 03

Julien Arino

September 2023

Sojourn times

Discrete-time Markov chains

Continuous time Markov chains

Sojourn times

Discrete-time Markov chains

Continuous time Markov chains

Time to events

We suppose that a system can be in two states, S_1 and S_2

- ▶ At time $t = 0$, the system is in state S_1 .
- ▶ An event happens at some time $t = \tau$, which triggers the switch from state S_1 to state S_2 .

Let us call T the random variable

“time spent in state S_1 before switching into state S_2 ”

The states can be anything:

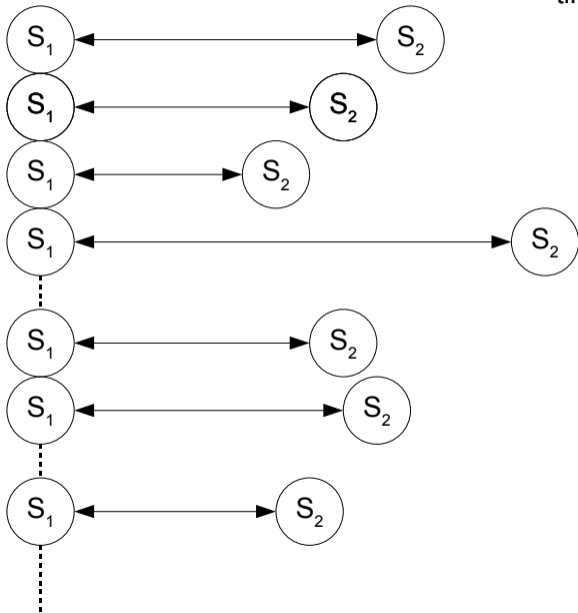
- ▶ S_1 : working, S_2 : broken;
- ▶ S_1 : infected, S_2 : recovered;
- ▶ S_1 : alive, S_2 : dead;
- ▶ ...

We take a collection of objects or individuals that are in state S_1 and want some law for the **distribution** of the times spent in S_1 , 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..

For this, we conduct an **infinite** number of experiments, and observe the time that it takes, in every experiment, to switch from S_1 to S_2

0  time



A distribution of probability is a model

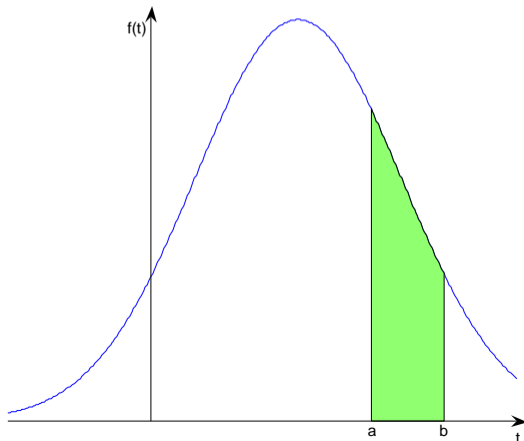
From the sequence of experiments, we deduce a model, which in this context is called a **probability distribution**

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

Probability density function

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

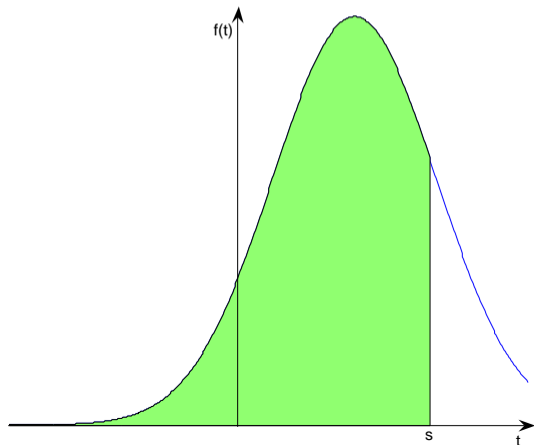
- ▶ $f \geq 0$
- ▶ $\int_{-\infty}^{+\infty} f(s) ds = 1$
- ▶ $\mathbb{P}(a \leq T \leq b) = \int_a^b f(t) dt$



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 S_1 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** τ in state S_1 is given by

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

Since $\lim_{t \rightarrow \infty} tS(t) = 0$, 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 $t + \Delta t$, 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)

Taux d'échec par type d'évènement

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, if a process is subject to two competing exponential risks with respective distributions with parameters θ_1 and θ_2 , the the mean sojourn time in the initial state before being affected by one of the two risks is

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

Sojourn times

Two “extreme” distributions

A simple cohort model with death

Sojourn times in an SIS disease transmission model

Conclusion

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 S_1 is of the form $\mathcal{S}(t) = e^{-\theta t}$, for $t \geq 0$, and the average sojourn time in state S_1 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 θ , 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 $\omega > 0$,

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

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

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

Sojourn times

Two “extreme” distributions

A simple cohort model with death

Sojourn times in an SIS disease transmission model

Conclusion

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 $P(t)$

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

$$N(t) = N_0 P(t) \tag{3}$$

$N_0 P(t)$ proportion of initial population 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 $P(t) = e^{-dt}$, and (3) takes the form

$$N(t) = N_0 e^{-dt} \quad (4)$$

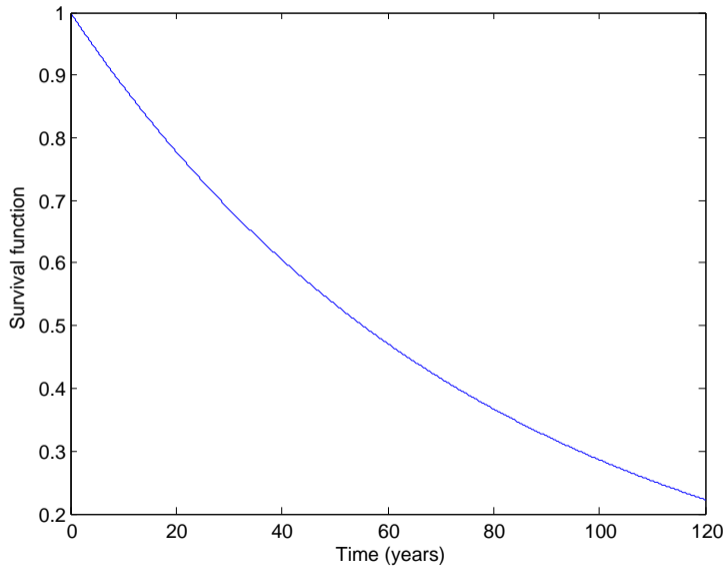
Now note that

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

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

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

Then (3) takes the form

$$N(t) = \begin{cases} N_0, & 0 \leq t \leq \omega, \\ 0, & t > \omega. \end{cases} \quad (5)$$

All individuals survive until time ω , then they all die at time ω .

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

Sojourn times

Two “extreme” distributions

A simple cohort model with death

Sojourn times in an SIS disease transmission model

Conclusion

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 γ
- ▶ Here, of the individuals that become infective at time t_0 , a fraction $P(t - t_0)$ remain infective at time $t \geq t_0$
- ▶ \Rightarrow For $t \geq 0$, $P(t)$ is a survival function. As such, it verifies $P(0) = 1$ and P 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)

$$I(t) = I_0(t) + \int_0^t \beta \frac{S(u)I(u)}{N} P(t-u) du \quad (6)$$

- ▶ $I_0(t)$ 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 $\lim_{t \rightarrow \infty} I_0(t) = 0$.
- ▶ $P(t-u)$ 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:

$$I(t) = I_0(t) + \int_0^t \beta \frac{(N - I(u))I(u)}{N} P(t - u) du \quad (6)$$

The term

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

- ▶ $\beta(N - I(u))I(u)/N$ is the rate at which new infectives are created, at time u ,
- ▶ multiplying by $P(t - u)$ 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 $P(t)$ such that sojourn time in the infective state has exponential distribution with mean $1/\gamma$, **i.e.**, $P(t) = e^{-\gamma t}$.

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 (3).

Equation (6) becomes

$$I(t) = I_0(0)e^{-\gamma t} + \int_0^t \beta \frac{(N - I(u))I(u)}{N} e^{-\gamma(t-u)} du. \quad (7)$$

Taking the time derivative of (7) yields

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

which 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

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

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

In this case (6) becomes

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

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, (8) 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)

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

Sojourn times

Two “extreme” distributions

A simple cohort model with death

Sojourn times in an SIS disease transmission model

Conclusion

Conclusion

- ▶ The time of sojourn in classes (compartments) plays an important role in determining the type of model that we deal with
- ▶ All ODE models, when they use terms of the form κX , make the assumption that the time of sojourn in compartments is exponentially distributed
- ▶ At the other end of the spectrum, delay differential with discrete delay make the assumption of a constant sojourn time, equal for all individuals
- ▶ Both can be true sometimes.. but reality is more likely somewhere in between

Sojourn times

Discrete-time Markov chains

Continuous time Markov chains

A discrete-time Markov chain takes the form

$$p(n+1) = p(n)P, \quad n = 1, 2, 3, \dots$$

where $p(n) = (p_1(n), p_2(n), \dots, p_r(n))$ is a (row) probability vector and $P = (p_{ij})$ is a $r \times r$ **transition matrix**

$$P = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1r} \\ p_{21} & p_{22} & \cdots & p_{2r} \\ p_{r1} & p_{r2} & \cdots & p_{rr} \end{pmatrix}$$

Stochastic matrices

Definition 1

The nonnegative $r \times r$ matrix M is **stochastic** if $\sum_{j=1}^r a_{ij} = 1$ for all $i = 1, 2, \dots, r$

Definition 2

Let M be a stochastic matrix M . Then all eigenvalues λ of M are such that $|\lambda| \leq 1$. Furthermore, $\lambda = 1$ is an eigenvalue of M

Theorem 3

If M, N are stochastic matrices, then MN is a stochastic matrix

Theorem 4

If M is a stochastic matrix, then for any $k \in \mathbb{N}$, M^k is a stochastic matrix

Asymptotic behavior

Let $p(0)$ be the initial distribution (row) vector. Then

$$\begin{aligned}p(1) &= p(0)P \\p(2) &= p(1)P \\&= (p(0)P)P \\&= p(0)P^2\end{aligned}$$

Iterating, we get that for any n ,

$$p(n) = p(0)P^n$$

Therefore,

$$\lim_{n \rightarrow +\infty} p(n) = \lim_{n \rightarrow +\infty} p(0)P^n = p(0) \lim_{n \rightarrow +\infty} P^n$$

Discrete-time Markov chains

- Regular DTMC

- Random walk v1.0 (regular case)

- Absorbing DTMC

Regular Markov chain

Definition 5

A **regular Markov chain** is one in which P^k is positive for some integer $k > 0$, i.e., P^k has only positive entries, no zero entries

Definition 6

A nonnegative matrix M is **primitive** if, and only if, there is an integer $k > 0$ such that M^k is positive

Theorem 7

A Markov chain is regular if, and only if, the transition matrix P is primitive

Important result for regular Markov chains

Theorem 8

If P is the transition matrix of a regular Markov chain, then

- 1. the powers P^n approach a stochastic matrix W*
- 2. each row of W is the same (row) vector $w = (w_1, \dots, w_r)$*
- 3. the components of w are positive*

So if the Markov chain is regular

$$\lim_{n \rightarrow +\infty} p(n) = p(0) \lim_{n \rightarrow +\infty} P^n = p(0)W$$

The vector w is the left eigenvector corresponding to the eigenvalue 1 of P . (We already know that the (right) eigenvector corresponding to 1 is $\mathbb{1}$.)

Indeed, if $p(n)$ converges, then $p(n+1) = p(n)P$, so w is a fixed point of the system. We thus write

$$wP = w$$

and solve for w , which amounts to finding w as the left eigenvector corresponding to the eigenvalue 1

Alternatively, we can find w as the (right) eigenvector associated to the eigenvalue 1 for the transpose of P

$$P^T w^T = w^T$$

(normalise if need be)

Linking matrix and graph theory

Definition 9

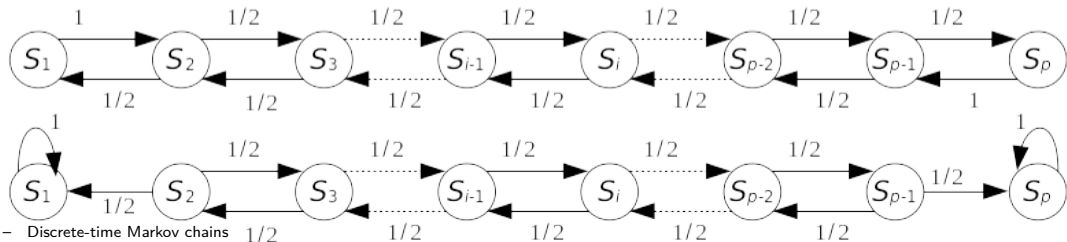
A digraph \mathcal{G} is **strongly connected** if there is a path between all pairs of vertices

Definition 10

A matrix $M \in \mathcal{M}_n$ is **irreducible** if there does not exist a matrix $P \in \mathcal{M}_n$ s.t. $P^{-1}AP$ block triangular

Theorem 11

$A \in \mathcal{M}_n$ irreducible $\iff \mathcal{G}(A)$ strongly connected



Discrete-time Markov chains

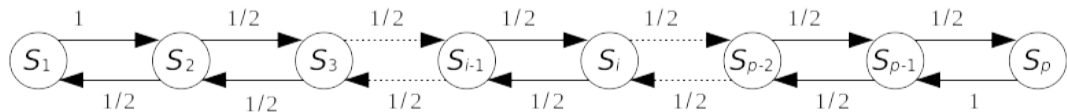
Regular DTMC

Random walk v1.0 (regular case)

Absorbing DTMC

Drunk man's walk 1.0 (regular case)

- ▶ chain of states S_1, \dots, S_p
- ▶ if in state S_i , $i = 2, \dots, p-1$, probability $1/2$ of going left (to S_{i-1}) and $1/2$ of going right (to S_{i+1})
- ▶ if in state S_1 , probability 1 of going to S_2
- ▶ if in state S_p , probability 1 of going to S_{p-1}



Transition matrix for DMW 1.0

$$P = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & \cdots & 0 \\ 1/2 & 0 & 1/2 & 0 & & & \\ 0 & 1/2 & 0 & 1/2 & & & \\ \vdots & & \ddots & \ddots & \ddots & & \vdots \\ & & & & 1/2 & 0 & 1/2 \\ & & & & 0 & 1 & 0 \end{pmatrix}$$

Clearly a primitive matrix, so a regular Markov chain. We find (easy to do by hand)

$$w^T = \left(\frac{1}{2(p-1)}, \frac{1}{p-1}, \dots, \frac{1}{p-1}, \frac{1}{2(p-1)} \right)$$

Setting up the transition matrix

```
# Total population
nb_states = 10 # Small so we can see output
# Parameters
proba_left = 0.5
proba_right = 0.5
proba_stay = 1-(proba_left+proba_right)
# Make the transition matrix
T = mat.or.vec(nr = nb_states, nc = nb_states)
for (row in 2:(nb_states-1)) {
  T[row,(row-1)] = proba_left
  T[row,(row+1)] = proba_right
  T[row, row] = proba_stay
}
# First row only has move right
T[1,2] = 1
# Last row only has move left
T[nb_states, (nb_states-1)] = 1
```

Analysis using markovchain library

```
library(markovchain)
mcRW <- new("markovchain",
            states = sprintf("S_%d", 1:nb_states),
            transitionMatrix = T,
            name = "RW_reg")
```

```
> summary(mcRW)
RW_reg Markov chain that is composed by:
Closed classes:
S_1 S_2 S_3 S_4 S_5 S_6 S_7 S_8 S_9 S_10
Recurrent classes:
{S_1,S_2,S_3,S_4,S_5,S_6,S_7,S_8,S_9,S_10}
Transient classes:
NONE
The Markov chain is irreducible
The absorbing states are: NONE
```

```

> steadyStates(mcRW)
      S_1      S_2      S_3      S_4      S_5      S_6      S_7      S_
8      S_9
[1,] 0.05555556 0.11111111 0.11111111 0.11111111 0.11111111 0.11111111 0.11111111 0.11111111
      0.11111111 0.11111111
      S_10
[1,] 0.05555556

```

Jives with

$$w^T = \left(\frac{1}{2(p-1)}, \frac{1}{p-1}, \dots, \frac{1}{p-1}, \frac{1}{2(p-1)} \right)$$

we had computed

`meanRecurrenceTime`: outputs a named vector with the expected time to first return to a state when the chain starts there. States present in the vector are only the recurrent ones. If the matrix is ergodic (i.e. irreducible), then all states are present in the output and order is the same as states order for the Markov chain

```
> meanRecurrenceTime(mcRW)
 S_1 S_2 S_3 S_4 S_5 S_6 S_7 S_8 S_9 S_10
 18  9  9  9  9  9  9  9  9  18
```

`period`: returns a integer number corresponding to the periodicity of the Markov chain (if it is irreducible)

```
> period(mcRW)
[1] 2
```

(period of state $x \in \mathcal{S}$ is $\gcd\{n \in \mathbb{N}_+ : T^n(x, x) > 0\}$)

meanFirstPassageTime: Given an irreducible (ergodic) markovchain object, this function calculates the expected number of steps to reach other states

```
> meanFirstPassageTime(mcRW)
      S_1 S_2 S_3 S_4 S_5 S_6 S_7 S_8 S_9 S_10
S_1    0  1  4  9 16 25 36 49 64 81
S_2   17  0  3  8 15 24 35 48 63 80
S_3   32 15  0  5 12 21 32 45 60 77
S_4   45 28 13  0  7 16 27 40 55 72
S_5   56 39 24 11  0  9 20 33 48 65
S_6   65 48 33 20  9  0 11 24 39 56
S_7   72 55 40 27 16  7  0 13 28 45
S_8   77 60 45 32 21 12  5  0 15 32
S_9   80 63 48 35 24 15  8  3  0 17
S_10  81 64 49 36 25 16  9  4  1  0
```

Discrete-time Markov chains

- Regular DTMC

- Random walk v1.0 (regular case)

- Absorbing DTMC

Absorbing states, absorbing chains

Definition 12

A state S_i in a Markov chain is **absorbing** if whenever it occurs on the n^{th} generation of the experiment, it then occurs on every subsequent step. In other words, S_i is absorbing if $p_{ii} = 1$ and $p_{ij} = 0$ for $i \neq j$

Definition 13

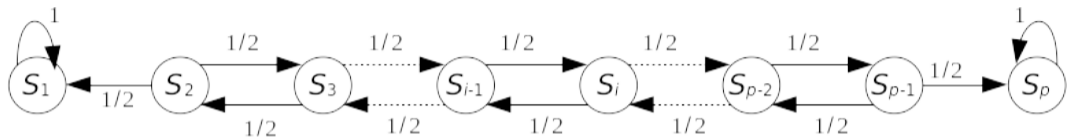
A **Markov chain is absorbing** if it has at least one absorbing state, and if from every state it is possible to go to an absorbing state

Definition 14

In an absorbing Markov chain, a state that is not absorbing is called **transient**

Some questions on absorbing chains

Suppose we have a chain like the following



1. Does the process eventually reach an absorbing state?
2. Average number of times spent in a transient state, if starting in a transient state?
3. Average number of steps before entering an absorbing state?
4. Probability of being absorbed by a given absorbing state, when there are more than one, when starting in a given transient state?

Reaching an absorbing state

Answer to question 1:

Theorem 15

In an absorbing Markov chain, the probability of reaching an absorbing state is 1

Standard form of the transition matrix

For an absorbing chain with k absorbing states and $r - k$ transient states, the transition matrix can be written as

$$P = \begin{pmatrix} \mathbb{I}_k & \mathbf{0} \\ R & Q \end{pmatrix}$$

	Absorbing states	Transient states
Absorbing states	\mathbb{I}_k	$\mathbf{0}$
Transient states	R	Q

\mathbb{I}_k the $k \times k$ identity, $\mathbf{0} \in \mathbb{R}^{k \times (r-k)}$, $R \in \mathbb{R}^{(r-k) \times k}$, $Q \in \mathbb{R}^{(r-k) \times (r-k)}$

The matrix $\mathbb{I}_{r-k} - Q$ is invertible. Let

- ▶ $N = (\mathbb{I}_{r-k} - Q)^{-1}$ be the **fundamental matrix** of the Markov chain
- ▶ T_i be the sum of the entries on row i of N
- ▶ $B = NR$

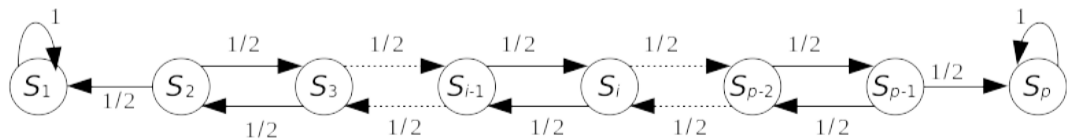
Answers to our remaining questions:

2. N_{ij} is the average number of times the process is in the j th transient state if it starts in the i th transient state
3. T_i is the average number of steps before the process enters an absorbing state if it starts in the i th transient state
4. B_{ij} is the probability of eventually entering the j th absorbing state if the process starts in the i th transient state

See for instance book of Kemeny and Snell

Drunk man's walk 2.0 (absorbing case)

- ▶ chain of states S_1, \dots, S_p
- ▶ if in state S_i , $i = 2, \dots, p-1$, probability $1/2$ of going left (to S_{i-1}) and $1/2$ of going right (to S_{i+1})
- ▶ if in state S_1 , probability 1 of going to S_1
- ▶ if in state S_p , probability 1 of going to S_p



Transition matrix for DMW 2.0

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 1/2 & 0 & 1/2 & 0 & & & \\ 0 & 1/2 & 0 & 1/2 & & & \\ \vdots & & \ddots & \ddots & \ddots & & \vdots \\ & & & & 1/2 & 0 & 1/2 \\ & & & & 0 & 0 & 1 \end{pmatrix}$$

Put P in standard form

Absorbing states are S_1 and S_p , write them first, then write other states

	S_1	S_p	S_2	S_3	S_4	\dots	S_{p-2}	S_{p-1}
S_1	1	0	0	0	0	\dots	0	0
S_p	0	1	0	0	0	\dots	0	0
S_2	$1/2$	0	0	$1/2$	0	\dots	0	0
S_3	0	0	$1/2$	0	$1/2$	\dots	0	0
\vdots								
S_{p-2}	0	0	0	0	0	\dots	0	$1/2$
S_{p-1}	0	$1/2$	0	0	0	\dots	$1/2$	0

So we find

$$P = \begin{pmatrix} \mathbb{I}_2 & \mathbf{0} \\ R & Q \end{pmatrix}$$

where $\mathbf{0}$ a $2 \times (p - 2)$ -matrix, R a $(p - 2) \times 2$ matrix and Q a $(p - 2) \times (p - 2)$ matrix

$$R = \begin{pmatrix} 1/2 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & 1/2 \end{pmatrix}$$

and

$$Q = \begin{pmatrix} 0 & 1/2 & 0 & & & \\ 1/2 & 0 & 1/2 & & & \\ 0 & 1/2 & 0 & & & \\ & & \ddots & \ddots & \ddots & \\ & & & & 1/2 & 0 & 1/2 \\ 0 & & & & & 1/2 & 0 \end{pmatrix}$$

$$\mathbb{I}_{p-2} - Q = \begin{pmatrix} 1 & -1/2 & 0 & & & \\ -1/2 & 1 & -1/2 & & & \\ 0 & -1/2 & 1 & & & \\ & & & \ddots & \ddots & \ddots \\ 0 & & & & -1/2 & 1 & -1/2 \\ 0 & & & & & -1/2 & 1 \end{pmatrix}$$

This is a **symmetric tridiagonal Toeplitz** matrix

(symmetric: obvious; tridiagonal: there are three diagonal bands; Toeplitz: each diagonal band is constant)

Could invert it explicitly, let us not bother

Setting up the transition matrix

```
# Total population
nb_states = 10 # Small so we see output
# Parameters
proba_left = 0.5
proba_right = 0.5
proba_stay = 1-(proba_left+proba_right)
# Make the transition matrix
T = mat.or.vec(nr = nb_states, nc = nb_states)
for (row in 2:(nb_states-1)) {
  T[row,(row-1)] = proba_left
  T[row,(row+1)] = proba_right
  T[row, row] = proba_stay
}
# First and last rows only have stay
T[1,1] = 1
T[nb_states, nb_states] = 1
```

Analysis using markovchain library

```
library(markovchain)
mcRW <- new("markovchain",
           states = sprintf("S_%d", 1:nb_states),
           transitionMatrix = T,
           name = "RW_abs")
```

```
> summary(mcRW)
RW_abs Markov chain that is composed by:
Closed classes:
S_1
S_10
Recurrent classes:
{S_1},{S_10}
Transient classes:
{S_2,S_3,S_4,S_5,S_6,S_7,S_8,S_9}
The Markov chain is not irreducible
The absorbing states are: S_1 S_10
```

```
> canonicForm(mcRW)
```

```
RW_abs
```

A 10 - dimensional discrete Markov Chain defined by the following states:

S_1, S_10, S_2, S_3, S_4, S_5, S_6, S_7, S_8, S_9

The transition matrix (by rows) is defined as follows:

	S_1	S_10	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9
S_1	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
S_10	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
S_2	0.5	0.0	0.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0
S_3	0.0	0.0	0.5	0.0	0.5	0.0	0.0	0.0	0.0	0.0
S_4	0.0	0.0	0.0	0.5	0.0	0.5	0.0	0.0	0.0	0.0
S_5	0.0	0.0	0.0	0.0	0.5	0.0	0.5	0.0	0.0	0.0
S_6	0.0	0.0	0.0	0.0	0.0	0.5	0.0	0.5	0.0	0.0
S_7	0.0	0.0	0.0	0.0	0.0	0.0	0.5	0.0	0.5	0.0
S_8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5	0.0	0.5
S_9	0.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.5	0.0

```
> meanAbsorptionTime(mcRW)
S_2 S_3 S_4 S_5 S_6 S_7 S_8 S_9
   8  14  18  20  20  18  14   8
> absorptionProbabilities(mcRW)
      S_1      S_10
S_2 0.8888889 0.1111111
S_3 0.7777778 0.2222222
S_4 0.6666667 0.3333333
S_5 0.5555556 0.4444444
S_6 0.4444444 0.5555556
S_7 0.3333333 0.6666667
S_8 0.2222222 0.7777778
S_9 0.1111111 0.8888889
```

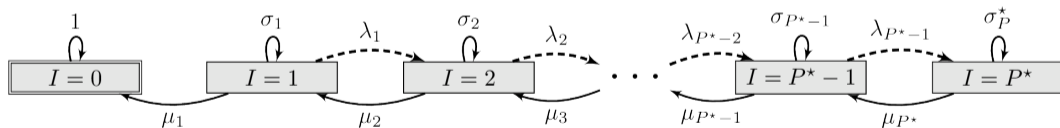
hittingProbabilities: given a markovchain object, this function calculates the probability of ever arriving from state i to j

```
> hittingProbabilities(mcRW)
      S_1  S_2  S_3  S_4  S_5  S_6  S_7  S_8  S_9
S_10
S_1  1.000000 0.0000 0.000000 0.000000 0.000 0.000 0.000000 0.000000 0.0000
0.000000
S_2  0.8888889 0.4375 0.500000 0.3333333 0.250 0.200 0.1666667 0.1428571 0.1250
0.1111111
S_3  0.7777778 0.8750 0.6785714 0.6666667 0.500 0.400 0.3333333 0.2857143 0.2500
0.2222222
S_4  0.6666667 0.7500 0.8571429 0.7500000 0.750 0.600 0.5000000 0.4285714 0.3750
0.3333333
S_5  0.5555556 0.6250 0.7142857 0.8333333 0.775 0.800 0.6666667 0.5714286 0.5000
0.4444444
S_6  0.4444444 0.5000 0.5714286 0.6666667 0.800 0.775 0.8333333 0.7142857 0.6250
0.5555556
S_7  0.3333333 0.3750 0.4285714 0.5000000 0.600 0.750 0.7500000 0.8571429 0.7500
0.6666667
S_8  0.2222222 0.2500 0.2857143 0.3333333 0.400 0.500 0.6666667 0.6785714 0.8750
0.7777778
S_9  0.1111111 0.1250 0.1428571 0.1666667 0.200 0.250 0.3333333 0.5000000 0.4375
```

DTMC SIS system

Since $S = P^* - I$, consider only the infected. To simulate as DTMC, consider a random walk on I (\simeq Gambler's ruin problem)

Denote $\lambda_I = \beta(P^* - I)\Delta t$, $\mu_I = \gamma I\Delta t$ and $\sigma_I = 1 - (\lambda_I + \mu_I)\Delta t$



To make things easy to see: Pop=5

```
# Make the transition matrix
T = mat.or.vec(nr = (Pop+1), nc = (Pop+1))
for (row in 2:Pop) {
  I = row-1
  mv_right = gamma*I*Delta_t # Recoveries
  mv_left = beta*I*(Pop-I)*Delta_t # Infections
  T[row,(row-1)] = mv_right
  T[row,(row+1)] = mv_left
}
# Last row only has move left
T[(Pop+1),Pop] = gamma*(Pop)*Delta_t
# Check that we don't have too large values
if (max(rowSums(T))>1) {
  T = T/max(rowSums(T))
}
diag(T) = 1-rowSums(T)
```


Analysis using markovchain library

```
library(markovchain)
mcSIS <- new("markovchain",
            states = sprintf("I_%d", 0:Pop),
            transitionMatrix = T,
            name = "SIS")
```

```
> summary(mcSIS)
SIS Markov chain that is composed by:
Closed classes:
I_0
Recurrent classes:
{I_0}
Transient classes:
{I_1,I_2,I_3,I_4,I_5}
The Markov chain is not irreducible
The absorbing states are: I_0
```

```
> canonicForm(mcSIS)
```

```
SIS
```

```
A 6 - dimensional discrete Markov Chain defined by the following states:
```

```
I_0, I_1, I_2, I_3, I_4, I_5
```

```
The transition matrix (by rows) is defined as follows:
```

	I_0	I_1	I_2	I_3	I_4	I_5
I_0	1.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
I_1	0.1666667	0.5000000	0.3333333	0.0000000	0.0000000	0.0000000
I_2	0.0000000	0.3333333	0.1666667	0.5000000	0.0000000	0.0000000
I_3	0.0000000	0.0000000	0.5000000	0.0000000	0.5000000	0.0000000
I_4	0.0000000	0.0000000	0.0000000	0.6666667	0.0000000	0.3333333
I_5	0.0000000	0.0000000	0.0000000	0.0000000	0.8333333	0.1666667

```
# The vector of steady states. Here, all mass should be in I_0
```

```
> steadyStates(mcSIS)
```

```
      I_0 I_1 I_2 I_3 I_4 I_5  
[1,]  1  0  0  0  0  0
```

```
> hittingProbabilities(mcSIS)
```

	I_0	I_1	I_2	I_3	I_4	I_5
I_0	1	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
I_1	1	0.8333333	0.6666667	0.5454545	0.4615385	0.3529412
I_2	1	1.0000000	0.8888889	0.8181818	0.6923077	0.5294118
I_3	1	1.0000000	1.0000000	0.9090909	0.8461538	0.6470588
I_4	1	1.0000000	1.0000000	1.0000000	0.8974359	0.7647059
I_5	1	1.0000000	1.0000000	1.0000000	1.0000000	0.8039216

Read by row: if the process starts in I_i (row $i - 1$), probability that state I_j (column $j - 1$) is visited

```
> meanAbsorptionTime(mcSIS)
  I_1  I_2  I_3  I_4  I_5
24.30 33.45 37.55 39.65 40.85
> absorptionProbabilities(mcSIS)
I_0
I_1  1
I_2  1
I_3  1
I_4  1
I_5  1
```

Sojourn times

Discrete-time Markov chains

Continuous time Markov chains

Continuous-time Markov chains

CTMC similar to DTMC except in way they handle time between events (transitions)

DTMC: transitions occur each Δt

CTMC: $\Delta t \rightarrow 0$ and transition times follow an exponential distribution parametrised by the state of the system

CTMC are roughly equivalent to ODE

Continuous time Markov chains

ODE \leftrightarrow CTMC

Simulating CTMC (in theory)

Simulating CTMC (in practice)

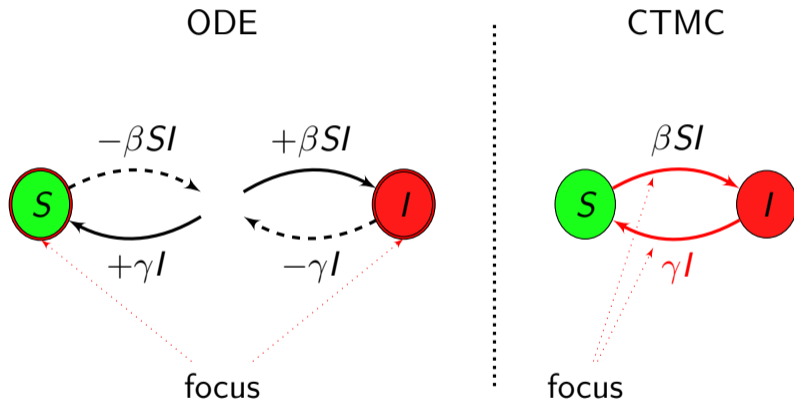
Parallelising your code in R

Converting your compartmental ODE model to CTMC

Easy as π :)

- ▶ 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



SIS without demography

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

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	βSI	$\frac{\beta SI}{b+d(S+I)+\beta SI+\gamma I}$
$I \rightarrow I - 1$	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	γI	$\frac{\gamma I}{b+d(S+I)+\beta SI+\gamma I}$

States are S, I

Kermack & McKendrick model

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

States are S, I, R

Continuous time Markov chains

ODE \leftrightarrow CTMC

Simulating CTMC (in theory)

Simulating CTMC (in practice)

Parallelising your code in R

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 τ_t from $T \sim \mathcal{E}(\xi_t)$

- Draw ζ_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 τ_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 ζ from $\mathcal{U}([0, 1])$ and solve $F(x, \xi_t) = \zeta$ for x

$$F(x, \xi_t) = \zeta \Leftrightarrow 1 - e^{-\xi_t x} = \zeta$$

$$\Leftrightarrow e^{-\xi_t x} = 1 - \zeta$$

$$\Leftrightarrow \xi_t x = -\ln(1 - \zeta)$$

$$\Leftrightarrow \boxed{x = \frac{-\ln(1 - \zeta)}{\xi_t}}$$

Gillespie's algorithm (SIS model with only I 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 τ_t from $T \sim \mathcal{E}(\xi_t)$

- $v \leftarrow [\beta(P^* - i)i, \xi_t] / \xi_t$

- Draw ζ_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 τ_t from $T \sim \mathcal{E}(\xi_t)$
- ▶ So the inter-event time $\tau_t \rightarrow 0$ if ξ_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 τ_t from $T \sim \mathcal{E}(\xi_t)$

- $v \leftarrow [bN(t), \xi_t] / \xi_t$

- Draw ζ_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, $N(t + \tau_t) = N(t) - 1$

- $t \leftarrow t + \tau_t$

```
b = 0.01 # Birth rate
d = 0.01 # Death rate
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

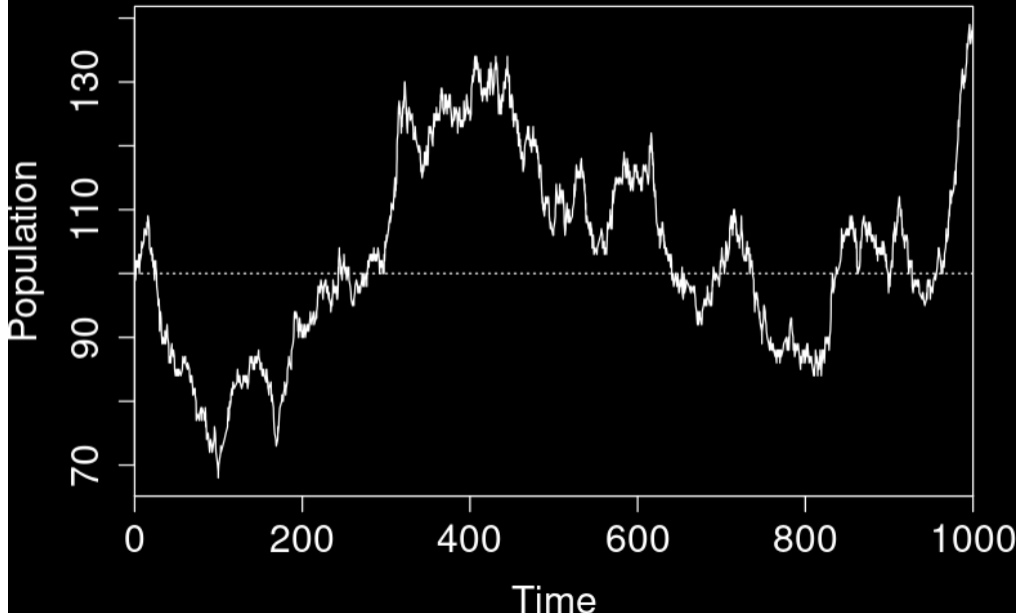
# We'll track the current time and state (could also just check last entry in t
# and N, but will take more operations)
t_curr = t_0
N_curr = N_0
```

```

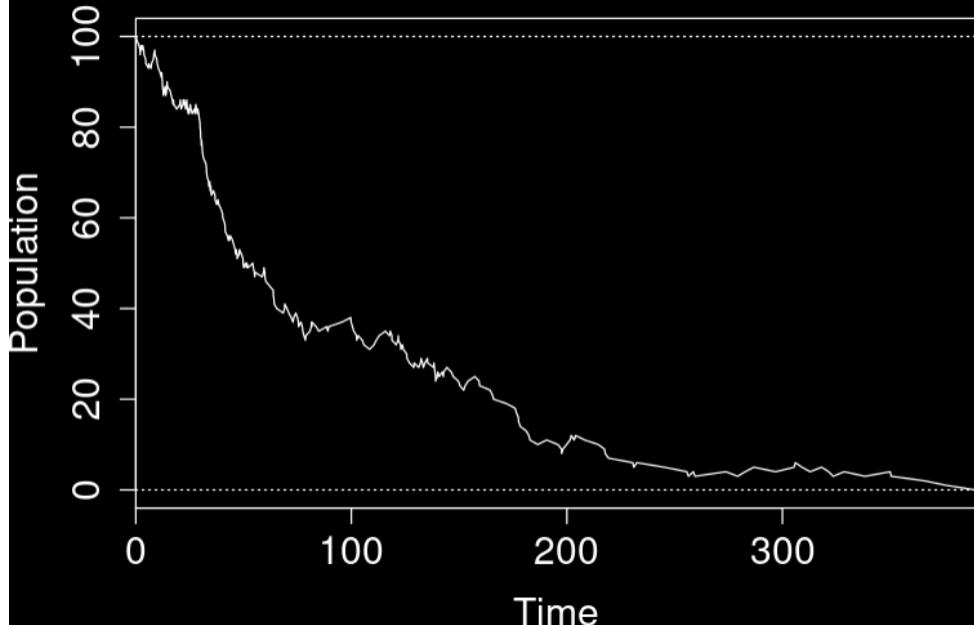
while (t_curr<=t_f) {
  xi_t = (b+d)*N_curr
  # The exponential number generator does not like a rate of 0 (when the
  # population crashes), so we check if we need to quit
  if (N_curr == 0) {
    break
  }
  tau_t = rexp(1, rate = xi_t)
  t_curr = t_curr+tau_t
  v = c(b*N_curr, xi_t)/xi_t
  zeta_t = 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)
}

```

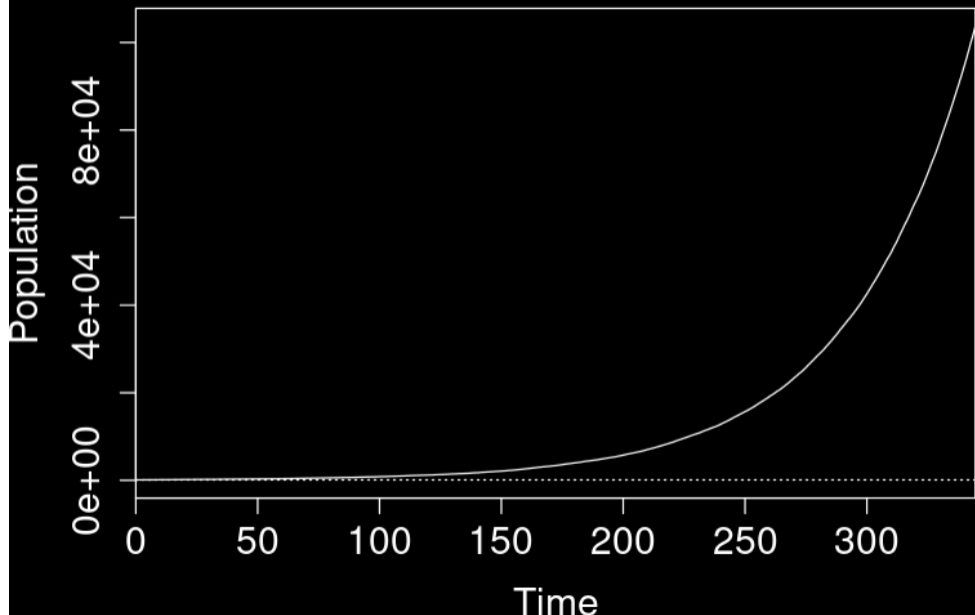
$b=0.01, d=0.01$



$b=0.01, d=0.02$



$b=0.03, d=0.01$

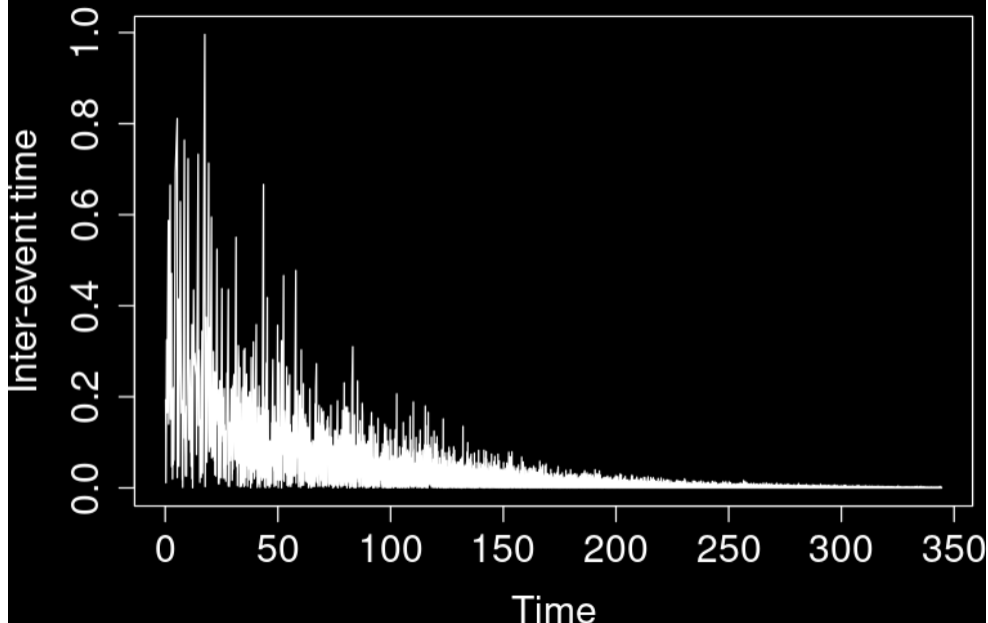


Last one did not go well

- ▶ Wanted 1000 time units (days?)
- ▶ Interrupted at $t = 344.4432$ because I lost patience
(Penultimate slide: sim stopped because the population went extinct, I did not stop it!)
- ▶ At stop time
 - ▶ $N = 103,646$
 - ▶ $|N| = 208,217$ (and $|t|$ as well, of course!)
 - ▶ time was moving slowly

```
> tail(diff(t))  
[1] 1.282040e-05 5.386999e-04 5.468540e-04 1.779985e-04 6.737294e-05 2.618084e-04
```

$b=0.03, d=0.01$



Continuous time Markov chains

ODE \leftrightarrow CTMC

Simulating CTMC (in theory)

Simulating CTMC (in practice)

Parallelising your code in R

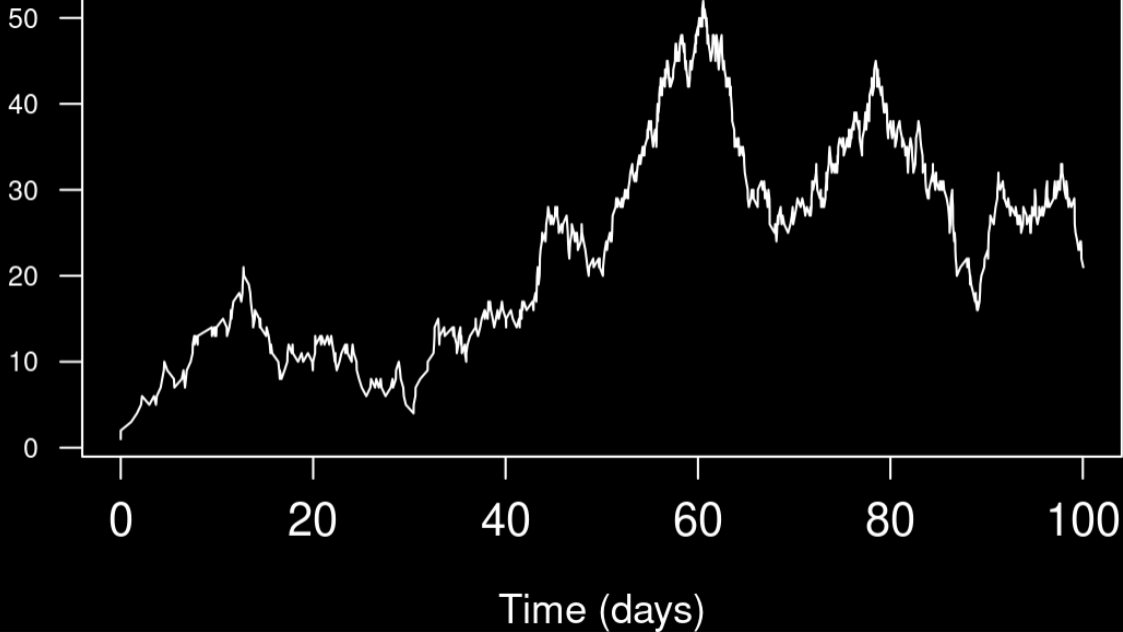
Tau-leaping (and packages) to the rescue!

- ▶ *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

Simulating a CTMC

```
library(GillespieSSA2)
IC <- c(S = (Pop-I_0), I = I_0)
params <- c(gamma = gamma, beta = beta)
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")
```

Prevalence



Continuous time Markov chains

ODE \leftrightarrow CTMC

Simulating CTMC (in theory)

Simulating CTMC (in practice)

Parallelising your code in R

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`

```

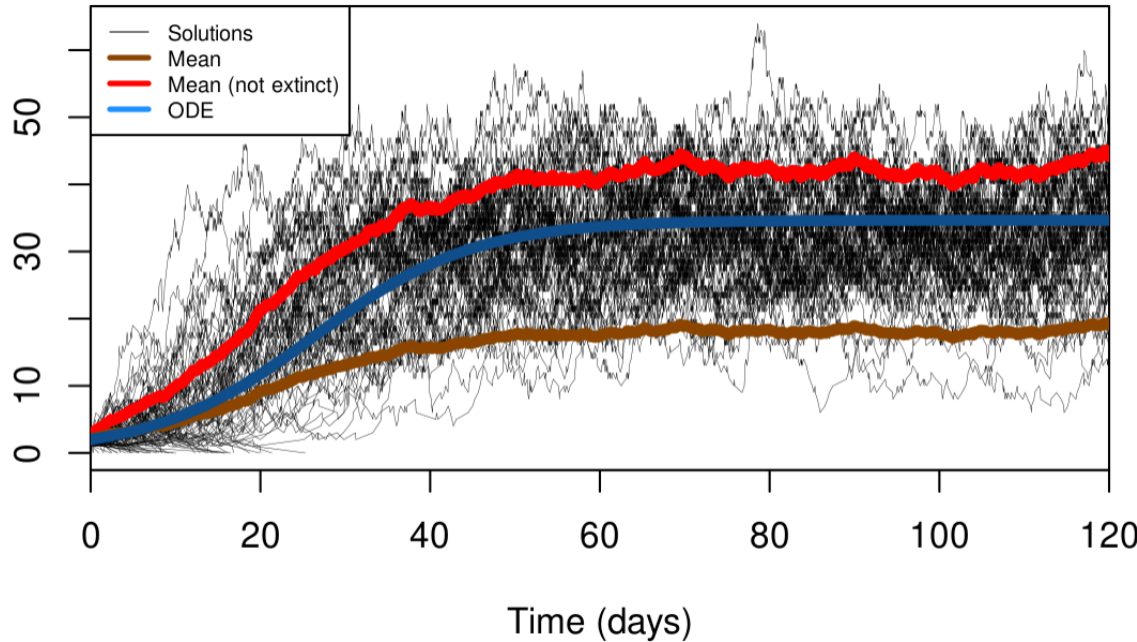
run_one_sim = function(params) {
  IC <- c(S = (params$Pop-params$I_0), I = params$I_0)
  params_local <- c(gamma = params$gamma, beta = params$beta)
  reactions <- list(
    # propensity function effects name for reaction
    reaction("beta*S*I", c(S=-1,I=+1), "new_infection"),
    reaction("gamma*I", c(S=+1,I=-1), "recovery")
  )
  set.seed(NULL)
  sol <- ssa(
    initial_state = IC,
    reactions = reactions,
    params = params_local,
    method = ssa_exact(),
    final_time = params$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)
})
clusterExport(cl,
              c("params",
                "run_one_sim"),
              envir = .GlobalEnv)
SIMS = parLapply(cl = cl,
                 X = 1:params$number_sims,
                 fun = function(x) run_one_sim(params))
stopCluster(cl)
```

Number infectious



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× faster) or '12.067 sec elapsed' versus '258.985 sec elapsed' on a 32C/64T AMD Ryzen Threadripper 3970X 32-Core Processor (21.46× faster !)