Poisson Process and its Application in Epidemic Simulation

Final Project

Course: High Dimensional Probability for Data Science 2020-21
Professor: Formentin Marco

Ma Jiawei

February 6, 2021

Abstract

In this project we aimed to study the notable Poisson process which is the most proper model for describing random events. Specifically, we proved that the waiting times are exponentially distributed and we illustrated a very important theorem regarding concurrent Poisson processes.
In the second part we applied the theoretical results for implementing a stochastic simulation's algorithm; Temporal Gillespie Algorithm. Finally, we run the algorithm along with an epidemic model (SIQRD) in order to analyse the italian COVID-19 contagion situation and the outcomes are very prominent.

1 Theory

In this section we present theoretical basis that will justify our choices in implementing the algorithm. The complete theory can be found in [1] and what we illustrate here is about *Homogeneous Poisson Processes* which is a subclass belongs to a more general mathematical model; *Continuous-Time Markov Models.*

1.1 Poisson Distribution

The notable *Poisson distribution* is a common model used for *counting of randomly independent occurring events in a fixed interval of time.* X is a Poisson random variable if it's a discrete random variable and, given $k \in \mathbb{N}$, we have that

$$\mathbb{P}(X = k) = \frac{\lambda^k}{k!} e^{-\lambda},$$

where λ is called *rate*. It's easy to see that $\mathbb{E}[X] = Var(X) = \lambda$. So we can say that λ is *average number of events occurred in an in a fixed interval of time.* One way to think of the Poisson distribution is as an approximation to the binomial distribution $Bin(n, p)$ when n is large and p is small. Indeed, if $X \sim Bin(n, p)$ then $\mathbb{P}(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$, for any $k \leqslant n$, defining $\lambda := pn$ we can write that

$$\mathbb{P}(X = k) = \binom{n}{k} p^k (1-p)^{n-k} = \frac{n!}{k!(n-k)!} \left(\frac{\lambda}{n}\right)^k \left(1 - \frac{\lambda}{n}\right)^{n-k} = \frac{\lambda^k}{k!} \left(1 - \frac{\lambda}{n}\right)^{n-k} \frac{n!}{(n-k)! n^k}.$$

Now we observe that

$$\frac{n!}{(n-k)!n^k} = \frac{n\cdot(n-1)\cdots(n-k+1)}{n^k} = 1\cdot\big(\frac{n-1}{n}\big)\cdots\big(\frac{n-k+1}{n}\big) \underset{n\ large}{\approx} 1$$

Thus,

$$\mathbb{P}(X=k) = \frac{\lambda^k}{k!}\left(1-\frac{\lambda}{n}\right)^{n-k} \underset{n\ large}{\approx} \frac{\lambda^k}{k!}e^{-\lambda}.$$

Remark. *Using* Central Limit Theorem*, applied to the binomial distribution as the sum of Bernoulli random variables, we can write also that* $X \sim N(\lambda,\lambda)$.

This can prove what we supposed at the beginning about random independent events; we divide the time in small intervals (n large) and during each period of time the occurrences of some events (0 and 1) can be thought as independent *Bernoulli random variables* with same parameter p.

1.2 Homogeneous Poisson Process

Homogeneous Poisson Process (HPP) is the core theoretical object we need. First of all we have to model the occurrence times of the events; $(T_n)_{n\geqslant 0}$. Then we give the following

Definition 1.1. *Let* $(T_n)_{n\geqslant 0}$ *be a sequence of nonnegative random variables, they are called* ***random point process*** *if, almost surely,*

1. $T_0 \equiv 0$,
2. $0 < T_1 < T2 < \cdots$,
3. $\lim_{n\to+\infty} T_n = +\infty$.

The second condition implies that *at most one event can take place in a general time* t. So the process is also called *simple point process*. On the other hand, the third condition means that time increases and the last event's time *explodes*. These conditions are suitable for us; we require that events occur indefinitely and there no more than one event takes place in the same time point.
We need another modelling object which is, essentially, a function that counts the events in a given interval of time:

$$N((a,b]) := \sum_{n\geqslant 1} \mathbf{1}_{(a,b]}(T_n).$$

Finally, we define

$$N(t) := N((0,t]),$$

for any $t \geqslant 0$. It's quite easy to see that $N(0)=0$ and $N(a,b] = N(b) - N(a)$. $N := (N(t))_{t\geqslant 0}$ is called the *counting process* of the point process $(T_n)_{n\geqslant 1}$.
Now we are ready to give the following

Definition 1.2. *Let* N *be a counting process of some point process* $(T_n)_{n\geqslant 0}$. N *is an HPP with intensity* $\lambda > 0$ *if*

1. *(Independent increments) for any times* t_i, $i = 1,\cdots,k$, *such that* $0 \leqslant t_1 \leqslant t_2 \leqslant \cdots \leqslant t_k$, *the random variables* $N(t_i, t_{i+1}]$ *are independent for any* $i = 1,\cdots,k-1$;
2. *for any interval* $(a,b] \subset \mathbb{R}_+$, $N(a,b]$ *has the Poisson distribution*

$$\mathbb{P}(N(a,b]=k) = e^{-\lambda(b-a)}\frac{[\lambda(b-a)]^k}{k!}.$$

Remark. *The definition above can be reformulated in a more general way without imposing that* $(T_n)_{n\geqslant 0}$ *is a point process. The proof is in [1]. We note that the explosion property is straightforward;* $\mathbb{E}[N(a)] = \lambda a < \infty$ *for any* $a \geqslant 0$ *then* $N(a) < \infty$ *a.s., the latter claim is equivalent to* $\lim_{n\to+\infty} T_n = +\infty$ *(by contradiction).*

1.3 Waiting time

Another very important concept related to random events is *waiting time*; amount of time computed starting from the last event fire time to the next event fire time.
We show a theorem that proves that the waiting time in our context (HPPs) has a continuous probability distribution: *exponential distribution.* This is a very convenient way to simulate, with a modern computer, the HPPs.

Theorem 1.3. *Let* $(S_n)_{n\geqslant 1}$ *be defined as*

$$S_n = T_n - T_{n-1},$$

where $(T_n)_{n\geqslant 0}$ *is a HPP with rate* $\lambda > 0$*. Then*

$$\mathbb{P}(S_n \leqslant t) = 1 - e^{-\lambda t},$$

for any $n \geqslant 1$ *and* $t \geqslant 0$*.*

Proof. Let $f_S(s_1, s_2, \cdots, s_n)$ be the joint density of $(S_i)_{0<i\leqslant n}$ and $f_T(t_1, t_2, \cdots, t_n)$ be the joint density of $(T_i)_{0<i\leqslant n}$. It's sufficient to prove that

$$f_S(s_1, s_2, \cdots, s_n) = \prod_{i=1}^{n} \lambda^n e^{-\lambda s_i} \mathbf{1}_{s_i>0} = \lambda^n e^{-\lambda \sum_{i=1}^n s_i} \mathbf{1}_{\{0<t_1<t_2<\cdots<t_n\}}, \tag{1}$$

where $t_1 = s_1, t_2 = s_1 + s_2, \cdots, t_n = s_1 + s_2 + \cdots + s_n$ (we note that it proves also the independence of the exponential random variables).
Thus, (1) is equivalent to

$$f_T(t_1, t_2, \cdots, t_n) = \lambda^n e^{-\lambda t_n} \mathbf{1}_{\{0<t_1<t_2<\cdots<t_n\}}.$$

Now suppose that $0 < t_1 < t_2 < \cdots < t_n$, we can compute the density as

$$\frac{\mathbb{P}\left(\bigcap_{i=1}^n T_i \in (t_i, t_1 + h_i]\right)}{\prod_{i=1}^n h_i}, \tag{2}$$

where $h_1, \cdots, h_n$ tend to zero. Keeping in mind the meaning of s and t we can deduce that

$$\bigcap_{i=1}^{n} \Big\{ T_i \in (t_i, t_1 + h_i] \Big\} =$$
$$= \{N(t_1) = 0\} \cap \left\{ \bigcap_{i=1}^{n-1} N(t_i, t_i + h_i] = 1, N(t_i + h_i, t_{i+1}] = 0 \right\} \cap \{N(t_n, t_n + h_n] \geqslant 1\},$$

for small h_i.
Due to definition of N we have that

$$\mathbb{P}\left(\bigcap_{i=1}^{n} T_i \in (t_i, t_1 + h_i] \right) = e^{-\lambda t_1} \prod_{i=1}^{n-1} (e^{\lambda h_i} \lambda h_i e^{-\lambda(t_{i+1} - t_i - h_i)})(1 - e^{-\lambda h_n}) =$$
$$= \lambda^{n-1} e^{-\lambda t_n} h_1 \cdots h_{n-1} (1 - e^{-\lambda h_n}).$$

Thus, (2) is equal to

$$\frac{\lambda^{n-1}e^{-\lambda t_n}(1-e^{-\lambda h_n})}{h_n} \underset{h_n\to 0}{\longrightarrow} \lambda^n e^{-\lambda t_n}.$$

□

Remark. *λ in HPPs is considered as* average number of events *taken place in a fixed interval of time which can* 1 *in the normalized situation. Then naively, we intuit that the waiting time between events is, in average, $1/\lambda$ and this is what Theorem 1.3 states.*

1.4 Concurrent Poisson Processes

Frequently, studying some phenomena, the easiest way to model the events is to consider them as multiple HPPs that run concurrently or *compete* with each other. In our case, epidemic model, the transitions can be considered as many Poisson process, but we have at least two problems:

- we desire that the *overlapping/sum* of the processes is still a Poisson process because we have many theoretical tools to deal with it;

- we would like to know when an event takes place and which is the process responsible for it. For instance, in the SIR model we may know that an event occurred but we also look for the fired transition; infection or recovery transition.

We solve these problems by using two theorems which are stated and proved in [1]. Here we give only the *finite version* of the two results since we have only to take into account finite states/transitions.

Theorem 1.4. *Let $(N_m)_{m\in\Omega}$ be a family of* ***independent*** *HPPs with respect rates $(\lambda_m)_{m\in\Omega}$ and Ω be a finite set. If $\Lambda := \sum_{m\in\Omega}\lambda_m$ then*

$$N(t) := \sum_{m\in\Omega} N_m(t)$$

defines the counting process of an HPP with rate Λ.

Proof. Trivial; sum of independent Poisson random variables is a Poisson random variables with rate equal to sum of the rates. □

The theorem is very powerful that allows us to only consider an *unique* Poisson process that includes all processes and it simplifies our model.
On the other hand doing so we need to understand, when a transition fired, how to choose the right one (second problem) and the solution is given by the following

Theorem 1.5. *Having the same assumptions as in Theorem 1.4 we define Z the random variable that gives the first event time of N and J the random variable that is the index of the HPP responsible for it. Then*

$$\mathbb{P}(J=m, Z\geqslant a) = \mathbb{P}(J=m)\mathbb{P}(Z\geqslant a) = \frac{\lambda_m}{\Lambda}e^{-\lambda a}. \tag{3}$$

In particular, J and Z are independent, $\mathbb{P}(J=m)=\lambda_m/\Lambda$ and Z is exponential with mean Λ^{-1}.

Proof. Without loss of generality we can assume that $J = 1$. We define $\|\Omega\| := K$ and $U := \inf(S_2, \cdots, S_K)$.
We can compute the first member on the left side of (3):

$$\begin{aligned}\mathbb{P}(J = 1, Z \geqslant a) &= \mathbb{P}(a \leqslant S_1 < U) = \\ &= \int_a^{+\infty} d\mathbb{P}(S_1 < U) = \int_a^{+\infty} \mathbb{P}(U > x)\lambda_1 e^{-\lambda_1 x} dx = \\ &= \int_a^{+\infty} \mathbb{P}\Big(\bigcap_{m=2}^{K} (S_m > a)\Big)\lambda_1 e^{-\lambda_1 x} dx = \int_a^{+\infty} e^{-(\lambda_2+\cdots+\lambda_K)x}\lambda_1 e^{-\lambda_1 x} dx = \\ &= \frac{\lambda_1}{\lambda_1 + \cdots + \lambda_K} e^{-(\lambda_1+\cdots+\lambda_K)a}.\end{aligned}$$

Finally, we know that

$$\begin{aligned}\mathbb{P}(Z \geqslant a) &= \mathbb{P}\Big(\bigcap_{m=1}^{K} (S_m \geqslant a)\Big) = e^{-(\lambda_1+\cdots+\lambda_K)a}, \\ \mathbb{P}(J = 1) &= \lim_{a\to+\infty} \mathbb{P}(J = 1, Z \geqslant a) = \frac{\lambda_1}{\lambda_1 + \cdots + \lambda_K}.\end{aligned}$$

□

The Theorem 1.5 claims, in practical terms, that the event occurred can be decided by *uniformly selecting among the events with probabilities* λ_m/Λ.
The two results are all we need to implement our simulation algorithm which is, essentially, an application of two theorems.

2 Application

Epidemiology has a long history starting from the $17th$ century([2]) and it's still a well-known field study, especially in this moment $(2020 - 2021)$, when the entire world is facing a new pandemic emergency: COVID-19.
This field research involves several other scientific branches; for instance, biology, chemistry, medicine and also mathematics. Traditional mathematical epidemiology aims to develop proper models in order to explain observations or predict outcomes, one of mainly used mathematical tools is *differential equation theory*. Although sometimes mathematical theories are able to carry out an explicit solution to a epidemic model problem (temporal function), nevertheless we should pay attention to what we can conclude observing the results, because the initial model, no matter how complicated, is *not perfect*.
The modern theoretical epidemiology relies on other available techniques and one of them is *network science*. There are many ways to exploit this young research area since it's quite intuitive how the *social interactions* are relevant in disease spreading, the most simple approach could be to perform *stochastic simulations*, it's a powerful way to study complex networks such as real-world social networks.
We implemented the *Temporal Gillespie Algorithm* [3] to simulate the so called *SIQRD* epidemic model which aims to describe the beginning of the *COVID-19 situation in Italy*.

2.1 SIQRD model

The SIQRD model is based on the notable SIR model and the derivation of this epidemic model can be found in [4]. The additional letters in the model's name have specific meaning: Q stands

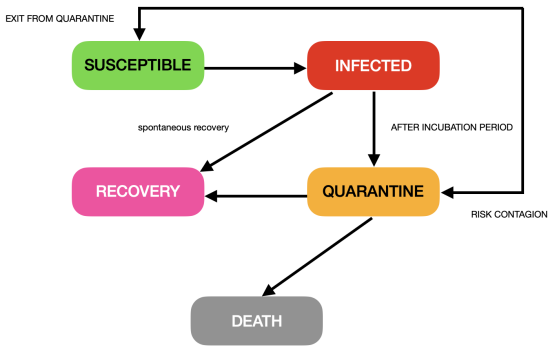

Figure 1: SIQRD model

for *quarantine* and D stands for *death*. This new model can be easily explained by the fig. 1 and below we describe each *state*:

- **Susceptible**(S): it's the initial state of every subject (node of the network) and the susceptible subjects can be infected or quarantined if they got in touch with infected people.

- **Infected**(I): the infected subjects can change the state to other three states: death, quarantine and recovery.

- **Quarantine**(Q): this extra state is to consider many real situations, for instance, one is quarantined whenever he/she had been in contact with an infected person or simply he/she is tested as infected.

- **Recovery**(R): once the subject is recovered he/she becomes immune to the virus.

- **Death**(D): we suppose that this state is followed only by the Q state because usually ill persons get cures before death.

2.2 Temporal Gillespie Algorithm

This method is flexible and applicable to many situation, we firstly describe the theoretical framework and then we'll give a pseudocode of our implementation to the SIQRD model.

Gillespie Algorithm

The Gillespie Algorithm (also known as *SSA* or *Gillespie's direct method*) made popular by *Daniel Gillespie* [5] for the simulation of coupled chemical reactions and then exploited by other scientific study (like economics or physics). It belongs to the big family of *stochastic simulations* and it's applicable to general *Poisson* processes with constant rate.
It's a suitable simulation algorithm for the spreading phenomena since the infection events could be considered random events and so they follow *Poisson distribution*. Thus, the main idea of this algorithm is to assume that *the waiting time between state transitions is exponentially distributed*

(Theorem 1.3).
Using the notation as in the theoretical part we can suppose that the possible events are finite set Ω and the waiting time τ for the first event follows the exponential distribution:

$$p(\tau) = \Lambda \exp(-\Lambda\tau).$$

Thus, according to Theorem 1.5, we consider the waiting time $\tau \sim Exp(\Lambda)$ for the next transition and the probability that m is the next event equal to $\pi_m = \lambda_m/\Lambda$.

Dynamic network

We haven't yet taken into account the dynamic structure of the real social networks and in doing so in [3] it was proposed an extension of Gillespie's algorithm: *Temporal Gillespie Algorithm*.
First of all we make events space Ω to be dependent on time and we denote the new set $\Omega(t)$ and following a similar approach to what we did in the precedent case we obtain the waiting time probability distribution:

$$p(\tau; t^*) \propto S(\tau; t^*) = \exp\left(\int_{t^*}^{t^{**}} \Lambda(t)dt\right),$$

where $t*$ is the time in which the last transition took place, $t^{**} = t^* + \tau$ is the time when the next transition takes place. We note that now $\Lambda(t)$ depends on time since the network's structure changes and so we have no analytical solution because the dynamics of the network considered is generally unknown or complicated.
Nevertheless, we can still implement the algorithm numerically. Algorithm 1 is a possible pseucode for simulating the SIQRD model, which can explain in a plain and simple way what we illustrated beforehand with mathematical language.

2.3 Experiments

Dynamic social network

Recently several research works about COVID-19 spreading have emerged and some of them leveraged the network structure of social interactions, generating a proper social network for simulations or even for theoretical purposes. In [6] a possible virus spreading network model is presented, such model is a *multi-layers network model* and each layer represents a particular social community such as workplaces and households. This idea was inspired by another work that simulated and analysed a virus (influenza) spreading phenomena [7]. See fig. 2 taken from the cited paper. The model considered human-human interactions (*social interaction network*) and also *lockdown* strategies that many countries adopted during the pandemic. In the paper many lockdown situations are analysed and some are revealed to be more effective than others in mitigating the spreading.
Here, based on the work done in [6], we modified the original static network into a *dynamic network*, dynamics are due to the random interactions between the populations while we kept fixed the layers. We built this network model by taking into account italian social structure and then we applied Temporal Gillespie Algorithm to the SIQRD model. We want to stress that it's difficult to map every single citizen because it would require enormous computational power, we believe that it's not necessary for studying the *general situation* of the pandemic. Thus, it's sufficient to build a *mini-scale* network which simulates the real-world society and at the end analysing the results carefully, keeping in mind the differences in term of networks' size.
Table 1 summarizes the features of the layers included in the model. The workplace layer was also intended to figure the school layer described by fig. 2.

Simulation method

We implemented and executed the algorithm, afterwords we compared the results of the sim-

Input: μ_0 : initial recovery rate, γ_0 : initial death rate and q_0 : initial quarantine rate.

1: Initialize the rates: $\mu \leftarrow \mu_0,\ \gamma \leftarrow \gamma_0,\ q \leftarrow q_0$.
2: New social contact network is created.
3: Draw a normalized waiting time until the first event from a standard exponential distribution, $\tau \sim Exp(1)$.
4: Find out the number of persons who are in contact with infected subjects and multiply this number by β_0. So we obtain β.
5: Compute $\Lambda = \mu + \beta + \gamma + q$.
6: **if** $\Lambda < \tau$ **then**
7: No events occurred before new contact network was created. Compute the difference of time: $\tau \leftarrow \tau - \Lambda$.
8: Return to step 4.
9: **else**
10: A transition occurred. Initialize remaining fraction of time before the change of social contact network: $r \leftarrow 1$.
11: **while** $r \times \Lambda \geq \tau$ **do**
12: Randomly select the transition $m \in \{I, R, D, Q\}$ with probability $\beta/\Lambda, \mu/\Lambda, \gamma/\Lambda, q/\Lambda$ respectively.
13: Update values and parameters for transition m. For instance, if the transition I takes place a subject is infected and update network's parameters.
14: Update β, μ, γ, q according to which event occurred and remaining time: $r \leftarrow r - \tau/\Lambda$.
15: Recompute β and Λ. Draw a normalized waiting time $\tau \sim Exp(1)$.
16: **end while**
17: Time expired and new social contact network is coming up. Return to step 2.
18: **end if**

Algorithm 1: Temporal Gillespie Algorithm applied to SIQRD model.

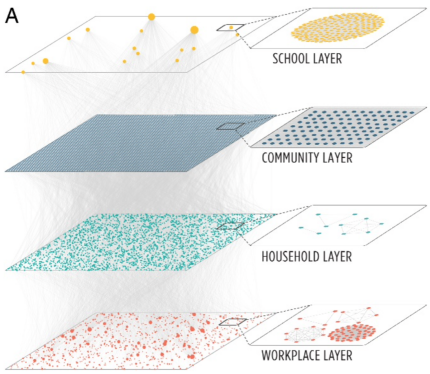

Figure 2: Example of a multilayer social network structure.

Layer	Type	Parameters
HOUSEHOLD	*Fully connected* between family members	Families proportion = [0.333, 0.271, 0.193, 0.151, 0.04, 0.013]
WORKPLACE	*scale-free* (non-ess.), *random* (ess.)	ba-degree = 4, ess-prop = 0.2
INTERACTIONS	*random*	interaction-prob. (with ess. workers) = 0.2, social-prob. = 0.01

Table 1: Layer type summary. See [8] for details about each network type.

ulations performed with the italian real contagion data[1]. As pointed out before we don't expect the model can 'predict' perfectly the contagion numbers, precisely the proportions w.r.t the real population, during the time but we hope that it can *capture the general trend* (increasing or decreasing) and the goal is actually achieved.
The period of time examined starts from 2020/02/21 to 2020/06/21 and we considered also the lockdown that the italian government imposed (2 months) in this interval of time. In some cases in order to eliminate randomness we ran the algorithm many times and computed the average results.

Other specifications

We want to stress also the fact that one subject (node) can be 'quarantined' only after a certain interval of time: *incubation period*, that has as mean value 14 days. Finally a node exits from quarantine status only if it's passed some time: observation period. The complete algorithm implementation can be found in the attached files.

Results

The fig. 3 contains what we obtained from 12 simulations. There is no perfect match between the real data and the simulation data but it's an expected outcome since the real contagion numbers could not be *real-time*; at the beginning of the pandemic there was no enough resources to test any people having the COVID symptoms (some patients were even asymptomatic) and so the number reported depended on how many tests were executed. In fact, as we can observe the contagion number had no drastic drop but it decreased over the time when more people were tested. The trend could be affected by other factors, for instance how the infection numbers are counted or how the data are collected. That's said, we can conclude that our model is *satisfactory* because it was able to capture the *increasing/decreasing trend* of the time series, especially the *contagion peak*. The model can even simulate the virus spreading during the lockdown. Indeed, the infection events can take place due to the interactions between people and essential workers or subjects within same family.
As last comments about the results, we can verify that the lockdown emergency protocol can be really effective as showed in [6] and by the real data, we can also note that at the end of the simulation there were almost no infected nodes, nevertheless we cannot conclude that the pandemic is over since the model considered *only a closed system* in which there are no external accesses to the system, this may cause a second contagion wave as happened in reality.

[1] https://lab24.ilsole24ore.com/coronavirus/

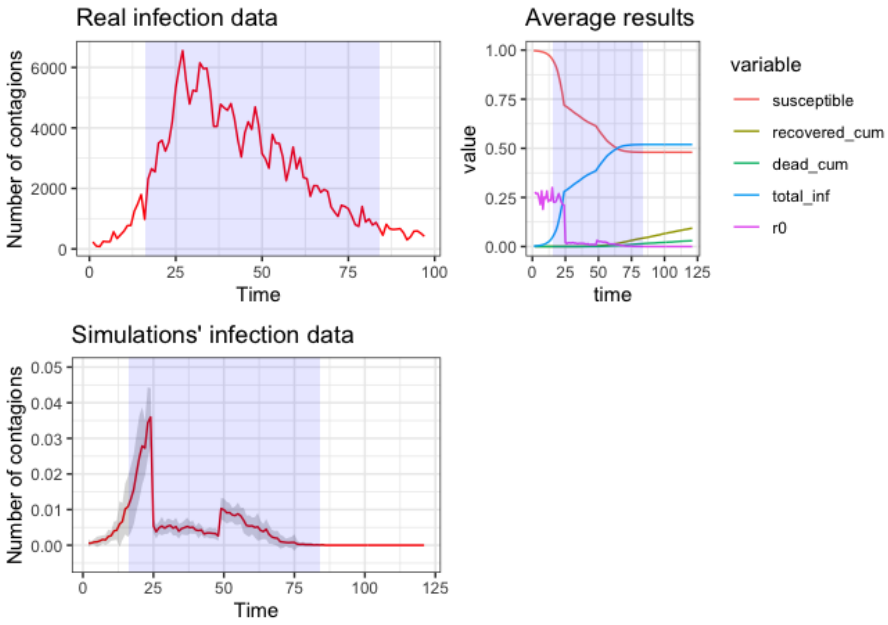

Figure 3: Average results from the performed simulations. with 1500 nodes. The simulation values are the percentages w.r.t the overall nodes. Shadow area indicates the lockdown period.

References

[1] P. Brémaud, *Markov chains: Gibbs fields, Monte Carlo simulation, and queues*. Springer Science & Business Media, 2013.

[2] Wikipedia contributors, "Epidemiology — Wikipedia, the free encyclopedia," 2021.

[3] C. L. Vestergaard and M. Génois, "Temporal gillespie algorithm: Fast simulation of contagion processes on time-varying networks," *PLOS Computational Biology*, vol. 11, pp. 1–28, 10 2015.

[4] I. Z. Kiss, J. C. Miller, P. L. Simon, *et al.*, *Mathematics of epidemics on networks*. Springer, 2017.

[5] D. T. Gillespie, "Exact stochastic simulation of coupled chemical reactions," *The Journal of Physical Chemistry*, vol. 81, no. 25, pp. 2340–2361, 1977.

[6] P. Maheshwari and R. Albert, "Network model and analysis of the spread of covid-19 with social distancing," *Applied network science*, vol. 5, no. 1, pp. 1–13, 2020.

[7] Q.-H. Liu, M. Ajelli, A. Aleta, S. Merler, Y. Moreno, and A. Vespignani, "Measurability of the epidemic reproduction number in data-driven contact networks," *Proceedings of the National Academy of Sciences*, vol. 115, no. 50, pp. 12680–12685, 2018.

[8] A. Barabasi and M. Posfai, *Network Science*. Cambridge University Press, 2016.