UNIVERSITY OF AMSTERDAM

REPORT

An exploration into nondeterministic modelling of infectious diseases

Students:
Bart Lankheet 12885797
Kyle Tunis 16245547
Lecturer: Valeria Krzhizhanovskaya

Course:
Introduction to Computational Science

October 26, 2025

Contents

1 Introduction

Computational modelling is an invaluable tool in the study of infectious diseases. Having previously explored Kermack & McKendrick's basic SIR (Susceptible/Infected/Recovered) framework [6], we now move on to more rigorous (and computationally demanding) investigations. While basic SIR equations and their extensions such as demographic effects or additional compartments (such as in the SEIRS or MSIR versions) can provide interesting and meaningful insights, they are also limited. These equations do not account for heterogeneous population mixing, nor for the ways in which random chance can influence the spread of a disease.

We will use stochastic, Event-Driven simulations to explore the effects that random chance can have on the dynamics of a SIR system, and discover the parameter configurations that can increase or decrease these effects. We will also investigate how the spread of disease changes when the naive assumption of homogeneous mixing is discarded by explicitly modelling different types of human networks. We will see how interventions can be made more effective by leveraging the network structure, even when the exact structure is unknown.

2 Theory

2.1 Event based modelling for the SIR model

2.1.1 The Stochastic SIR Model

The Stochastic SIR model shares most basic features with the deterministic SIR model as originally proposed by Kermack and McKendrick [6], including categorizing individuals as either Susceptible individuals who are able to be infected if exposed, Infected individuals who are currently infected with the disease and able to infect others, and Recovered individuals who have previously been exposed to the disease (or, in some cases, a vaccine) and are immune from infection. The equations for the deterministic SIR model are explained in subsubsection 3.1.1. The main distinction of the Stochastic SIR model is that, rather than individuals moving continuously between states at rates determined entirely by the solutions to ordinary differential equations, stochastic noise is introduced so that the rate that individuals move between states is nondeterministic. This noise can be introduced in a variety of ways, including Observational Noise in which the "observed" number of individuals in each state is noisy, or Process Noise in which the transition rates themselves are assumed to be noisy.

To model the SIR equations stochastically, we use an Event Driven Model based on Gillespie's First Reaction Algorithm (implementation details in subsubsection 3.1.2). This is an example of introducing Process Noise, in which the rate at which each event occurs is determined partially by the SIR differential equations and partially by a Random Number Generator. The result is a model which generally approximates the deterministic equations, but also introduces the possibility of stochastic extinction, in which the number of infected individuals drops to zero and no further infections are possible, as well as stochastic harmonic oscillations, in which stochastic effects cause oscillations which amplify existing oscillatory behaviour in the base equations.

These stochastic effects are the most noticeable when the population size, N, is smaller. As N increases, the underlying equations dominate the stochastic effects and the Stochastic SIR model becomes a closer approximation of the deterministic SIR model. Figure 1 shows how the stochastic effects decrease for larger values of N, with a near-perfect approximation of the deterministic results when $N = 1,000,000$. The effects of Stochastic Harmonic Oscillations can also be clearly observed in the first subplot where the noisy peaks and troughs match those of the deterministic model, but with increased amplitude.

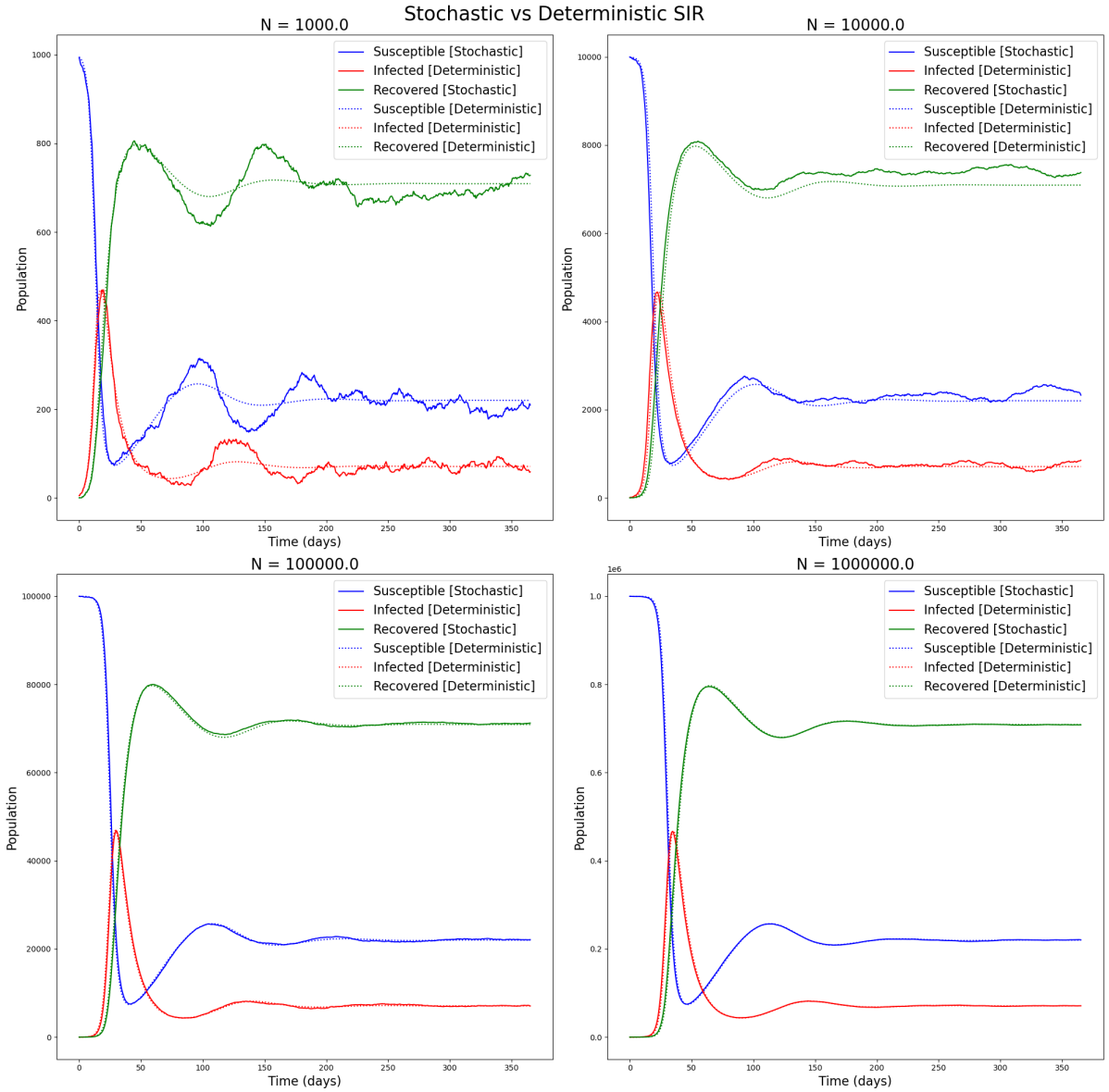

Figure 1: The results of a single iteration of the stochastic model compared to the equivalent deterministic result for various values of N. As N increases, the effects of noise are diminished and the stochastic model begins to closely resemble the deterministic results. Simulated with $\beta = 0.5, \gamma = 0.1, \mu = 0.01$ ($R_0 = 4.55$).

2.2 Spreading of disease in networks

In this report we will also take a more detailed approach by studying the spread of diseases in networks. Networks allow us to study the stochastic process of spreading through a specific social network with a resolution of a single person.

A network is a system of nodes (N), connected to one another by links (L). In our use-case these nodes will represent persons, and links will represent a connection between two persons that could cause a disease to be spread between them. A simple example of a social network with 4 nodes and 3 links is shown in figure 2, where people are represented by nodes 0 through 5, and their social connections by the links between them[2].

Networks can be very versatile, making use of single direction links, or weighted links between nodes. In our case, we will use simple, bidirectional, unweighted links.

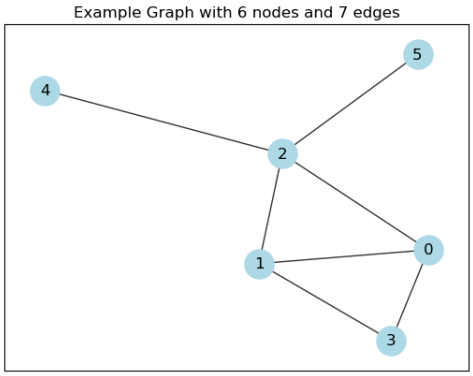

Figure 2: A simple network of 6 nodes and 7 links

2.2.1 Centrality measures

There are a number of interesting statistics one can use to analyse a given network. A few of these are useful to study when considering disease spread. We will discuss those in the list below.

- **Node degree** - A node has degree k where k is the amount of links attached to the node. Useful for our research because nodes with high k have a high chance of having an infected neighbour [7].
- **Node clustering coefficient** - The local clustering coefficient C is given by $C = l_{occupied}/l_{possible}$ where $l_{occupied}$ is the links that exist between neighbouring nodes and $l_{possible}$ is the total possible links between neighbouring nodes. If a node has a degree of k, the local clustering coefficient is given by $C = \frac{l_{occupied}}{k(k-1)}$. The more clustered a graph is the more interconnected its nodes are, and the higher we expect disease to spread[7].
- **Diameter** - The diameter of a network measures the maximum of shortest paths in the network. We expect a large diameter to slow disease spread[7].
- **Node betweenness** - A nodes betweenness is given by $g = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$ where σ_{st} is the total count of shortest paths between nodes s and t, and $\sigma_{st}(v)$ is the count of shortest paths between nodes s and t that go through node v. A node with high betweenness centrality is a very central node. A high average betweenness might indicate the network[7].
- **Closeness** - The normalised closeness C of node u is given by $C(u) = \frac{N-1}{\Sigma d(u,v)}$, where $d(u, v)$ is the shortest path distance between node u and node v. A network with high average closeness should spread an infection rapidly[7].

2.2.2 Network types

There are many different shapes a network can take. We will investigate the disease spread on 3 network types that all share that they can be randomly generated using an algorithm and one or two predefined parameters.

- Erdös-Rényi (ER) networks
  This network type, named after mathematicians Paul Erdös and Alfréd Rényi, is generated by letting n nodes exist, and giving each connection between any nodes have a p chance to exist. This results in a random network with very little structure to it. The degree distribution of an ER network looks looks like a binomial distribution. Figure 3 shows an exemplary ER network. Around the circumference of the graph float the orphaned

nodes which are not in any way connected to any other node. There are also smaller islands of nodes which are connected amongst each other but not to the rest of the graph. The largest of these islands is called the giant component. This part contains the bulk of the nodes and will be the part we're most interested in studying. This is also the part we consider for the diameter calculations, because conventionally the path length is considered to be infinite if there is no path between the two nodes, and that would set the diameter of most ER networks to infinity, which would not be a very useful statistic.[7]

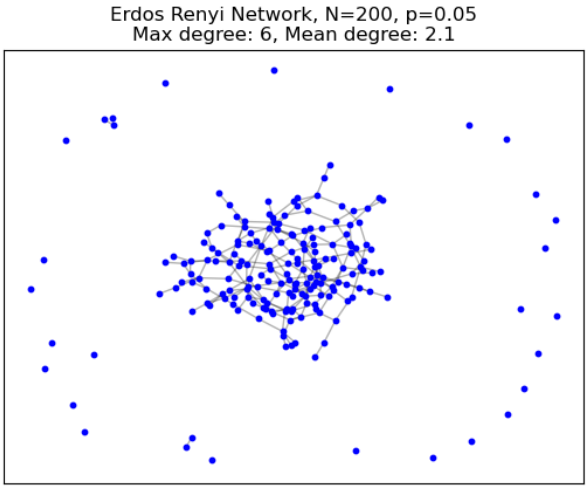

Figure 3: A randomly generated Erdös-Rényi network, with N=200, p=0.05

- Barabási-Albert (BA) Networks
  Albert-László Barabási writes in *Network science (2016)*[2] that most real life networks take the approximate form of a scale free network, which is a network for which the degree distribution scales as a power law: $p_k \sim k^{-\gamma}$, which means that there are very many nodes with a low degree, and few nodes with a very high degree. The generator for a BA network takes 2 parameters: the amount of nodes N and the amount of connections for each node m. Generating a BA network requires it to grow from 1 starting node. Each iteration a node is added which makes m connections to other nodes, where the probability of connecting to node i is given by $\Pi(k_i) = \frac{k_i}{\sum_{j=1} k_j}$, so: the more connections a node already has, the higher the probability the new node will connect to it as well[2].

  A very interesting property of BA networks is that they contain hubs: nodes with a very high connection degree. These have a very small chance of showing up in ER networks. Figure 4 shows a BA network with N=100 where the node with the highest degree centrality has 19 connections, while the highest measured degree in the ER network is only 6. These hubs will be very important in the spreading of disease.

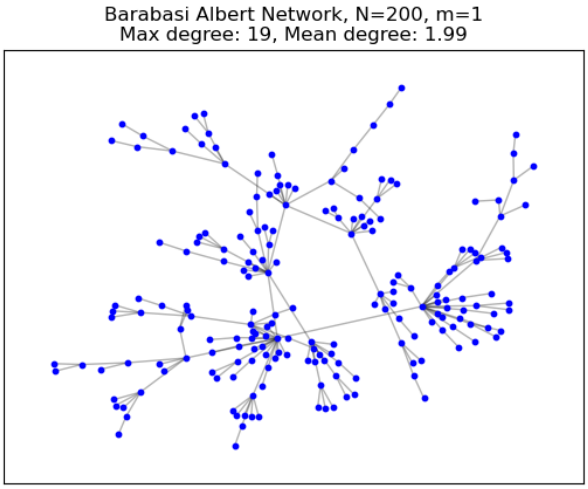

Figure 4: A Barabási-Albért network with N=200 and m=1. One can clearly see the hubs in this network which have many connections, and will be vital facilitators of disease spread.

- Watts-Strogatz (WS) networks
  We also studied WS networks, because in their article *Collective dynamics of 'small-world' networks* (1998)[11] Duncan J. Watts and Steven H. Strogatz propose a network structure which models social networks. We think that infectuous diseases could spread via the structure of such social networks. The WS network is generated by first generating a ring of N nodes, where each node is connected to their k nearest neighbours, and giving each node a probability p to create a shortcut to a completely random other node. This creates a network with relatively small diameter, and high clustering. An example is shown in figure 5. What should immediately be clear is that a network like this has a very specific degree distribution, with most nodes having a degree centrality of 5, and only some some having one of 6.

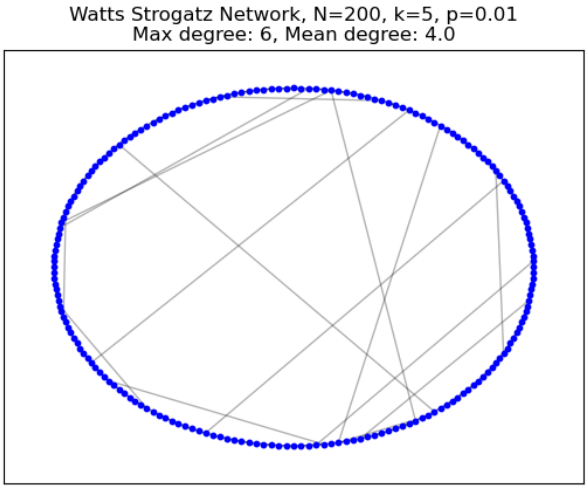

Figure 5: WS network with N=200, k=5 and p=0.01. The shortcuts are clearly visible crossing through the middle of the graph.

3 Numerical methods

3.1 Event-based Approach

3.1.1 Deterministic SIR Equations

The deterministic SIR model is defined by the following equations:

$$\frac{dS}{dt} = \mu - \beta SI - \mu S \tag{1a}$$

$$\frac{dI}{dt} = \beta SI - \gamma I - \mu I \tag{1b}$$

$$\frac{dR}{dt} = \gamma I - \mu R \tag{1c}$$

where β is the rate of infection, γ is the recovery rate, and μ represents the birth and death rate. In these equations, $R_0 = \frac{\beta}{\gamma+\mu}$.

3.1.2 Gillespie's First Reaction Algorithm

There are many versions of Gillespie's Algorithm for Event-Driven models, including the Direct method, First Reaction method, and others [5]. We have used the First Reaction method, which is both simple to implement and also sacrifices no accuracy, making it a theoretically robust approach.

We implemented the First Reaction algorithm using object-oriented Python, with an Event class storing each event's rate function and result function, so that the main code running the simulation can calculate rates and execute the results of the selected event while remaining agnostic to which event is being considered. The parent Experiment object stores original and current state of the population, experiment parameters, results, and other important information. This approach allowed us to quickly create, run, and iterate experiments.

The name, rate function, and result function for each event can be seen in Table 1:

Event Name	Rate Function	Result Function
Birth	μN	$X++$
Transmission	$\beta X \frac{Y}{N}$	$X--$, $Y++$
Recovery	γY	$Y--$, $Z++$
Death X	μX	$X--$
Death Y	μY	$Y--$
Death Z	μZ	$Z--$
Import	$\epsilon\sqrt{N}$	$X--$, $Y++$

Table 1: List of Events used in Gillespie's First Reaction algorithm with their associated rate and result functions.

β, γ, and μ have the same meaning as in Equation 1, and ϵ is the Import Rate, assuming an external force of infection. This rate is proportional to the square root of the population size, which has been found to be accurate in real-world populations [3]. Thus, smaller populations experience a greater relative Import Rate than larger populations.

3.1.3 Termination Condition

For all event-based simulations, we generally run many iterations of the same configuration to examine a range of possible results. The termination condition for these iterations is based on the Relative Standard Error (RSE), or $\frac{std\_err}{mean}$. The RSE is averaged across all time steps and the simulation is terminated when the averaged RSE is less than 5% %. This condition

is bounded, with a minimum of 5 iterations and a maximum of 100. Figure 6 contains a plot showing how various versions of the model converge over multiple iterations.

3.2 Integration Method

We perform all mathematical integrations using `solve_ivp` from the `scipy` Python package, which integrates ordinary differential equations (ODEs) using 4th order Runge-Kutta integration [10]. This is the same method used by Keeling and Rohani, as "it provides a good balance between accuracy and simplicity" [5].

3.3 Network Generation and Simulation

To import and generate our networks we used the `Networkx` python package. We used the generator functions for the relevant network types, and the `from_numpy_array` function to import the connectivity matrix describing the Sociopatterns data [1].

To simulate the spread of an infectuous disease in a network we will use the SIR compartment model in its most simple form. We will be disregarding demographics and infection induced mortality for simplicity. That leaves only the parameters β for infection rate and γ for recovery rate. We also set the weight of each link to 1.

To simulate the SIR models applied on these networks we used the python package `NDlib`. Specifically the `model` class which keeps track of the different compartments and the equations relating to the SIR model, and the `model.iterate` method to take one simulation step. Each step `NDlib` evaluates every node. A susceptible node will have a probability of $k_I\beta$ to get infected and an infected node has a probability γ to recover [8]. All network simulations were run for a predetermined amount of timesteps and multiple iterations were run with newly generated networks, after which the results were averaged.

To simulate disease spread using the SIR model on our networks we chose the initial condition of having a random 5% of the population be infected. This was to guarantee an epidemic and to prevent premature extinction.

3.4 Vaccination Algorithms

The vaccination algorithms' implementations all make use of a test function which takes a test accuracy between 0 and 1 as a parameter, and then returns the real status of the tested node if a random number falls below the accuracy parameter, and otherwise it returns the wrong wrong status. If a node is in the 'recovered' compartment it would not get vaccinated, because we asume that in a real life scenario a person would know they have been infected in the past.

3.5 Figure Generation

Representations of networks were made using the `Networkx` [1] and `matplotlib` [4] python packages. All other figures are generated using the `matplotlib` Python package [4].

4 Results and discussion

4.1 Parameter Influence on Stochastic SIR

To explore the Stochastic SIR Event-Driven model, we first run the model through a range of possible parameters, varying R_0, γ, and N to observe the resulting population dynamics. We were particularly interested in which parameters had the strongest effect on the stochastic noise, as well as the effect of changing the parameters on the S/I covariance.

4.1.1 Estimating Parameter Influence with RSE

Before diving into the specific results, it is interesting to first consider the RSE convergence discussed in subsubsection 3.1.3. Increased variability and transients will increase the RSE and delay convergence, while a configuration that closely approximates the deterministic results will have a very low RSE that converges quickly. The plots in Figure 6 therefore provide an interesting proxy for examining the impact that parameters have on the model dynamics.

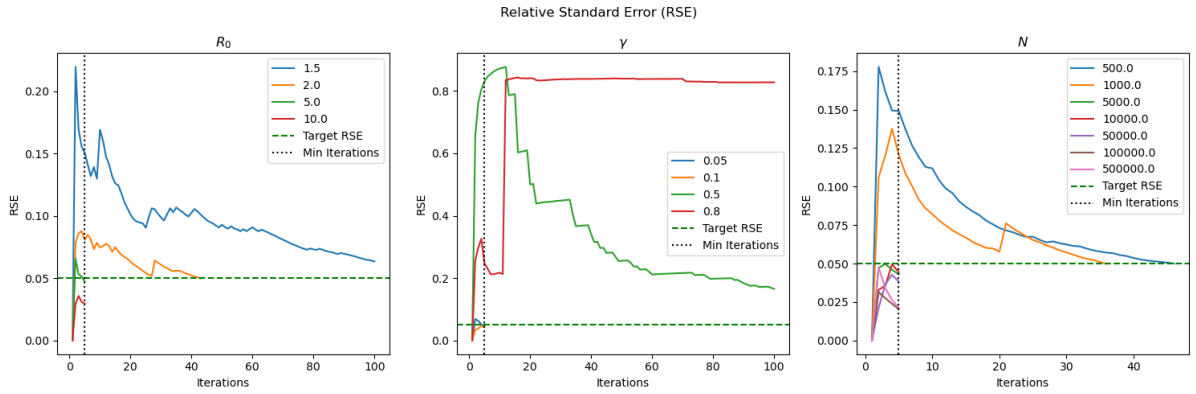

Figure 6: Relative Standard Error at each iteration of various runs of the Stochastic SIR model, showing how various parameters impact the rate at which iterations of the model converge on a mean. Simulated with $\mu = 0.01$ and $\epsilon = 0$ (no imports) for all versions, and $N = 5000$, $R_0 = 5.0$, and $\gamma = 0.1$ when not the examined parameter.

We can observe that for low values of R_0 the model experiences significant variability, while for even moderate values the RSE converges quickly, suggesting a nearly deterministic behaviour. Similar results can be seen when increasing N, although even low values of N exhibit less volatility than seen in R_0. However, γ is entirely different. Here, we see that *lower* values actually demonstrate more consistent behaviour, and as the value increases to even moderate values of 0.5 or 0.8 (meaning an infectious period of 1.25 or 2 days), the RSE increases dramatically and does not even come close to converging at 5% within the 100 iteration limit. This suggests that the stochastic model is extremely sensitive to variations in γ. This is consistent with our intuitive understanding, if we consider that infected individuals spending less time in the Infected state means that there is simply less time for stochastic noise to "smooth out" before individuals recover.

How fascinating that we can glean such insights from a single statistical measure, before even examining the results themselves! That said, let us proceed to the *real* results.

4.1.2 Parameter Influence

To examine our parameters of interest R_0, γ, and N, we run the simulation through a range of values for one parameter at a time, while holding other parameters fixed. When not the subject of the given experiment, these values are fixed at $R_0 = 5.0$, $N = 5000$, and $\gamma = 0.1$. For all simulations, $\mu = 0.01$ and $\epsilon = 0$, meaning the disease is never imported from external sources. If stochastic extinction occurs, the disease will not return for the duration of that simulation. We begin each simulation iteration with 5 infected individuals.

The first parameter we examine is R_0, the basic reproductive number. Figure 7 shows the S/I covariance of R_0 values between 0.5 and 10, as well as the infections over time of selected values in that range. We can see that the covariance is 0 for the nonepidemic states when $R_0 <= 1$. As R_0 increases, the covariance drops precipitously before slowly climbing again, with the minimum value when $R_0 = 3.5$. Looking at the plot of infected individuals over time,

we can see that larger R_0 values have lower transients compared to the relatively volatile lower values. This is consistent with the conclusions drawn from the RSE plots in subsubsection 4.1.1.

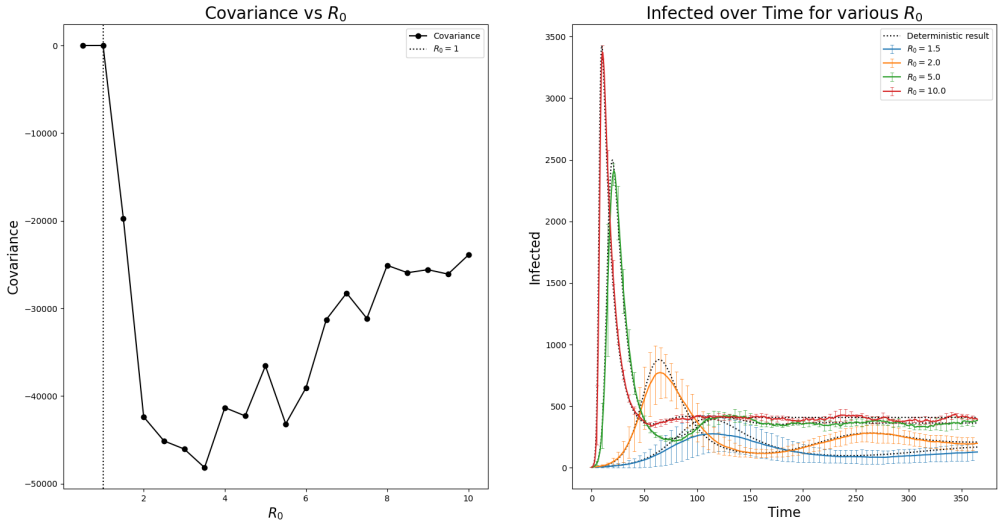

Figure 7: On the left, S/I covariance plotted for values of R_0 from 0.5 to 10. On the right is the plot of infected individuals over time for selected R_0 values. The mean result across multiple iterations is shown in the solid colour, with error bars of 1σ. The dotted lines represent the deterministic result for each parameter combination.

Next, we examine the effect of γ on the model dynamics. Following the same approach as with R_0, we simulate γ values from 0.05 to 1.0. The results of this experiment are shown in Figure 8. The covariance graph shows an interesting phenomenon, where covariance starts extremely low and increases, before hitting a critical threshold (here 0.4) where it drops dramatically, before eventually starting to climb again for higher γ values.

We have plotted the infections over time twice for the γ exploration, as both the linear and log scales reveal interesting findings. The linear scale (centre) plot shows large transients for both high and low γ values, with moderate values showing less variation. This finding can help explain the behaviour shown in the covariance plot.

The log-scale plot shows another interesting finding: that the γ values of 0.5 and 0.8 fail entirely to track their deterministic results. The $\gamma = 0.5$ line falls consistently below its deterministic counterpart after the first oscillation, while the $\gamma = 0.8$ line collapses entirely due to stochastic extinction occurring within 150 days in every iteration of the simulation.

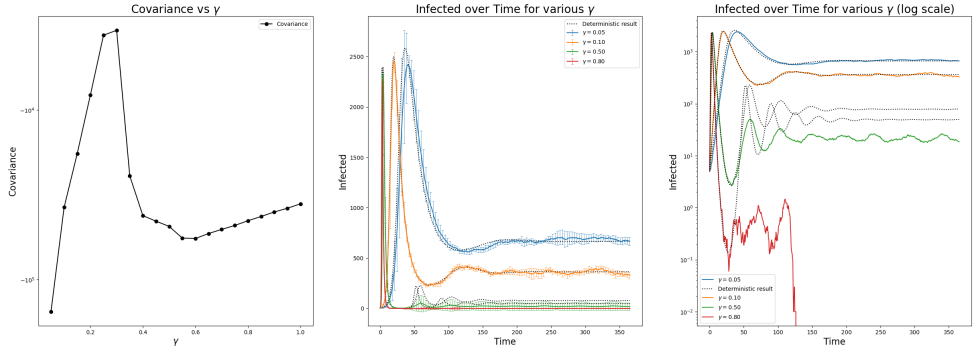

Figure 8: On the left, S/I covariance plotted for values of γ from 0.05 to 1. In the centre is the plot of infected individuals over time for selected γ values. The mean result across multiple iterations is shown in the solid colour, with error bars of 1σ. The dotted lines represent the deterministic result for each parameter combination. The left data shows the same data as the centre, but plotted on a log scale and without error bars to more clearly observe the data close to the x axis.

Finally, we examine the impact that various values of N have on the model. This is a simpler case to interpret than R_0 or γ. The covariance plot in Figure 9 shows that covariance decreases on a power law trajectory as N increases, while the infection-over-time plot shows that variance and transients both decrease as N increases. This is consistent with our interpretation of the RSE plots in Figure 6 as well as the conclusions drawn from Figure 1.

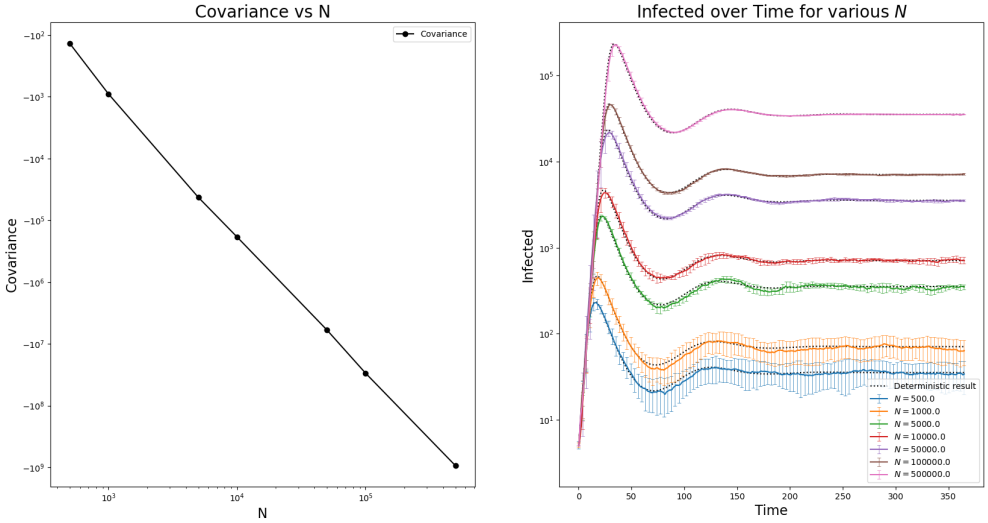

Figure 9: On the left, S/I covariance plotted for values of N from 500 to 500,000. On the right is the plot of infected individuals over time for selected N values. The mean result across multiple iterations is shown in the solid colour, with error bars of 1σ. The dotted lines represent the deterministic result for each parameter combination.

To validate our conclusions, we can construct a set of parameters that we predict to yield highly variable results, and another predicted to yield low-variance, nearly deterministic results. We ran two sets of simulations for 20 iterations each:

- High variance: $R_0 = 1.2$, $\gamma = .99$, $\mu = 0.01$, $N = 1000$, $\epsilon = 0$

- Low variance: $R_0 = 10.0$, $\gamma = .09$, $\mu = 0.01$, $N = 500,000$, $\epsilon = 0$

Plotting each iteration separately to visualize the resulting variance, we get the results in Figure 10. This confirms our conclusions, definitively showing that stochastic influences are more powerful for lower values of R_0 and N and higher values of γ.

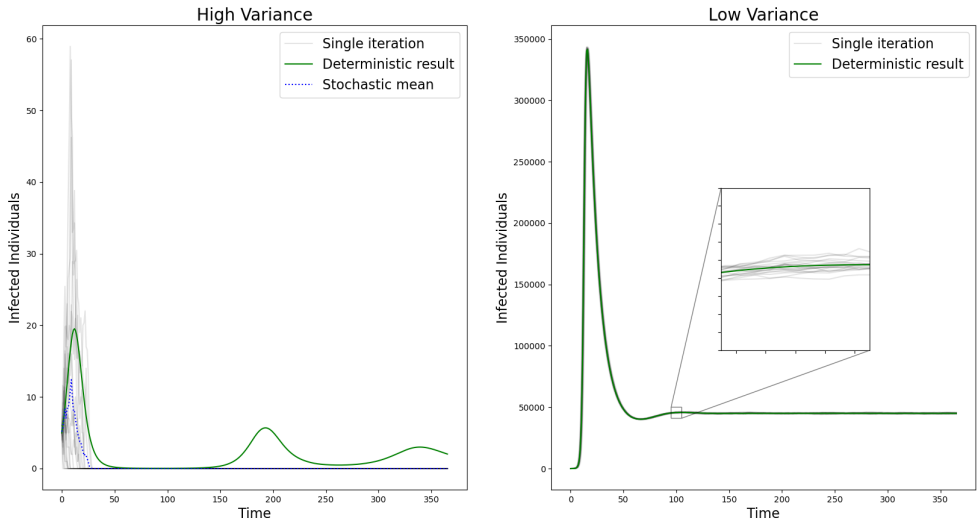

Figure 10: Plot of 20 individual iterations of the stochastic model with two parameter configurations, one predicted to result in high variance (left) and the other predicted to result in low variance(right). The equivalent deterministic results are shown, as well as the stochastic mean result for the high-variance configuration.

4.1.3 Stochastic Extinction and Critical Community Size

When using stochastic models to study diseases, one important metric to consider is the Critical Community Size (CCS). This is the population size threshold above which the disease should not be expected to go extinct. For our purposes, we define the CCS as the community size at which fewer than 1 extinction occurs per year on average, including extinctions which occur after the disease is re-introduced by an import after previously having gone extinct. For that reason, we now use $\epsilon \neq 0$, resulting in an import rate of $\epsilon\sqrt{N}$.

Figure 11 shows the number of extinctions and time spent extinct per year at various populations sizes. These charts show a clear relationship where both number of extinctions and time spent extinct decrease as N increases. Here, we see that the CCS is approximately 500 individuals.

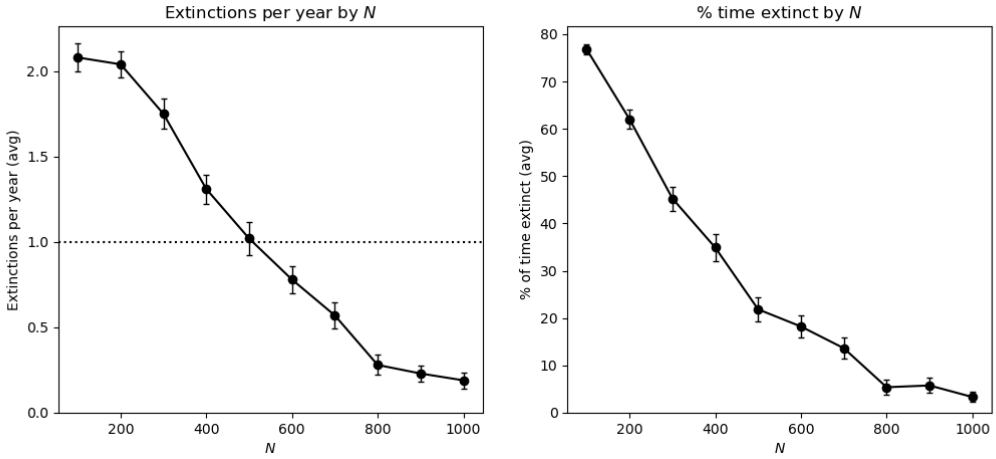

Figure 11: The left plot shows the number of extinctions per year as a function of N. The right plot shows the percentage of time the disease was extinct for each N value. Error bars represent the standard error. Simulated with $R_0 = 5$, $\gamma = 0.2$, $\mu = 0.01$, $\epsilon = 0.2$.

We also perform a similar experiment by setting $N = 500$ (the approximate CCS from the previous example) and varying R_0 instead. This answers the question of "How infectious would a disease have to be to not go extinct in this population?" Unsurprisingly, we find the threshold to be approximately $R_0 = 5$, the same value that resulted in this CCS. This provides a handy validation that we can find the same threshold point ($R_0 = 5, N = 500$) by approaching the problem from either side. The results of this experiment are shown in Figure 12.

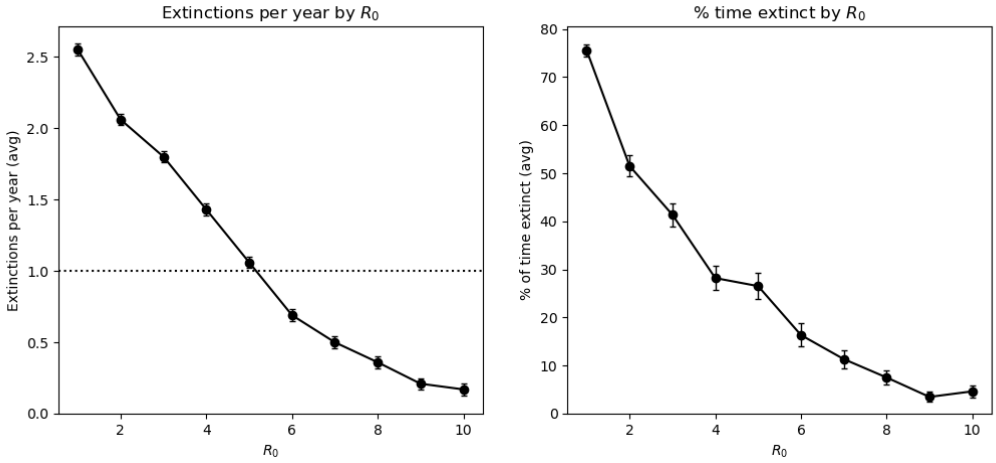

Figure 12: The left plot shows the number of extinctions per year as a function of R_0. The right plot shows the percentage of time the disease was extinct for each R_0 value. Error bars represent the standard error. Simulated with $N = 500$, $\gamma = 0.2$, $\mu = 0.01$, $\epsilon = 0.2$.

4.2 Comparing Network Types

We studied the properties of three different types of randomly generated graphs: Erdös-Rényi, Barabási-Albért and Watts-Strogatz networks. To gain insight in how these networks are structured we plotted the centrality measures mentioned in section 2.2.1 versus the amount of nodes N in the generated network. The results are visible in figures 15-17. We did this for multiple

name(nodes,links)	Sociopatterns(375,1639)	ER(375,1615)	BA(375,1484)	WS(375,1500)
avg degree	8.74	8.61	7.91	8.0
avg clustering	0.556	0.0233	0.0664	0.146
diameter	2	5	5	5
avg betweenness	0.00262	0.00533	0.00497	0.00603
avg closeness	0.507	0.336	0.353	0.308

Table 2: Comparison of various graph metrics for real-world (Sociopatterns) and randomly generated network models (ER, BA, WS). Parameters: ER: $(N, p) = (375, 0.0234)$, BA: $(N, m) = (375, 4)$, WS: $(N, k, p) = (375, 8, 0.4)$

values of the second parameter. For the WS network we chose to keep the amount of neighbours k constant and vary the probability of forming shortcuts p, because the shortcuts are what makes these networks interesting for study.

To simulate see which of these three network types simulates real life data the best we compared our selection of randomly generated networks with a network provided by Sociopatterns [9]. We approximated the Sociopatterns network by choosing fitting parameters for our models. For the ER network we calculated p via $p = \frac{2*L}{N*(N-1)}$ which comes down to dividing the amount of present links by the total amount of possible links. For our BA network we approximated the m parameter by dividing the target average degree by 2, to simulate incoming and outgoing directions in the graph generator process, and rounding to the closest integer. For our WS network we set k as close to our target degree as possible and set the p parameter as low as we could while preserving a minimal diameter.

The recovered parameters and the network properties of these networks are shown in Table 2. We kept the node size the same for all networks and tried to approximate the average degree. The differences in network structure become clear when one compares how similar the degree, and the clustering. The clustering of the randomly generated networks clearly differs from the real-world data, but there are also differences between the properties of these synthetic models themselves.

To see which network type could be used to approximate the real life data the best we simulated the SIR model on each of them and plotted the resulting graphs in figure 13. The network with the most similar trajectory is the BA network. It has both the closest peak time and peak intensity.

4.3 Vaccination Strategies on a Network based Approach

Our network based approach really shows its usefulness in designing and analysing research strategies. When an epidemic occurs it is important to know who to vaccinate to make the biggest impact with the available resources. We will test 3 vaccination strategies with different amounts of available infection tests, test accuracies and vaccination availability.

- Random vaccination
  With random vaccination we will vaccinate v nodes at random. We see this as the baseline vaccination strategy.

- First friend vaccination
  We will select a random node, and vaccinate one of its neighbours. This is intended to make use of the scale-free nature of BA and real-life networks where it is likely that a neighbouring node has a higher degree.

- Second friend vaccination
  Like the first friend vaccination strategy, we will select a random node, and select at random one of its neighbours, but then we will select one of its neighbours, and vaccinate

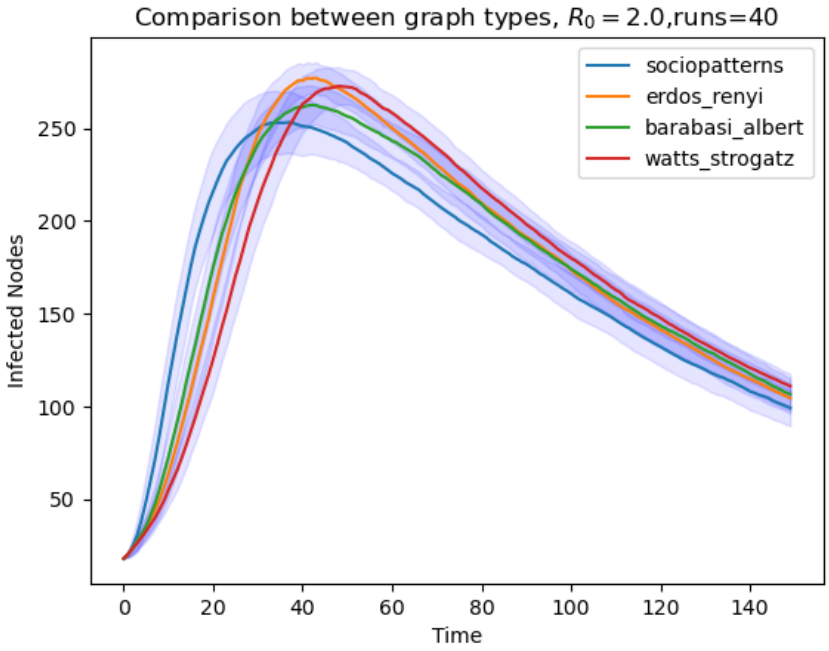

Figure 13: Results of SIR simulation on the real-world network and our three randomly generated networks. Parameters: $\beta = 0.02, \gamma = 0.01$. 40 runs, error bands display one standard deviation.

them. The idea behind this is that if it is likely that a neighbour is better connected, then should a neighbour of a neighbour not be even more connected?

We ran our experiment with available vaccinations from the set $(0, 1, 3, 5, 10)$ and test accuracies from the set $(1, 0.75, 0.5)$. The results of our vaccination experiments are shown in figure 14. The largest impact is given by the first friend strategy with a test accuracy of 1. Not surprisingly higher test accuracy seems to result in a higher strategy effectiveness. What is also clear from this plot is that the first friend strategy seems to make the biggest difference. One reason for the second friend strategy underperforming could be that the Sociopatterns network is so connected that on average a locally largest hub has already been reached with just one step.

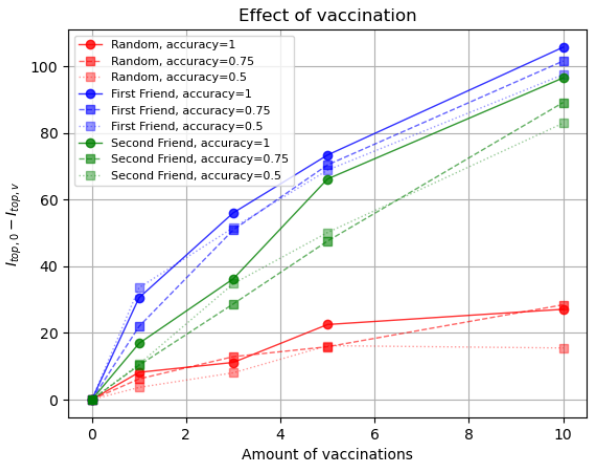

Figure 14: Effect $I_{top,0} - I_{top,v}$ of vaccination on infection peaks. Disease parameters: $\beta = 0.2, \gamma = 0.1$. Average over 100 runs. Infection curves of each vaccination strategy can be found in the appendix in section 6.2.

5 Conclusions

In our investigation of stochastic disease modelling, we were able to successfully isolate the effect that modifying R_0, γ, and N have on the disease dynamics. We determined, using analysis of both the simulation results and various statistical measures, that a larger population, a more infectious disease, and a longer infectious period all contribute to a stochastic result that more closely approximates the deterministic results of the equivalent SIR equations evaluated numerically. Simulations using opposite parameters will likewise contain more variance and larger transients. These conclusions are confirmed by our results in Figure 10. We also examined the system when import of the disease from external sources is considered, successfully identifying the critical community size for a disease with a specific parameter configuration.

A potential area of future research would be to create an approach to determine the CCS for *any* parameter configuration of the model, without the need to guess and check each potential community size. Additionally, while the use of Gillespie's First Reaction algorithm was appropriate for this level of investigation, a deeper dive including larger populations or evaluating along wider ranges of parameters would benefit greatly from a more efficient algorithm, such as Gillespie's Direct Method.

Our investigation of networked systems also yielded interesting findings. Our results show that is possible to approximate real-world networks using randomly generated networks. Barabási-Albért networks are shown to be most suited for our application, showing the biggest similarity to real-world networks in SIR model experiments. We used our real-world network to experiment with vaccination strategies and found that our first friend strategy described in section 4.3 worked the best for this network. It is clear that test accuracy matters when it comes to vaccinating strategies, and that high accuracy is preferable. For future research we recommend quantitively testing the similarity of BA networks to real-world networks in SIR model experiments to see if vaccination strategies could be tested on BA network based models of the real-world. Another topic for future research is the effectiveness of the second friend vaccination strategy in larger networks.

References

[1] Daniel A. Schult Aric A. Hagberg and Pieter J. Swart. Exploring network structure, dynamics, and function using networkx. *Proceedings of the 7th Python in Science Conference (SciPy2008)*, page pp. 11–15, Aug 2008.

[2] Albert-László Barabási and Márton Pósfai. *Network science*. Cambridge University Press, Cambridge, 2016.

[3] M S Bartlett. Measles periodicity and community size. *J. R. Stat. Soc. Ser. A*, 120(1):48, 1957.

[4] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[5] Matt J. Keeling and Pejman Rohani. *Modeling Infectious Diseases in Humans and Animals*. Princeton University Press, 2008.

[6] William Ogilvy Kermack and A. G. McKendrick. A contribution to the mathematical theory of epidemics. *Royal Society of London, Series A*, 115(772), 1927.

[7] Dr. V. Krzhizhanovskaya. Introduction to computational science, September 2025.

[8] Giulio Rossetti, Letizia Milli, Salvatore Rinzivillo, Alina Sîrbu, Dino Pedreschi, and Fosca Giannotti. Ndlib: a python library to model and analyze diffusion processes over complex networks. *International Journal of Data Science and Analytics*, 5(1):61–79, December 2017.

[9] Juliette Stehlé, Nicolas Voirin, Alain Barrat, Ciro Cattuto, Vittoria Colizza, Lorenzo Isella, Corinne Regis, Jean-François Pinton, Nagham Khanafer, Wouter Van den Broeck, and Philippe Vanhems. Simulation of an seir infectious disease model on the dynamic contact network of conference attendees. *BMC Medicine*, 9(87), July 2011.

[10] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[11] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, Jun 1998.

6 Appendix

6.1 Network properties

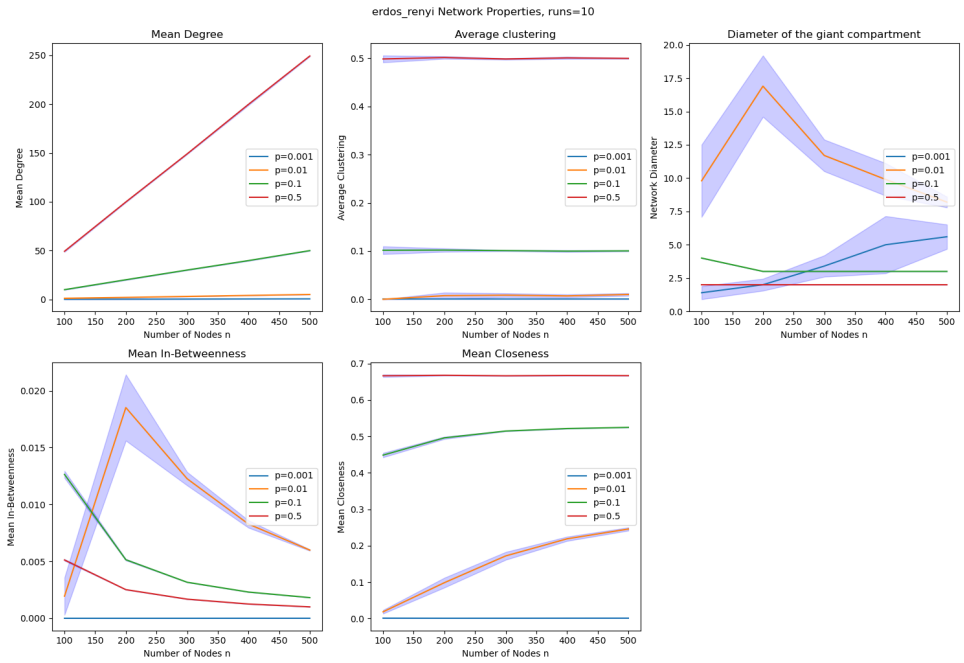

Figure 15: Network properties for Erdös-Rényi networks. Averaged over 10 runs. Error bands show one standard deviation.

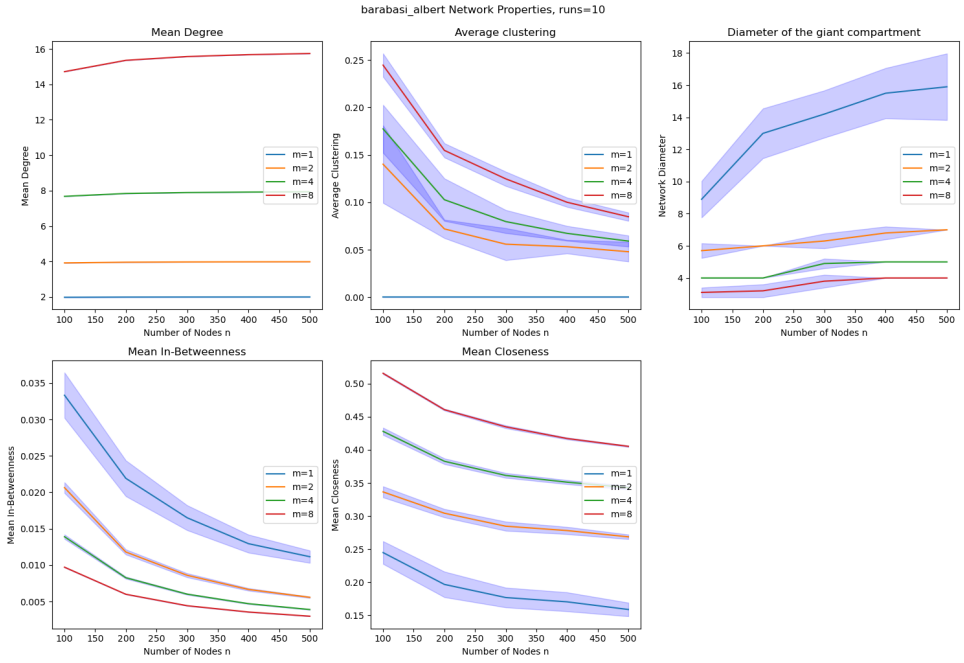

Figure 16: Network properties for Barabási-Albért networks. Averaged over 10 runs. Error bands show one standard deviation.

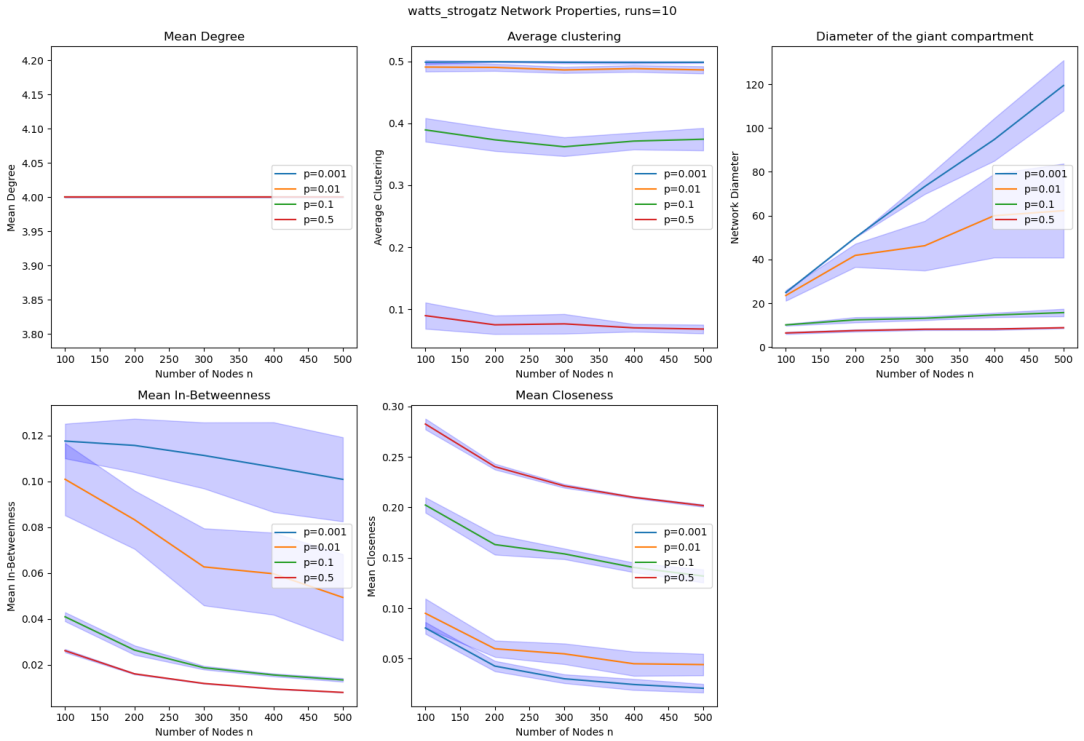

Figure 17: Network properties for Watts-Strogatz networks. Averaged over 10 runs. Error bands show one standard deviation.

6.2 Vaccination experiments

6.2.1 Test accuracy = 1

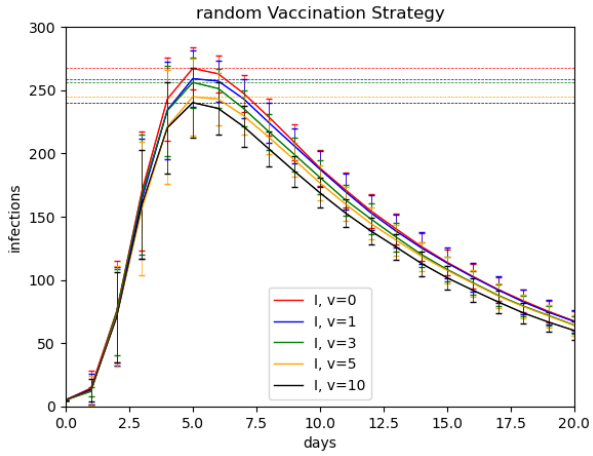

Figure 18: Vaccination strategy experiment. Disease parameters: $\beta = 0.2, \gamma = 0.1$

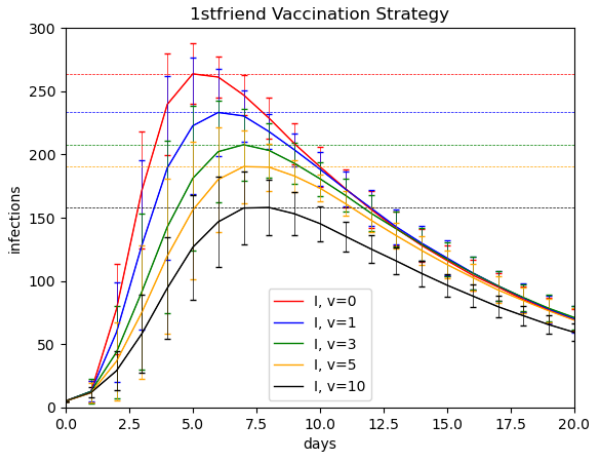

Figure 19: Vaccination strategy experiment. Disease parameters: $\beta = 0.2, \gamma = 0.1$

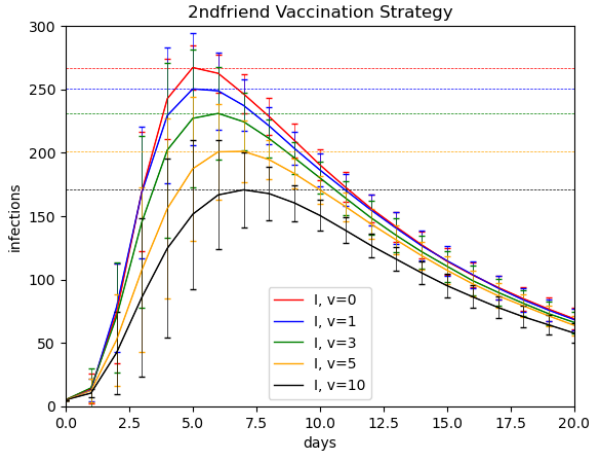

Figure 20: Vaccination strategy experiment. Disease parameters: $\beta = 0.2, \gamma = 0.1$

6.2.2 Test accuracy = 0.75

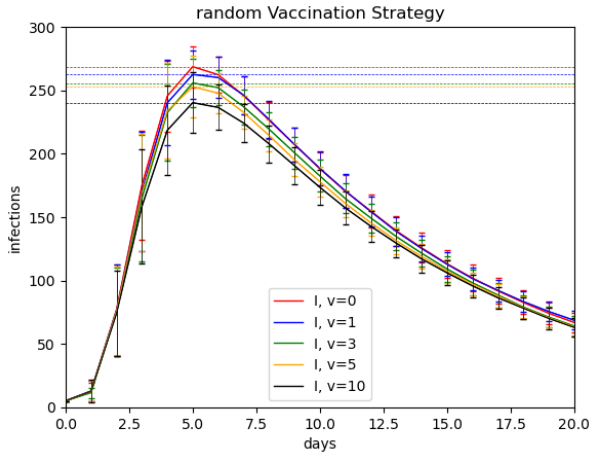

Figure 21: Vaccination strategy experiment. Disease parameters: $\beta = 0.2, \gamma = 0.1$

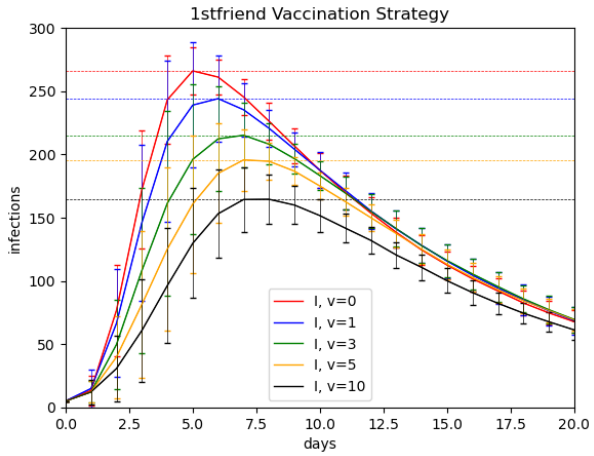

Figure 22: Vaccination strategy experiment. Disease parameters: $\beta = 0.2, \gamma = 0.1$

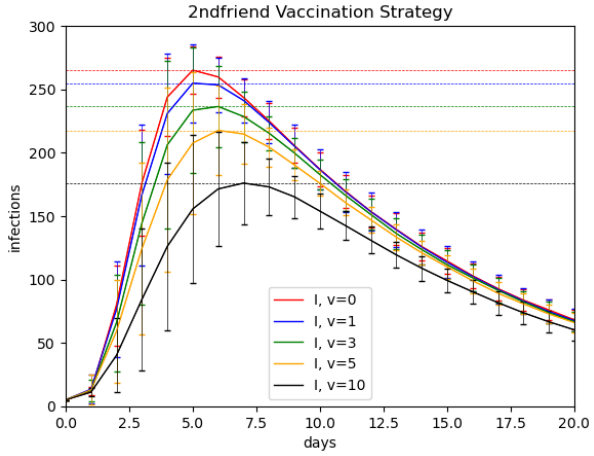

Figure 23: Vaccination strategy experiment. Disease parameters: $\beta = 0.2, \gamma = 0.1$

6.2.3 Test accuracy = 0.5

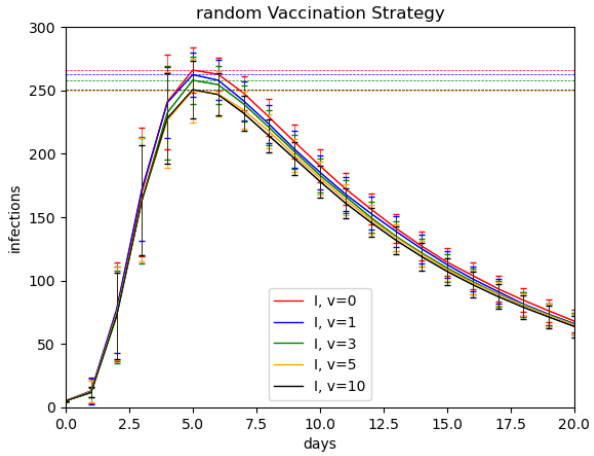

Figure 24: Vaccination strategy experiment. Disease parameters: $\beta = 0.2, \gamma = 0.1$

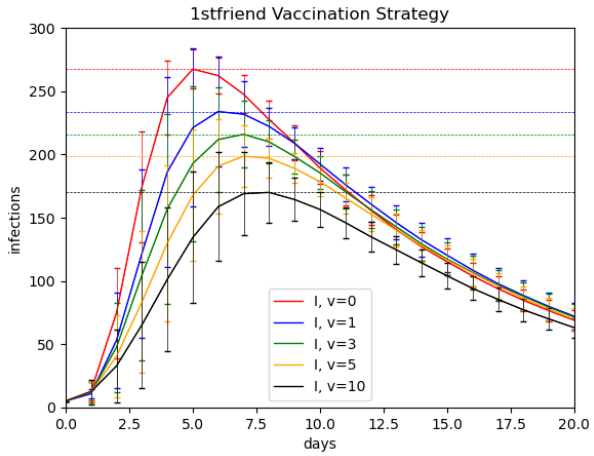

Figure 25: Vaccination strategy experiment. Disease parameters: $\beta = 0.2, \gamma = 0.1$

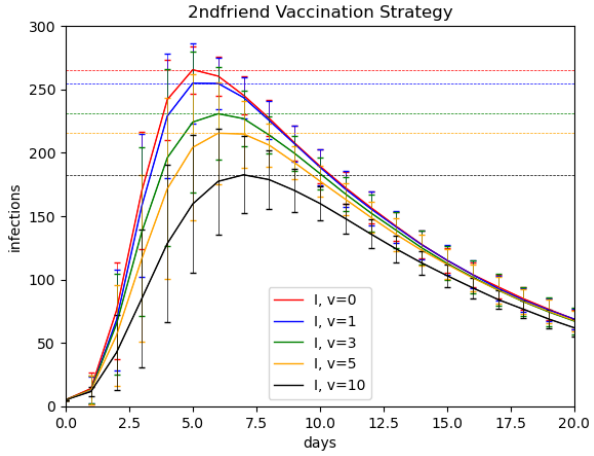

Figure 26: Vaccination strategy experiment. Disease parameters: $\beta = 0.2, \gamma = 0.1$