Modeling Epidemiological Spread on Contact Networks

Elina Shirolkar

University of North Carolina at Chapel Hill
Department of Statistics and Operations Research

Thesis Advisor: Professor Shankar Bhamidi
April 21, 2022

Contents

Chapter 1

Network Diffusion

1.1 Contact Networks

A contact network describes information about the location and proximity between individuals over a fixed period of time. As individuals come in contact, they are able to spread both intangible material, such as information and ideologies, and tangible material, such as bacteria and viruses. Contact networks can describe spread through a variety of sources, such as airline networks, animal populations, plant populations, and computers. [Kleinberg, 2010] In the majority of cases, the manner of spread will determine the intensity of the spread. For example, in the case of an epidemic, an airborne virus will spread rapidly in comparison to a disease requiring close contact. Much like the diffusion of a disease, the diffusion of ideas also spread with differing intensities. However, the key differentiator between the two is the process of decision-making that occurs when an individual chooses to adopt an idea. This difference outlines two approaches to diffusion within contact networks: decision-based diffusion and probabilistic diffusion. An idea will disperse through decision-based diffusion, while a disease will spread through probabilistic diffusion. Random modeling using probabilities can be applied to assess the spread of disease. The simplest way to model this probabilistic diffusion is by using the branching process model.

1.2 The Branching Process

The branching process utilizes a tree-like network to visualize the spread of a contagion through numerous individuals. Each of these individuals represent a node in the tree. [Ethen, 2018] A node is a basic unit of a data structure that is able to link to other nodes; each link between these nodes is implemented by a pointer system. The branching process tree consists of a single node at the top, labeled the 'root node'. The root node will then connect to a set of nodes in the row below it. Every node in the tree will connect to a node in the level above it, aside from the root node.

Additionally, the contagion will have a specific contagion probability, which will determine if an infection will widely spread, or ultimately die out. This probability will aid in determining the number of nodes in every subsequent row of the tree. A higher contagion probability will result in a larger average number of nodes in each row of the tree. If the number of nodes in each row remains consistent throughout the branching process, the total number of nodes will continue to grow exponentially. There are two common possibilities for a contagion in a branching process model – in the first possibility, the branching process will reach a round in which no individual is infected, which will result in the contagion dying out within a finite number of steps. The second possibility is that the contagion will continue to infect individuals in each wave, which will result in an infinite spread of the infection through the contact network.

The probabilistic branching process relies on the basic productive number, denoted R_0. R_0 is the expected number of new individuals that have been infected by a single individual in the previous round. R_0 is also known as the measure of 'average contagiousness' of a pathogen. For example, if an individual affects an average of 2 individuals in each round, $R_0 = 2$. R_0 can be found by multiplying the dispersion of the pathogen, or the total number of new people that an individual has come in contact, k, with by the contagion probability, p. We can calculate the average contagiousness as $\mathrm{R}_0 = pk$.

We are able to come to a variety of conclusions by analyzing R_0. If $R_0 < 1$, the contagion is 100% likely to die out after a finite number of rounds. If $R_0 > 1$, then there is a probability greater than 0% that the contagion will continue to infect a minimum of 1 person in each round. While the differences in R_0 may seem to be minute, having R_0 fall just above or just below 1 will largely impact whether a contagion spreads rapidly through a contact network. In the case of epidemic spread, an $R_0 > 1$ may induce the transformation from a small-scale disease outbreak into a large epidemic.

There are multiple precautionary measures that can be taken to reduce the R_0 in the case of an epidemic. To reduce the quantity k, individuals can be encouraged to quarantine and social-distance. As a greater percentage of the population takes these precautions, the total number of people that an individual will come in contact with will also be reduced. To reduce the contagion probability p, a community can encourage better sanitary practices to reduce the spread of germs, such as regularly washing hands and wearing face coverings. When computing the R_0, it is essential to recognize that these preventive measures are purely voluntary, and may not be executed by the entire population.

There are many simple examples attributed to the common use of branching processes, one being the popular question proposed by the reputable explorer and anthropologist, Francis Galton: *"How many male children must each generation of the family have for the family name to continue forever?"* [Zitkovic, 2016] We can assess many differing situations for simple branching processes to discern Galton's proposition. For these situations, let us begin with one individual, at time $n = 0$. We can keep track of the total number of individuals, Z, in each round. At time $n = 0$, $Z_0 = 1$. Suppose that this first individual reproduces a certain number of times, and then dies. The total number of individuals at time $n = 1$ will vary based on the contagion probability. If $Z_1 = 0$, the population will die out and there would be no more reproduction possible. If $Z_1 > 0$, Z_1 would reproduce a random number of times based on the contagion probability, and then die. It is possible to calculate the total number of individuals in the second generation using the offspring distribution, as follows: $Z_2 = \sum Z(1, k)$. This process would continue for all living individuals until $Z_m = 0$ for all $m >= n$. At this point, the population would go extinct.

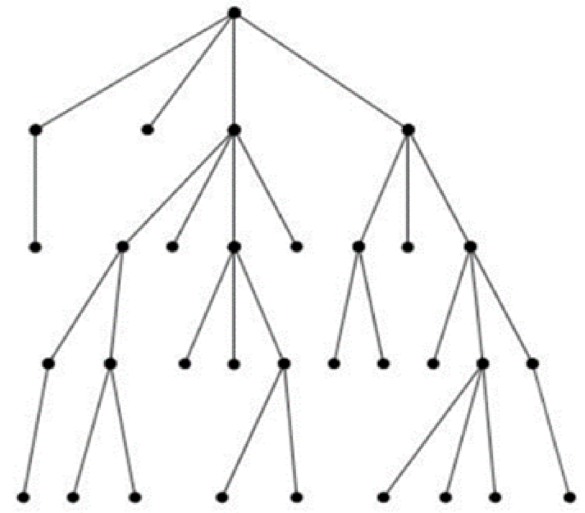

Figure 1.1: This figure visualizes generational growth beginning at a root node at time $n = 0$. [Kumar, 2016]

If the offspring distribution above were to be given by $P(s)$, the generating function to find the total population size Z_n at a time n would be $P_{Zn}(s) = P(P(...P(s)...))$, for $n >= 1$ with n number of P's. Suppose we have $p_0 = 1$ and $p_n = 0$. In this case, the population would become extinct after the first generation of individuals. Suppose we have $p_0 = 0$, $p_1 = 1$, and $p_n = 0$ for $n >= 2$. In this case, the population size would remain at $Z_n = 1$.

1.3 Cluster Diffusion

Many viruses, such as the virus COVID-19, may be ‘over-dispersed’, meaning that they spread in the form of clusters. We can analyze the spread of a contagion in the form of clusters. For example, a virus contagion may disperse through the individuals of a single household. This household could be identified as a single cluster. The virus would then spread to other individuals who may also be members of a different households, or different clusters of individuals.

Suppose we have an individual, i, living in a house with an additional individual, j. [Borowiak *et al.*, 2020] We can identify p_H as the probability that individual i infects individual j with the virus. We could also identify p_G as the probability of infecting an individual j who does not live in the same house, or cluster, as individual i. Households will contain differing numbers of people. Due to this fluctuation in cluster size, we can identify $(pi)_k$ as the probability that individual i belongs to a cluster of size k. We can also find the average cluster size by finding the mean of all cluster sizes, $\mu = \sum_k (kp_k)$. Therefore, the generating function for a cluster of size k would be $G_H(z) = \sum_k (p_k z^k)$. For a population of size $N = mn$, with m being the number of households and n being the number of people in each household, the average measure of contagiousness would be calculated as $R_0 = \mu N p_G$.

Each individual i will have a unique generating function. Suppose we have two individuals, X and Y. If we were to find $X + Y$, or the combined generating function of the cluster size that is impacted by the interaction between X and Y, we would have to multiply the generating functions of individuals X and Y. This would result in an exponential increase of the cluster size, depending on the number of individuals involved in the interaction. As interactions between individuals increase, the total cluster size that is impacted by the contagion rises at an exponentially higher rate.

Chapter 2

Game Theory in Networks

2.1 Virus Diffusion and Contagion Interaction

As the virus contagion disperses at a high rate, the likelihood of attaining epidemic or pandemic level increases. During a pandemic, there is little to no herd immunity within the community. [Wey *et al.*, 2020] Herd immunity will develop when there are enough individuals who are immune to a disease, making an escalated spread of the virus unlikely. Herd immunity is beneficial to the community due to the lessened risk of those who are not immune to the virus contracting the virus. The most common way to achieve herd immunity is through either vaccinating the community, or allowing the virus to spread naturally through enough individuals to reach herd immunity.

If there is no vaccination available for a virus, the virus will naturally spread through the community by human interaction. We can measure the spread of this virus using the mathematical models developed above. However, many of these calculations, such as the average contagiousness R_0, will be crude measures of the true impact of the virus. The true values of spread will be dependent on both the environment and the behavior of the infectious individuals who have contracted the virus. For example, if an individual with a virus is in a tightly packed space with poor ventilation, nearly all members of the room will contract the virus. In this case, the true average contagiousness will be much higher than the R_0 value.

Individual behavior is also crucial in determining the true R_0 value. In a common workplace, there will be two types of workers. The first will be the 'transparents', who will take precautions if they have contracted the virus. These precautions will include regular testing and safety measures. The second will be the 'opaques', who will not take precautions if they have contracted the virus. Of all infected individuals, approximately $10 - 20\%$ of individuals, who are more likely to be opaques, may be responsible for as much as $80 - 90\%$ of virus transmission. [Tufekci, 2020]

This leads to a circumstance comparable to the 'prisoner's dilemma'. [Asu Ozdaglar, n.d.] In the case of a workplace, there will be two conflicting necessities – the safety of the workers, and the productivity of the company. If the R_0 value of the virus is initially low, the company may want to remain open to maximize its economic incentives. However, if the workers were to get infected with the virus, both the company and the employees will be negatively affected.

The prisoner's dilemma takes into account the selfish incentives of users. This brings about the concept of 'game theory', an efficient manner of assessing the decision-making process of individuals. [Chang *et al.*, 2020] Many decisions will be beneficial to one individual, but have a negative impact on another. The common usage for game theory involves assessing the levels of benefit for each individual in comparison to the harm that the decision will cause. Each 'player' will attempt to choose the option that will result in the optimal benefit to themselves. This benefit is referred to as the 'utility' of the player. Moreover, the best strategy will be dependent on the choices of the other players. In the case of epidemic spread, the carrier of the virus will be at a gain, while the people who are infected by the virus will be at a loss. Within a group of people, there will always be those who are solely interested in their own benefit, and will not make decisions that will lead to the socially optimal result. For example, while choosing to quarantine is the decision that will be optimal for all parties, there will be those who select their personal enjoyment over the safety of themselves and the surrounding community.

2.2 Decision-Making Modeling

This decision-making process can be modeled in multiple ways. The first manner is through 'self-learning', in which individuals will develop their own understanding of the disease through gaining knowledge. The second manner is through imitation, in which people are influenced by the decisions of the surrounding population. This is determined by the way in which the contact network is constructed. For example, to model the distribution of a vaccine, one can simulate the ‘committed vaccinators’, who will choose to take the vaccine without considering the benefits or losses of taking it. The surrounding population will watch this action and likely mimic this behavior. Over time, the vaccine will be taken by much of the population. This demonstrates the development of ‘behavioral clusters’. These behavioral clusters could also have a negative impact on the community – those who do not take safety precautions against the virus may inspire others to also not take precautions.

This behavior and interaction develops contact networks through the community. These networks are influenced by a multitude of factors, including the social, cultural, and geographic characteristics of the community. The interactions within these networks can be represented using network links. These networks can be either static or dynamic. A static network will remain unchanging as a contagion spreads through the network, while a dynamic network will continue to grow in size as the contagion spreads. The spread of an epidemic can be modeled using a dynamic network.

Within a network model, individual behavior is represented using ‘utility functions’. These utility functions demonstrate the game-theoretical decision-making process. The network model for a virus will contain an adjustable level of transparency, level of interaction between individuals, and average contagiousness R_0. A common method of modeling decision-making in this interactive manner is by using agent-based modeling.

Chapter 3

Agent-Based Models

3.1 Agent-Based Modeling Concepts

An agent-based model is a model composed of multiple entities called ‘agents’. Each agent is able to make decisions based on its current situation and a set of pre-determined rules. Agent-based modeling is used to emulate and analyze real-world systems. [Bonabeau, 2002] This form of modeling is extremely useful for hypothesis testing, which is a statistical manner of analyzing samples from a population to make estimates about the behavior of the entirety of the population. Hypothesis testing is used to make estimations about real-world situations.

Agent-based modeling is commonly used to make predictions in many real-world situations. For example, this technique is used to assess traffic and customer flow, stock market risk analysis, collective panic behavior, and even the length of rides at amusement parks. Agent-based models are utilized in situations involving changing behaviors and strategies. For example, fluctuations in the financial market are influenced by a variety of conditions. The agent-based model allows users to examine a variety of strategies and the resulting behaviors of the agents. This allows for the faster implementation of new protocols, and lessened risk in the real financial market. Agent-based models are also an excellent way to limit the spread of COVID-19 within the workplace. Many workplaces have faced the difficult decision between employee safety and monetary optimization. However, the likelihood of being infected with COVID-19 within a facility is very high, due to the close-knit contact networks within the workplace and the high average contagiousness of COVID-19. By using agent-based modeling, facilities will be able to simulate the spread of COVID-19 under different conditions and take the correct precautions in order to find the optimal balance between employee safety and economic standing.

There are many benefits of using agent-based modeling. These models provide a method of examining 'emergence', or behaviors that are only seen when the parts of a system interact with each other. Agent-based models also allow users to simulate and visualize complex problems, which are often difficult to solve using real-world data. It is also very simple to add and subtract agents from the model, and to change the behaviors of the agents. When running agent-based models, a simulation of real-world data is created for analyzation. Agent-based models are also useful in assessing natural systems. For example, it is unethical to perform a real-world human test to collect information about virus infection. In this case, it is only possible to perform this test using an agent-based simulation.

Agent-based models are often utilized when the behavior of individuals is ‘nonlinear’, meaning the change in the behavior of each output is not proportional to the change in the behavior of each individual. A nonlinear system is often created when the behavior of individuals involves learning and adaptation. A nonlinear system results in a high level of unpredictability, and is only easily analyzed through simulation.

An important step for creating an agent-based model is developing a strong conceptual model. [Auchincloss & Garcia, 2015] The conceptual model will consist many parts, including the experiment plan, the agents and their characteristics, the design of the simulation space, and the behavior of the agents.

3.2 Building an Agent-Based Model

The model will begin with a group of randomly chosen agents, each placed in a specific position within the simulation space. These agents will follow a specific set of rules that have been determined by the user. These rules will influence the position of each agent and the relationship with the surrounding agents. Agent behaviors, also known as the utility functions of the model, will allow the agent to practice decision-making. The agent will be given multiple choices and will rank these options to make a behavioral decision. Agent interactions are determined through stochasticity, or randomness. [Jalayer *et al.*, 2020] As agents interact, they will be put in various situations in which they must make decisions. The agents will produce a variety of responses based on their environment.

An example of an agent-based model is the COVID-19 model. [Cuevas, 2020] This model contains two types of agents, agents A_k (susceptible individuals) and B_k (infected individuals). Each time the simulation is rerun, the characteristics of the agents are randomly reassigned. As the simulation is run, the agents will make either local or long-distance moves. The probability of moving locally is $1 - a$. The a-value for long-distance displacement will generally fall between 0.6 and 0.8. Therefore, it is more likely for the agent to move locally. The probability of being infected, or P_i, is impacted by the health condition characteristics and the prevention measurement characteristics of the agents.

Each agent also has a risk characteristic, r, describing the level of contagiousness of the individual. r is a random number generated within a uniform distribution. During each k iteration of the simulation, a random number r between 0 and 1 is selected for P_1. The value of r will determine the risk associated with each agent. r values that are closer to 0 represent agents that are more immune. r values close to 0.2 represent the normal transmission rate of an individual. A high transmission rate will have r values closer to 1. When the simulation is run, A_k will be infected if its r value is less than

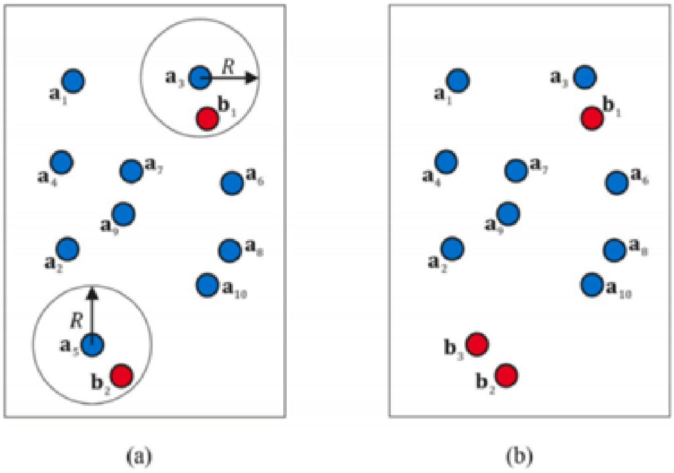

Figure 3.1: The value of R describes the level of contact under a certain risk contagiousness r. This figure demonstrates the (a) initial and (b) final configurations of infected agent b_j inside an enclosed area. The decision process will determine whether agent a_i is infected. [Cuevas, 2020]

or equal to the probability of infection P_i. Every agent A that gets infected is deleted from A and added to B. This model will be run many times to produce a distribution of outcomes.

3.3 Network Analysis Using Agent-Based Models

The use of agent-based models is an increasingly common approach within epidemiological research. The application of ABMs closely coincides with a second approach, called social network analysis. [El-Sayed *et al.*, 2012] Social network analysis involves the analysis of social networks in order to understand the impacts of different characteristics of the network on the risk characteristic, r, of the network. This approach takes an observational direction, gathering data through methods such as surveys and random sampling. However, social network analysis has its limitations. The largest limitation is the trade-off between the quality of data collection and the level of generalizability of the data. Carrying out surveys amongst large samples of individuals may not collect data with enough technicality to apply the results to an entire population. Contrarily, focusing on the detail in the surveys may introduce the confounding variable of homophily to the data, decreasing its generalizability to the larger population. The social concept of homophily involves a higher frequency of communication between the people in a network who are similar (in their daily routines and liking), encouraging a level of homogeneity to the data. [Khanam *et al.*, 2020] This may result in a homoscedastic distribution of the data, which would fail the regression diagnostic assumption of having a constant variance within the data. Therefore, we take the original approach of using a stochastic agent-based model to analyze our network data.

The combination of network analysis with the use of ABMs allows us to deeply understand the spread of a contagion through a population with a recognizable degree of social interaction. ABMs allow researchers to simulate this analysis to produce inferences about the population through the use of programming. We will be using the EpiModel [Jenness *et al.*, 2018a] package within R to carry out our agent-based modeling.

Chapter 4

EpiModel for Agent-Based Modeling

4.1 What is EpiModel?

EpiModel is an open-source software package that is used in the R programming platform. [Jenness *et al.*, 2018b] R is a language that is used to conduct statistical analysis and create mathematical graphics. It is commonly used by statisticians and data analysts and is supported by the R Foundation for Statistical Computing. R provides an easy way to produce high-quality graphics and plots. The R environment simplifies the mining and manipulation of data by providing a variety of effective tools for data analysis.

The Epimodel package allows its users to simulate and analyze infectious diseases by using a mathematical model. EpiModel was created in order to further the exploration of modeling amongst students and advanced researchers. This package will allow us to simulate agent interactions through the use of stochastic agent-based models. These network models are created using the methods of Exponential Random Graph Models, or ERGMs, from the suite of software packages in R called **statnet**.

The EpiModel Package characterizes the following processes that establish the transmission of an infection through a population [Jenness *et al.*, 2018a]:

1. The Contact Process
2. The Infection Process
3. The Demographic Process

4.2 The EpiModel Network Model

This paper will highlight the use of the EpiModel network model. This network model provides an adjustable and controllable structure for showing the contact between individuals in a population. The network simulation in EpiModel is implemented using the **ergm** package, specifically the Markov Chain Monte Carlo algorithm. The Monte Carlo algorithm is used to simulate the probabilities of differing outcomes that are difficult to predict due to the presence of other random variables in the data. [Kenton, 2020] This algorithm allows the network model to vary stochastically (randomly) over time.

Each epidemiological model type in EpiModel, called the base model, accepts a different set of parameters. These parameters include the rate of infection, rate of recovery, the number of time steps in the simulation, and the size of the simulated population. The simulated examples within this paper will follow the Susceptible-Infectious-Recovered/Immune (SIR) compartmental model type. [Jenness *et al.*, 2018a] The SIR model type is a compartmental model consisting of the following three states:

- State 1: Those who are susceptible (S) to being infected by the disease. If a susceptible individual comes into contact with an infected individual, the susceptible individual will contract the disease according to a certain infection probability.

- State 2: Those who are currently infected (I) with the disease. Infected individuals are able to infect others with a specific infection probability.

- State 3: Those who have already recovered (R) from the disease and are now immune to the disease, or the deceased who have been removed from the system.

Examples of diseases that follow the SIR model type include influenza, measles, and COVID-19. Using EpiModel to analyze the SIR model of a specific disease allows us to predict the manner in which the disease will spread through a population and observe the impacts of various public health interventions on the epidemic.

The stochastic network model that will be highlighted in this paper involves an arrangement of discrete individuals, acting as nodes within the system, and the partnerships between the individuals, acting as the edges between the nodes. One node is capable of having multiple edges. Each partnership has a certain dissolution rate, which will determine the average duration of a partnership. Furthermore, transitions between states are stochastic, meaning individual transitions are randomly assigned based on a distribution that is determined by the chosen set of parameters. Additionally, the network model is represented through discrete time, meaning multiple events can sequentially occur in a single time step.

4.3 Exponential Random Graph Models

The package statnet implements Exponential Random Graph Models (ERGMs) in order to simulate network frameworks according to specific nodal patterns. [Wasserman & Pattison, 1996] These patterns include the density of the simulated network, triad closure due to homophily between nodes, and other features.

ERGMs are used to predict the probability of the existence of an edge between two nodes. [soc, n.d.] Many may resort to the use of standard regression methods, however ERGMs allow us to analyze the interactions between nodes, edges, and the predictor network at a much deeper level. When analyzing social interactions, it is not enough to view only the binary relationships between the pairs of nodes. As once stated by Jacob Levy Moreno and Helen Hall Jennings, “a pertinent form of statistical treatment would be one which deals with social configurations as wholes, and not with single series of facts, more or less artificially separated from the total picture.” Therefore, we must look at the larger picture of human social interactions. Using standard regression would violate the basic assumption of independence, since ties between individuals in the real world are often brought about through homophily and association, and are often clustered. ERGMs allow for the modeling of these dependent nodes.

This simulation framework supposes the random graph model of “Erdős-Rényi”, which acts as the null hypothesis for our ERGM framework. It is assumed that any variation from this random graph model and the simulation is due to chance. [Hayes, 2021] The Erdős-Rényi random graph is a network of n nodes. The edges between these nodes form with a probability p. This formation of nodes follows a Bernoulli distribution, [Upe, n.d.] which is the discrete probability distribution of a random variable taking the value 1 with probability p, and the value 0 with probablility $1 - p$. In an Erdős-Rényi random graph, this probability p is found by calculating the network density of the graph. This value is found by dividing the total number of edges in the network by the number of possible edges in the network.

One can model this random graph as a $G(n, p)$ model. [gee, 2017] In this model, a network graph is created by randomly connecting the nodes using edges that occur with a probability p. Each edge is created independently.

As the value of p increases, the graph becomes denser, meaning the nodes are interconnected to a higher degree. Another manner of modeling the Erdős-Rényi random graph is with a $G(n, M)$ model. In this model, a single graph is chosen from the set of all graphs with n nodes and M edges.

Chapter 5

Epidemic Modeling of COVID-19

5.1 Model Background

The epidemic simulation discussed in this chapter will be created using the EpiModel package available in the R library. This simulation will model the spread of the COVID-19 virus through a population of 1,000 individuals over a span of 100 days.

The COVID-19 epidemic has taken the world by storm. With over 500,000 coronavirus deaths in the United States alone, the country has been tasked with the difficult job of updating its national safety standards and precautionary health measures in order to reduce the number of COVID-19 cases. We will model the epidemic spread of this virus through a population and assess the effectiveness of the precautionary measures taken to inhibit the spread of this virus. We are assessing the following protocols popularly implemented during the COVID-19 epidemic:

1. The Mask Mandate
2. The Social Distancing Protocol
3. Reduction of Initial Infected Population Size (ex. Travel Restrictions)
4. National Vaccination

Our network model utilizes the following formation formula:

```
formation <- ~edges
```

This formation model includes the network property of density, the ERGM term for which is edges. The density property is the count of the number of edges that are created in this network. This network model is modeled as an Erdős-Rényi random graph with a $G(n, M)$ model. We set the number of nodes n to be 1,000. We calculate the number of edges in the system using the following formula:

2 $*$ (number of edges) = (average number of individuals with whom interaction occurs) $*$ (total number of individuals in the network)

We calculate the total number of edges M to be 4,000 by assuming that the average American has approximately $8-9$ close friends. [Team, 2019] Using the above statistics, we produce a connected network with 1,000 nodes and 4,000 edges. One example of such a network is the following:

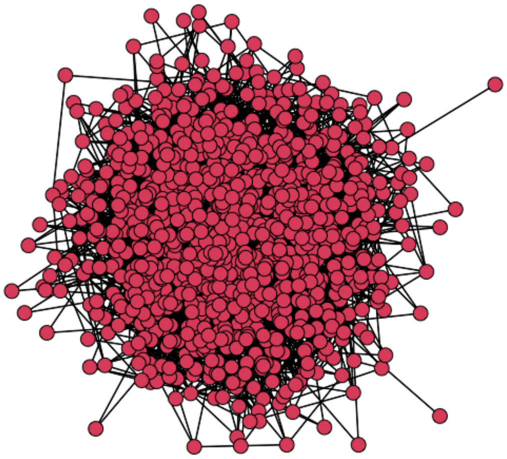

Figure 5.1: This figure visualizes a network of 1,000 nodes and 4,000 edges.

We will run 10 total simulations of the epidemic spread of the COVID-19 virus over the span of 100 time steps. The R code for the simulation that is discussed can be found in the Appendix. After running 10 simulations of our epidemic model, we refer to the plot formation statistics to find that the number of edges M does remain evenly distributed around the value of 4000 for all 100 timesteps:

Variations in the number of edges are due to the arrival of new individuals, the death of individuals due to the virus, and the dissolution coefficient. The dissolution coefficient refers to the probability of edge dissolution at each time step. We set this dissolution to a high value to minimize the number of edge dissolutions in our model.

This EpiModel simulation requires the use of the following functions:

1. The **netest** function generates an approximated model for the dynamic

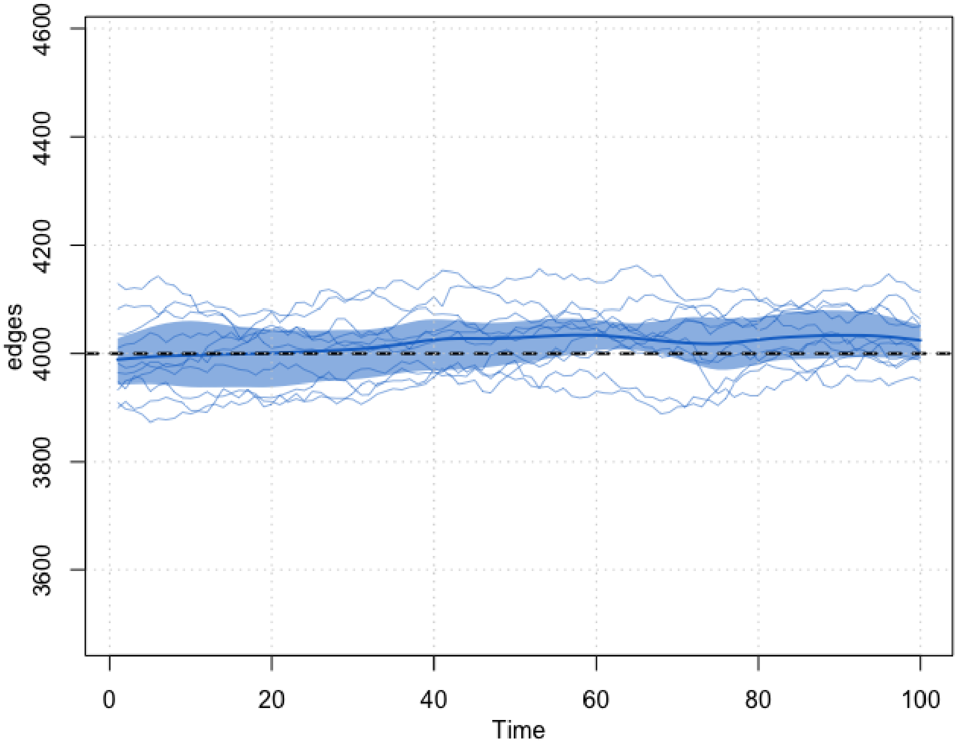

Figure 5.2: Number of edges within the network model at each timestep.

network.

2. The **netdx** function simulates the dynamic network model created using **netest** multiple times to run diagnostics on whether the dynamic network model can be replicated over time.

3. The **netsim** function uses the dynamic network model created using **netest** to run the stochastic epidemic simulation. In our independent model, the epidemic simulation will be run after the dynamic network model is created.

5.2 Control Model

We begin with a control model with 5 individuals initially infected. We assign the infection probability to be 0.015, [Staff, 2020] referring to a 1.5% risk of the transmission of COVID-19 between a contagion carrier and a non-carrier, both wearing facial masks. We will also set the rate of exposure to a low value of 2, assuming that the two individuals are social distancing to a certain extent. We set the recovery rate to be 1/17, referring to an average incubation period of 6 days [hea, 2020] in addition to a period of 10 days [Lee, 2020] until the symptoms have passed. An individual will not be a coronavirus carrier after an average period of 17 days and will shift to the recovered state.

We run the EpiModel simulation with the above parameters and produce the following model at times 1, 25, 50, and 100:

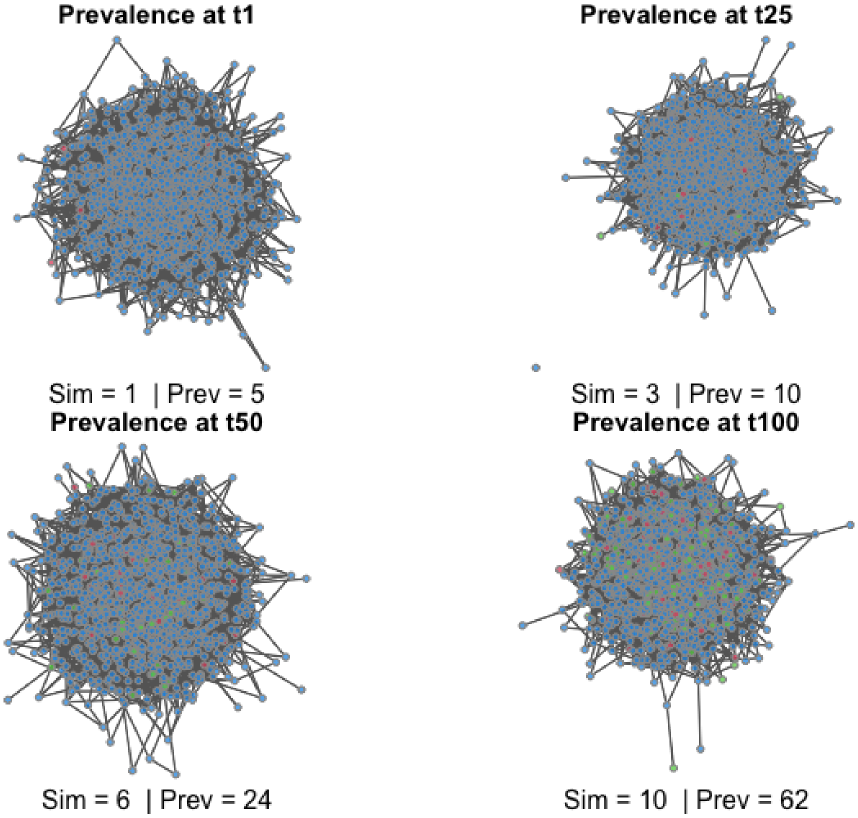

Figure 5.3: This figure visualizes a network of 1,000 nodes at times 1, 25, 50, and 100. The network consists of nodes of 3 colors: blue (susceptible), red (infected), and green (recovered).

Over a span of 100 days, a total of 251 individuals have been infected by COVID-19 with the above protocols in place. 25.1% of the population has been infected.

5.3 Removal of Mask Mandate

We now simulate the results of removing the population-wide mask mandate. With no facial masks, the risk of transmission between a coronavirus carrier and non-carrier will increase to approximately 70%. [Staff, 2020]

We run the EpiModel simulation with the infection probability set to a value of 0.7 and produce the following model at times 1, 25, 50, and 100:

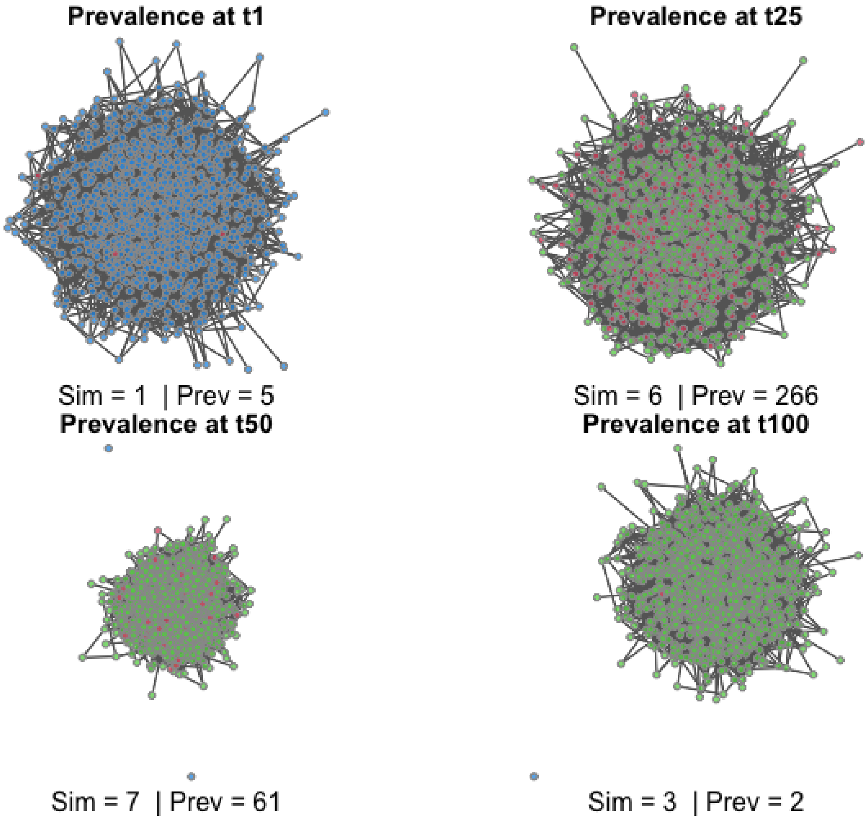

Figure 5.4: This figure visualizes a network of 1,000 nodes at times 1, 25, 50, and 100. The network consists of nodes of 3 colors: blue (susceptible), red (infected), and green (recovered).

In a span of 100 days, a total of 998 individuals have been infected by COVID-19 with the above protocols in place. 99.8% of the population has been infected over the length of this simulation.

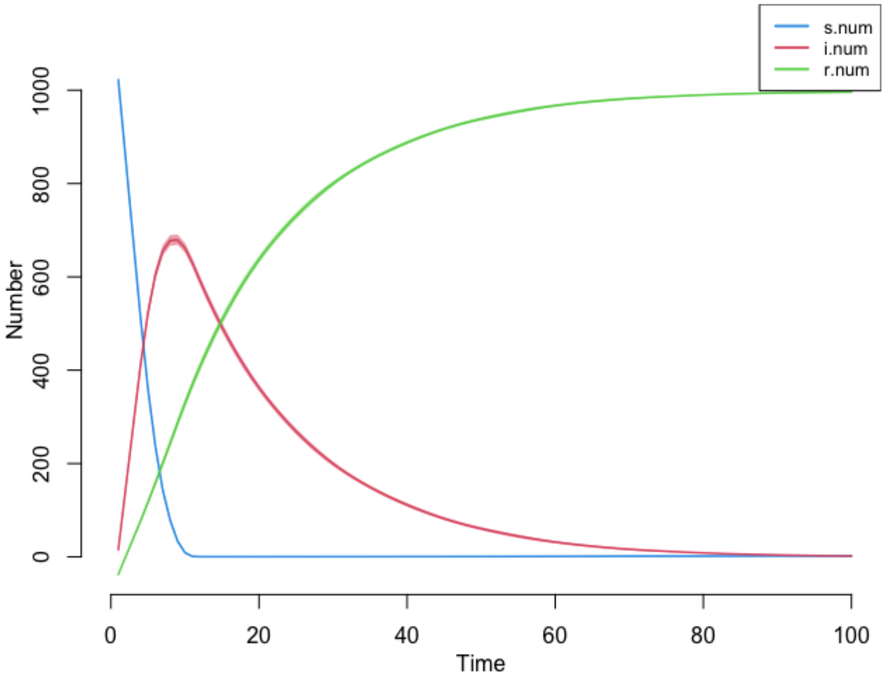

Figure 5.5: This graph visualizes the count of cases for each time step, separated by susceptible, infected, and recovered individuals. The network consists of 3 states: blue (susceptible), red (infected), and green (recovered).

Analyzing the line graph above, we can observe that the highest rate of infection occurs within the first 10 days of the simulation. Therefore, we can conclude that the failure to implement a mask mandate will correlate with an extremely sudden rise in COVID-19 cases throughout the population.

5.4 Removal of Social Distancing Protocol

We now simulate the results of removing the social-distancing protocol, with the mask mandate in place and the initial number of infected individuals being 5. With no social distancing, the rate of exposure between individuals is increased. We increase this exposure rate, or the number of “acts”, or “actions”, between the individuals from 1 to 5.

We run the EpiModel simulation and produce the following model at times 1, 25, 50, and 100:

Over a span of 100 days, approximately 989 individuals have been infected by COVID-19 with triple the initial population and the above protocols in place. 98.9% of the population has been infected in this simulation.

The above graph visualizes the count of cases for each time step. Analyzing the line graph above, we can see that the highest rate of infection occurs within the first 20 days of the simulation, which spans twice the length of time as the peak number of infections with the removal of the mask mandate. We can conclude that the failure to implement the social distancing protocol will correlate with a high rise in COVID-19 cases throughout the population.

5.5 Increased Initial Infection

We now simulate the results of increasing the initial number of infected individuals, with the mask mandate and social distancing protocol in place. This increased count of initial infections moderately mimics the lifting of travel

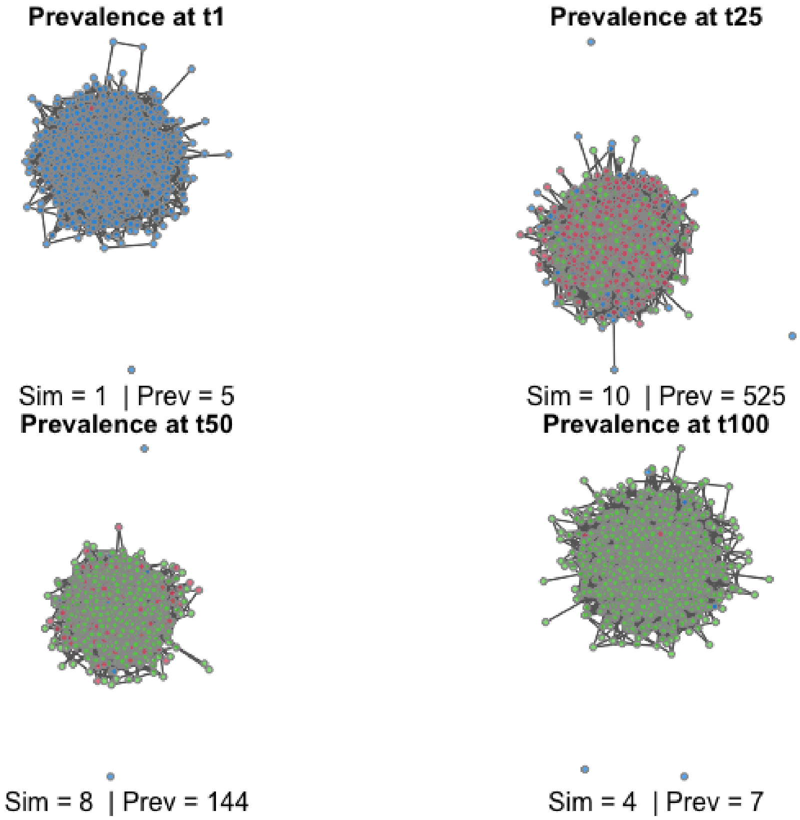

Figure 5.6: This figure visualizes a network of 1,000 nodes at times 1, 25, 50, and 100. The network consists of nodes of 3 colors: blue (susceptible), red (infected), and green (recovered).

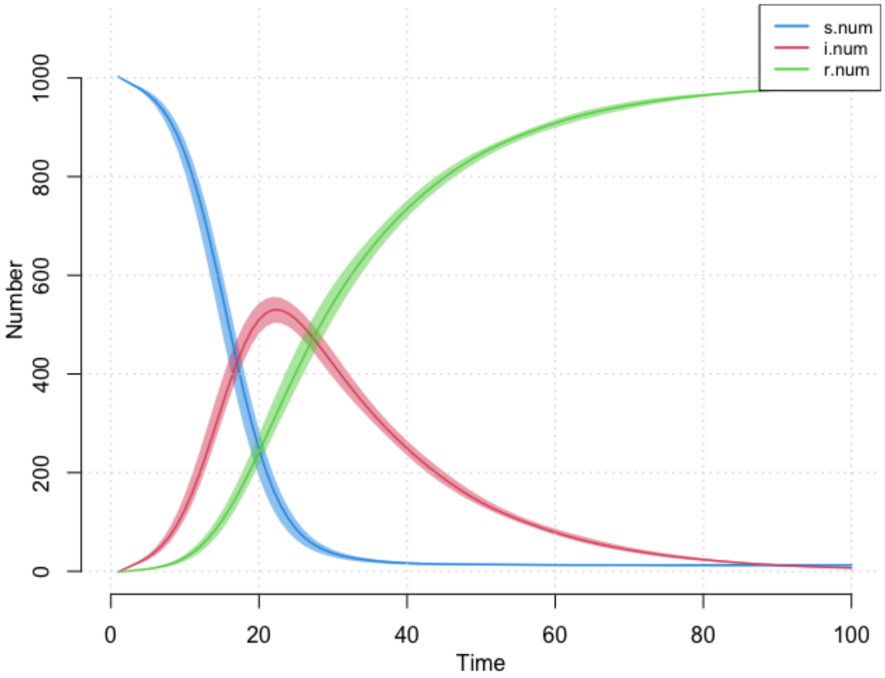

Figure 5.7: This graph visualizes the count of cases for each time step, separated by susceptible, infected, and recovered individuals. The network consists of 3 states: blue (susceptible), red (infected), and green (recovered).

restrictions, which subsequently increases the risk of case importation. We will triple the initial number of infected individuals from 5 to 15.

We run the EpiModel simulation and produce the following model at times 1, 25, 50, and 100:

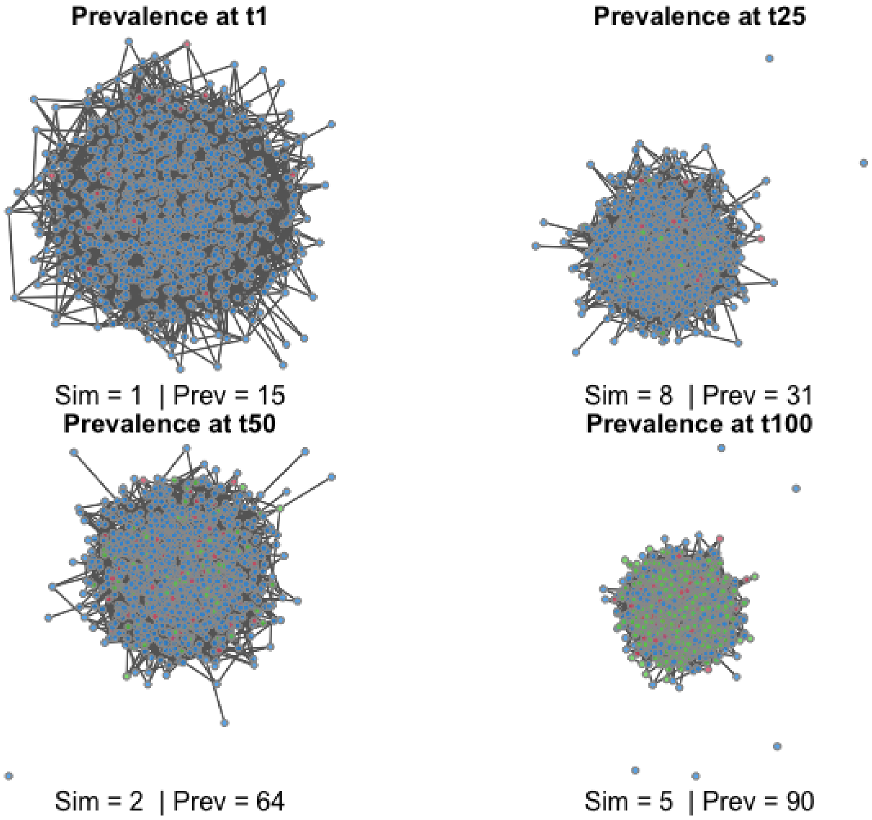

Figure 5.8: This figure visualizes a network of 1,000 nodes at times 1, 25, 50, and 100. The network consists of nodes of 3 colors: blue (susceptible), red (infected), and green (recovered).

Over a span of 100 days, approximately 446 individuals have been infected by COVID-19 with three times the initial infected population and the above protocols in place. 44.6% of the population has been infected in this simulation.

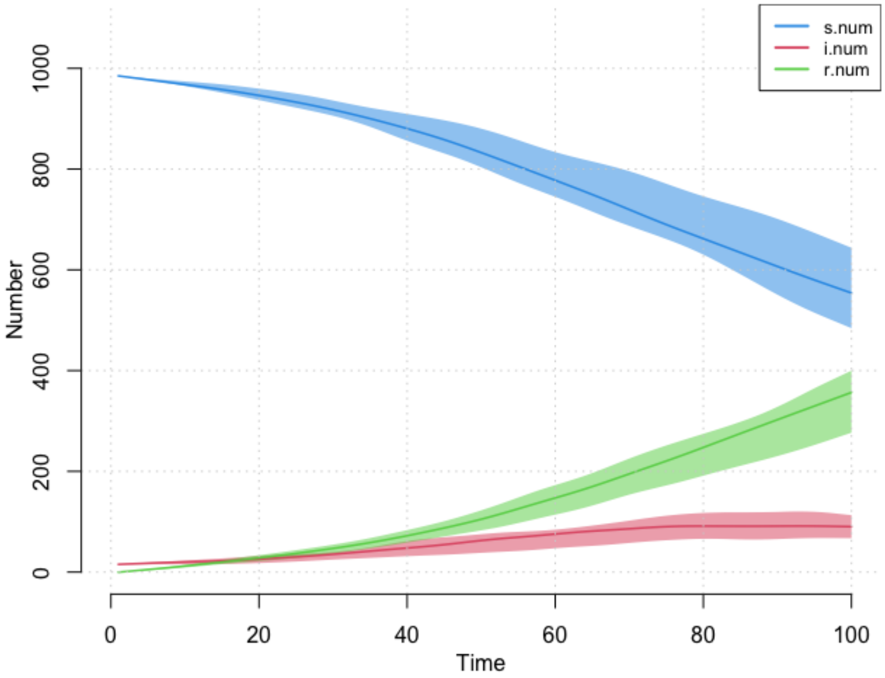

Figure 5.9: This graph visualizes the count of cases for each time step, separated by susceptible, infected, and recovered individuals. The network consists of 3 states: blue (susceptible), red (infected), and green (recovered).

The above graph visualizes the count of cases for each time step, showing a nearly constant, linear increase in the number of infected individuals over time. This pattern of increase vastly differs from the sudden, exponential increase seen subsequent to removal of the mask mandate. An increase in the initially infected population will correlate to a steady increase in coronavirus cases through the population.

5.6 Modeling Vaccination

We now simulate the spread of coronavirus through a fully vaccinated population, with the implementation of the mask mandate and social distancing protocols. We mimic the implementation of a vaccine by greatly increasing an individual's rate of recovery.

We run the EpiModel simulation with the recovery rate set to a value of 1/5 and produce the following model at times 1, 25, 50, and 100:

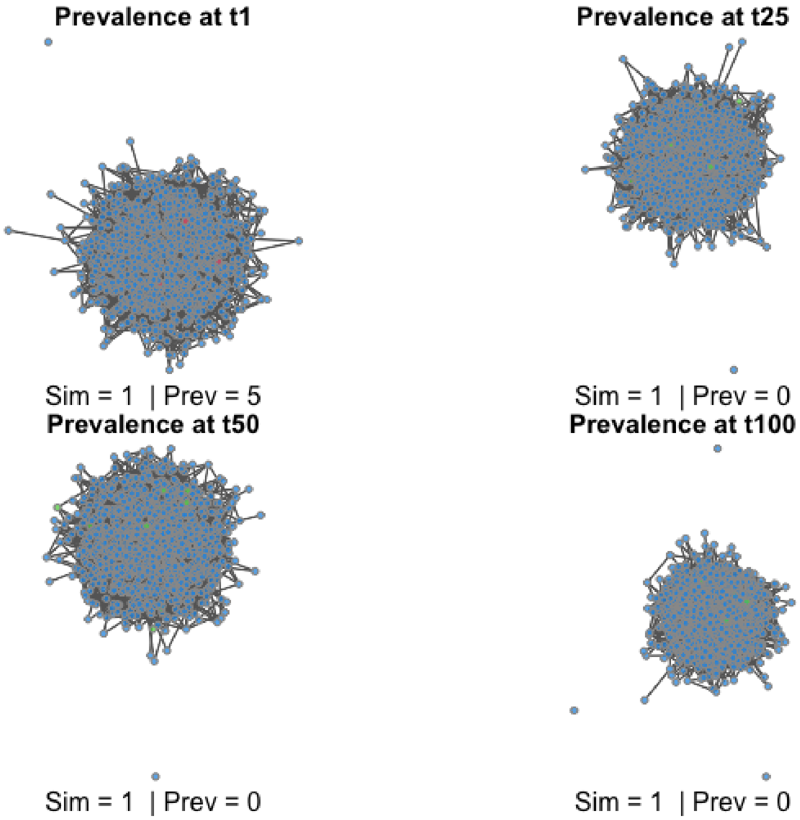

Figure 5.10: This figure visualizes a network of 1,000 nodes at times 1, 25, 50, and 100. The network consists of nodes of 3 colors: blue (susceptible), red (infected), and green (recovered).

Over a span of 100 days, approximately 10 individuals have been infected by COVID-19 with the vaccination of the complete population and the above protocols in place. Only 5 individuals have been infected by carrier-non carrier interaction from an initial count of 5 infected individuals over the span of 100 days. Therefore, we can conclude that the vaccination of a population does in fact correlate to a drastic reduction in the epidemic spread of COVID-19.

Appendix A

Appendix

A.1 R Code: Epidemic Spread of COVID-19

Below is a simple implementation of creating an epidemic simulation using the EpiModel package available within the R library. [Jenness *et al.*, 2018a] This simulation models the spread of the COVID-19 virus through a population of 1,000 individuals over a span of 100 days.

The R code below requires the following parameters:

Parameters		
Term	Value	Notes
nsteps	100	Number of time steps in one simulation.
nsims	10	Number of simulations to be conducted.
s.num	995	Initial susceptible population count.
r.num	0	Initial recovered population count.
i.num	5	Initial infected population count.
inf.prob	0.015	The infection probability.
act.rate	1	Rate of exposure.
rec.rate	1/17	Rate of recovery.
a.rate	(10.5/365)/1000	Rate of arrival of new individuals.
ds.rate	(7/365)/1000	Daily death rate of susceptible individuals.
di.rate	(14/365)/1000	Daily death rate of infected individuals.
dr.rate	(7/365)/1000)	Daily death rate of recovered individuals.
n	1000	Number of individuals in the starting population.
M	4000	Total number of edges in the network.

Please note that the values selected for each parameter are changeable and subject to the status of the epidemic that you are modeling. Each value above was chosen according to further research conducted on the COVID-19 virus epidemic. All references to this research can be found within the Bibliography.

```
#Control Parameters (model type, # time steps in simulation,
number of independent simulations)
control <- control.net(type = "SIR", nsteps = 100, nsims = 10)

#Initial Population (#susceptible, infected, and recovered)
init <- init.net(s.num = 995, i.num = 5, r.num = 0)

#Parameters (probability of infection from an encounter, rate of
exposure, rate of recovery, rate of arrival of new individuals,
daily (individual) death rate of susceptible individuals, and
daily death rate for infected individuals, death rate for recovered
individuals )
param <- param.net(inf.prob = 0.015, act.rate = 1, rec.rate = 1/17,
                   a.rate = (10.5/365)/1000, ds.rate = (7/365)/1000,
                   di.rate = (14/365)/1000, dr.rate = (7/365)/1000)

#Initialize the network (number of individuals in the population)
nw <- network::network.initialize(n = 1000, directed = FALSE)

#STERGM Formation formula
formation <- ~ edges

#Total number of edges in the network
target.stats <- c(4000)
```

```
#The likelihood of an edge to dissolve at a certain time step
# (ex. mean duration of 80 time steps = 1/80 = 1.25% risk of
dissolving)
coef.diss <- dissolution_coefs(dissolution = ~ offset(edges),
duration = 80)

#Estimation of the statistical network model
est <- netest(nw, formation , target.stats , coef.diss)

#Simulate the stochastic network epidemic model
simulation <- netsim(est , param , init , control)
simulation

# Plot formation statistics
par(mfrow = c(1,1), mar = c(3,3,1,1), mgp = c(2,1,0))
plot(simulation , type = "formation", grid = TRUE)

#Plot the graph of the simulation (time vs. cases)
plot(simulation)

#Summarize the simulation at time 100
summary(simulation , at = 100)
```

```
#Visualize the simulation as a network diagram at time 1 and
time 100
par(mfrow = c(2, 2), mar = c(1, 0, 1, 0), mgp = c(1, 1, 0))
plot(simulation, type = "network", at = 1, sims = "mean",
     col.status = TRUE, main = "Prevalence␣at␣t1")
plot(simulation, type = "network", at = 25, sims = "mean",
     col.status = TRUE, main = "Prevalence␣at␣t25")
plot(simulation, type = "network", at = 50, sims = "mean",
     col.status = TRUE, main = "Prevalence␣at␣t50")
plot(simulation, type = "network", at = 100, sims = "mean",
     col.status = TRUE, main = "Prevalence␣at␣t100")
```

A.2 R Code: Community Network Structure Simulator Application

Below is a simple implementation of creating a community network structure simulator using the RShiny and iGraph packages available within the R library. [RStudio, Inc, 2013] [Csardi & Nepusz, 2006] This simulation models a variety of network structures given a set of parameters. Network structure matrices are created in block sizes of 10 and 100. Network visualizations are created in block sizes of 10.

The R code below requires the following parameters:

Parameters		
Term	Value	Notes
prob.1	0.5	Probability of spread within Group 1.
prob.between	0.05	Probability of spread between groups.
prob.2	0.5	Probability of spread within Group 2.
blocksize	50	Size of block 1.
exp.tau	-0.4	Tau value used to create exponential structure.
power.tau	-2	Tau value used to create power-log structure.

Please note that the values selected for each parameter are changeable and subject to the status of the network structure that you are simulating.

```
#Load the EpiModel library (make sure EpiModel is already installed)
suppressMessages(library(tidyverse))
suppressMessages(library(ggraph))
suppressMessages(library(igraph))
suppressMessages(library(tidygraph))
suppressMessages(library(dplyr))
suppressMessages(library(shiny))
suppressMessages(library(shinythemes))

#Given 90% chance of being a pro-masker
masks <- rbinom(10, 1, 0.9)
nodes <- data.frame(masks)
nodes[’Mask_Bin’] <- masks
nodes[’Mask’] <- masks
nodes$Mask[nodes$Mask_Bin==1] <- "Pro"
nodes$Mask[nodes$Mask_Bin==0] <- "Anti"


#Build the Shiny App
ui <- fluidPage(
theme = shinytheme("sandstone"),
titlePanel("Community␣Network␣Structure␣Simulator"),
#Input Functions
#Output Functions
```

```
sidebarPanel(
sliderInput(inputId = "prob.1",
label = "SBM:␣Probability␣of␣Spread␣within␣Group␣1",
value = 0.5, min = 0, max = 1),
sliderInput(inputId = "prob.2",
label = "SBM:␣Probability␣of␣Spread␣within␣Group␣2",
value = 0.5, min = 0, max = 1),
sliderInput(inputId = "prob.between",
label = "SBM:␣Probability␣of␣Spread␣between␣Groups",
value = 0.05, min = 0, max = 1),
sliderInput(inputId = "blocksize",
label = "SBM:␣Block␣1␣Size",
value = 50, min = 0, max = 100),
sliderInput(inputId = "exp.tau",
label = "EXP:␣Exponential␣Tau",
value = -0.4, min = -0.75, max = -0.04),
sliderInput(inputId = "power.tau",
label = "PWR:␣Power-Law␣Tau",
value = -2, min = -2.4, max = -2),
actionButton("startsim", "Start␣Simulation")
),

mainPanel(
tabsetPanel(type = "tabs",
tabPanel("SBM␣100", plotOutput("sbm100plot")),
tabPanel("SBM␣10", plotOutput("sbm10plot")),
tabPanel("SBM␣Net", plotOutput("sbmNetplot")),
tabPanel("EXP␣100", plotOutput("exp100plot")),
tabPanel("EXP␣10", plotOutput("exp10plot")),
tabPanel("EXP␣Net", plotOutput("expNetplot")),
tabPanel("PWR␣100", plotOutput("power100plot")),
tabPanel("PWR␣10", plotOutput("power10plot")),
tabPanel("PWR␣Net", plotOutput("pwrNetplot"))
    )
  )
)

server <- function(input, output) {
observeEvent(input$startsim, {

output$sbm100plot <- renderPlot({
#SBM - Sampling from the stochastic block model of networks
pm <- cbind( c(input$prob.1, input$prob.between), c(input$prob.between,
input$prob.2) )
g <- sample_sbm(100, pref.matrix=pm, block.sizes=c(input$blocksize, (100
- input$blocksize)))
if (require(Matrix)) { image(g[]) }
})
```

```
output$sbm10plot <- renderPlot({
#SBM - Sampling from the stochastic block model of networks
pm <- cbind( c(input$prob.1, input$prob.between), c(input$prob.between,
input$prob.2) )
g <- sample_sbm(10, pref.matrix=pm, block.sizes=c(ceiling(
input$blocksize / 10), 10-ceiling(input$blocksize / 10)))
if (require(Matrix)) { image(g[]) }
})

output$sbmNetplot <- renderPlot({
#SBM - Sampling from the stochastic block model of networks
pm <- cbind( c(input$prob.1, input$prob.between), c(input$prob.between,
input$prob.2) )
g <- sample_sbm(10, pref.matrix=pm, block.sizes=c(ceiling(
input$blocksize / 10), 10-ceiling(input$blocksize / 10)))
#concatenate g into dataframe
g_df <- data.frame(g[1], g[2], g[3], g[4], g[5], g[6], g[7], g[8],
g[9], g[10])
#Loop through g_df and make source-node table
source = data.frame()
target = data.frame()
for (i in 1:10) {
for (x in g_df[i]) {
```

```
for (j in 1:10){
if(g_df[i,j]==1){
source = rbind(source, i)
target = rbind(target, j)
}
}
}
}
links <- data.frame(source, target)
colnames(links) <- c(’source’, ’target’)
links <- links %>%
filter(!(source == target))
links <- dplyr::distinct(links)
#Creating Graph
social_net_tbls <- tbl_graph(nodes = nodes,
edges = links,
directed = FALSE)
social_net <- ggraph(social_net_tbls, layout = "stress") +
geom_node_point(size = 2) +
#geom_node_text(aes(label = Mask), nudge_y = 0.05, nudge_x = 0.2)+
geom_edge_link() +
theme_void()
# Render the network
show(social_net)
```

```
})

output$exp100plot <- renderPlot({
degs <- sample(1:100, 100, replace=TRUE, prob=exp(
input$exp.tau*(1:100)))
if (sum(degs) %% 2 != 0) { degs[1] <- degs[1] + 1 }
g <- sample_degseq(degs, method="vl")
if (require(Matrix)) { image(g[]) }
})

output$exp10plot <- renderPlot({
degs <- sample(1:10, 10, replace=TRUE, prob=exp(
input$exp.tau*(1:10)))
if (sum(degs) %% 2 != 0) { degs[1] <- degs[1] + 1 }
g <- sample_degseq(degs, method="vl")
if (require(Matrix)) { image(g[]) }
})

output$expNetplot <- renderPlot({
degs <- sample(1:10, 10, replace=TRUE, prob=exp(
input$exp.tau*(1:10)))
if (sum(degs) %% 2 != 0) { degs[1] <- degs[1] + 1 }
g <- sample_degseq(degs, method="vl")
#concatenate g into dataframe
```

```
g_df <- data.frame(g[1], g[2], g[3], g[4], g[5], g[6], g[7], g[8],
g[9], g[10])
#Loop through g_df and make source-node table
source = data.frame()
target = data.frame()
for (i in 1:10) {
for (x in g_df[i]) {
for (j in 1:10){
if(g_df[i,j]==1){
source = rbind(source, i)
target = rbind(target, j)
}
}
}
}
links <- data.frame(source, target)
colnames(links) <- c('source', 'target')
links <- links %>%
filter(!(source == target))
links <- dplyr::distinct(links)
#Creating Graph
social_net_tbls <- tbl_graph(nodes = nodes,
edges = links,
directed = FALSE)
```

```
social_net <- ggraph(social_net_tbls, layout = "stress") +
geom_node_point(size = 2) +
#geom_node_text(aes(label = Mask), nudge_y = 0.05, nudge_x = 0.2)+
geom_edge_link() +
theme_void()
# Render the network
show(social_net)
})

output$power100plot <- renderPlot({
degs <- sample(1:100, 100, replace=TRUE, prob=(1:100)^input$power.tau)
if (sum(degs) %% 2 != 0) { degs[1] <- degs[1] + 1 }
g <- sample_degseq(degs, method="vl")
if (require(Matrix)) { image(g[]) }
})


output$power10plot <- renderPlot({
# Power-law degree distribution
degs <- sample(1:10, 10, replace=TRUE, prob=(1:10)^input$power.tau)
if (sum(degs) %% 2 != 0) { degs[1] <- degs[1] + 1 }
g <- sample_degseq(degs, method="vl")
if (require(Matrix)) { image(g[]) }
})
```

```
output$pwrNetplot <- renderPlot({
degs <- sample(1:10, 10, replace=TRUE,
prob=(1:10)^input$power.tau)if (sum(degs) %% 2 != 0) { degs[1]
<- degs[1] + 1 }g <- sample_degseq(degs, method="vl")
#concatenate g into dataframe
g_df <- data.frame(g[1], g[2], g[3], g[4], g[5], g[6], g[7], g[8],
g[9], g[10])
#Loop through g_df and make source-node table
source = data.frame()
target = data.frame()
for (i in 1:10) {
for (x in g_df[i]) {
for (j in 1:10){
if(g_df[i,j]==1){
source = rbind(source, i)target = rbind(target, j)}}}}
links <- data.frame(source, target)
colnames(links) <- c(’source’, ’target’)
links <- links %>%
filter(!(source == target))
links <- dplyr::distinct(links)
#Creating Graph
social_net_tbls <- tbl_graph(nodes = nodes, edges = links,
directed = FALSE)
social_net <- ggraph(social_net_tbls, layout = "stress") +
geom_node_point(size = 2) +
#geom_node_text(aes(label = Mask), nudge_y = 0.05, nudge_x = 0.2)+
geom_edge_link() +
theme_void()
# Render the network
show(social_net)
})
})
}

shinyApp(ui = ui, server = server)
```

Bibliography

[Upe, n.d.] Introduction to mathematical probability. *McGraw-Hill, New York.*

[soc, n.d.] *Social Network Analysis for Anthropologists - Introduction to ERGMs.*

[gee, 2017] 2017 (Sep). *Erdos Renyl Model (for generating Random Graphs).*

[hea, 2020] 2020 (Dec). *Exposed to COVID? Here's When to Get Tested - COVID-19, Featured, Health Topics.*

[Asu Ozdaglar, n.d.] Asu Ozdaglar, title = Networks' Challenge: Where Game Theory Meets Network Optimization, month = July year = 2008.

[Auchincloss & Garcia, 2015] Auchincloss, Amy H., & Garcia, Leandro Martin Totaro. 2015. Brief introductory guide to agent-based modeling and an illustration from urban health research. *Cad Saude Publica*, 65–78.

[Bonabeau, 2002] Bonabeau, Eric. 2002. Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences of the United States of America*, **99**, 7280–7287.

[Borowiak *et al.*, 2020] Borowiak, Molly, Ning, Fayfay, Pei, Justin, Zhao, Sarah, Tung, Hwai-Ray, & Durrett, Rick. 2020. *Controlling the spread of COVID-19 on college campuses.*

[Chang *et al.*, 2020] Chang, Sheryl L., Piraveenan, Mahendra, Pattison, Philippa, & Prokopenko, Mikhail. 2020. Game theoretic modelling of infectious disease dynamics and intervention methods: a review. *Journal of Biological Dynamics*, **14**(1), 57–89.

[Csardi & Nepusz, 2006] Csardi, Gabor, & Nepusz, Tamas. 2006. The igraph software package for complex network research. *InterJournal*, **Complex Systems**, 1695.

[Cuevas, 2020] Cuevas, Erik. 2020. An agent-based model to evaluate the COVID-19 transmission risks in facilities. *Computers in Biology and Medicine*, 103827.

[El-Sayed *et al.*, 2012] El-Sayed, Abdulrahman, Scarborough, Peter, Seemann, Lars, & Galea, Sandro. 2012. Social network analysis and agent-based modeling in social epidemiology. *Epidemiologic Perspectives Innovations : EP+I*, **9**.

[Ethen, 2018] Ethen. 2018. *Submodular Optimization Influence Maximization.* https://ethen8181.github.io/machine-learning/networkx/max$_i nfluence/max_i nfluence.html$.

[Hayes, 2021] Hayes, Adam. 2021 (Apr). *Null Hypothesis Definition.*

[Jalayer *et al.*, 2020] Jalayer, Masoud, Orsenigo, Carlotta, & Vercellis, Carlo. 2020. CoV-ABM: A stochastic discrete-event agent-based framework to simulate spatiotemporal dynamics of COVID-19. *arXiv preprint arXiv:2007.13231.*

[Jenness *et al.*, 2018a] Jenness, Samuel M., Goodreau, Steven M., & Morris, Martina. 2018a. EpiModel: An R Package for Mathematical Modeling of Infectious Disease over Networks. *Journal of Statistical Software, Articles*, **84**(8), 1–47.

[Jenness *et al.*, 2018b] Jenness, Samuel M., Goodreau, Steven M., & Morris, Martina. 2018b (Nov). *EpiModel: Mathematical Modeling of Infectious Disease Dynamics.* https://cran.r-project.org/web/packages/EpiModel/vignettes/Intro.html.

[Kenton, 2020] Kenton, Will. 2020 (Dec). *Monte Carlo Simulation.*

[Khanam *et al.*, 2020] Khanam, Kazi Ainab, Srivastava, Gautam, & Mago, Vijay. 2020. The Homophily Principle in Social Network Analysis. *Proc. ACM Meas. Anal. Comput. Syst.*, **4**(111).

[Kleinberg, 2010] Kleinberg, Jon, and Easley David. 2010. *Networks, Crowds, and Markets: Reasoning about a Highly Connected World*. Cambridge University Press.

[Kumar, 2016] Kumar, Balasubramanian. 2016. *Branching Processes and Their Applications*.

[Lee, 2020] Lee, Bruce Y. 2020 (May). *How Long Does It Take To Recover From COVID-19 Coronavirus And Return To Work?*

[RStudio, Inc, 2013] RStudio, Inc. 2013. *Easy web applications in R.* URL: http://www.rstudio.com/shiny/.

[Staff, 2020] Staff, Reuters. 2020 (Apr). *Partly false claim: Wear a face mask; COVID-19 risk reduced by up to 98.5.*

[Team, 2019] Team, The GoodTherapy. 2019 (Feb). *How Many Friends Does the Average Person Have?*

[Tufekci, 2020] Tufekci, Zeynep. 2020. This Overlooked Variable Is the Key to the Pandemic. *The Atlantic.*

[Wasserman & Pattison, 1996] Wasserman, Stanley, & Pattison, Philippa. 1996. Logit models and logistic regressions for social networks: I. An introduction to Markov graphs andp. *Psychometrika*, **61**, 401–425.

[Wey *et al.*, 2020] Wey, Arkady, Champneys, Alan, Dyson, Rosemary J, Alwan, Nisreen A, & Barker, Mary. 2020. The benefits of peer transparency in safe workplace operation post pandemic lockdown. *arXiv preprint arXiv:2007.03283*.

[Zitkovic, 2016] Zitkovic, Gordan. 2016 (March). *Lecture 7 Branching Processes*.