# MODELLING OF DISEASE ON RANDOM GRAPHS

## ABSTRACT

This research paper explores random graphs and using them to model disease. The first section of the paper focuses on the models of random graphs and includes information on their thresholds and properties. We also illustrate some properties that are useful for later in the paper. The second section of the paper focuses on the modelling of disease and delves into some Erdos-Renyi graphs that can be used. During this section we also explore modelling a specific disease, COVID-19. We use information on the contact rate, transmissibility and infection period along with a set number of nodes to create a model. The final section discusses the measures we can take to prevent spreading of disease and again uses COVID-19 as a specific disease. The two methods we discuss include vaccination and social distancing.

## INTRODUCTION

With COVID-19 transforming the world, researching disease has never been more exciting. This paper discusses disease in a non-medical direction, instead exploring how disease can be modelled using graph theory. Random graphs are an intersection of graph theory and probability theory, making them perfect to model networks and several scenarios including the spread of disease. Using these models, we can also discuss the benefits of different types of prevention and demonstrate the effectiveness of the prevention. These models can be used by doctors and organisations to decide which methods of intervention they should use to prevent the spread of different diseases. Some examples of prevention that can be discussed using these models include vaccination, social-distancing and travelling restrictions. This paper will attempt to show the applications of random graphs in modelling disease and provide some information on effectiveness of different types of intervention, specifically using the example of COVID-19.

## 1. RANDOM GRAPHS

In the first section we will explore *random graphs*. We will look into their models and relationships and then their thresholds and sharp thresholds, while also discussing notation. This will provide a foundation of knowledge to apply to disease later in the paper. The start of this section will explain some fundamental graph theory that must be understood in order to lay a foundation for introducing random graphs.

A *graph* in its most abstract form is a set of objects and a set of connections between groups or pairs of these objects (Maricq, 2014). A *graph* is an ordered pair **G = (V, E)** where **V** is a set of *vertices* and **E** is a set of subsets **e** $\subset$ **V** of *cardinality* 1 or 2 (Maricq, 2014). *Cardinality* is the number of elements in a set. The *order* of a graph **G** is the *cardinality* of its vertex set, and the *size* of a graph is the *cardinality* of its edge set. A directed graph or digraph, G = (V, E), is a graph in which the elements of E or the edges have a direction. This means that they go from

one vertex to another in one way. An undirected graph, G = (V, E), is a graph in which the elements of E or the edges have no direction.

The degree of a vertex is the number of edges in which the vertex is contained in. For example, the degree of $v_n$ would be the number of edges in **E** attached to $v_n$ when $v_n \in$ **V**. The *eccentricity* of the vertex **v** is the greatest distance between that vertex and every other vertex. Its symbol is **ecc(v)**. The minimum eccentricity of any vertex in a graph is known as the *radius* of **G**, indicated by the symbol **rad(G)**. A graph's *diameter*, given by the symbol **diam(G)**, is the value of any vertex's maximum eccentricity. The vertex or set of vertices, **v**, such that **ecc(v) = rad(G)**, is the centre of the graph **G** (Harris, 2008).

A *random graph* is a *graph* in which properties such as the number of *graph vertices*, *graph edges*, and *connections* between them are determined in some random way (Weisstein, 2022). **Definition 1.1:** The term random graph refers to probability distributions over graphs. Random graphs may be described simply by a probability distribution, or by a random process which generates them (Wikipedia, 2022).

*Figure 1.1* shows the three possible *graphs* on the *vertex set* **[3] = {1, 2, 3}** with **2** *edges*. The *probability distribution* is the uniform, namely, each graph has the same *probability* **1/3** to be sampled.
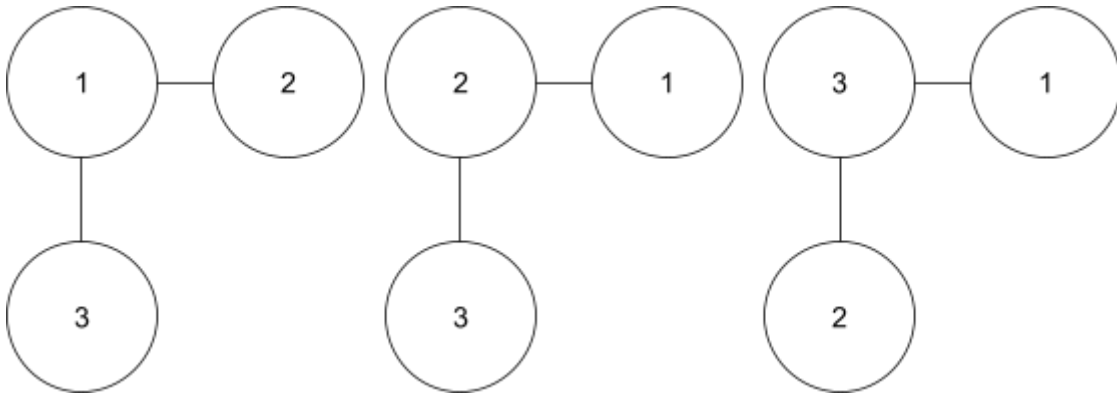


Figure 1.1: Random graph on *{1, 2, 3}* with *2 edges* with the uniform distribution.

Two models we will look into are the *random binomial graphs* and *uniform random graphs*. The *random binomial graph* is denoted by **G(n, p)**. This model has two parameters, the number of vertices **n** and a probability parameter **0 ≤ p ≤ 1**. Let **G₁** be the family of all possible labelled graphs on the vertex set **[n]**. Notice **|G₁| = 2^(nC2)**. The **G(n, p)** model assigns to a graph **G ∈ G₁** the following probability:

$$Pr[G] = p^{|E(G)|} (1-p)^{\binom{n}{2} - |E(G)|}.$$

The second model is the *uniform random graph* model. It is denoted by **G(n, m)**. This model has two parameters, the number of *vertices* **n** and the number of *edges* **m**, where **0 ≤ m ≤ nC2**. This model assigns to all labelled graphs on the vertex set **[n]** with exactly **m** *edges* equal probability. This can be shown as:

$$Pr[G] = \begin{cases} \dfrac{1}{\dbinom{\binom{n}{2}}{m}} & if \ \ |E(G)| = m \\[2em] 0 & if \ \ |E(G)| \neq m \end{cases}$$

.

**Theorem 1.1:** A random graph **G(n, p)** with **m** *edges* is equally likely to be any of the **((nC2)Cm)** graphs with **m** *edges*.

**Proof:** Consider any graph with **m** *edges*, call it **G**.

$$Pr[G(n,p) = G \big| |E(G(n,p))| = m = \frac{Pr[G(n,p) = G]}{Pr[\,|E(G(n,p))| = m}$$

$$= \frac{p^m (1-p)^{\binom{n}{2} - m}}{\dbinom{\binom{n}{2}}{m} p^m (1-p)^{\binom{n}{2} - m}}$$

$$= \frac{1}{\dbinom{\binom{n}{2}}{m}}$$

Thus **G(n, p)** conditioned on the event **{ G(n, p) has m** *edges* **}** is equal in distribution to **G(n, m)**, the graph chosen uniformly at random from all graphs with **m** *edges*. The biggest differentiator between the two models is that in **G(n, m)** we choose the number of *edges*, while in the case of **G(n, p)** the number of edges is the binomial random variable with the parameters **nC2** and **p**. For large **n** random graphs **G(n, m)** and **G(n, p)** should behave in a similar fashion when the number of edges **m** in **G(n, m)** equals or is almost equal to the expected number of edges of **G(n, p)**. For example, when

$$m = \binom{n}{2} p \approx \frac{n^2 p}{2},$$

or, when the edge probability in **G(n, p)**

$$p \approx \frac{2m}{n^2}$$ (Fierze, 2015).

A technique used in forming random graphs is called *coupling*. To generate **G(n, p)** using *coupling* we first will suppose that $p_1 > p$ and $p_2$ is defined by the equation:

$$1 - p = (1 - p_1)(1 - p_2),$$

or,

$$p = p_1 + p_2 - p_1 p_2.$$

Thus if an edge is not included in either of **G(n, p₁)** or **G(n, p₂)**, it is also not included in **G(n, p)**. It follows that

$$G(n, p) = G(n, p_1) \cup G(n, p_2),$$

where the two graphs **G(n, p₁)** and **G(n, p₂)** are independent. So when we write

$$G(n, p_1) \subseteq G(n, p),$$

we mean that the two graphs are coupled so that **G(n, p)** is obtained from **G(n, p₁)** by superimposing it with **G(n, p₂)** and replacing eventual double edges with single ones. We can also use this technique for **G(n, m)**. Coupling random graphs **G(n, m₁)** and **G(n, m₂)** where $m_2 \geq m_1$ via

$$G(n, m_1) = G(n, m_2) \cup H.$$

Here **H** is the random graph on the vertex set **[n]** that has **m = m2 − m1** edges chosen uniformly at random from **([n]C2) / E$_{n,m1}$** .

**Definition 1.2** Define a graph property **P₁** as a *subset* of all possible labelled graphs. Namely **P** ⊆ **2[n]C2** (Tsourakakis, 2015).
We will call a property **P₁** as *monotone increasing* if **G ∈ P₁** implies **G + e ∈ P₁**. Similarly, we will call a property **P₁** as *monotone decreasing* if **G ∈ P₁** implies **G − e ∈ P₁**. graph property **P₁** is monotone increasing if and only if its *complement* is monotone decreasing. Not all graph properties are monotone. For example having at least half of the *vertices* having a given fixed *degree* **d** is not monotone (Frieze, 2022).

From the coupling argument it follows that if **P₁** is a monotone increasing property then, whenever **p < p' or m < m'**,

$$\Pr(G_{n,p} \in P_1) \leq \Pr(G_{n,p'} \in P_1),$$
$$\Pr(G_{n,m} \in P_1) \leq P(G_{n,m'} \in P_1),$$

respectively.

The 'abrupt' nature of the arrival and absence of certain graph properties is one of the most notable observations about the asymptotic properties of random graphs. We introduce threshold functions (or just thresholds) for monotone graph properties in order to define this phenomenon more explicitly. We begin by defining a threshold for a monotone increasing graph property **P₁** formally (Frieze, 2022).

**Definition 1.3:** A function **p\* = p(n)** is a threshold for a monotone increasing property **P₁** in **G(n, p)** if

$$\lim_{n \to \infty} Pr[G(n,p) \in P_1] = \begin{cases} 0 & if \quad p/p^* \to 0 \\ 1 & if \quad p/p^* \to \infty \end{cases},$$

as **n→∞**. Furthermore for a monotone decreasing property, the cases will be flipped.

**Definition 1.4:** A function **m\* = p(m)** is a threshold for a monotone increasing property **P₁** in **G(n, m)** if

$$\lim_{n \to \infty} Pr[G(n,m) \in P_1] = \begin{cases} 0 & if \quad m/m^* \to 0 \\ 1 & if \quad m/m^* \to \infty \end{cases},$$

as **n→∞**. Furthermore for a monotone decreasing property, the cases will be flipped.

**Definition 1.5:** A function **p\* = p(n)** is a sharp threshold for a monotone increasing property **P₁** in **G(n, p)** if for every **ε > 0**,

$$\lim_{n \to \infty} Pr[G(n,p) \in P_1] = \begin{cases} 0 & if \quad p/p^* \leq 1-\varepsilon \\ 1 & if \quad p/p^* \geq 1+\varepsilon \end{cases},$$

as **n→∞**. Furthermore for a monotone decreasing property, the cases will be flipped.

**Definition 1.6:** A function **m\* = p(m)** is a sharp threshold for a monotone increasing property **P₁** in **G(n, m)** if for every **ε > 0**,

$$\lim_{n \to \infty} Pr[G(n,m) \in P_1] = \begin{cases} 0 & if \quad m/m^* \leq 1 - \varepsilon \\ 1 & if \quad m/m^* \geq 1 + \varepsilon \end{cases}$$

,

as **n→∞**. Furthermore for a monotone decreasing property, the cases will be flipped.

As discussed, a graph in **G(n, p)** has on average **(nC2)p** edges. The distribution of the degree of any particular vertex is binomial:

$$Pr[deg(v) = k] = \binom{n-1}{k} p^k (1-p)^{n-1-k}$$

,

where n is the total number of vertices in the graph. Since:

$$Pr[deg(v) = k] \to \frac{(np)^k e^{-np}}{k!}$$

as **n→∞** and **np** is a constant. This means this distribution becomes *Poisson*. Furthermore, other properties proven for **G(n, p)**'s behaviour as **p** varies include:

- If **np < 1**, then a graph in **G(n, p)** will almost surely have no connected components of size larger than the order of **log(n)**.

- If **np = 1**, then a graph in **G(n, p)** will almost surely have a largest component whose size is of **order $n^{2/3}$**.

## 2. APPLICATIONS TO DISEASE

In the first section we have seen how we can use probabilities to create random graphs with differing edges depending on their nodes. We can use this to create models of graphs of diseases by considering their contact rate, transmissibility and infective period. We also consider the population size (considering either individuals or households).
With these assumptions, we can create different random graphs and then also consider the differences when we use different nodes that have varying degrees. The variance of degrees are due to the differences in social behaviour.

To generate these models we will use a python library. The code we will use is the following:

*#importing the networkx library*
*>>> import networkx as nx*
*#importing the matplotlib library for plotting the graph*
*>>> import matplotlib.pyplot as plt*
*>>> G= nx.erdos_renyi_graph(x, y)*

```
>>> nx.draw(G, with_labels=True)
>>> plt.show()
```

This will create a **G(n, p)** graph with **x** nodes and **y** probability. The probability will be the contact rate, and we will then apply the transmissibility to determine after contact the chances of the infected node becoming infected. We can then use the infective period to determine how long that node will stay infected. We will determine each model for a certain number of times. We will have patient zero be a randomised node for the starting model, but will change this when looking into prevention.
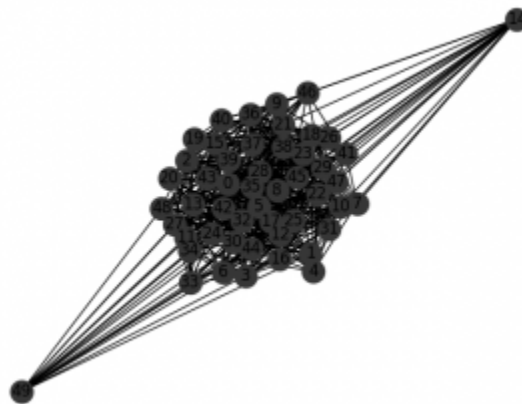
An example of a model created with this code is:



*Figure 2.1: **G(50, 0.5)***

This is a random graph created with 50 nodes and a contact rate of 0.5.
Another example is:



*Figure 2.2: **G(10, 0.25)***

This is a random graph created with 10 nodes and a contact rate of 0.25.
Next we will apply this to information from a specific disease to show a model. In this paper we will focus on COVID-19. The information on contact rate, transmissibility and infective period are all quite new as the virus is novel:
Contact rate: **p = .003** to model a contact rate in a public space (Sheely, 2020).

Transmissibility: **T = .066** based on in depth surveillance study in Shenzhen, China (Sheely, 2020).

Infective period: **α = 10**, based on study of viral loads indicating constant infectivity for the 3 days prior to and 7 days after onset of symptoms (Sheely, 2020).

Let's test this model with a population of 3000 nodes.
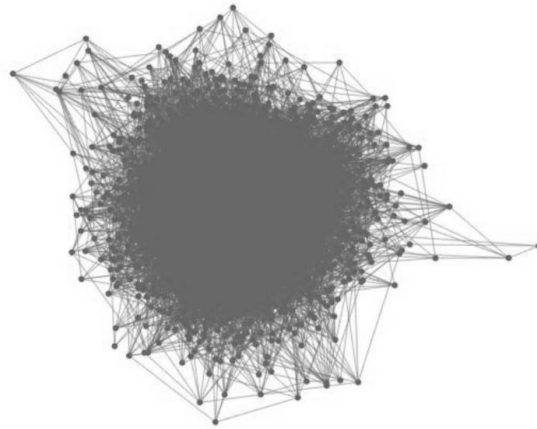
The graph generated will look like this:



*Figure 2.3: Random graph taken from Sheely, 2020.*

This random graph was generated using the same parameters, however the plotting was not possible on my computer and so I used the plotting of someone else's graph.

Now, we factor in how the transmission rate and infective period will affect the model and how this will show the infected nodes. We will use **T' = (1 − (1 − T )$^α$)**, where **T'** is the transmission rate. **T' = 0.49479** to **5 d.p**.

We will select a random node out of the population and consider it patient zero. This patient zero will have a probability of **0.49479** of infecting any node that has an edge connection. Each time an infection stage happens, a time period passes. We can compute this and change conditions. We repeated the computation 6 times, with two separate random graphs. Three computations each of the graphs with different nodes as patient zero. We consider three different scenarios, one where we use the node with maximum degree, one where we use the node with minimum degree and one where we use a randomly selected node.
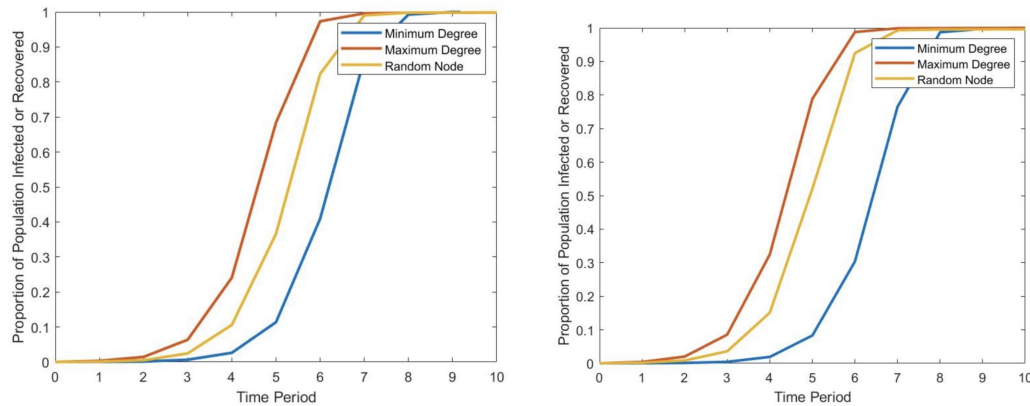
*Figure 2.4: Graph of simulations (Sheely, 2020).*

Both show that all three initial conditions led to all or nearly all of the individuals being infected. In both computations, the models seem to lag behind one another in order of greatest initial degree to least.

There are some obvious patterns even if the results of each simulation vary somewhat because of stochastic effects. A disease's trajectory can be greatly influenced by its initial circumstances, particularly early on. However, if preventative precautions aren't taken, the disease will eventually affect a person who has a lot of contacts, and the number of cases will rise.

## 3. PREVENTION

In this section we will discuss two preventative methods that can be used: vaccination and social-distancing. The goal of prevention is to decrease the number of infections. Two ways we can decrease the infection rate is by decreasing the contact rate or by decreasing the transmission rate. Social distancing will decrease the contact rate, while successful vaccination will decrease the transmission rate.

First let's consider vaccinations. Vaccinations will lead to a decrease in transmission rate. Let's consider the Pfizer vaccine for COVID-19. The vaccine had an efficacy of 95 percent on transmission rate (Tartof, 2021). This means it will decrease the transmission rate, or **T'** to **0.02474**. Considering the previous model of 3000 nodes, if we have 1500 nodes vaccinated; this will lead to the time period decreasing very significantly.

However, when computed, even with a 50 percent vaccination rate, eventually almost all the nodes are still affected. We can show vaccination as having thinned nodes, with the edges having lower transmission rates:

*Figure 3.1: Random graph with thinned random graph.*

Another way that the COVID-19 vaccinations work is decreasing the severity of the effects of the infection and so this may lead to a lower effect of the disease, but when it comes to changing the effect on infections, the vaccine will only increase the time period. The time period increases to over 80 when 60 percent of the population is vaccinated.

However, when modelling vaccinations we can also consider a perfect vaccine. For example the smallpox vaccine is a perfect vaccine, a vaccine that provides full and permanent immunity to the disease. Modelling this type of vaccination will lead to all the edges of a node that is vaccinated being removed, which we can also consider as just removing the node:
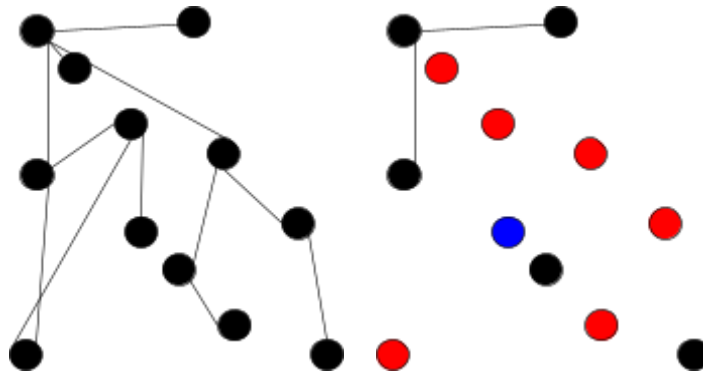


*Figure 3.2: Random Graph **G(12, 0.3)** before and after thinning.*

Figure 3.2 shows the effect of vaccinating half the population when considering a disease with a perfect vaccine.
This demonstrates that although the vaccine for COVID-19 may not be an extremely effective way of decreasing the number of  infections, in other diseases with perfect vaccines it would be.

Next we will consider social distancing as a way of preventing infection. Social-distancing will not affect the transmission rate, but will instead change the contact rate. This is as distancing will lead to removal of contact with nearby individuals (nodes). We can see that this will lead to less infections as there will be far less edges which have a chance of transmitting the disease:
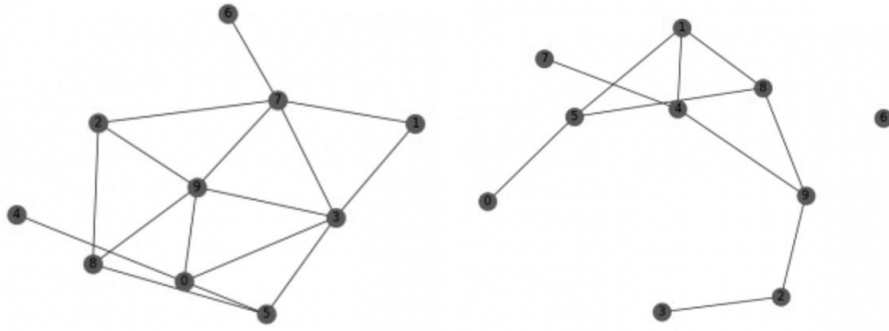
*Figure 3.3: Random graphs with **n=10** and right has **p=0.5** while left has **p=0.25**.*

Figure 3.3 illustrates this. When we compute this for greater values of **n** this remains true, however it is not as easily viewable. When the contact rate is decreased by 50 percent for the COVID-19 model for 3000 nodes, the graph when viewed looks the same. However, when computed with the transmission rate the graph changes drastically.
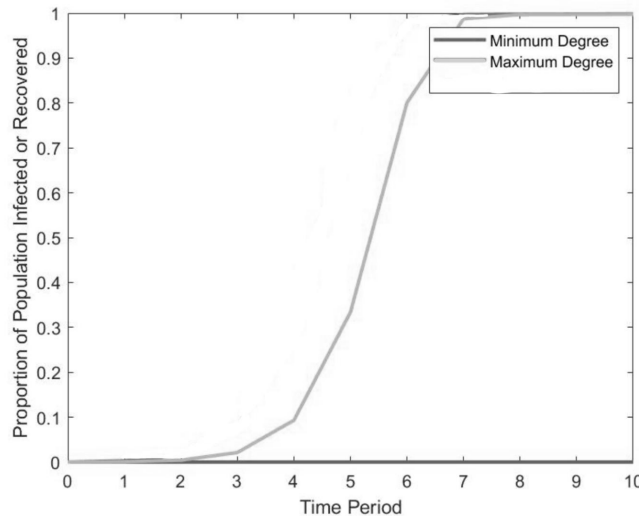


*Figure 3.4: Graph of simulation with social-distancing.*

Figure 3.4 illustrates that social isolation will cause a shift in time period when considering the vertex with the maximum degree as patient zero, but with the vertex with the minimum degree, there will be almost no spread. The model shows that social-distancing can be used to protect the population from COVID-19.

## CONCLUSION

In this paper, we have effectively introduced the concept of random graphs, discussed the properties and thresholds of random graphs, modelled disease on random graphs and evaluated methods of prevention using the models. We modelled COVID-19 on a random graph and were able to show the effects of vaccination and social-distancing. However, we were unable to generate the models as graphs with large numbers of nodes and were forced to use

the results provided by another paper. We were also unable to discuss how reducing the severity of disease by means of vaccination is another benefit of the strategy. We also explored how transmission rate can be approximated and how the contact rate along with the transmission rate can be used to evaluate the methods of preventing infections. We also generated random graphs using python and networkx. We also demonstrated how the patient zero's level of social activity makes a large difference in the initial spread of the disease.

Random graphs can be used in several modelling situations and can be used differently to model different types of diseases. The limit theorems and the monotone increasing and decreasing properties become more useful when considering the different components and constructions of random graphs. In the future, I will investigate the differential form of random graphs and how it can be used to model diseases more accurately and also how there would be no need for two separate random graph workings for contact rate and transmission rate, instead only using graphs to model the disease and the spread of the disease.

**Bibliography:**

Weisstein, Eric W. (2022) Random Graph, Wolfram-MathWorld

Wikipedia (2022) Random graph, Wikimedia Foundation

Tsourakakis, Charalampos E. (2015) Lecture 1: Random Graphs, C284r: Social Data Mining

Frieze, Alan and Karonski, Michał (2022) An introduction to Random Graphs, CMU

Maricq, Aleksander (2014), Applications of Expander Graphs in Cryptography, Whitman

Sheely, Joseph (2020) Random Networks and their Applications to the COVID-19 Pandemic, Brown University

Bisht, Jayant (2022) Erdos Renyl Model, Geeksforgeeks.org

Harris, John M. Hirst, Jeffry L. Mossinghoff, Michael J. (2008), Combinatorics and Graph Theory, Second Edition, Springer

Tartof, SY (2021) Effectiveness of mRNA BNT162b2 COVID-19 vaccine up to 6 months in a large integrated health system in the USA: a retrospective cohort study, The Lancet