An Agent-Based SEIR Model
Project 1
CS 575

Mike Goodrich and Nick Smith

Winter 2024

Abstract

The goal of Project 1 is to learn about how simple contagions spread on networks. You will experiment with various disease parameters including *transmissibility* and *infection duration* over seven networks and three infection characteristics. You will present your results as time-series plots, and interpret the results in terms of the effects of degree distribution, average degree, graph diameter, and graph density. You should include a few time series plots to convince me that you've successfully implemented the SEIR model. Most of your results should be presented in summary form, with the summary including peak infections, time to peak infection, the time until no new agents are infected, and the percentage of agents who were not infected.

1 COVID-19 Characteristics

> "COVID-19 is a disease caused by SARS-CoV-2 that can trigger what doctors call a respiratory tract infection. It can affect your upper respiratory tract (sinuses, nose, and throat) or lower respiratory tract (windpipe and lungs). It spreads the same way other coronaviruses do, mainly through person-to-person contact. Infections range from mild to deadly.
>
> "SARS-CoV-2 is one of seven types of coronavirus, including the ones that cause severe diseases like Middle East respiratory syndrome (MERS) and sudden acute respiratory syndrome (SARS). The other coronaviruses cause most of the colds that affect us during the year but aren't a serious threat for otherwise healthy people" [1].

Three important characteristics of the SARS-CoV-2 virus have contributed to its spread. First, the virus is airborne[2]. Second, the virus can be spread by non-symptomatic persons[3]. Third, there is an incubation period in which an infected person does not show symptoms and is not infectious to others[4][5]. The compartment model[6] that best fits these characteristics is the SEIR model.

[1] https://www.webmd.com/lung/coronavirus, accessed early January 2022.

[2] "COVID-19 is spread in three main ways:

- Breathing in air when close to an infected person who is exhaling small droplets and particles that contain the virus.
- Having these small droplets and particles that contain virus land on the eyes, nose, or mouth, especially through splashes and sprays like a cough or sneeze.
- Touching eyes, nose, or mouth with hands that have the virus on them."

From `https://www.cdc.gov/coronavirus/2019-ncov/faq.html`.

[3] See NYT Opinion on why there were many mistakes initially with the virus `https://www.nytimes.com/2021/05/07/opinion/coronavirus-airborne-transmission.html`.

[4] "The estimated incubation period is between 2 and 14 days with a median of 5 days. It is important to note that some people become infected and do not develop any symptoms or feel ill" `https://www.cdc.gov/coronavirus/2019-ncov/hcp/non-us-settings/overview/index.html`.

[5] "The Omicron variant has an incubation period of about three days, which is shorter than the incubation period for other variants of the virus that cause COVID-19." `https://www.health.com/condition/infectious-diseases/coronavirus/what-is-incubation-period-omicron-covid-19` accessed Jan 11, 2023

[6] https://en.wikipedia.org/wiki/Compartmental_models_in_epidemiology

> Donald S. Burke ... gave a lecture ... in 1997 in which he listed the criteria that might implicate certain kinds of viruses as likeliest candidates to cause a new pandemic. "The first criterion is the most obvious: recent pandemics in human history " ... That would point to ... the influenzas and ... the retroviruses ... "The second criterion is the proven ability to cause major epidemics in non-human animal populations" [which] would spotlight ... the coronaviruses, such as the virus later known as SARS-CoV ... [The] third criterion was "intrinsic evolvability," meaning readiness to mutate and to recombine (or reassort) which "confers on a virus the potential to emerge into and to cause pandemics in human populations ... Some of these viruses," ... citing coronaviruses in particular, "should be considered as serious threats to human health" [p.512, Quammen, D. (2012). *Spillover: Animal infections and the next human pandemic.* WW Norton & Company.]

In summer 2019, I worked with Dr. Chantel Sloan, an epidemiologist from the BYU Department of Health Sciences, and Dr. Candace Barrett from the BYU Statistics department to understand more about how COVID-19 might spread over the networks associated with BYU. Dr. Barrett reviewed any study she could find on COVID-19 and performed a Bayesian analysis to estimate parameters for how the virus could spread from person-to person under various conditions (e.g., classroom versus shared living space). She was not satisfied with the results of her model and I'm sure she'd revise the model now, but she did an excellent job given the data available. We'll use slightly modified versions of her model choices for the first part of the experiment. The modifications will reflect the characteristics of the Omicron variant of SARS-CoV-2.

There are three important model decisions: how long does the virus incubate before the person becomes infectious, how long is a person infectious, and how transmissible is the virus between an infectious person and a susceptible person.

1.1 How Long Does the Virus Incubate?

Dr. Sloan modeled the incubation period using a lognormal distribution; see Figure 1a. Notice how the figure agrees with the CDC estimates, "The estimated incubation period is between 2 and 14 days with a median of 5 days. It is important to note that some people become infected and do not develop any symptoms or feel ill" [7]. The lognormal distribution takes two parameters, mean μ and standard deviation σ.

1.2 How Long is a Person Infectious?

The number of days that a person is infectious is also modeled as a lognormal distribution; see Figure 1b. Notice that the model agrees with the CDC information, "Available data suggest that patients with mild-to-moderate COVID-19 remain infectious no longer than 10 days after symptom onset"[8]. Notice that the mean and standard deviation parameters differ between the incubation period model and the infectious period model.

1.3 How Infectious is a Person?

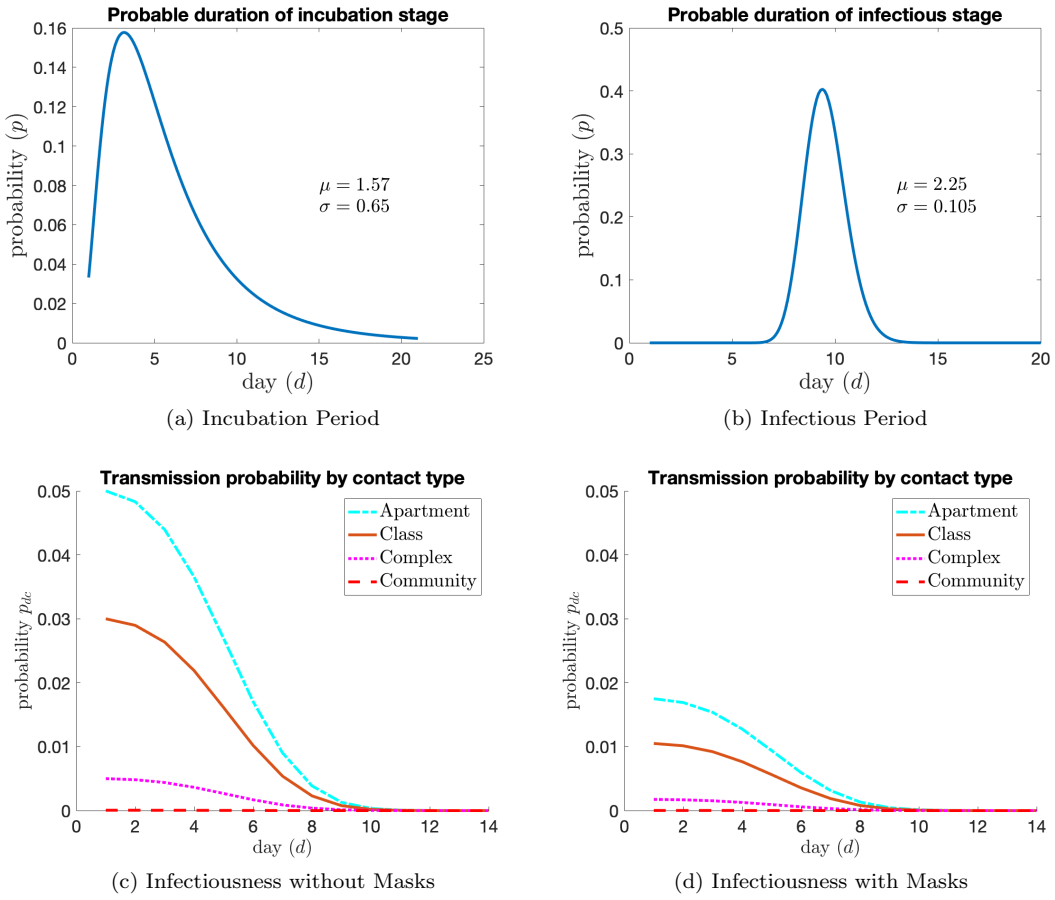
The infectiousness of a person decreases over time. Infectiousness also depends both on the type of contact (in an apartment, in a classroom) and whether or not a person is wearing a mask; see Figures 1c and 1d. The curves in the figure were generated using the Bayesian estimate from studies available as of July 2019. The probability is given by the function,

$$p_j^{\text{inf}}(d) = \frac{\frac{p_{1c}}{1-p_{1c}} \exp[\beta(d^3-1)]}{1 + \frac{p_{1c}}{1-p_{1c}} \exp[\beta(d^3-1)]} \tag{1}$$

The function is read as the infectiousness of agent j on day d as a function of the type of exposure p_{1c}. The parameter β was estimated to be

$$\beta = -0.0050367, \tag{2}$$

[7] `https://www.cdc.gov/coronavirus/2019-ncov/hcp/non-us-settings/overview/index.html`.
[8] `https://www.cdc.gov/coronavirus/2019-ncov/hcp/duration-isolation.html`.



(a) Incubation Period

(b) Infectious Period

(c) Infectiousness without Masks

(d) Infectiousness with Masks

Figure 1: Model choices for the COVID-19 virus.

and the estimates for p_{1c} as a function of contact type are given in Table 1. There is a range of values for p_{1c} and there is quite a bit of uncertainty (and low confidence) in the values. The curves in Figure 1c are obtained using Equation (1) and the best guess from Table 1. The curves in Figure 1d decrease p_{1c} by multiplying it by 0.35, which is based on a very crude estimate that infectiousness while wearing masks is only 35% of what it is without masks.

Contact Type	**Contact Frequency**	**Strength of Contact**	**Estimate of p_{1c}**		
			Low	**Best Guess**	**High**
Household	Daily	High	0.037	0.038	0.041
Apartment Complex	Daily	Low	0.001	0.005	0.02
Classroom	MWF/TTh	Moderate	0.01	0.03	0.06
Community	Daily	Very Low	0.0000324	0.0000542	0.0000976

Table 1: Estimates of p_{1c} for various types of contact.

1.4 Parameters to Use in Experiments

The parameters specified in this subsection are not based in solid analysis but rather modifications to Dr. Sloan's original data that I made up to roughly match reports from reputable sites. The parameters are not reliable or accurate, and are intended to illustrate simple contagion rather than to illustrate epidemic patterns for SARS-CoV-2.

Exposed Duration. The plot in Figure 1a uses the parameters that Dr. Sloan estimated on the original variant using difficult-to-collect data. Given recent evidence that the Omicron variant

incubates for only about three days, use the parameters $\mu^E = 1.0$ and $\sigma^E = 1.0$ in the experiments below, which yields an average incubation time of three days.

Infectious Duration. The data I gleaned from reliable websites in January of 2023 suggest that people can remain infectious for up to ten days. For the experiments, use the parameters $\mu^I = 2.25$ and $\sigma^I = 0.105$ as illustrated in Figure 1b.

Infectiousness Level. Reports on the infectious level of the Omicron variant compared to the original variant are imprecise. Use $p_{1c} = 0.12$ and $\beta = -0.00504$ in your experiments.

Exposed Duration		Infectious Duration		Infectiousness	
$\mu^E = 1.0$	$\sigma^E = 1.0$	$\mu^I = 2.25$	$\sigma^I = 0.105$	$p_{1c} = 0.12$	$\beta = -0.00504$

Table 2: Parameters to use in experiments.

2 SEIR Model Elements

The SEIR compartment represents the spread of the pathogen as a state machine. See Figure 2.

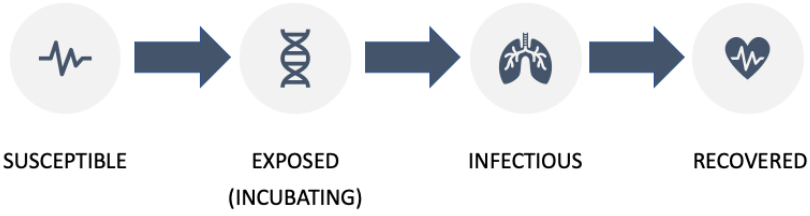


Figure 2: The four states are Susceptible, Exposed, Infectious, and Recovered.

2.1 States

The agent-based SEIR model has four states:

- Susceptible: an agent in this state can be infected with the disease.
- Exposed: an agent in this state is infected but cannot infect others because the pathogen is incubating.
- Infectious: an agent in this state can infect others.
- Removed: an agent in this state has recovered from the disease.

2.2 Agent Initialization

Before running the SEIR model, it is efficient to determine how long each agent will be in the exposed stage if they become infected and how long they will be in the infectious stage once they are through with the exposed stage. If I were doing this project, I'd create an agent class with private variables `countdown_to_infectious`, `countdown_to_recovered`, `days_spent_infectious`, `initial_state`. I'd also figure out a way to keep track of what state the agent is in, with possible states $\mathbb{S} = \{S, E, I, R\}$.

When the agent is initialized, I'd do the following:

1. create a sample d^E from the lognormal distribution that governs how long an agent stays in the exposed stage using $\mu^E = 1.0$ and $\sigma^E = 1.0$.
2. round d^E up to the nearest integer and set `countdown_to_infectious`$= \lceil d^E \rceil$. This sets the time an agent spends in the exposed state to a whole number
3. create a sample d^I from the lognormal distribution that governs how long an agent stays in the infectious stage using $\mu^I = 2.25$ and $\sigma^I = 0.105$.

4. round d^I up to the nearest integer and set `countdown_to_recovered`= $\lceil d^I \rceil$. This sets the time an agent spends in the infectious state to a whole number

5. set `days_spent_infectious`= 0. This variable will track how many days the agent has been infectious.

6. set `initial_state`= S unless told to initialize to a different state

2.3 Population Initialization

Create a class that manages the population. Initialize 90% of the population of agents to the susceptible state, 5% to exposed, and 5% to infectious.

2.4 Behavior of Agents in Different States

Each day,

- a susceptible agent monitors its neighbors to see if any are in the infectious state and otherwise does nothing,
- an exposed agent decreases the `countdown_to_infectious` variable by one,
- an infectious agent decreases the `countdown_to_recovered` variable by one, increases its `days_spent_infectious` by one, and computes its infectiousness level using Equation (1) with d =`days_spent_infectious`,
- a recovered agent does nothing.

2.5 State Transitions

Only the following transitions are allowed:

- Susceptible to Exposed
- Exposed to Infectious
- Infectious to Recovered
- Remain in current state.

Any transition not shown in the figure is a self-loop from state back to itself.

2.5.1 Recovered

An agent in the recovered state stays there until the end of the simulation.

2.5.2 Infectious

An agent in the infectious state stays in that state until the `countdown_to_recovered` variable is zero, at which time it transitions to the recovered state.

2.5.3 Exposed

An agent in the exposed state stays in that state until the `countdown_to_infectious` variable is zero, at which time it transitions to the infectious state.

2.5.4 Susceptible

An agent in the susceptible state will poll each of its neighbors and identify which neighbors are infectious. Let $N(j)$ denote the neighbors of agent j. The polling process is:

1. for each neighbor i in $N(j)$
   (a) if i is not infectious, continue.
   (b) else look up p_i^{inf} for agent i with d set to `days_spent_infectious` of agent i.
   (c) Draw a random sample r from the uniform distribution over the interval $[0, 1]$. If $r < p_i^{\text{inf}}$ then agent j is infected by agent i so it transitions to the exposed state

3 Networks

Conduct experiments with the following seven networks.

1. a circulant graph with 20 vertices where each vertex is attached to two neighbors on either side (`nx.circulant_graph(20,[1,2,3,4])`)
2. a circulant graph with 20 vertices where each vertex is attached to four neighbors on either side (`nx.circulant_graph(20,[1,2])`
3. a complete graph with 100 vertices (`nx.complete_graph(100)`)
4. a 10×10 lattice with 100 vertices (`nx.grid_2d_graph(10,10)`)
5. a scale-free network with 100 vertices (`nx.barabasi_albert_graph(100,???)`
6. a scale-free network with 410 vertices (`nx.barabasi_albert_graph(410,???)`
7. `infect-dublin` network, which has 410 vertices; described and downloadable here:

   `https://networkrepository.com/ia-infect-dublin.php`

3.1 Circulant Graphs

Recall from class that a *cycle graph* is a graph where every vertex is connected to its two neighbors, forming a ring. A circulant graph is a "cousin" to the cycle graph, but you can control how many of its nearest neighbors it is connected to. In the circulant graph in Figure 3a, each agent is connected to its neighbors on either side of it, but also their nearest neighbor making a total of four nearest neighbors. In the circulant graph in Figure 3b, each agent is connected to its eight nearest neighbors.

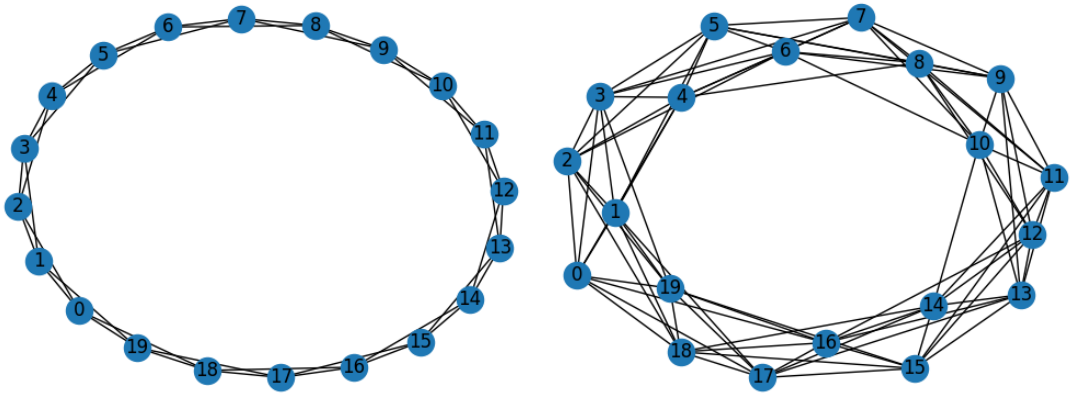
You do not need to report experiment results for the circulant graphs. I'm including them in the project because they are really useful for evaluating whether your code is working. I suggest using the color-map parameter to animate how the virus spreads over the network. I set a color for each state (for example, blue for susceptible, yellow for exposed, red for infectious, and green for removed) and then changed the colors as the infection spread through the network using:

`nx.draw(G,pos,with_labels=True,node_color = color_map)`

See the Tips section on Animation for hints on making this efficient.

When you are using these networks to test your code, you will likely see the following:

- More agents are infected in the network with the eight nearest neighbors than in the network with the four nearest neighbors, using the nominal parameters above.
- The rate of infection is faster for the network with eight nearest neighbors than for the network with four nearest neighbors.
- Most agents get infected.



(a) Circulant Graph with Nearest Four Neighbors (b) Circulant Graph with Nearest Eight Neighbors

Figure 3: Two circulant graphs intended to help you calibrate whether your code is working correctly.

3.2 Complete and Lattice Graphs

The complete and lattice graphs are formed using the same methods you used in Homework 1. They are to serve as your baseline for your experiment results. The diameter of the complete graph is one so the disease should propagate most rapidly and most completely through this network. The lattice graph has larger diameter, so this network should be more resistant to spread than the complete graph.

3.3 Scale-Free Graphs

Two scale-free graphs are to be used: one with 100 vertices and one with 410 vertices. The graph with 100 vertices allows for fair comparisons with with complete and lattice graphs, since each graph has 100 vertices. The graph with 410 vertices has the same number as the `dublin` graph to help make comparisons.

The scale-free network is implemented using the Barabasi-Albert model, implemented using the barabasi-albert method in Networkx. The number of vertices will be either 100 or 410, but the barabasi-albert method requires another parameter. You are to choose that parameter, and you should justify your choice in your report. *Hint: you want your degree distribution to have the signature of a scale-free network, namely the straight line in the log-log plot.*

What hypotheses would you make about spread of the virus through scale-free graphs? Recall from class that scale-free graphs have a degree distribution that follows a power law, creating some vertices with high degree (think *superspreaders*). Scale-free graphs also tend to have a modest diameter and radius.

3.4 Dublin Network

The `infect-dublin` network can be read in using the method shown below, where `filename = ia-infect-dublin.mtx`.

```
def _read_graph_from_file(filename):
    fo = open(filename,'r')
    line = fo.readline()  # Read file header
    line = fo.readline()  # Number of vertices and edges
    if not line:
        print('error -- illegal format for input')
        return
    v = line.split(" ")
    numVertices = int(v[0])
    G = nx.Graph()
    G.add_nodes_from(range(1,numVertices+1))
    while True:
```

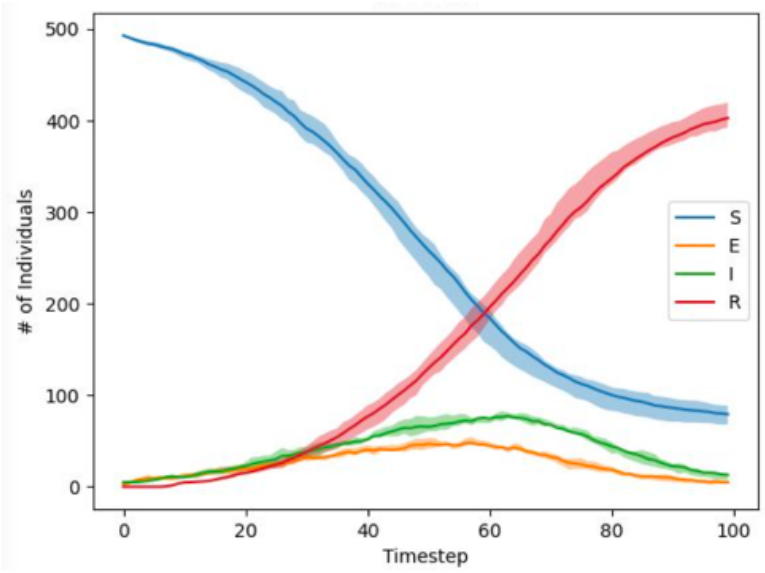


Figure 4: Time series summary

```
        line = fo.readline()
        if not line:
            break
        #print("Line{}: {}".format(count,line.strip()))
        v = line.split(" ")
        v1 = int(v[0])
        v2 = int(v[1])
        G.add_edge(v1,v2)
        G.add_edge(v2,v1)
    fo.close()
    return G
```

Read a little about the network here `https://networkrepository.com/ia-infect-dublin.php` (notice the summary of network characteristics) and here `http://www.sociopatterns.org/datasets/infectious-sociopatterns-dynamic-contact-networks/` Given the network characteristics, what would you hypothesize about how the virus would spread over the network? How are your hypotheses similar to or different from the hypotheses for the complete, lattice, and scale-free networks?

3.5 Experiments to Conduct and Reporting Results

You will perform a 5×2 experiment, with five networks {complete, lattice, scale-free-100, scale-free-410, and dublin} and two disease parameter sets. The two disease parameter sets you will use are the default values specified above and values that you choose because you think they might be interesting.

4 Presenting Results

4.1 Time Series Data

Please show a few time series plots like those in Figure 4. Please observe that the y-axis is the number of individuals and the x-axis is the number of days. The number of the agents in each state is shown in a time series. Multiple trials were conducted for each condition since there are probabilities in the SEIR model that cause each trial to differ. The solid lines represents the mean over each trial and the shaded regions represent the interquartile range (see `https://en.wikipedia.org/wiki/Interquartile_range`).

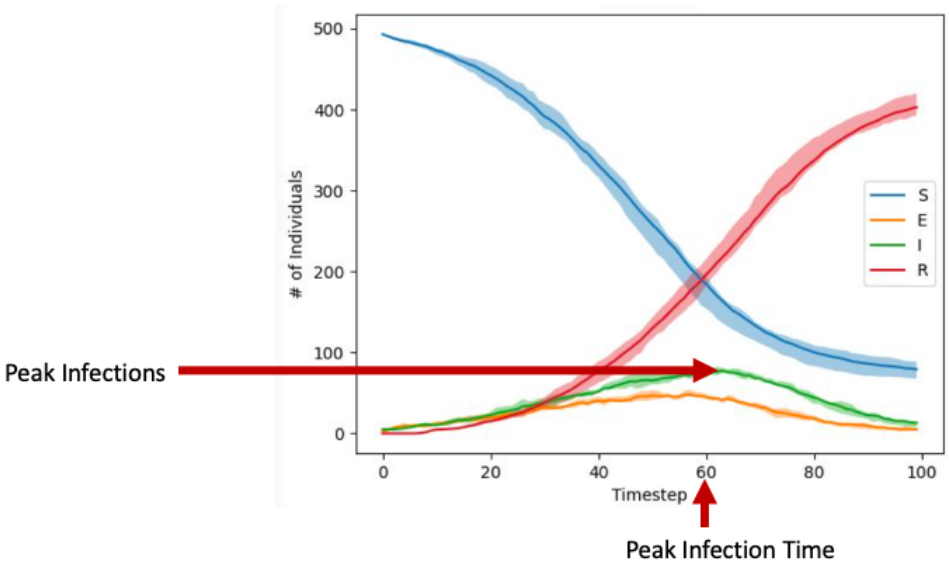


Figure 5: Time series summary

Do not use standard deviation to represent the variation since a plot with plus-or-minus one standard deviation can show a range that is negative, meaning some negative number of agents are, for example, removed.

You will not be able to show the time series over all experiments, both because there isn't enough space and because they aren't necessary for telling an interesting story. Choose those that you think are most useful.

Please note two important items that will make your report much better.

- Use the same time-scale (e.g., x-axis) for all plots. That allows each plot to be compared without having to do a mental transformation.
- Set the maximum value for the x-axis to a value that is large enough such that the experiment that takes the most amount of time reaches steady state (e.g., the time series aren't changing much anymore). In the example plot above, I'd have preferred if the result went to 150 seconds so that we could see the steady state reached.

4.2 Summarizing Results

Since there is not enough space to include all time series plots, please provide a summary of the time series plots. Three useful summary variables are *time-to-peak* infection, *peak infections*, *time until no new infections*, and number of uninfected individuals at the end of the experiment. Figure 5 shows what I mean by *time-to-peak* and *peak infections*. The time to reach steady state is the number of days until there are no more agents in the infectious or exposed states. The number of uninfected individuals will be the average number of agents in the susceptible state at the end of the simulation.

The networks were chosen to represent different interesting conditions. Please include the following information about each network: a figure or a description of the degree distribution, the maximum degree, the average degree, the diameter of the graph, the radius of the graph, and the density (connectance) of the graph. Note that networkx has methods for computing maximum degree, average degree, graph diameter, graph radius, and graph density.

A useful way to present results would therefore be a table with columns {network, max degree, avg degree, diameter, radius, density, time-to-peak infections, peak infections, time-to-steady-state, uninfected individuals}. There would be one row per each of the five networks.

5 What to Submit

Write a report of what you did and what you learned, and upload the report to Learning Suite. Please do not write more than four pages of text, but your figures and tables can make your report longer than four pages. If you'd like you can submit a Jupyter notebook that generates the figures and provides explanations of the figures. You do not need to summarize the experiment conditions since I'm giving those to you. Please include the following:

- Names: Names of everyone on your team. Please only submit one report per group on Learning Suite.
- Abstract: Summarize what you did and what you learned.
- Introduction: Short summary of why the problem is interesting to you (if it's not interesting to you, make something up), what you did, and how you organized your report.
- Experiment Conditions: I specified most of the experiment conditions, but I asked you to do two things: first, choose parameters for your scale-free networks and justify those choices, and second, run some experiments where you choose the infectiousness levels, time spent infectious, and time spent exposed.
- Hypotheses: Make some hypotheses about how the virus will propagate across the different networks under the different conditions. Give some justification for your hypotheses in terms of network characteristics (e.g., the metrics in your summary table).
- Discussion: Summarize which hypotheses were supported by data and which were not supported. Explain why you think the results came out the way they did. If you are speculating about why, state that you are hypothesizing a possible explanation. If the reason why is justified by the data, tell me how the data supports your explanation.
- Future Work: Tell me what you wish you had done or could do now that the project is over.
- You do not need a conclusion or summary.

6 Tips

6.1 Animation

- When animating your simulations, it may be easier to run the entire simulation and record the nodes in each sub-population (S, E, I, R) at each time step and then animate it, rather than try to animate it while you are running the simulation. That way you can rerun the animation or save your images without having to rerun the whole simulation.
- It will be more efficient to only redraw nodes that have actually changed state rather than redraw the entire graph each time. You can accomplish this using a command like the following:

```
nx.draw_networkx_nodes(G, pos, nodelist=changed_nodes, node_color=new_colors)
```

6.2 Time-Series Plots

- Remember that for the Time-Series plots you will need to repeatedly run your simulation and keep track of how many people are Susceptible, Exposed, Infected and Recovered at each time step. Then you plot either the average or median number of people infected at each time step for each graph.
- You can get the lower and upper bounds using the `np.quantile` function and then make the plot using:

```
plt.fill_between(x, lbounds, ubounds, color=fill_color)
plt.plot(x, means, color=color, label=label)
```