# Disease Spread Through Different Communities

*Original Authors: Sam Wisnoski, Charlie Mawn*

## Section I: Modeling Question

### Modeling Question:

- **How does a disease spread through different types of quarantined communities?**
- Our modeling question aims to identify **HOW** the relations of people in different communities affect the transmission of disease. Because our question examines how different variables (pod size) affect an outcome (disease spread), it is an **explanation based modeling question**.

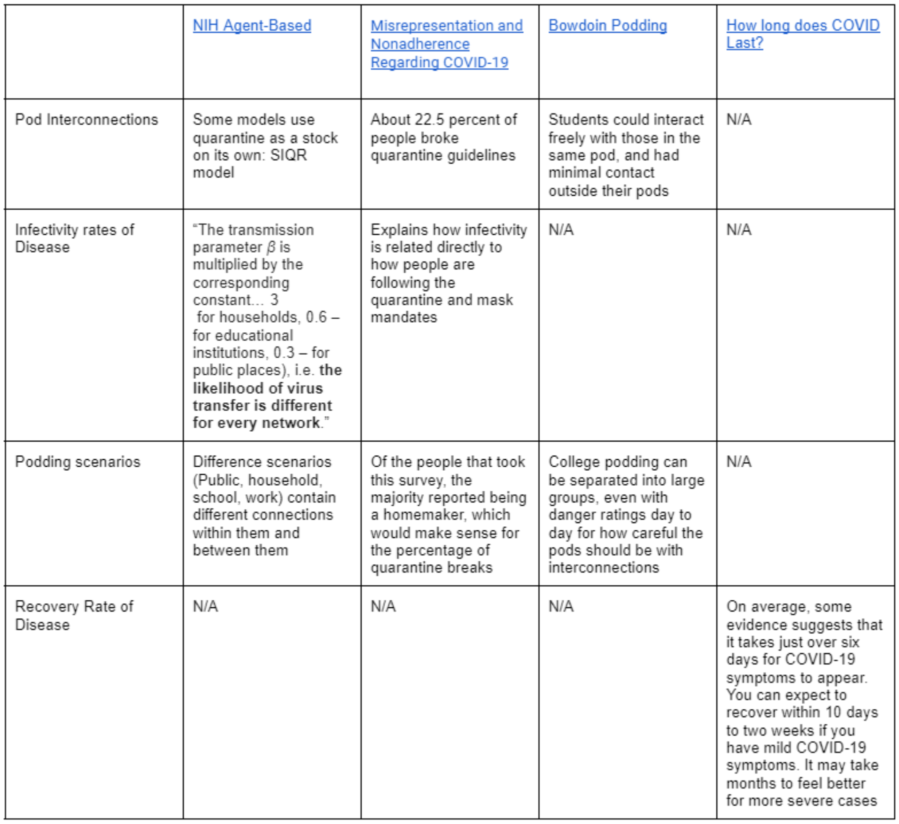
### Background Information:

In March of 2020, America shut down as schools, businesses, and public places closed in an attempt to slow the spread of COVID-19. This method of precaution, known as podding, or quarantine, limits the spread of disease by limiting social interaction. Podding involves quarantining people in "pods", often within their houses, apartments, or places of work. In this paper, we investigate the spread of disease through four different communities: residential, apartmental, collegiate, and a control community. However, these pods are not completely airtight; realistically, although a quarantine is in place, people still need to leave their pods for groceries and other essential supplies.

### Importance/Interest:

In any pandemic, the goal is to limit the spread and destruction of a disease; however, the measures we take can often have unintended consequences. In 2020, quarantine led to economic collapse, soaring rates of depression, and a lack of quality education/opportunities for America's youth. Our model aims to demonstrate how effective various levels of podding are -- specifically, we aim to measure the peak level of infected people and the timestep at which the disease peak occurs. With the information our model produces, we can make better informed decisions; maximizing safety during an outbreak while minimizing negative psychological, economic, and educational effects.

### Research Synthesis:



| | NIH Agent-Based | Misrepresentation and Nonadherence Regarding COVID-19 | Bowdoin Podding | How long does COVID Last? |
|---|---|---|---|---|
| Pod Interconnections | Some models use quarantine as a stock on its own: SIQR model | About 22.5 percent of people broke quarantine guidelines | Students could interact freely with those in the same pod, and had minimal contact outside their pods | N/A |
| Infectivity rates of Disease | "The transmission parameter $\beta$ is multiplied by the corresponding constant… 3 for households, 0.6 – for educational institutions, 0.3 – for public places), i.e. **the likelihood of virus transfer is different for every network**." | Explains how infectivity is related directly to how people are following the quarantine and mask mandates | N/A | N/A |
| Podding scenarios | Difference scenarios (Public, household, school, work) contain different connections within them and between them | Of the people that took this survey, the majority reported being a homemaker, which would make sense for the percentage of quarantine breaks | College podding can be separated into large groups, even with danger ratings day to day for how careful the pods should be with interconnections | N/A |
| Recovery Rate of Disease | N/A | N/A | N/A | On average, some evidence suggests that it takes just over six days for COVID-19 symptoms to appear. You can expect to recover within 10 days to two weeks if you have mild COVID-19 symptoms. It may take months to feel better for more severe cases |

### References:

NIH Agent-Based

Misreprisentation and Nonadherence

Bowdoin Podding

How long does COVID last?

### Parameters:

- **the rate of infection is 0.005 (chance of an infected person infecting another person they have contact with per timestep).** We chose this value when considering how covid acts in the real world and comparing disease spread in different communities. In our control model, when the rate of infection is 0.005, the peak number of people infected is 355/400, but everyone eventually becomes infected. This is similar to the real world "worst-case" scenario - in the unlikely scenario that 400 people were trapped in a room and refused to quarantine the sick, the infection would spread rapidly, but not rapidly enough to affect everyone simultaneously. Compared to our residential model, where the peak is 137/400, and not everyone is infected at the end of our simulation. This also mimics the real world - in our recent quarantine, COVID quickly spread through many households, but many others were also cautious enough and delayed infection until after quarantine ended. Overall, we chose to use an infection rate of 0.005 because it mimics how infections spread in the real world.

- **the rate of recovery is 0.05 (chance of an infected person recovering per timestep).** We chose this value after extensive research. According to the Mayo Clinic, in their article "How long does COVID last?", they found that people are infected but asymptomatic for 6 days on average, and then infected with symptoms for 10-14 days on average, depending on the person and severity of disease. In total, people have covid for between 16 and 20 days on average. We chose the upper bound for our study given that in extreme cases, it may take months to fully recover due to an effect called "long covid". To model an average of 20 days to recover, we gave people a 1/20 (or 0.05) chance of recovering on any given day.
- **In our model, 25% of people in each pod break quarantine.** We chose 25% after referencing the study "Misreprisentation and Nonadherence Regarding COVID-19", where roughly 22.5% of people reported breaking quarantine.
- Our Residential Model is representative of a small suburban neighborhood with **one hundred houses, each with four people** (two parents, two children) in each house. One member from each house **(25%) leaves the pod and becomes exposed each timestep.** Based of sources "NIH Agent-Based" and "Misreprisentation and Nonadherence"
- Our Urban Model is representative of an urban street with **twenty apartment complexes, each with twenty people** (or five families) in each building. Five members from each building **(25%) leave the pod and become exposed each timestep.** Based off "NIH Agent-Based"
- Our Collegiate Model is representative of the dorms in a small college setting with **four dorms, each with one hundred people** in each building. Twenty-five people from each building **(25%) leave the pod and become exposed each timestep.** Based off "NIH Agent-Based" and "Bowdoin Podding"
- Our Control Model is representative of an unquarantined community **(one pod) with four hundred people** all of whom **(100%) are trapped in a large room together and must interact. This is the "worst case scenario" in terms of podding.**

# Section II: Model and Methodology

## Description:

Our Model and Methodology section is broken into four sections: Creation of Pods, Running ABSIR to Simulate Infection, Creating Multiple Realizations, and Model Validation. Each section goes into further detail about the choices we made when building that section, and a broad overview of model and methodology is below:

1.) Creation of Pods: We build our four pods using for loops and represent them using a heatmap.

2.) Running ABSIR to Simulate Infection: Using the simulate_absir function runs a single realization, demonstrating the mechanics of the function.

3.) Creating Multiple Realizations: Using the simulate_absir function to run multiple realizations, allowing us to analyze the mean data of the realizations for a more accurate data set.

4.) Model Validation: Comparing our model against other research studies and ensuring our model makes sense with respect to the real world.

## Underlying Assumptions:

- We assume that all pods in any given model are exactly the same size (i.e. exactly 4 people, exactly 20 people, etc.). It would be significantly more difficult to randomize the pods in each community,
- We assume that everyone within the pod has an equal likelihood of infecting others within that pod. For our model to work, those inside of the pods and those outside of the pods need to have consistent infection rates.
- We assume that the same individuals leave the pods each timestep and are equally exposed to the other people who leave the pods each timestep. In a real world scenario, the same person(s) would leave the pod, allowing others in the pod to be completely isolated from the rest of society.
- We assume that people are not able to get reinfected after recovering. This would also significantly (and unnecessarily) complicate the model.
- We assume that everyone is equally able to recover (although this does NOT mean that everyone recovers in the same timespan).

## Creation of Pods:

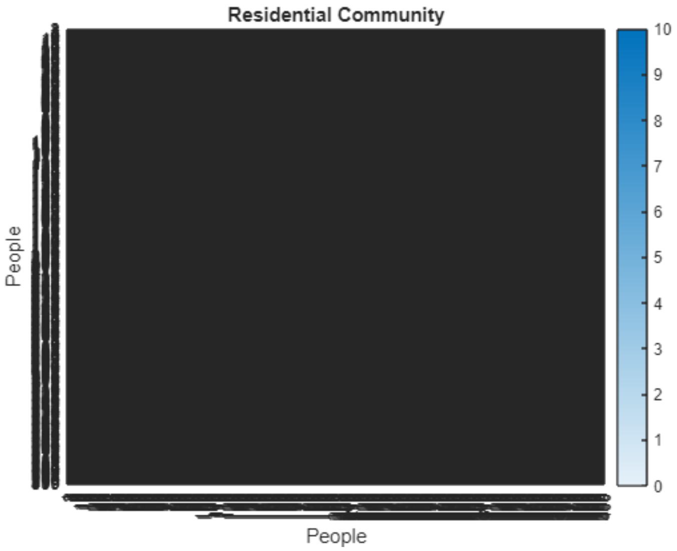
Creation of pods involves creating smaller "base pods", and then using for loops to create a 400x400 matrix composed of base pods. We can then visualize the connections within these pods by using a heatmap.

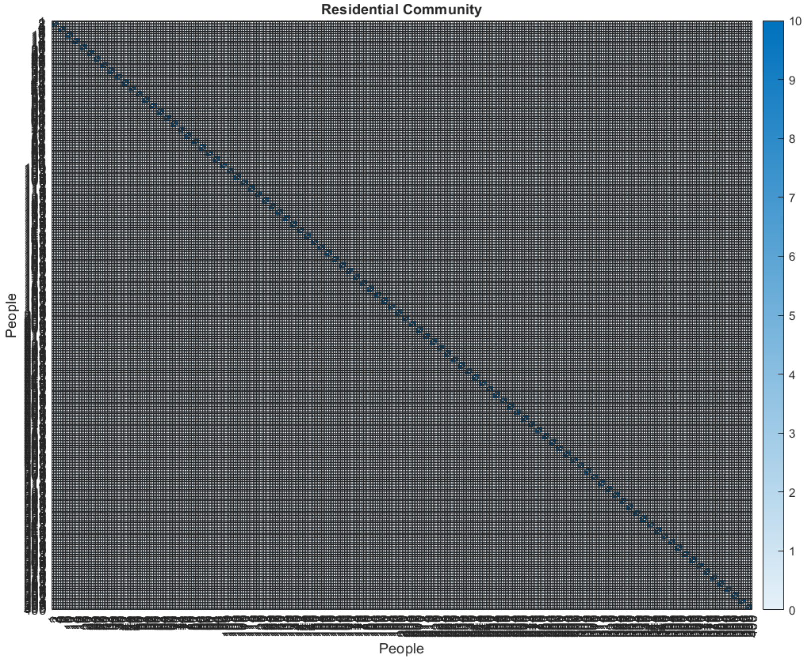
IMPORTANT NOTE: MATLAB's large heatmaps render as unreadable black squares. To combat this, we have attached a "close-up" image of each of our pods below each of our heatmaps.

### Residential Pod (100 Pods, 4 People Each)

```
%Residential Pods: 100 pods, 4 ppl each, 1 person outside contact per pod
pRes = zeros(400,400); %making whole matrix
pRes_4 = 10 * ~eye(4); %making residential base pod matrix
pBase = zeros(4,4); %making 25% connected matrix
pBase(1,1) = 2; %making connections for base 25% matrix
for k = 1:100 %setting up an entire matrix where 25% of people are connected by a factor of 2
    slice_begin2 = (k-1) * 4 + 1;
    slice_end2 = slice_begin2 + 4 -1;
    for i = 1:100
        slice_begin = (i - 1) * 4 + 1;
        slice_end = slice_begin + 4 - 1;
        pRes(slice_begin2:slice_end2, slice_begin:slice_end) = pBase;
    end
end

for i = 1:100 %adding our 100 residential pods to the matrix
    slice_begin = (i - 1) * 4 + 1;
    slice_end = slice_begin + 4 - 1;
    pRes(slice_begin:slice_end, slice_begin:slice_end) = pRes_4;
end
```

```
figure()
heatmap(pRes) %visualizing via a heatmap (a zoomed in picture is attached)
title('Residential Community')
ylabel('People'); xlabel('People');
```

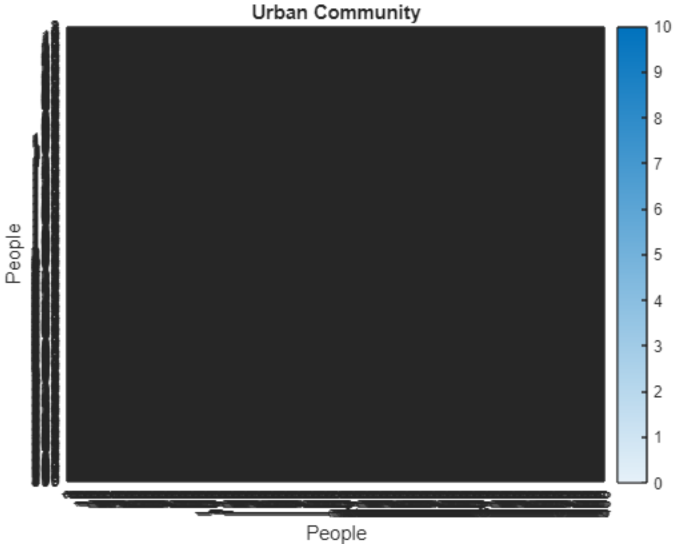


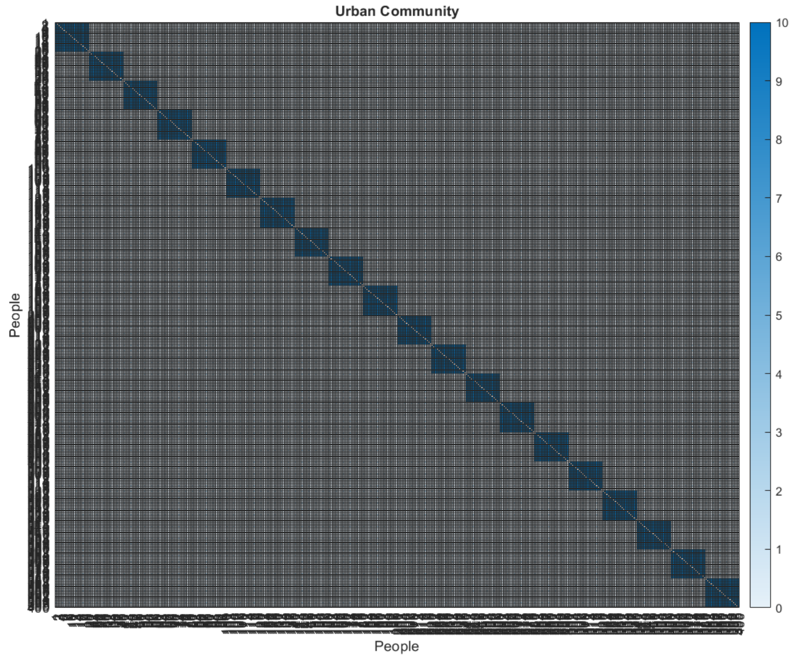


*This is a heatmap of our residential pods. Each axis is 400 people, and each box is a connection between two people. For example, box (34, 57) is the connection between the 34th person and the 57th person. The boxes are colored based on the level of connection - darker boxes indicate higher connection. In this heatmap, we can clearly see the 100 pods that have higher connection along the diagonal.*

## Urban Pod (20 Pods, 20 People Each)

```
%Urban Pods: 20 pods, 20 ppl each, 5 person outside contact per pod
pUrb = zeros(400,400); %making whole matrix
pUrb_20 = 10 * ~eye(20); %making apartmental pod matrix
pBase = zeros(4,4); %making 25% connected matrix
pBase(1,1) = 2; %making connections for base 25% matrix
for k = 1:100 %setting up an entire matrix where 25% of people are connected by a factor of 2
    slice_begin2 = (k-1) * 4 + 1;
    slice_end2 = slice_begin2 + 4 -1;
    for i = 1:100
        slice_begin = (i - 1) * 4 + 1;
        slice_end = slice_begin + 4 - 1;
        pUrb(slice_begin2:slice_end2, slice_begin:slice_end) = pBase;
    end
end

for i = 1:20 %adding our 20 apartmental pods to the matrix
    slice_begin = (i - 1) * 20 + 1;
    slice_end = slice_begin + 20 - 1;
    pUrb(slice_begin:slice_end, slice_begin:slice_end) = pUrb_20;
end

figure()
heatmap(pUrb); %visualizing via a heatmap (a zoomed in picture is attached)
title('Urban Community')
ylabel('People'); xlabel('People');
```



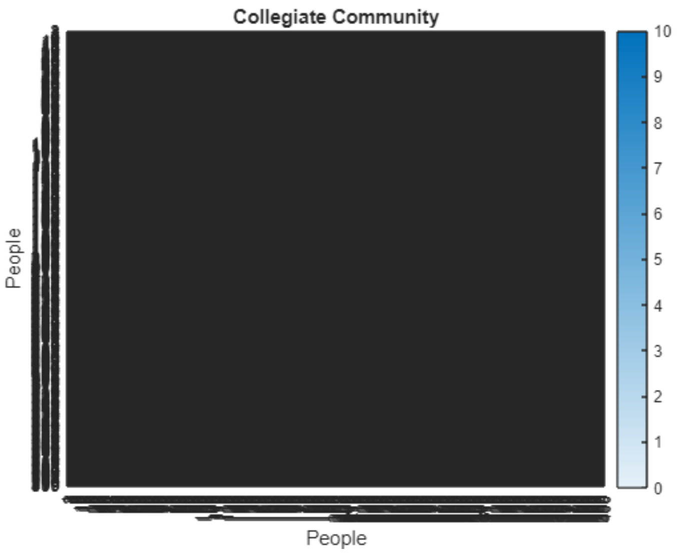


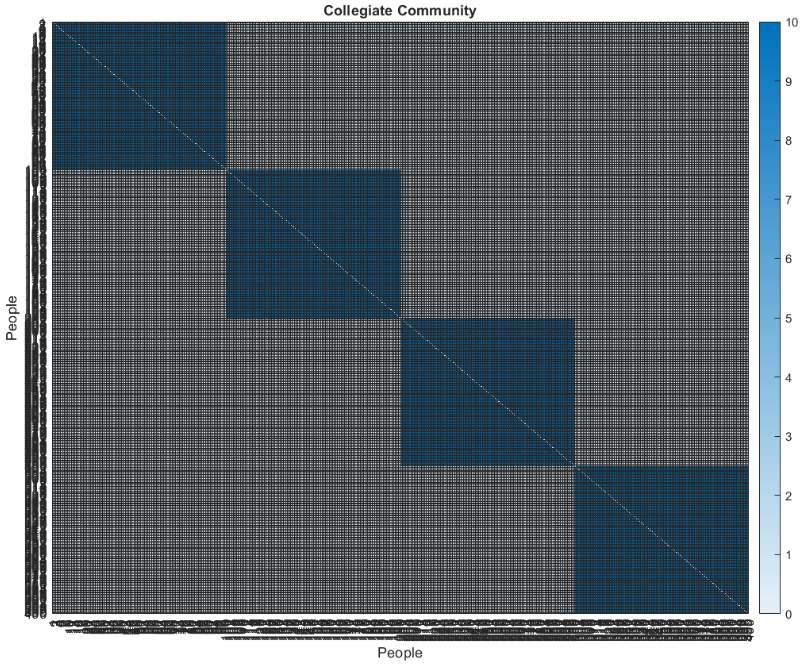
*This is a heatmap of our urban pods. Each axis is 400 people, and each box is a connection between two people. For example, box (34, 57) is the connection between the 34th person and the 57th person. The boxes are colored based on the level of connection - darker boxes indicate higher connection. In this heatmap, we can clearly see the 20 pods that have higher connection along the diagonal.*

### Collegiate Pod (4 Pods, 100 People Each)

```
%College Dorms: 4 pods, 100 people each, 25 have outside contact per pod
pCol = zeros(400,400); %making whole matrix
pCol_100 = 10 * ~eye(100); %making collegiate pod matrix
pBase = zeros(4,4); %making 25% connected matrix
pBase(1,1) = 2; %making connections for base 25% matrix
for k = 1:100 %setting up an entire matrix where 25% of people are connected by a factor of 2
    slice_begin2 = (k-1) * 4 + 1;
    slice_end2 = slice_begin2 + 4 -1;
    for i = 1:100
        slice_begin = (i - 1) * 4 + 1;
        slice_end = slice_begin + 4 - 1;
        pCol(slice_begin2:slice_end2, slice_begin:slice_end) = pBase;
    end
end
```

```
for i = 1:4 %adding our 4 collegiate pods to the matrix
    slice_begin = (i - 1) * 100 + 1;
    slice_end = slice_begin + 100 - 1;
    pCol(slice_begin:slice_end, slice_begin:slice_end) = pCol_100;
end

figure()
heatmap(pCol) %A zoomed in picture will be attached
title('Collegiate Community')
ylabel('People'); xlabel('People');
```

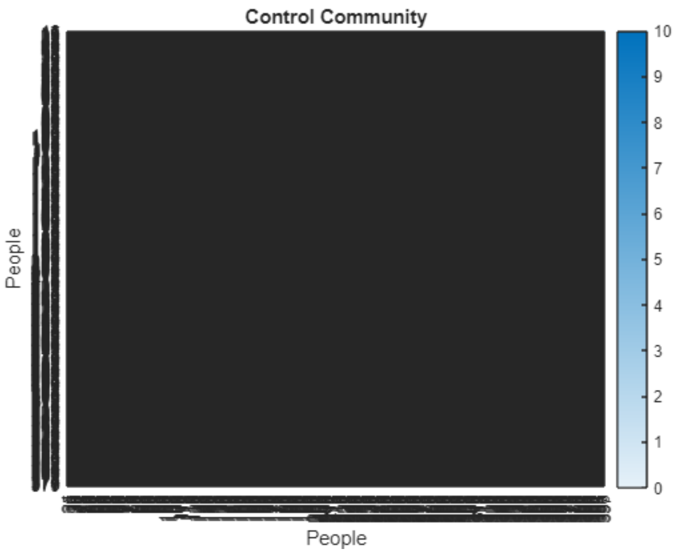


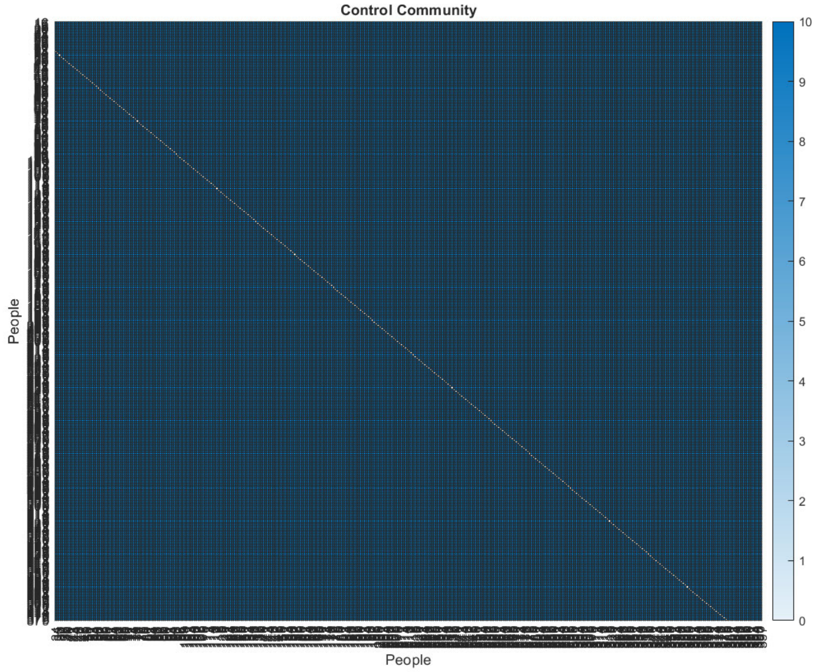


*This is a heatmap of our collegiate pods. Each axis is 400 people, and each box is a connection between two people. For example, box (34, 57) is the connection between the 34th person and the 57th person. The boxes are colored based on the level of connection - darker boxes indicate higher connection. In this heatmap, we can clearly see the 20 pods that have higher connection along the diagonal.*

## Control Pod (No Podding)

```
%Control Pod with no podding, everyone has contact with everyone
pCon = 10*ones(400,400) - 10*eye(400,400); %making whole matrix

figure()
heatmap(pCon) %A zoomed in picture will be attached
title('Control Community')
ylabel('People'); xlabel('People');
```





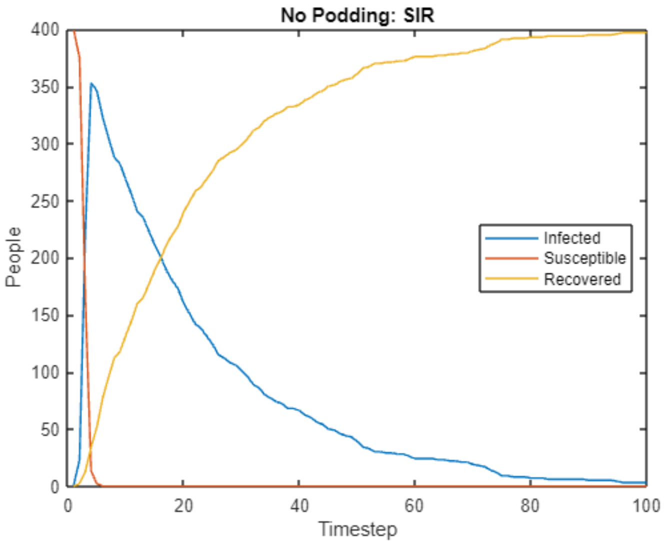
*This is a heatmap of our control community. Each axis is 400 people, and each box is a connection between two people. For example, box (34, 57) is the connection between the 34th person and the 57th person. The boxes are colored based on the level of connection - darker boxes indicate higher connection. In this heatmap, we can clearly see that everyone is in a single pod that has high connection.*

## Running ABSIR to Simulate Infection:

In order to demonstrate what the function simulate_absir.m does, we are running a single realization over the control matrix. This will help us visualize our future results; however, this run of the model has no bearing on our final results.

```
%set parameter values
infection_rate = .005; %setting our infection rate (chance of an infected person infecting another person they have contact with per times
recovery_rate = .05; %setting our recovery rate (change that someone will recover per timestep)

Iv1 = zeros(400,1); %setting up our base infection matrix
Iv1(1) = 1; %infecting a single individual in our base infection matrix
time = [1:100]; %creating a matrix for our timesteps
[Sh, Ih, Rh] = simulate_absir(pCon, Iv1, 100, infection_rate, recovery_rate); %running simulate_absir to create a single realization
I_count = sum(Ih, 1); %summing our number of infection people each timestep
S_count = sum(Sh, 1); %summing the number of susceptible people each timestep
R_count = sum(Rh, 1); %summing the number of recovered people each timestep

%plotting results
figure()
plot(time, I_count, time, S_count, time, R_count); hold on;
axis([0,100,0,400]);
legend('Infected', 'Susceptible', 'Recovered', 'Location','east')
title('No Podding: SIR')
```

```
ylabel('People')
xlabel('Timestep')
```



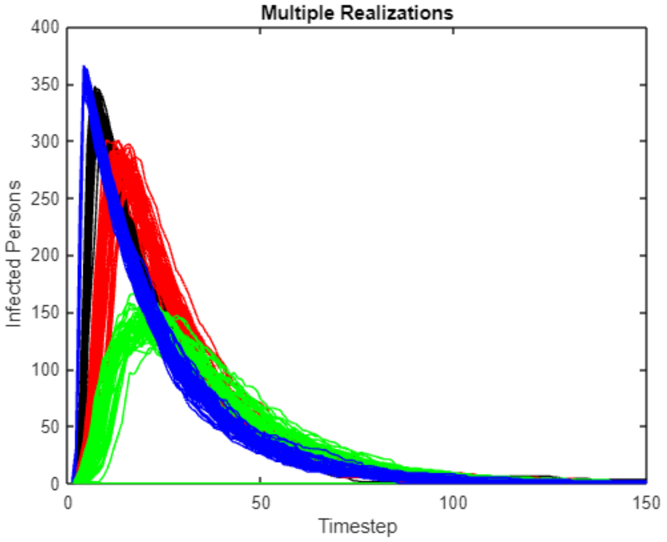
*This graph is a single realization from simulate_absir. It demonstrates how susceptible, infected, and recovered people fluctuate over time for our control community (400 people in a single pod).*

## Creating Multiple Realizations:

In order to maximize the accuracy of our data, we are running simulate_absir multiple times to create several realizations of our data. This is due to the fact that simulate_absir utilizes randomness within each realization that it runs. To find the most accurate/applicable data, we are finding the mean of our realizations. This allows us to better visualize our data and increase the consistency of our data. Additionally, to verify our code, we have several assert statements in this section to ensure the model is running correctly.

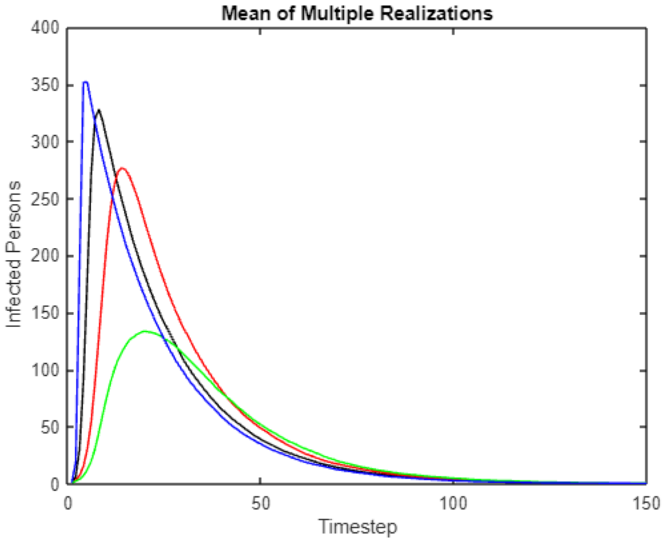
```
% Set the number of replications for the agent-based model
n_realizations = 50;

% Run each simulation for 150 timesteps
T_simulation = 150;

% Reserve space for simulation results
I_ensembleCol = zeros(T_simulation, n_realizations);
I_ensembleRes = zeros(T_simulation, n_realizations);
I_ensembleUrb = zeros(T_simulation, n_realizations);
I_ensembleCon = zeros(T_simulation, n_realizations);
R_ensembleCol = zeros(T_simulation, n_realizations);
R_ensembleRes = zeros(T_simulation, n_realizations);
R_ensembleUrb = zeros(T_simulation, n_realizations);
R_ensembleCon = zeros(T_simulation, n_realizations);
S_ensembleCol = zeros(T_simulation, n_realizations);
S_ensembleRes = zeros(T_simulation, n_realizations);
S_ensembleUrb = zeros(T_simulation, n_realizations);
S_ensembleCon = zeros(T_simulation, n_realizations);
I_ensembleColMax = zeros(1, n_realizations);
I_ensembleResMax = zeros(1, n_realizations);
I_ensembleUrbMax = zeros(1, n_realizations);
I_ensembleConMax = zeros(1, n_realizations);

% Set the random seed.
rng(101)


% Generate the ensembles of results
for i = 1 : n_realizations
    % Run a single simulation
    [Sh, Ih, Rh] = simulate_absir(pCol, Iv1, T_simulation, infection_rate, recovery_rate);
    % Summarize and store the infection history
    I_ensembleCol(:, i) = sum(Ih, 1);
    I_ensembleColMax(i) = max(I_ensembleCol(:, i));

    % Assert that the number of recovered people only ever increases
    % Assert that the number of susceptible people only ever decreases
    R_ensembleCol(:,i) = sum(Rh,1);
    S_ensembleCol(:,i) = sum(Sh,1);
    for t = 2:T_simulation
        assert(R_ensembleCol(t, i) >= R_ensembleCol(t - 1, i));
        assert(S_ensembleCol(t, i) <= S_ensembleCol(t - 1, i));
    end
    % Assert that every person in each timestep for the collegiate community is either susceptible, infected, or recovered
    assert(isequal(ones(400,T_simulation),(Sh + Ih + Rh)))
end

for i = 1 : n_realizations
    % Run a single simulation
    [Sh, Ih, Rh] = simulate_absir(pUrb, Iv1, T_simulation, infection_rate, recovery_rate);
    % Summarize and store the infection history
    I_ensembleUrb(:, i) = sum(Ih, 1);
    I_ensembleUrbMax(i) = max(I_ensembleUrb(:, i));

    % Assert that the number of recovered people only ever increases
    % Assert that the number of susceptible people only ever decreases
    R_ensembleUrb(:,i) = sum(Rh,1);
    S_ensembleUrb(:,i) = sum(Sh,1);
    for t = 2:T_simulation
        assert(R_ensembleUrb(t, i) >= R_ensembleUrb(t - 1, i));
        assert(S_ensembleUrb(t, i) <= S_ensembleUrb(t - 1, i));
    end

    % Assert that every person in each timestep for the urban community is either susceptible, infected, or recovered
    assert(isequal(ones(400,T_simulation),(Sh + Ih + Rh)))
end


for i = 1 : n_realizations
    % Run a single simulation
    [Sh, Ih, Rh] = simulate_absir(pRes, Iv1, T_simulation, infection_rate, recovery_rate);
    % Summarize and store the infection history
    I_ensembleRes(:, i) = sum(Ih, 1);
    I_ensembleResMax(i) = max(I_ensembleRes(:, i));

    % Assert that the number of recovered people only ever increases
    % Assert that the number of susceptible people only ever decreases
    R_ensembleRes(:,i) = sum(Rh,1);
    S_ensembleRes(:,i) = sum(Sh,1);
    for t = 2:T_simulation
        assert(R_ensembleRes(t, i) >= R_ensembleRes(t - 1, i));
        assert(S_ensembleRes(t, i) <= S_ensembleRes(t - 1, i));
    end

    % Assert that every person in each timestep for the residential community is either susceptible, infected, or recovered
    assert(isequal(ones(400,T_simulation),(Sh + Ih + Rh)))
end


for i = 1 : n_realizations
    % Run a single simulation
    [Sh, Ih, Rh] = simulate_absir(pCon, Iv1, T_simulation, infection_rate, recovery_rate);
    % Summarize and store the infection history
    I_ensembleCon(:, i) = sum(Ih, 1);
    I_ensembleConMax(i) = max(I_ensembleCon(:, i));

    % Assert that the number of recovered people only ever increases
    % Assert that the number of susceptible people only ever decreases
    R_ensembleCon(:,i) = sum(Rh,1);
    S_ensembleCon(:,i) = sum(Sh,1);
    for t = 2:T_simulation
        assert(R_ensembleCon(t, i) >= R_ensembleCon(t - 1, i));
        assert(S_ensembleCon(t, i) <= S_ensembleCon(t - 1, i));
    end

    % Assert that every person in each timestep for the control is either susceptible, infected, or recovered
    assert(isequal(ones(400,T_simulation),(Sh + Ih + Rh)))
end


%find the mean curve of each set of realizations
I_mean_Con = mean(I_ensembleCon, 2);
I_mean_Col = mean(I_ensembleCol, 2);
I_mean_Urb = mean(I_ensembleUrb, 2);
I_mean_Res = mean(I_ensembleRes, 2);

% Visualize the ensembles
figure()
plot(I_ensembleCol, 'k-', 'LineWidth', 0.1); hold on;
plot(I_ensembleUrb, 'r-', 'LineWidth', 0.1);
plot(I_ensembleRes, 'g-', 'LineWidth', 0.1);
plot(I_ensembleCon, 'b-', 'LineWidth', 0.1);
xlabel('Timestep')
```

```
ylabel('Infected Persons')
ylim([0 400])
title('Multiple Realizations')

hold off
```



*This graph plots four ensembles in relation to eachother. Each line is the number of infected people in a community over time. The blue ensemble is our control community, the black is our collegiate community, the red is our urban community, and the green is our residential community. The lines are messy, in order to better visualize the data, we can find and plot the mean.*

```
% Visualize the mean curves
figure()
plot(I_mean_Col, 'k-', 'LineWidth', 0.1, 'HandleVisibility', 'off'); hold on;
plot(I_mean_Urb, 'r-', 'LineWidth', 0.1, 'HandleVisibility', 'off');
plot(I_mean_Res, 'g-', 'LineWidth', 0.1, 'HandleVisibility', 'off');
plot(I_mean_Con, 'b-', 'LineWidth', 0.1, 'HandleVisibility', 'off');
xlabel('Timestep')
ylabel('Infected Persons')
title('Mean of Multiple Realizations')
ylim([0 400])
```



*This graph plots the means of four ensembles in relation to eachother. Each line is the number of infected people in a community over time. The blue ensemble is our control community, the black is our collegiate community, the red is our urban community, and the green is our residential community.*

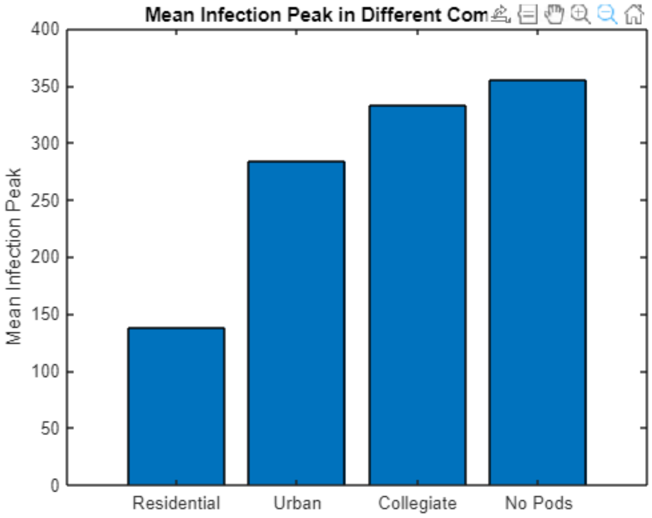
## Metrics:

In order to have concrete comparisons between the different pods, we have chosen to examine the differences in peak number of infected people and the timestep that the peak number of infected people occurs. To do this, we are taking the peak (number of infected people and timestep) of each realization and finding the mean of that data for each different community. We then use bar graphs to help best visualize our results.

### Finding Peak Infected in Each Community

```
% finding the max points for each community
```

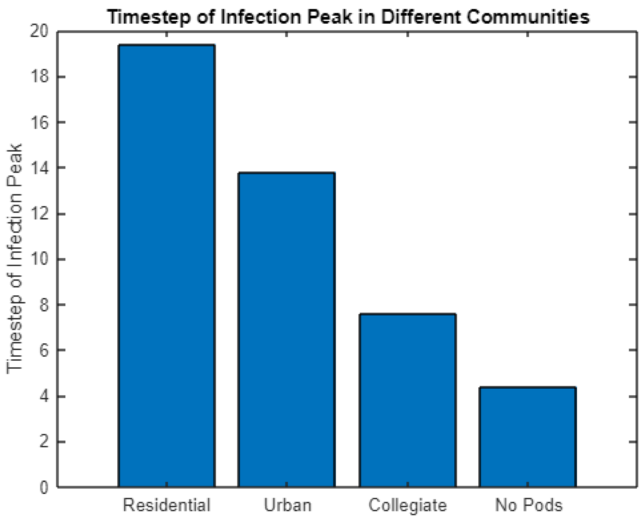
```
[I_max_Res, T_max_Res] = max(I_ensembleRes, [], 1);
[I_max_Urb, T_max_Urb] = max(I_ensembleUrb, [], 1);
[I_max_Col, T_max_Col] = max(I_ensembleCol, [], 1);
[I_max_Con, T_max_Con] = max(I_ensembleCon, [], 1);

% finding the peak number of infected people for each community
I_max_data = [
    I_max_Res;
    I_max_Urb;
    I_max_Col;
    I_max_Con];

% creating labels for the communities for our bar graphs
X = categorical({'Residential', 'Urban', 'Collegiate', 'No Pods'});
X = reordercats(X, {'Residential', 'Urban', 'Collegiate', 'No Pods'});

% visualizing the data and graphing the mean
figure(); clf;
mean_I_max = mean(I_max_data, 2);
bar(X, mean_I_max)
ylabel('Mean Infection Peak')
title('Mean Infection Peak in Different Communities')
```



*This graph compares the mean peak number of infected people from each community ensemble.*

### Finding Timestep of Peak Infected in Each Community

```
% finding the timestep of the peak number of infected people for each community
T_max_data = [
    T_max_Res;
    T_max_Urb;
    T_max_Col;
    T_max_Con];

% visualizing the data and graphing the mean
figure(); clf;
mean_T_Max = mean(T_max_data, 2);
bar(X, mean_T_Max)
ylabel('Timestep of Infection Peak')
title('Timestep of Infection Peak in Different Communities')
```



*This graph compares the mean timestep at which the peak number of infected people occurs from each community ensemble.*

## Model Validation:

Although there is little "real data" research that has been conducted on the topics of community podding, the research paper "Social network-based distancing strategies to flatten the COVID 19 curve in a post-lockdown world" closely mimics the results found in our model. In an article that analyzes the study, they reported that "compared to no social distancing, quarantine bubbles would delay the peak of infections by 37%, decrease the height of the peak by 60% and result in 30% fewer infected individuals overall." The quarantine bubbles, or quaranteams, referenced in the article refer to "quaranteams are founded on the idea that people can interact freely within a group, but that group stays isolated from other people as much as possible". This matches nearly exactly with the data we found for our residential model, where the peak infection was (137.72/355.10), 38.7834413% of the control data, or decreased by 61.22%.

This article can be found at: Quarantine bubbles – when done right – limit coronavirus risk and help fight loneliness (theconversation.com)

This paper can be found at: (PDF) Social network-based distancing strategies to flatten the COVID 19 curve in a post-lockdown world (researchgate.net)

# Section III: Results

### Metrics and Sweeps:

To best answer our question, we need a parameter sweep over pod size, each pod representing a different community. We chose two main metrics to study for each of our pods: peak number of infected persons and timestep of peak infected persons. Here are our results for each metric:

- Peak number of infected persons:
- Residential..........137.72 persons
- Urban..................283.66 persons
- Collegiate............332.60 persons
- Control................355.10 persons
- Timestep of peak infected persons:
- Residential............19.42 timesteps
- Urban....................13.80 timesteps
- Collegiate................7.58 timesteps
- Control....................4.38 timesteps

### Anaylsis:

There is a direct positive correlation between pod size and the peak number of infected people. As the pods get smaller, the peak number of infected people shrinks. Each community increases the pod size by five times the last community (four people, then twenty, then one hundred). The first increase, from residential to urban, the peak number of infected people more than doubled (145.94 people). However, the next increase, from urban to collegiate, only increases the peak number of infected people by 17% (48.94 people). And the final increase, comparing collegiate to control, is only a 6% increase (22.50 people). Therefore, having increasingly smaller pods exponentially decreases the peak number of infected people. **This suggests podding is only significantly effective at decreasing the peak number of infected people if the pods are very small and the quarantine is highly controlled, as in our model.**

There is a direct negative correlation between pod size and the timestep where the peak number of infected people occurs. **Even though the smaller pods had a lower peak number of infected people, it still took longer to reach that peak than the larger pods.** Again, we see significant change from the residential community to the urban community, with the residential community taking 5.62 timesteps longer to reach its peak. However, the change from urban to collegiate is actually bigger than the change from residential to urban, with an urban community taking an additional 6.22 timesteps to reach its peak. The change from collegiate to control drops back down again, with the collegiate community taking only 3.20 timesteps longer to reach its peak. Therefore, while having smaller pods always increases the time it takes to peak, it is not an exponential increase. **This suggests that podding is effective at prolonging the peak number of infected people in all of our modeled communities.**

# Section IV: Interpretation

### Original Question:

How does a disease spread through different types of quarantined communities?

### ABT:

Podding or quarantining is one way to limit the spread of disease, but it is difficult to discern how effective these methods are in real life, especially given the different circumstances of people's lives. Therefore, we tested various podding models and discovered that less people per pod in a given community leads to a lower peak number of people infected and a delayed peak. Therefore, a community with many small pods, such as a suburban neighborhood, is best suited for quarantine, but a community with large pods, such as college dorms, would be better off using other methods of health safety.

### Limitations:

Fundamentally, our model is quite limited. It goes without saying that people do not perfectly follow the predictions and assumptions we made in the model. For example, we assumed that exactly once every timestep, exactly 25% of the pod interacted with 25% of every other pod. This was done to model things like going grocery shopping, but exhibits a clear limitation of the model - people are not perfect and dont follow code like our model does. A time where our model would not hold includes 'super-spreader' events where there is a large gathering and a lot of people are in contact with each other. In our model, this simply doesn't exist.

### Next Steps in Model Development:

Future work should build upon the base model as well as create more certainty in the model. For us, future work could include a model person-specific infection chance rather than the base chances we gave them. For example, this would mean that old or very young people would have greater infection chances, which would increase the certainty of our model by better representing real communities. This could be expanded upon to represent more specific types of communities like daycare centers or nursing homes (both places with higher chances of infections).

```
function [Sh, Ih, Rh] = simulate_absir(M, Iv0, T, infection_rate, recovery_rate)
% Simulate agent-based transmission model. Uses a graph to represent social
% connectivity between agents.
%
% Note: You may find it helpful to summarize the state history matrices via
% summation. For instance sum(Ih, 1) will return the total number of
% infected persons at each timestep in the simulation.
%
% Inputs
%   M (square matrix): Adjacency matrix
%   Iv0 (column vector): Initial infection state
%   T (integer): Number of timesteps to simulate
%   infection_rate (float): Infection rate (probabilistic)
%   recovery_rate (float): Recovery rate (probabilistic)
%
% Returns
%   Sh (matrix): Susceptible state history
%   Ih (matrix): Infected state history
%   Rh (matrix): Recovered state history

    % Setup
    dim = length(Iv0);  % Dimensions of initial state
    Ih = zeros(dim, T); % Infection history
    Ih(:, 1) = Iv0; % Record the initial state to infection history
    Rh = zeros(dim, T); % Recovery history

    % Construct an "action" helper function
    function [I, R] = action(I, R)
        % Compute infection probabilities based on the social graph
        v_eff = infection_rate * M * I;
        v_pr = v_eff ./ (1 + v_eff);

        % Draw random values
        v_infect = rand(dim, 1) <= v_pr;
        v_recover = rand(dim, 1) <= recovery_rate;

        % Infect non-recovered individuals
        I = I | v_infect & (~R);
        % Recover infected individuals
        R = R | (I & v_recover);
        I = I & (~R);
    end

    % Run simulation
    for i = 2:T
        [Ih(:, i), Rh(:, i)] = action(Ih(:, i-1), Rh(:, i-1));
    end

    % Compute susceptible history
    Sh = ones(dim, T) - Ih - Rh;
end
```