Masks Effectitivity Using Modified SIR Model

Original Authors: Sam Wisnoski, Minu Rajasingh, Ahan Trivedi

Covid-19 has created a revival in daily public health interventions. We will research how use protective masks can stop the spread of a pandemic, and we will use SIR models to determine if masks can truly keep people safer. In order to demonstrate this, we will research the infectivity rate, recovery rate, and re-infection rate of the Covid-19 virus. We will create an SIR model using our findings and use a parameter sweep over mask effectivity to answer the question: **How effective do masks need to be to cut the peak number of infected persons by 50%?**

Section I: Question

Modeling Question

- ***Research Question:*** *How effective do masks need to be to cut the peak number of infected persons by 50%?*
- ***Type of question:*** *Our question is an explanitory question; it aims to identify how masks effectivness is related to the peak number of infected persons.*

Importance/Interest

This research is important because it will demonstrate the importance of masks in everyday life. Our model runs under the assumption that everyone is wears a mask in real life - however, this is not the case. Researching masks and running a simulation based on real world data, proving the importance of using masks, will serve as another reason for people to mask up and slow the spread of real world infectious diseases. .

Research Synthesis

Topic	Eikenberry et al.	CDC	Krista Conger
Importance of Masking	Using masks can decrease the transmission rate as well as hospitalizations and deaths	As younger adults were less likely to take proper masking precautions in late 2022 in terms of covid 19, they had higher re-infection rates of Covid-19.	In high population areas where social distancing or other interventions can be difficult, masks are a cheap and effective way to reduce the spread of Covid-19.
Effectiveness of masks	An 80% use of masks in New York at a mask effectivity of 50% decreased peak daily deaths by 34-58%	Masks have been able to decrease the infection and the reinfection rate of covid-19, which has been shown in re-infection rates throughout the variants. Often though, other variables can confound the effect of masks.	In a study with more than 340,000 people, it was found that 40% masking lowers Covid infection by 12%. However, these rates were significantly higher for the elderly: 40% masking lowered infection by 35%.
Covid Infectivity	They simulated using values ranging from .5-1.5 for the infectious contact rate; how many people the infected person would infect.	Reinfection rate of covid-19 has decreased substantially to around a 2.7% reinfection rate.	“7.6% of people in the intervention villages reported COVID-19…during the eight-week study period…” equating to roughly 1% of the population being infected each week.

Background Information

- the infection rate of covid is around 1%
- and the resusceptibility rate of covid is around 2.7%
- in real life, only a fraction of the population wears a mask

References

Eikenberry et al.

https://www.sciencedirect.com/science/article/pii/S2468042720300117

CDC:

https://www.cdc.gov/mmwr/volumes/72/wr/mm7225a3.htm#:~:text=Reinfections%20represented%202.7%25%20of%20all,4%2FBA.\

Krista Conger:

Surgical masks reduce COVID-19 spread, large-scale study shows | News Center | Stanford Medicine

Section II: Model and Methodology

Description

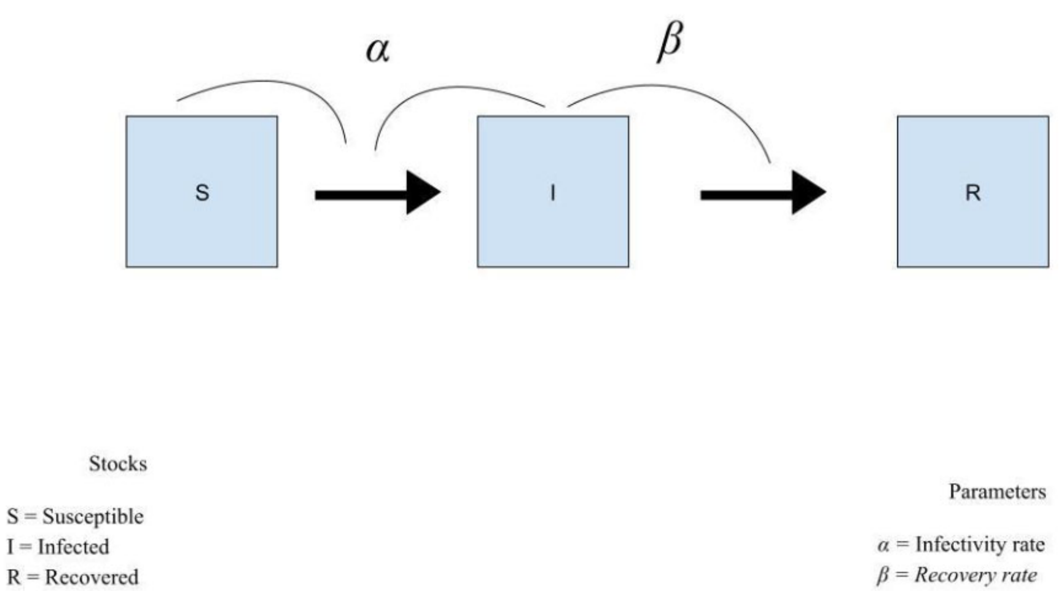
Our models are based on online data for the infection rate and re-susceptibility rate of covid-19. For our first model, we are using a basic SIR model. The first model, althought inaccurate, serves as a solid bases for our second model and allows for a deeper understanding of how diseases spread. For our second model, we will use a modified version of the SIR model. Our second model will include a flow from recovered people back into susceptible people to mimic how, in real life, people are able to become infected with Covid repeatedly. In order to answer our research question, we will perform a parameter sweep over mask efficiency in our second model to determine how effective masks need to be to cut the peak number of infected people by 50%.

Assumptions

- Our alpha value (infectivity rate) is 1%
- Our beta value (recovery rate) is 25%
- Our gamma value (re-susceptibility rate) is 2%
- Everyone wears a mask
- Everyone will recover, and everyone will take the same time to recover
- Everyone susceptible is equally susceptile to be infected

Model 1: Basic SIR Model

Stock and Flow Model (Model 1)



Update Equations (Model 1)

How we computed updates to infections and recoveries

infected = beta * i * s;

recovered = gamma * i;

How we updated the state of each stock

s_n = s - infected;

i_n = i + infected - recovered;

r_n = r + recovered;

Model 1 (Basic SIR Model)

Our initial state is a population of 100 persons: 99 susceptible, 1 infectious, 0 recovered

```
s_1 = 99;
i_1 = 1;
r_1 = 0;
```

Let's interpret t to be Weeks. Thus, we have the following parameter interpretations:

Infection rate: Each infected person has a chance to infect 1 susceptible person out of 100

Recovery rate: The infection lasts 4 weeks before recovery

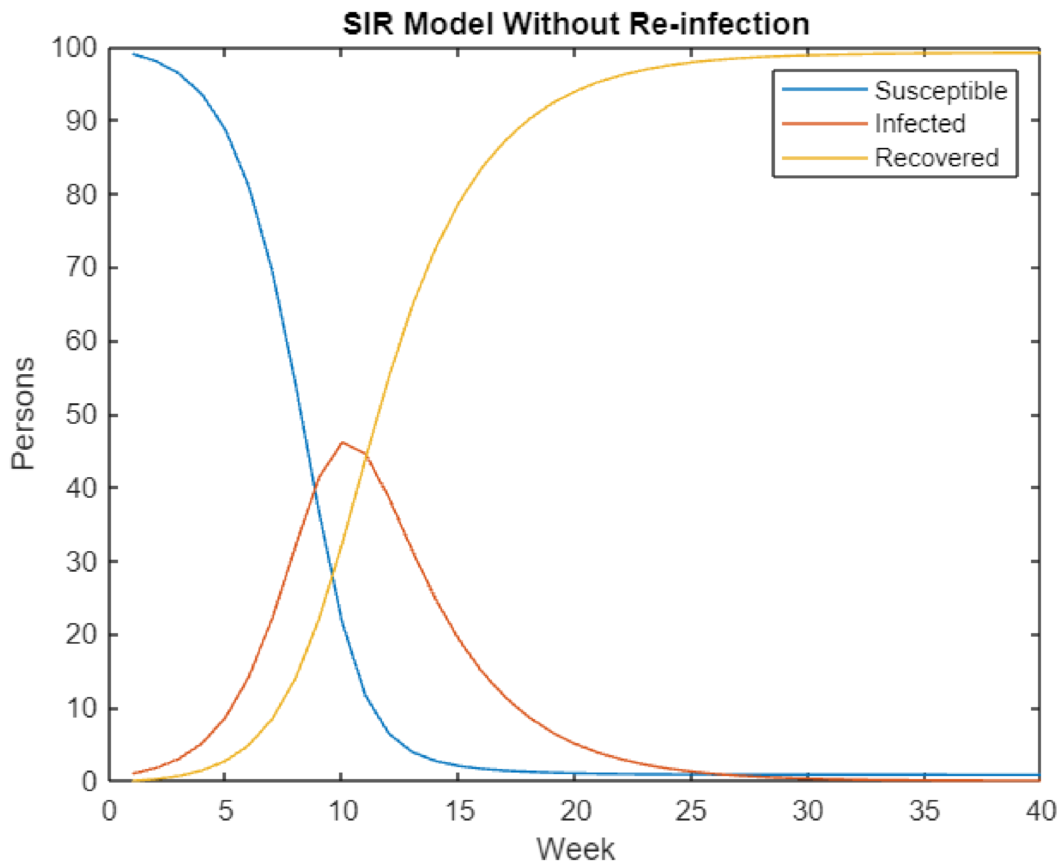
```
beta = 1 / 100; % Infection rate
gamma = 1 / 4; % Recovery rate
```

Run simulation over forty weeks

```
[S, I, R, W] = simulate_sir(s_1, i_1, r_1, beta, gamma, 40);
```

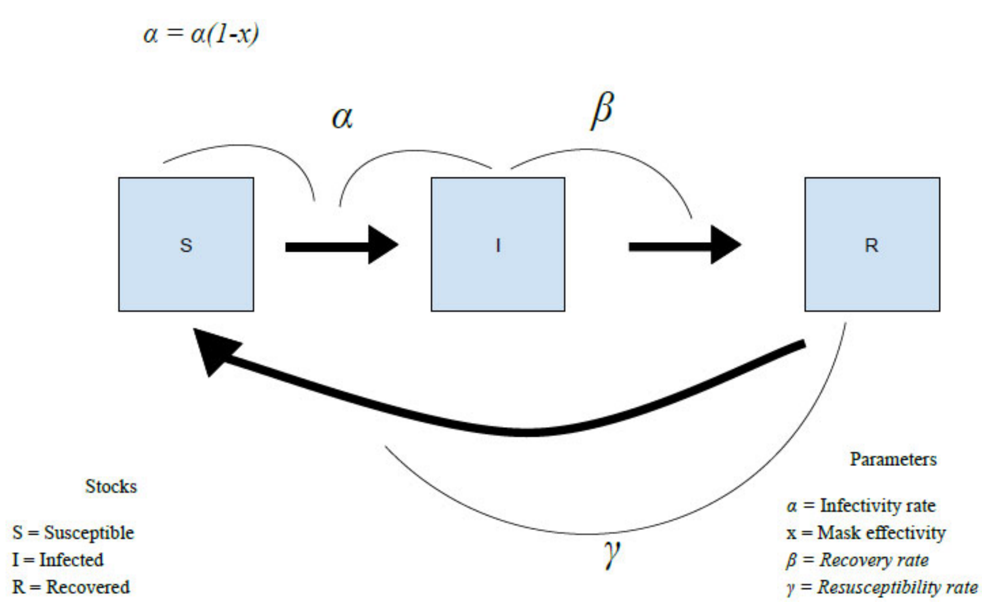
Plot results of the simulated model

```
figure()
plot(W, S, 'DisplayName', 'Susceptible'); hold on
plot(W, I, 'DisplayName', 'Infected')
plot(W, R, 'DisplayName', 'Recovered')
xlabel("Week")
ylabel("Persons")
title("SIR Model Without Re-infection")
legend()
hold off
```



Model 2: SIR Model with Re-infection

Stock and Flow Model (Model 2)



Update Equations (Model 2)

*How we computed updates to infections and recoveries newInfected = alpha * i * s

newRecovered = beta * i

newSusceptible = gamma * r

<u>*How we updated the state of each stock</u>

s_n = s - newInfected +newSusceptible

i_n = i + newInfected - newRecovered

r_n = r + newRecovered - newSusceptible

Model 2a (SIR Model with Re-infection)

Our initial state is again population of 100 persons: 99 susceptible, 1 infectious, 0 recovered

```
s_1 = 99;
i_1 = 1;
r_1 = 0;
```

Let's interpret t to be Weeks. Thus, we have the following parameter interpretations:

Infection rate: Each infected person has a chance to infect 1 susceptible person out of 100

Recovery rate: The infection lasts 4 weeks before recovery

Re-susceptibility rate: Each person recovered has a

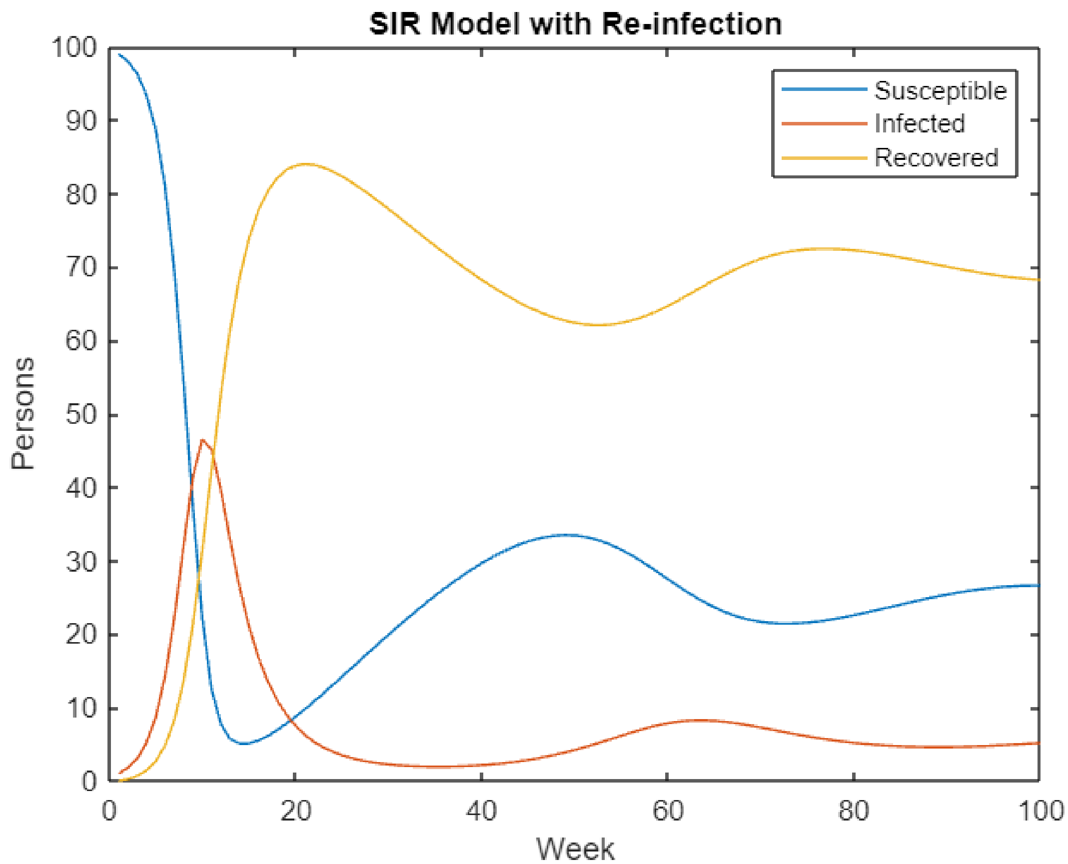
```
alpha1 = 1 / 100; %Infectivity rate
beta = 1 / 4; %Recovery rate
gamma = 1 / 50; %Re-susceptibility rate
```

Run simulation over 100 weeks

```
[S_sim, I_sim, R_sim, W_sim] = simulate_sir2(s_1, i_1, r_1, alpha1, beta, gamma, 100);
```

Plot results of simulated model

```
plot(W_sim, S_sim, 'DisplayName', 'Susceptible'); hold on
plot(W_sim, I_sim, 'DisplayName', 'Infected')
plot(W_sim, R_sim, 'DisplayName', 'Recovered')
xlabel("Week")
ylabel("Persons")
legend()
title("SIR Model with Re-infection")
```



Model 2b (Parameter Sweep for Mask Effectivness)

First, we must define our matrices for our x_all (Mask Effectiveness for each Simulation), alpha_metric (Infectivty Rate for each Simulation), and i_metric (Peak Number of Infected Persons for Each Simulation)

```
x_all = linspace(0,1,101);
alpha_metric = zeros(size(x_all));
i_metric = zeros(size(x_all));
```

Next, we will run a parameter sweep over our alpha_metric, which will run 101 simulations for each level of Mask Effectiveness

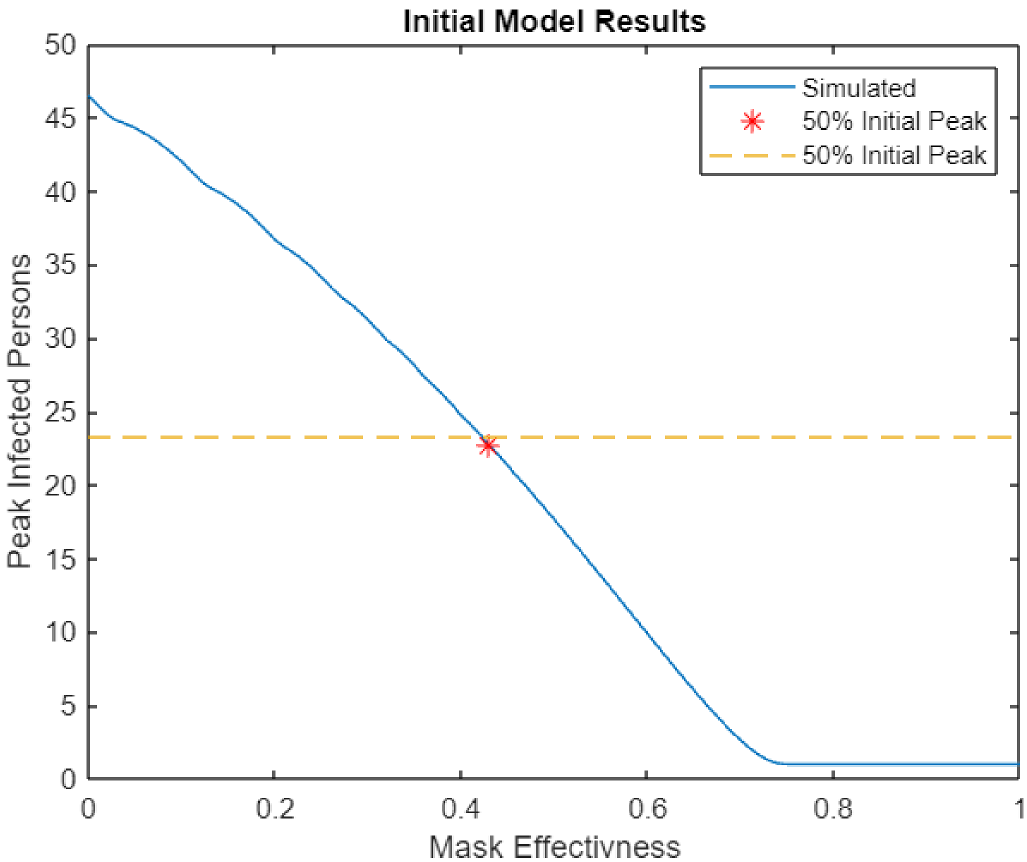
```
for i = 1:length(x_all)
    alpha_temp = alpha1 * (1 - x_all(i));
    [S_sim, I_sim, R_sim, W_sim] = simulate_sir2(s_1, i_1, r_1, alpha_temp, beta, gamma, 50);
    alpha_metric(i) = alpha_temp;
    i_metric(i) = max(I_sim);
end
```

Plot Results of parameter sweep using a graph of Mask Effectiveness v.s. Peak Infected

```
figure()
plot(x_all, i_metric, 'DisplayName', 'Simulated')
xlabel("Mask Effectivness")
ylabel("Peak Infected Persons")
legend()
title("Initial Model Results")
hold on;
```

In order to help answer our modeling question, we will also add a line representing 50% of the peak number of infected people when no masks are used, and a point representing the first effectivity of masks below that line.

```
g = max(i_metric)/2;
h = i_metric < g;
i = x_all(h);
j = i_metric(h);
k = j(1);
plot(i(1), k, 'r*', 'MarkerSize', 8, 'DisplayName', '50% Initial Peak');
plot([0,1], [g,g],'--', 'DisplayName', '50% Initial Peak');
```



Section III: Result

Metrics and Sweeps:

Our main metric measured the max amount of people who were infected in each simulation. We chose this metric as we needed to cut our peak number of infected persons by half to answer our initial modeling question. To do this we did a parameter sweep on the percent effectiveness of masks to find which effectiveness cut the peak infection by half.

Results Our Model Generated:

According to our model there is a negative correlation between peak number of infected persons and mask effectiveness. With our parameter sweep, as we used more effective masks, the peak number of infected persons decreased. **In order to decrease the peak number of infected persons by 50%, our masks needed to be at least 43% effective**.

Section IV: Interpretation

Original Question

How effective do masks need to be to cut the peak number of infected persons by 50%

ABT

Masks are an effective way to limit the spread of disease, but it is hard to determine how effective masks are in a pandemic setting. Using a parameter sweep we were able to determine that masks need to be 43% effective in order to cut the peak number of infected people by 50%. **However**, since not everyone in the real world wears masks, in order to truly cut the number of infected persons by 50%, masks would likely need to be more than 43% effective.

Limitations

Our model can't take into account how other public health interventions, such as vaccines, which would also impact infectivity and the peak number of infected persons. Since we are also assuming that everyone masks, our model doesn't accurately represent the true infectivity of the disease in the real world.

Next Steps in Model Development

In future developments of our models, we could implement the shortcomings of our current model, such as not everyone wearing masks. We could also attempt to model other forms of public health interventions. However, for this specific modeling question, we want to avoid making the model any more complicated than it needs to be.

```
function [s_n, i_n, r_n] = myaction_sir2(s,i,r,alpha,beta,gamma)
%Updates the amount of susceptible, infected, and recovered people based on
%the SIR model for each timestamp
%
%Usage
%[s_n, i_n, r_n] = myaction_sir2(s,i,r,alpha,beta,gamma)
%
%Inputs
% s = current number of susceptible people
% i = current number of infected people
% r = current number of recovered people
%
% alpha = infection rate of parameter
% beta = recovery rate of parameter
% gamma = re-susceptibility rate of parameter
% x = effectiness of masks
%
%Outputs
% s_n = the new amount of susceptible people
% i_n = the new amount of infected people
% r_n = the new amount of recovered people

%compute infections/recoveries
newInfected = alpha * s * i;
newRecovered = beta * i;
newSusceptible = gamma * r;

%compute new amount of susceptible, infected, and recovered people
s_n = s - newInfected + newSusceptible;
i_n = i + newInfected - newRecovered;
r_n = newRecovered + r - newSusceptible;

% Enforce invariants; necessary since we're doing a discrete approx
s_n = max(s_n,0);
i_n = max(i_n,0);
r_n = max(r_n,0);
%s_n = min(s_n,100);
%i_n = min(i_n,100);
%r_n = min(r_n,100);

end
```

```
function [S, I, R, W] = simulate_sir2(s_0, i_0, r_0, alpha, beta, gamma, num_steps)
% Simulate a SIR model
%
% Usage
%   [S, I, R, W] = fcn_simulate(s_0, i_0, r_0, beta, gamma, num_steps)
%
% Arguments
%   s_0 = initial number of susceptible individuals
%   i_0 = initial number of infected individuals
%   r_0 = initial number of recovered individuals
%
% alpha = infection rate of parameter
% beta = recovery rate of parameter
% gamma = re-susceptibility rate of parameter
% x = effectiness of masks
%
%   num_steps = number of simulation steps to simulate
%
% Returns
%   S = simulation history of susceptible individuals; vector
%   I = simulation history of infected individuals; vector
%   R = simulation history of recovered individuals; vector
%   W = simulation week; vector

% Setup
S = zeros(1, num_steps); S(1) = s_0;
I = zeros(1, num_steps); I(1) = i_0;
R = zeros(1, num_steps); R(1) = r_0;
W = 1 : num_steps;

% Run simulation
for step = 1 : (num_steps - 1)
    [S(step+1), I(step+1), R(step+1)] = myaction_sir2(S(step), I(step), R(step), ↙
alpha, beta, gamma);
end

end
```

```
function [s_n, i_n, r_n] = action_sir(s, i, r, beta, gamma)
% Advance an SIR model one timestep
%
% Usage
%   [s_n, i_n, r_n] = action_sir(s, i, r, beta, gamma)
%
% Arguments
%   s = current number of susceptible individuals
%   i = current number of infected individuals
%   r = current number of recovered individuals
%
%   beta = infection rate parameter
%   gamma = recovery rate paramter
%
% Returns
%   s_n = next number of susceptible individuals
%   i_n = next number of infected individuals
%   r_n = next number of recovered individuals

% compute new infections and recoveries
infected = beta * i * s;
recovered = gamma * i;

% Update state
s_n = s - infected;
i_n = i + infected - recovered;
r_n = r + recovered;

% Enforce invariants; necessary since we're doing a discrete approx.
s_n = max(s_n, 0);
i_n = max(i_n, 0);
r_n = max(r_n, 0);

end
```

```
function [S, I, R, W] = simulate_sir(s_0, i_0, r_0, beta, gamma, num_steps)
% Simulate a SIR model
%
% Usage
%   [S, I, R, W] = fcn_simulate(s_0, i_0, r_0, beta, gamma, num_steps)
%
% Arguments
%   s_0 = initial number of susceptible individuals
%   i_0 = initial number of infected individuals
%   r_0 = initial number of recovered individuals
%
%   beta = infection rate parameter
%   gamma = recovery rate paramter
%
%   num_steps = number of simulation steps to simulate
%
% Returns
%   S = simulation history of susceptible individuals; vector
%   I = simulation history of infected individuals; vector
%   R = simulation history of recovered individuals; vector
%   W = simulation week; vector

% Setup
S = zeros(1, num_steps); S(1) = s_0;
I = zeros(1, num_steps); I(1) = i_0;
R = zeros(1, num_steps); R(1) = r_0;
W = 1 : num_steps;

% Run simulation
for step = 1 : (num_steps - 1)
    [S(step+1), I(step+1), R(step+1)] = action_sir(S(step), I(step), R(step), beta, 
gamma);
end

end
```