Epidemiology – Human and Water Borne Illnesses

Marc Los Huertos

March 18, 2020

COVID-19 began spreading rapidly around the world in late 2019. And by the spring, Italy went into lock down, California declared a state of emergency, schools and universities around the globe suspended in person classes and events, and businesses have reduced travel and pushed tele-work policies. All of this is designed to slow the spread of the disease. These efforts are broadly referred to as social distancing.

The idea is to reduce person-to-person contact in order to make spreading the disease less likely. The effects of this are often illustrated in images such as those in the chart below, where the red plot is flattened to spread out the disease as much as possible. This helps to ensure that there are sufficient resources available for a sick population, which will help improve survival rates.

Flattening the curve to keep infection manageable (Source: Fast.ai). How do we determine the value of such distancing strategies and model this spread?

We walk through a SEIR epidemiological model and simulate it with R. The first model is the basic SEIR without social distancing, then we add social distancing to show how the potential effectiveness of these strategies.

The SEIR model is a compartmental model for modeling how a disease spreads through a population. It's an acronym for Susceptible, Exposed, Infected, Recovered. When a disease is introduced to a population, the people move from one of these classes (or compartments) to the next. When they reach the R state, theyre no longer able to be infected, depending on your interpretation, they either survived the disease and are now immune or succumbed to the illness and are out of the population.

This is an extension of the classic SIR model and simply adds one more equation to show those who are exposed. The full model is given below:

We have four ODE's in the time domain, with three parameters: α, β, γ.

- α is the inverse of the incubation period ($1/t_{incubation}$)
- β is the average contact rate in the population
- γ is the inverse of the mean infectious period ($1/t_{infectious}$)

Equation (1) is the change in people susceptible to the disease and is moderated by the number of infected people and their contact with the infected.

Equation (2) gives the people who have been exposed to the disease. It grows based on the contact rate and decreases based on the incubation period whereby people then become infected.

1 Models with Susceptibility and Infection

1.1 Modeling S and I with No Behavior Changes

Unfortunately, I have defined the array to track subject, thus, have to do some backflips to track overlap at each time stamp to see if people are sharing space and pathogens.

```
coord_range = c(0, 960, 0, 540)
N = 100
tstep = 2000
location = 5 # number of parameters to track
stationary = .4 # as a percent of N
speed = rep(5,N); speed[sample(N, round(N*stationary, 0))]=0; speed
Infect_distance = 5
(data_arr = array(dim=c(tstep,location, N)))
dimnames(data_arr)[[2]] <- c("x", "y", "theta", "speed", "status")

# Initialize Start Locations and Characteristics
subj_x = runif(N, coord_range[1],  coord_range[2]); subj_x
subj_y = runif(N, coord_range[3],  coord_range[4]); subj_y
theta = round(runif(N, 0, 360),0); theta

SERI = rep(1, N); SERI[sample(N, 2)] = 3; SERI

data_arr[1,,1]

#create function to move the subject
move_x = function(tstep, subj){
data_arr[tstep, 4, subj] * cos(data_arr[tstep,3, subj]*pi/180)}
move_y = function(tstep, subj){
data_arr[tstep, 4, subj] * sin(data_arr[tstep,3, subj]*pi/180)}

# check function
# move_y(1, 1); move_x(1,1)

# move function
move = function(j, i){
  c(data_arr[j-1, 1, i]+ move_x(j-1,i),
    data_arr[j-1, 2, i]+ move_y(j-1,i),
    data_arr[j-1, 3, i],
```

```
    data_arr[j-1, 4, i],
    data_arr[j-1, 5, i])
}
# check addressing
# data_arr[1,, 2]

for(i in 1:N){
  # Initial Locations
  data_arr[1,,i] = c(subj_x[i], subj_y[i], theta[i], speed[i], SERI[i])
  for(j in 2:tstep){
    # move subjects based on theta and speed
    data_arr[j,,i] = move(j, i)
    # coarse corrections when hitting a boundary
    # Min x-boundary
    if(data_arr[j,1,i] < coord_range[1]){
      data_arr[j-1,3,i]=180-data_arr[j-1,3,i]
      data_arr[j,,i] = move(j,i)
    }
    #Max x-boundary
    if(data_arr[j,1,i] > coord_range[2]){
      data_arr[j-1,3,i]=180-data_arr[j-1,3,i]
      data_arr[j,,i] = move(j,i)
    }
    #Min y-boundary
    if(data_arr[j,2,i] < coord_range[3]){
      data_arr[j-1,3,i]=360-data_arr[j-1,3,i]
      data_arr[j,,i] = move(j,i)
    }
    #Max y-boundary
    if(data_arr[j,2,i] > coord_range[4]){
      data_arr[j-1,3,i]=360-data_arr[j-1,3,i]
      data_arr[j,,i] = move(j,i)
    }
   # print(data_arr)
  }
  #print(data_arr)
}
```

1.2 Tracking the Disease Transfer

Test for proximity at each time step. I have developed an increbily ineffient way of doing this – I will have to some work for find a better method – here's where coding experience is useful!

```
# data_arr
euclid = array(data=NA, dim=c(tstep, 4, N, N))
for(i in 1:tstep){
for(j in 1:N){
for(k in 1:N){
euclid[i,1,j,k]=i; euclid[i,2,j,k]=j; euclid[i, 3, j,k]=k
euclid[i,4,j,k] = sqrt((data_arr[i,1,j]-data_arr[i, 1, k])^2+(data_arr[i,2,j]-data_arr[i, 2,
# Test Distance -- May not work as expected -- infections with stationary subjects can stil
if(euclid[i,4,j,k] < Infect_distance){
  if(data_arr[i,4,j] != 0){
    if(data_arr[i,5,k] == 3) data_arr[i:tstep,5,j]=3
    print(data_arr[i,,k])
  }
}

}}}

# I had to hard code in stationary - non-infections -- totally lame
#if(data_arr[,4,] == 0) data_arr[20,5,] <-1

# head(euclid)
```

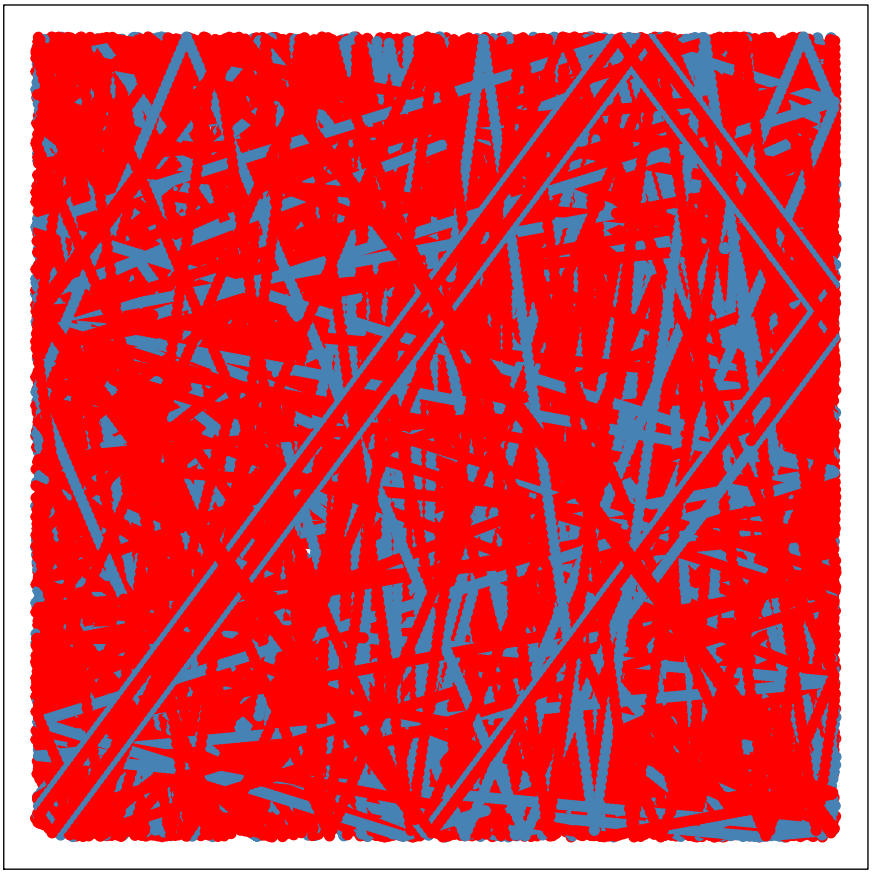
2 Plot Results

2.1 static plot

Because we are looking at a dynamic system, I don't find the static plot all that useful.

```
# data_arr
# plot results
par(mar=c(1,1,1,1))
SEIRcol = c("steelblue", "orange","red", "green")
plot(x=coord_range[1:2], y=coord_range[3:4], type='n', xlab='', ylab='', xaxt='n', yaxt='n')

for(i in 1:N){
  for(j in 1:tstep){
points(data_arr[j,1,i], data_arr[j,2,i], pch=19, col=SEIRcol[data_arr[j,5,i]])
  }
}
```



2.2 Animating the Results