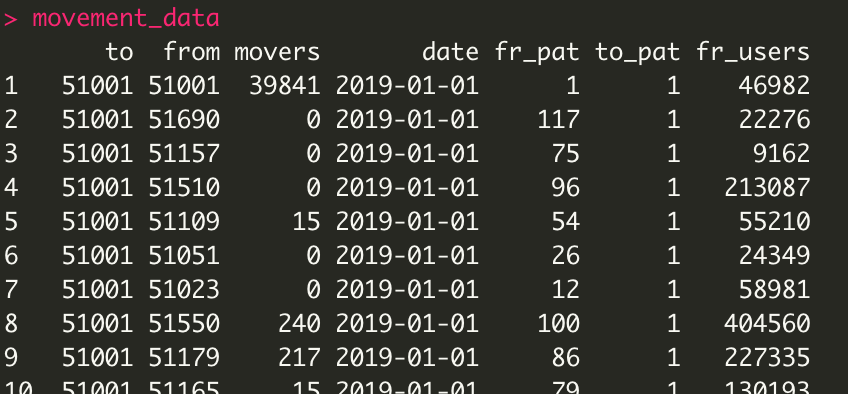
**Introduction to Spatial Models 2**

Today, we will be working with real data! The data for this assignment originate from a study in Virginia, in the USA, where we used cell phone data to quantify mobility of people between counties across the state. You will be mostly familiarizing yourself with this model, and then performing a change to see how it impacts the spread of disease.

The mobility patterns in this worksheet are from SafeGraph, from early 2020 (and so reflect some of the COVID-19 pandemic). Here, you can use the code to simulate the spread of a respiratory disease, and play around with the parameters.

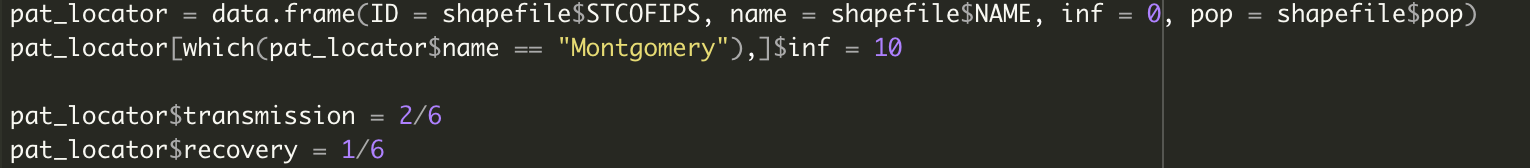
For today’s exercise, please look through the code and get a simulation to run, and look at the spread of disease after 50 days of simulation.

Roughly, this code first reads in a shapefile for Virginia that includes county boundaries and populations, and then a mobility data file similar to the one we worked with before:



Here, 39841 out of 46982 people in the county with ID 51001 stayed home (first row). Zero people moved from 51690 to 51001, out of 22276 people in county 51690.

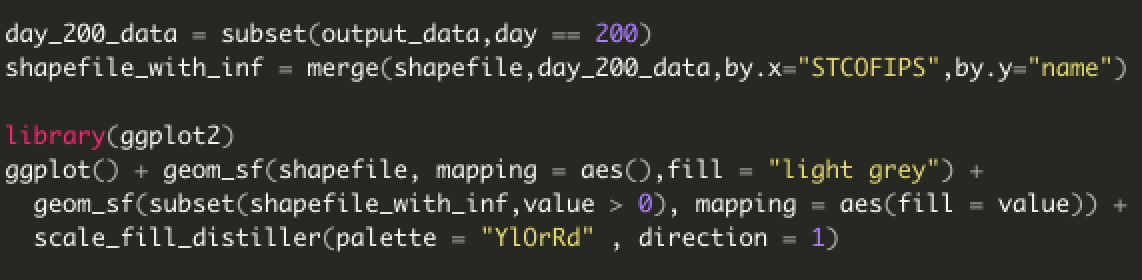
We then use this to create a data.frame that keeps track of our patches, the population size in each, IDs, the number of people initially infected, and the transmission/recovery rates. Later, you will modify this.



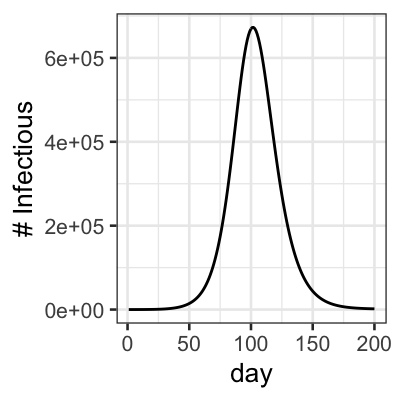
Here, we set the initial infected population to 0 everywhere except in Montgomery county, where we have 10 infected people.

Then, we initialize the population just as with the theoretical model, and run the model. As before, this creates two plots: An epidemic curve for each patch, then an overall epidemic curve for the entire state.

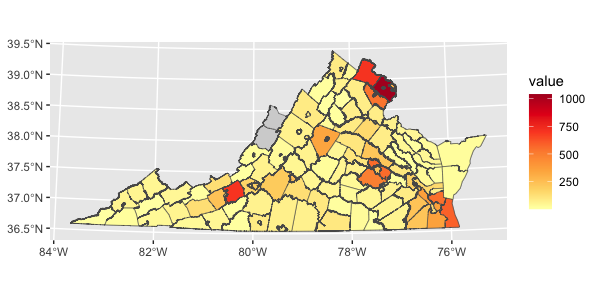
We also have some lines of code that subset the results from the 200th day of simulation, binds them to the shapefile, and then plots the shapefile:



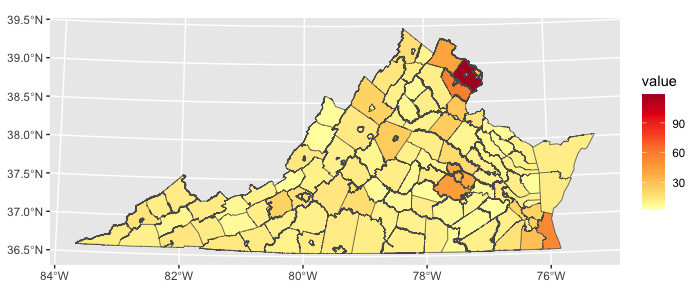
**TASK 1:** Run the model through this line and paste the figures showing the overall epidemic curve, and the map at 50 and 200 days, below (you will need to edit the code slightly to plot the 50th day’s data):



50 days:



200 days:

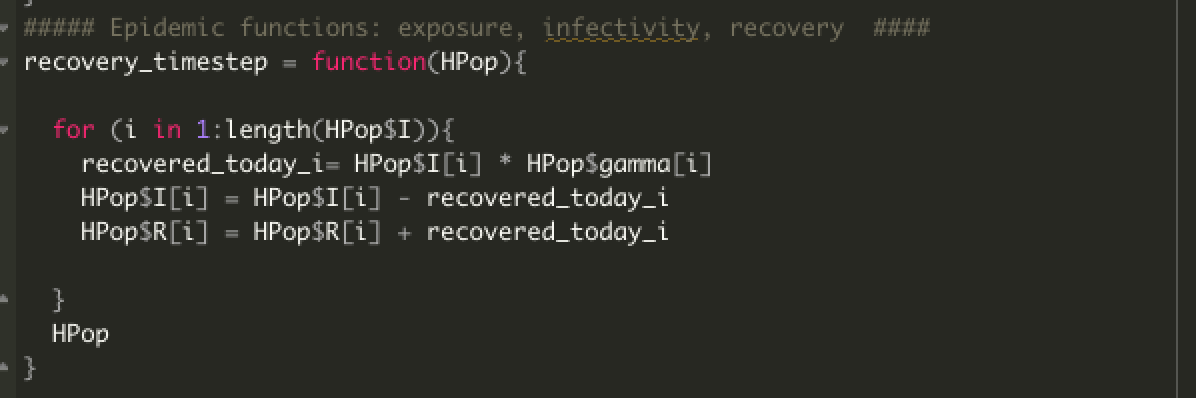


Now, let’s make a small change to the model. Right now, the model features lifelong immunity, where people move into the Recovered class and never leave. Here are the equations this model approximates:

**TASK 2:** If we wanted to include a loss of immunity, indicating that people are immune for 5 days on average (time to loss of immunity: 5 days), what would be the corresponding **rate** of losing immunity?

.2 (1/5)

Let’s implement this change, using the rate of losing immunity of .2. We can do this by opening the modelfx.R file, and adding some code to the recovery\_timestep() function, which governs what happens when people get over their infection:



Right now, people who recover at a rate gamma move into the Recovered class, and nothing else happens to them. This function works by calculating the number of people recovering in each patch i, as recovered\_today\_i= HPop$I[i] \* HPop$gamma[i]. We will add a step to this, which will first cause some recovered to lose their immunity and return to Susceptible, at a rate of .2. We will do this by adding a second “for” loop, which does this step before the addition of new recovered people:

for (i in 1:length(HPop$R)){

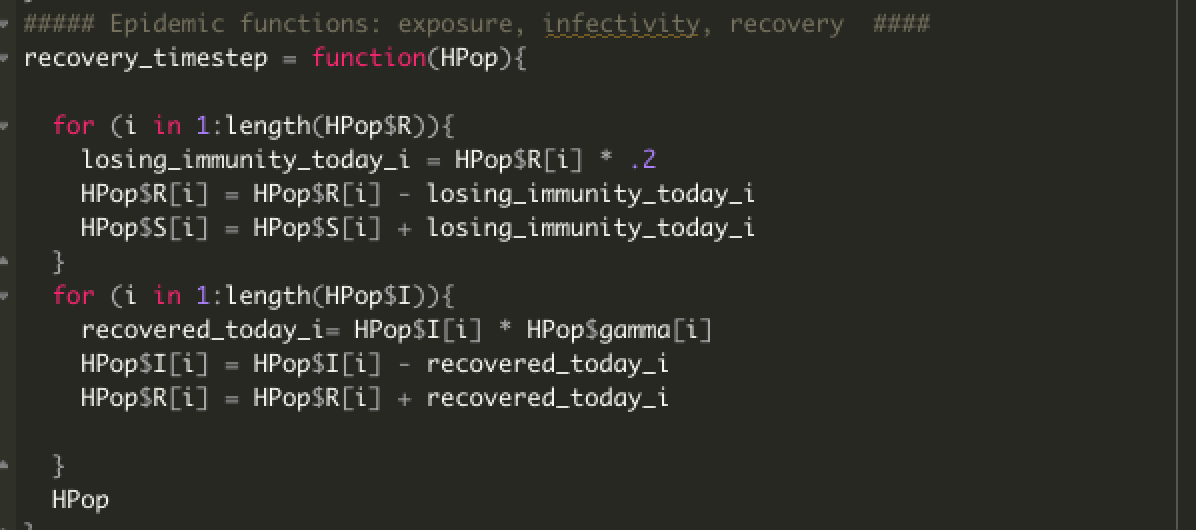
losing\_immunity\_today\_i = HPop$R[i] \* .2

HPop$R[i] = HPop$R[i] - losing\_immunity\_today\_i

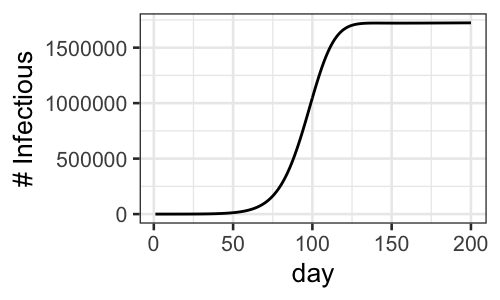
HPop$S[i] = HPop$S[i] + losing\_immunity\_today\_i

}

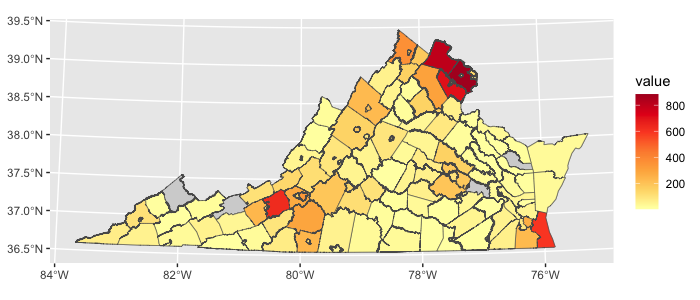
At the end of this, your function should look like this:



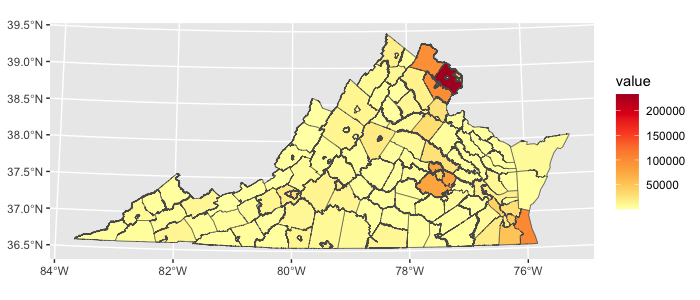
**TASK 3:** Implement this change, and rerun your simulation, plotting the output similarly for the entire epidemic curve, then for 50 and 200 days as before:



Day 50:



Day 200:



How is this different?

The outbreak didn’t die out, and the number of people infected at 200 days was therefore much larger.