VA mobility modeling

Using a model

- Introduce model, data, parameters
- Run baseline simulation and create epidemic curves
- Modify transmission/recovery
- Modify mobility

Using a Model

- The model we are working with is a directly-transmitted SIRS model
  - Eulerian movement
  - Many respiratory diseases, including COVID-19, influenza

β_i: $patch\ specific\ transmission\ rate$
γ_i: patch specific recovery rate
ε: $loss\ of\ immunity\ rate$
m_{ji}: $movement\ rate\ from\ i\ to\ j$

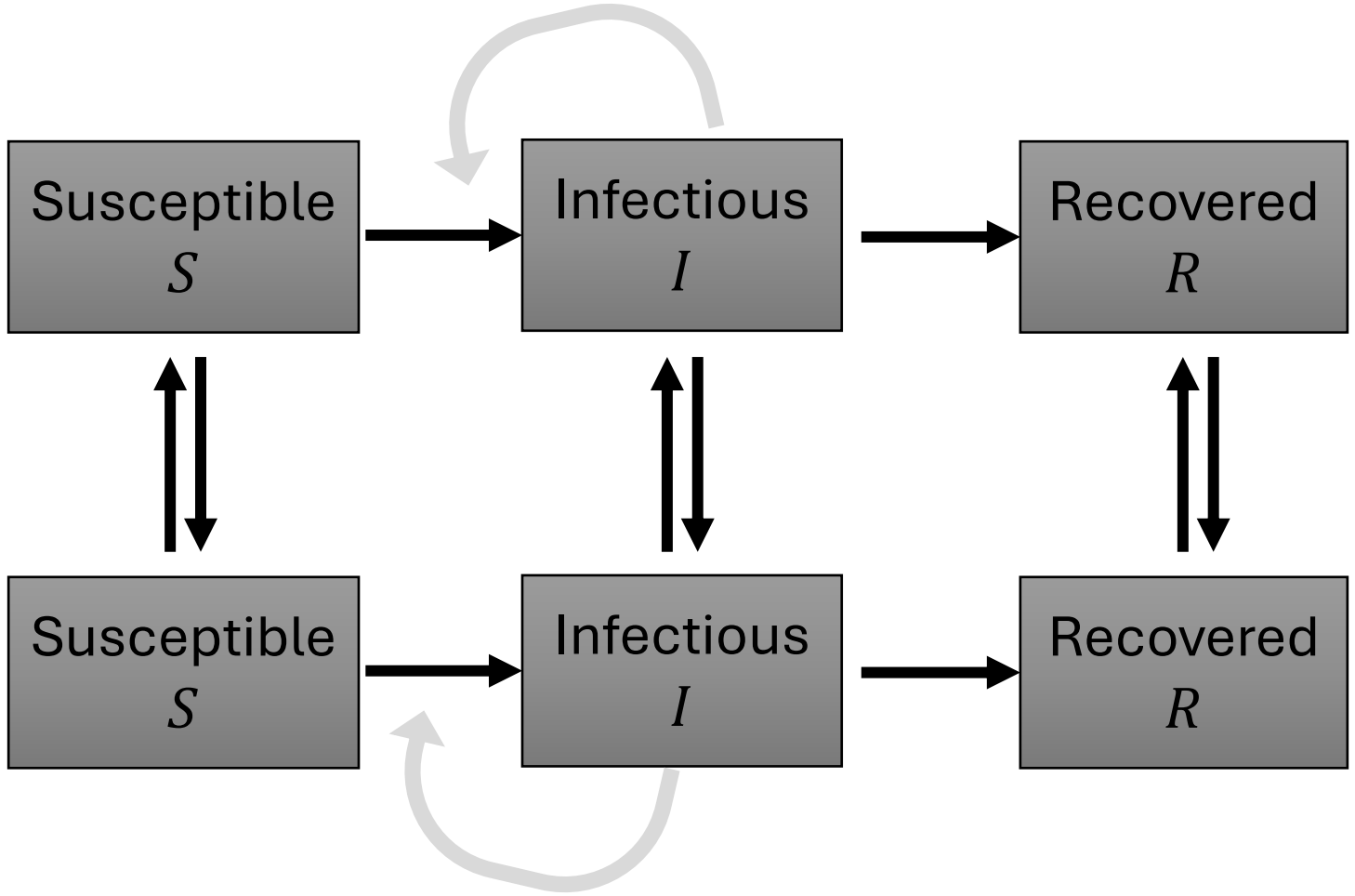
$$S_i(t+1) = S_i(t) - \beta_i \frac{S_i(t)I_i(t)}{N_i} + \varepsilon R_i(t) + \sum_j m_{ij}\, S_j(t) - \sum_j m_{ji}\, S_i(t)$$

$$I_i(t+1) = I_i(t) + \beta_i \frac{S_i(t)I_i(t)}{N_i} - \gamma_i I_i(t) + \sum_j m_{ij}\, I_j(t) - \sum_j m_{ji}\, I_i(t)$$

$$R_i(t+1) = \gamma_i I_i(t) - \varepsilon R_i(t) + \sum_j m_{ij}\, R_j(t) - \sum_j m_{ji}\, R_i(t)$$

Steps to the model

- Initialize a human population
  - How many patches? How many people in each patch? Who is initially infected?
- Use mobility data to create adjacency matrix
- Begin simulation
  - Infection/recovery are separate steps from movement
  - Within-patch transmission, then between-patch movement
- Human population object changes; use this to plot epidemic curve etc



Step 1: Run baseline

- First, create human population object
  - Define landscape, named “pat_locator”
  - InitiatePop() is then used to create human population object
- Use mobility data to create adjacency matrix
- Then, run the baseline simulation
  - runSim()
- Then, use some of the plotting functions on the final human population object
  - Plot_total_curve()
  - Plot_patch_curves()
  - Plot_map_day()

Working with the data

- Data is provided as number of people traveling from one place to another, out of a total population
- Use this to create a matrix
- Can create off-diagonal movement rates
  - This may be too high (people may take multiple trips, getting double counted); can use a multiplier to reduce
- Then, we fill in the diagonal

origin	dest.	movers	users
Montgomery	Roanoke	42	1200
Montgomery	Fairfax	12	1200
Roanoke	Montgomery	55	1800
Fairfax	Montgomery	8	3600

$$M[i,j] = movers_{ij} / users_i$$

$$M[i,i] = 1 - sum(M[i,-i])$$

```
movement_matrix <- matrix(0, n_patches, n_patches)
movement_matrix[cbind(fr_pat, to_pat)] <- movers / fr_users
movement_matrix <- movement_matrix * 0.1 * movement_multiplier
diag(movement_matrix) <- 1 - rowSums(movement_matrix)
```

Option 1: Start changing epidemiological parameters

- First, we will change factors across the entire landscape (global parameters)
  - What happens if the transmission rate of the disease changes?

```
# Since gamma = 1/6, beta/gamma is
approximately R0.
# Baseline: beta/gamma = 2

lower_R0 <- simulate_scenario(
  beta = 1.2/6,
  gamma = 1/6,
  days = 120
)

plot_total_curve(lower_R0,
  "Lower transmission: R0 approximately
1.2")
```

Option 2: Modify mobility

- What happens if overall movement rates change?
- Two scenarios: no mobility at all, and then higher than original levels of mobility

```
no_mobility <-
simulate_scenario(movement_multiplier =
0, days = 120)
high_mobility <-
simulate_scenario(movement_multiplier =
0.25, days = 120)

compare_mobility <- rbind(
  data.frame(no_mobility$epidemic_curve,
scenario = "No mobility"),
  data.frame(baseline$epidemic_curve,
scenario = "Baseline mobility"),

data.frame(high_mobility$epidemic_curve,
scenario = "High mobility")
)

ggplot(compare_mobility, aes(day, inf,
linetype = scenario)) +
  geom_line(linewidth = 1) +
theme_bw(base_size = 16)
```

Option 3: Spatial targeting

- Reduce transmission in certain counties

```
target_source <- simulate_scenario(
  seed_county = "Montgomery",
  intervention_counties = "Montgomery",
  beta_reduction = 0.5,
  days = 120
)

compare_intervention <- rbind(
  data.frame(baseline$epidemic_curve,
scenario = "No intervention"),
  data.frame(target_source$epidemic_curve,
scenario = "Reduce beta in source county")
)

ggplot(compare_intervention, aes(day, inf,
linetype = scenario)) +
  geom_line(linewidth = 1) +
theme_bw(base_size = 16)
```