Lydia Teinfalt, 2/11/2025, HW3: Build a Schelling Segregation from Scratch
specs: Python 3x on Google Collab and Spyder

Output of Schelling Segregation Model

Input parameters

- grid_size = 10 (2D grid is 10x10)
- 80% of grid initialized with an agent
- each cell holds one agent
- von Neumann neighborhood
- threshold = 62.5% of like-neighbor
- number of runs = 10
- max number of moves an agent to find a spot to be happy = 10,000. This constraint is necessary as it is the only way to stop the simulation since I did not have a graphical user interface that allows a user to stop the simulation runs.

Sample output of single run

```
Initial grid for run 10:
O X X . O O O . X X
X . O O X O X . . O
X . . O O O X X X .
O X O O X O O X . O
O O O X . . O O O X
. X O O O X X O X X
O X O O O O . O O O
O X . O . O X X . .
O O O O . X . O O X
X O . X X X O O O O

Final grid for run 10:
X X X X X X X X X X
X X X X X X X X X X
X X X X X X X X X X
O O O O O O O O O O
O O O O O O O O O O
O O O O O O O O O O
O O O O O O O O O O
O O O X O O O O O O
. . . . . . . . . .
. . . . . . . . . .

Run 10 reached the maximum move limit of 10000.
Average like neighbors: 0.86
Model execution time (HH:MM:SS) is: 0:00:02.104953
```

Depending on the threshold, the max total number of moves is reached for all agents to be happy. The lower the threshold, generally, the lower the number of times an agent needs to move to be happy. Below is the result where the threshold is 67.5%.

```
Simulation results over 10 runs:
Run 1: Total moves = 10000
Run 2: Total moves = 10000
Run 3: Total moves = 10000
Run 4: Total moves = 10000
Run 5: Total moves = 10000
Run 6: Total moves = 310
Run 7: Total moves = 10000
Run 8: Total moves = 10000
Run 9: Total moves = 10000
Run 10: Total moves = 10000
```

With the same input parameters, but varying thresholds, I had similar results as displayed in Chapter 2. With a larger grid, my results should mat

Preferences for like neighbors	Final average # of like neighbors	Increase of average over preferences	Fraction of possible increase realized
25%	51%	105.20%	71.29%
37.5%	73%	93.87%	66.58%
50%	72%	44.00%	113.64%
62.5%	83%	33.44%	112.14%
75%	85%	12.93%	193.35%

Table 1: 10x10 grid, 80% of grids have agent, limit on agents moving no more than 10,000

Increasing grid-size to 25, the time to run takes longer than 10 grid size.

```
Final grid for run 8:
O O X X X X O O O O O X X X X X X O O X O X O O
O O X X X X X O O O O O O O O X X O O O X X X X X
O X . X X X X O O X O O O O O O O O O X X X X X O
O X X X O O O X X X O O O X O O O O O O X X O O X O
O X X X X X X X X X O X O X X X X O X O X O O X O
O X O O X X X O O X O X O O O X X O X O O X O O O
O O X O X X O O X X O X X X O X X X O X O X X O O
O X X X X X X X X O O O O X O X O X O X O O X O O
X X O O X X X X X O O X O O X X O O O X X O O . O
X X . O O X . X . X . X X . X X X . . O X . X X O
X . O O . X . X X X O O . . X O O . O O X O O . O
X X O O O X O O O X X X . . . O O O . O X . O O O
X X X X O O O O O O O X O O . . X X X X X O O . . O
O O . X . O . O O . O O . . O X . . O X O X X . X
X X X O O . . . O X X . . . O . X . O O X X O O X
O O . X O O X . O X X . X O O X X O . X X O O X X
. O X X O X X X X X X . X . O X O O O O X . . X .
X X X X X O X X X . X X . O O O . O O O O O X O .
. . . X . O O . X X O . . X X O O O . . . O X O .
. O O X . X X O O . O . . . . . . . X X . . . O O
. . . O . . X . O . O O . . . X X X O O . . X O O
. X X O O X X X . O X O X . . O O O X X O X X X .
. . X O O . O . O O X X X . . . O O X . O . . O O
O X X . O O O X O . . X . X X . O O X O . . . O .
O O X O O . . X X . . O X . X X X X X O O O . . .

Average like neighbors: 0.52
Model execution time (HH:MM:SS) is: 0:00:00.046408
```

Preferences for like neighbors	Final average # of like neighbors	Increase of average over preferences	Fraction of possible increase realized
25%	54%	117.78%	63.68%
37.5%	77%	105.33%	59.22%
50%	76%	52.20%	95.79%
62.5%	93%	48.64%	77.10%
75%	91%	21.60%	115.74%

Table 2: 25x25 grid, 80% of grids have agent, limit on agents moving no more than 10,000

Randomly selecting an open spot strategy

When switching from agents choosing a spot to move to where they would be happy and randomly choose an available spot. Preliminary analysis I find that results were very similar to the previous runs where the agent chooses a happy spot to move to.

Repository

https://github.com/lydiateinfalt/CSS610-AgentBasedModelingSimulation-Spring2025/blob/main/SchellingSegregation_ABM.py