Agent-Based Modeling: Week 2

Winter 2023

Jean Clipperton

Agenda:

- ODD + D framework
- Design choices -- what does it mean to have different agents / actors
- Schelling

Grimm and Volker: ODD + D

- Puzzle / question
- Theory / reasoning
- Framework
- Do I buy it / so what?

GV: Puzzle

Paper seeks to explore how we can best standardize design.

GV: Theory / Reasoning

Without some kind of framework, it's tough to make sense of models. For example, it's challenging to compare / contrast different models and it can make any sort of broader / collaborative conversation really challenging.

GV: ODD + D Framework

Overview	Purpose
	State variables and scales
	Process overview and scheduling
Design Concepts	Design concepts
Details	Implementation details
	Initialization
	Input
	Submodels

GV: ODD + D Overview

- **Purpose** Who is the model for? (e.g. scientists, students / teachers, stakeholders, decision makers?)
- **Entities, state variables, and scales**: What kinds of entties are in the model, what state variables an dparameters do they have?
- **Process overview and scheduling** Describe scheduling, names for the processes, how the update process works.

GV: Design Concepts

Here, give an overview of all the pieces that come together to create the model. For example, you will want to discuss What the agents are, how they interact with their environments, any learning they are able to (particularly w/r/t their environment), and what random processes exist within the model.

GV: Details

This is where you provide the technical details of the model. Think of it as the actual recipe: **can someone read this component and recreate the model themselves?** *If the answer is no, you need to provide addtional detail*

GV: Details example

Consider

- Agents on a grid interact with one another in each step.
- Agents are placed at random on a grid and choose to stay or move based on their immediate neighbors (8 surrounding cells).
    - Note that we still haven't included the activation scheme!

Discussion

What do we think / what is the value of Grimm and Volker?

Schelling

Incredibly famous; broad implications and applications

- First, cover the scenario
- Next, going to use this as an opportunity to think about good design
- Finally, adjust code to accommodate expansion

Example: Schelling's segregation model

- The model seeks to **explore why we might see stark segregation**, even if focusing *solely* on actor behavior / motivation between two groups, and assuming somewhat benign intentions (no explicit dislike / hatred).
- Schelling's model is** *one piece* in exploring and understanding patterns** we see in housing markets and neighborhood patterns.
- Rothstein's Color of Law is great additional reading.

Reminder: ABM setup and components:

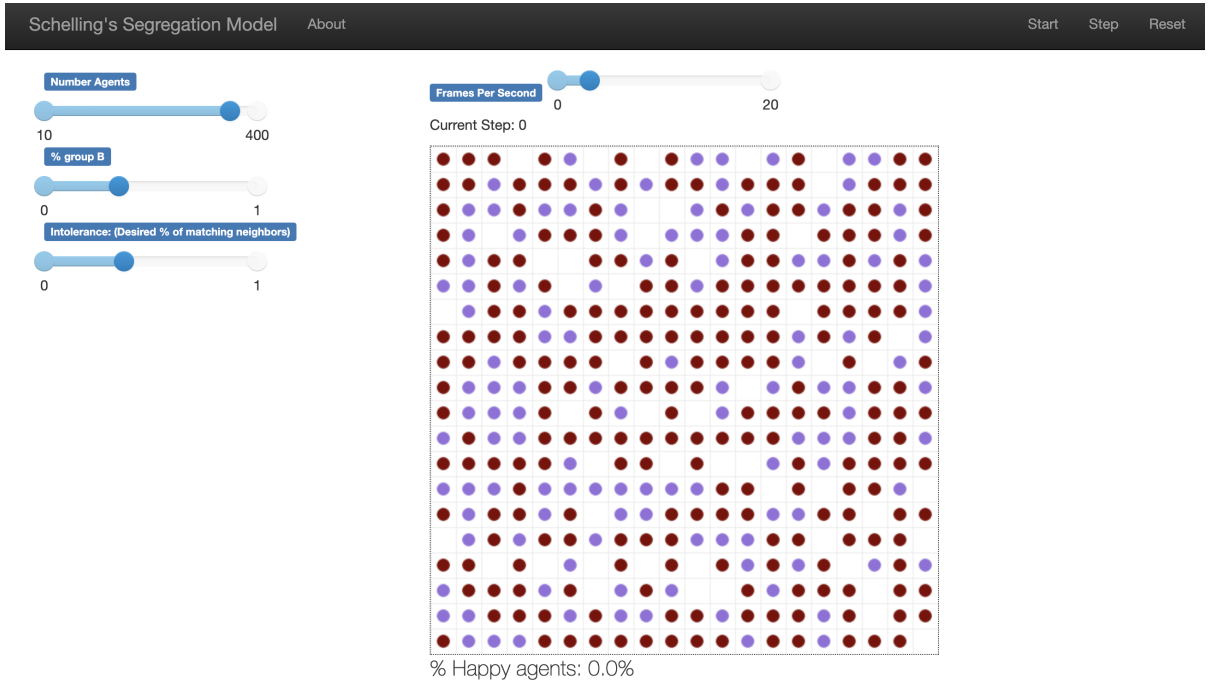
Who does what, how?

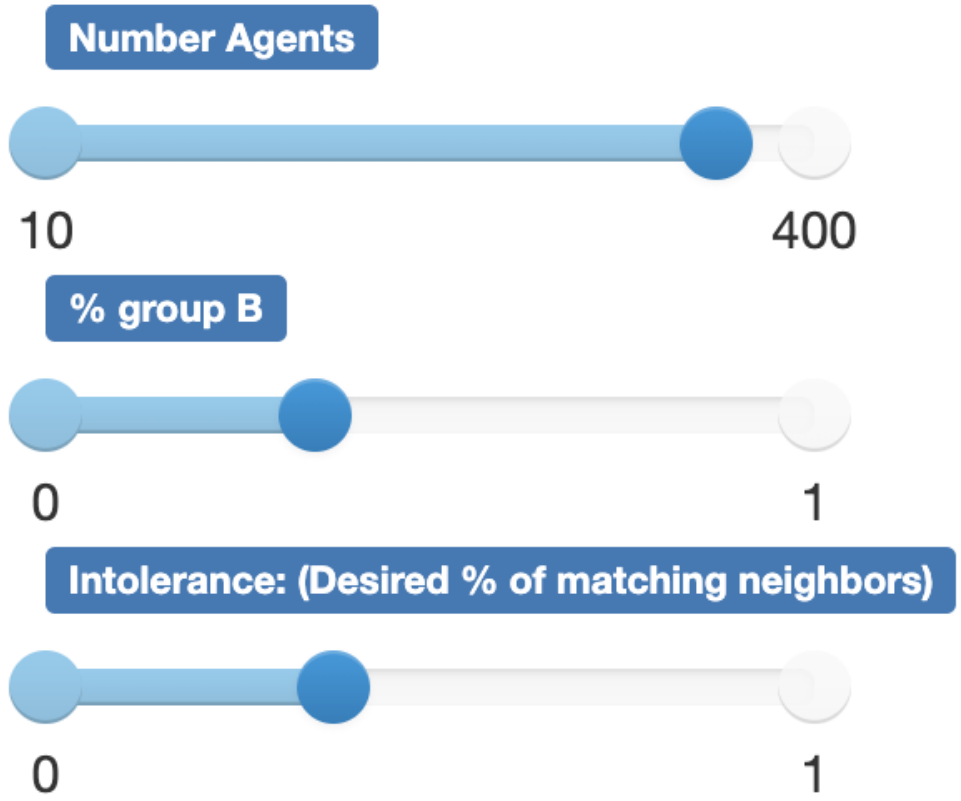
- **Actors** these can be people, animals, businesses, etc. The *units* you care about.
- **Parameters** (AKA *variables*) these are the things you're going to vary.
- **Framework** structure for interaction.
  -Structure ecosystem / environment for actor.
    - Define movement / behavior (where are they placed, how they interact, move, etc.)

Example: ABM setup (Schelling)

- **Actors** (what are the units / agents /actors)
  - Need two types of actors (why two?)
- **Parameters** (AKA *variables*) (what are the possible values/items that can be incorporated)
  - Agent type.
  - Agent tolerance threshold.
- **Framework** structure for interaction (in what way are actors aggregating information from their environment)
  - Start with random placement.
  - Neighborhood / area to scan.
  - Evaluate their satisfaction.
  - Move somewhere if unsatisfied.

Schelling's model: preview





.

- **Num Agents**: how many agents.
- **% Group B**: the percentage of total agents in Group B.
- **Intolerance**: proportion of neighbors each agent wants to have in their

Schelling's model: preview

```
knitr::include_graphics("images/schell1_overview_start.png")
```

Schelling: So what?

With your group mates, try varying the parameters and noticing how the pieces fit together.

- What do we observe from how the model runs?
- How does varying the paramters affect the outcomes?
- What contributions does the model make to the literature?

Questions / Comments / Concerns?

Designing the model: pseudocode for model (agent class)

```
import Mesa

class (agent)

  initialize agent
    (details for agent)

    define step
        (details for step for agent)

    define any other agent actions
        (moving, etc.)
```

Untangling code

Notice that this model has TWO types of agents. In our code, we have two files, schelling_basic and schelling_complex. Let's take some time to look over these files and see what we think about them.

Two agent types: the mechanics of design

How do we think about when / how to split our agent types? This might seem like a pedantic question, but if we care about having a good and well-designed model, this is something that matters and that we'll want to get right the first time.

Groups: Making a better model

In groups, **DESIGN** (do not yet implement) a schelling model with THREE groups and add some complication.

Some ideas:

- Different thresholds
- Different definition of neighbors
- Adjusting the step function
- Something else???

A brief interlude: getting started on visualizing

Basics (more to come!!)

We're going to cover the bare foundation to get us started and then next week, we'll go into greater depth.

Adding vizualizer tools / bars

This model has more sophisiticated visualization and tracking than we've had in the past. Next week, we'll go into this in greater depth.

For now, we're going to cover the basics so you can add in something for the additional agent types you're creating.

Sliders, tools, and bars

Back to groups: IMPLEMENT your plan

Create an extension of the model that addresses one of the criteria below:

- Different thresholds
- Different definition of neighbors
- Adjusting the step function
- Something else???

Pulling it all together

Do a short one-page writeup of your model, addressing the ODD + D protocol