Reward signal for ABM

Previous work

Reward Signal

TODO:
- Code in this reward signal for each agent in `agent_step!()`
- If $r > 0$: agent stays
- If $r \leq 0$: agent leaves

$$r = PR + NR + ER$$

Place Reward

$$PR = w_1$$

Reward for place attachment

Neighbor Reward

$$NR = w_2 \cdot \left(1 - \frac{n_t}{n_0}\right)$$

Reward for neighborhood; e.g., don't want to live in abandoned neighborhood

n_t: Number of neighbors at time t
n_0: Number of neighbors at start of simulation

Exposure Reward
(or penalty)

$$\mathrm{E}R = -w_3 \cdot p_e$$

Negative reward for exposure

Weights:
- $w_1 = 0.1$
- $w_2 = 0.05$
- $w_3 = 1.0$

Notes:
- n_t is accessible through `agent.state`
- n_0 is accessible through `agent.neighborhood_original`
- p_e is the percent of year exposed that you used before

Utility theory for ABM

Utility theory

Agents make decisions based on “utility”

Example from Wikipedia (https://en.wikipedia.org/wiki/Utility)

- Agent has utility function $U = \sqrt{xy}$ where x is number of apples and y is number of chocolates.
- Agent is given two options:
  - A: 9 apples and 16 chocolates
  - B: 13 apples and 13 chocolates
- $U_A = \sqrt{xy} = \sqrt{9 \cdot 16} = 12$
- $U_B = \sqrt{xy} = \sqrt{13 \cdot 13} = 13$
- Agent decides option B, 13 apples and 13 chocolates

Want to do the same for migration ABM
For example, each Agent has two options

- A: do nothing (e.g., stay)
- B: leave

Cobb-Douglas Utility Function - General

$$U_a = \prod_{i=1}^{n} P_i^{\alpha_i} = \left(P_1^{\alpha_1}\right) \cdot \left(P_2^{\alpha_2}\right) \cdots P_n^{\alpha_n}$$

U_a Utility if taking action a

P_i Some feature/variable, e.g., percent of year exposed

α_i Weights the importance of the above feature to the agent; between 0 (less important) and 1(more important)

Cobb-Douglas Utility Function - Applied to Galveston

$$U_{stay} = P_{place}^{\alpha_1} \cdot P_{neighbor}^{\alpha_2}$$

$$U_{leave} = P_{exposure}^{\alpha_3}$$

Features/variables to consider

$P_{place} = 100$	Set this to 100, the weighting of this will be done through α_1
$P_{neighbor} = \frac{N_t}{N_0} \cdot 100$	Same ratio as before (number of agents at time t to number of neighbors at time 0); multiply by 100
$P_{exposure} = p_e \cdot 100$	Same percent of year exposed as before, but $\times 100$

Cobb-Douglas Utility Function - Applied to Galveston

$$U_{stay} = P_{place}^{\alpha_1} \cdot P_{neighbor}^{\alpha_2}$$

$$U_{leave} = P_{exposure}^{\alpha_3}$$

Dylan Added `alpha_calc()` function to `misc_funcs.jl`

Use this to determine alpha values for each agent

Nat: Next Steps

Update the `ResidentialAgent` struct to carry α values
Add α values to all agents in function `add_agents!()`
- Each agent should have different α values

In `agent_step!()`
- Compute U_{stay} and U_{leave}
- If $U_{stay} > U_{leave}$ then agent stays
- If $U_{leave} > U_{stay}$ then agent leaves

Extra:
- In input.csv, there is a variable called n_iterations; change this to 100 to run the model 100 times
    - This is called Monte-Carlo simulation (https://en.wikipedia.org/wiki/Monte_Carlo_method)
    - It's used to understand uncertainty

I won't be available for meetings June 21-25; however, you can text me. I'll respond when I can.