8/31/21

# Assignment 1

## Please use this Word file template, follow (and retain) the instructions in gray text, and insert your work in black where indicated. Keep in mind the evaluation matrix at the end as you do the work and use it to guide what you submit. Use no more than 4 pages of 12-point text excluding figures, the gray instructions, and appendices. You can add as many as many appendices as you like. These will be read on an as-needed basis.

## The Given Problem

Go [here](https://mesa.readthedocs.io/en/master/tutorials/intro_tutorial.html#sample-model-description) to the Mesa tutorial and follow the creation of the “wealth-distribution” agents[[1]](#footnote-1). Show that you have set up the MoneyAgent and MoneyModel classes, as well as the RandomActivation scheduler, and that you can run it with every agent outputting its ID and current wealth. For this part, no explanation is required.

**Screen shot of output for #1:**

Text

Description automatically generated

## Reproducing the Model

Show, by means of a screenshot of your output, that you have successfully run the model up to the point in the tutorial that says “This outcome might be surprising. Despite the fact that all agents, on average, give and receive one unit of money every step, the model converges to a state where most agents have a small amount of money and a small number have a lot of money.” Explain why you think this is the outcome.

**Screen shot of output for #2**: Chart, histogram

Description automatically generated

**Explanation of the imbalanced outcome**:

The imbalanced outcome arises because after each round every agent always gives away one unit of wealth (if they have any to spare) and the odds that an agent receives a unit of wealth are 1 in 10 and they get N attempts at this per round, where N is the number of agents who have wealth > 0. This means that the chances an agent receives wealth decreases every round because there are more agents worth 0 each round. Therefore, as each round progresses less and less wealth is transferred from those who acquired it earlier, so the wealth begins to concentrate into fewer and fewer hands since there is never any new wealth generated: it is a zero sum game.

## A Fairer Model

Show the results of giving away a *proportion* of an agent’s wealth. You can parameterize this. Show this via the key code, as well as screenshots of your output for various parameter values. Explain.

**Key code**:

Text

Description automatically generated

Key Code Explanation: When creating the model, the user now specifies the percentage, as an integer (ex: 50 for 50%), of the agent’s wealth that they will give away each step. This becomes an attribute for the agent and is used in the agent’s step function instead of simply using 1. The outputs now show that the smaller proportion of wealth lessens the number of agents who end up poorer, more evenly distributing the wealth. It centers the data around 1 unit of wealth as well as decreases the maximum amount of wealth the wealthiest agent has (see; 10%, everyone is below 2.0 units). This is because the smaller the proportion, the less wealth is transferred over each round.

**Screen shot of output for #3**:

Giving away 50% of their wealth:

Chart, histogram

Description automatically generated

20% of their wealth:

Chart, histogram

Description automatically generated

10% of their wealth:

Chart, histogram

Description automatically generated

## Exploration

This part of the assignment is open-ended. We want you to explore 1-3 variations on the model that you come up with (i.e., not in the tutorial or known to you as published). For example, the effects of changing the scheduler, or of a measure of “happiness” of agents that affects their giving habits. Explain.

(1) **Description of variation 1**: Changed the scheduler from RandomActivation to BaseScheduler.

**Effect** **of variation 1**: The did not change the output over a large scale. This because all that RandomActivation does is complete the step() call in random order, whereas the BaseScheduler simply does it in the order agents were added to it.

(2) **Description** **of variation 2**: Changed proportion of giving if the agent has less than 1 unit of wealth or greater than 2 (Halved and doubled, respectively).

**Effect** **of variation 2**: This drastically changed the data, placing most of the agents (~85%) having wealth less than 1. The rest had a much higher amount of wealth, the most being 10 units of wealth. There were no detectable agents with wealth greater than one and less than seven! So the wealth became hyper-concentrated.

(3) **Description** **of variation 3**: Randomized the proportion given each time step() is called.

**Effect** **of variation 3**: This essentially undid what Q2 of the lab did. The vast majority of agents had almost no wealth and it was an exponential decrease from 0 to 7 or 8 depending on the iteration. The only difference between this variation and the first step is that no agent is worth 0 and each agent has wealth stored as a float, not an integer.

# Evaluation



1. There are some number of agents. All agents begin with 1 unit of money. At every step of the model, an agent gives 1 unit of money (if they have it) to some other agent. [↑](#footnote-ref-1)