**Instructor Guide: Emergent Income Inequality using Agent-Based Modeling**

**Goals for the assignment:**

1. Create a simple economic system of interacting agents.
2. Study the trends that emerge from this kind of system with different rules and initial conditions.
3. Plot and visualize these trends in a variety of ways.
4. Learn how to use two key metrics of income inequality, the Lorenz curve and the GINI coefficient, to explain the distribution of wealth within economic systems.

**What the student should be able to do after completing the assignment(*Enduring understanding*):**

1. After completing the assignment the students should be able to explain what ABM is and create different models of agents interacting with differing degrees of complexness.
2. The students should be able to code a standard “yard sale” economic model with ABM and further develop its complexity into a model which reflects a simplified version of a real world economic system. The tasks will also make the student capable of describing the emergent results from the models as well as give an explanation as to why this behaviour happens.
3. The students should be able to reflect upon the results that they achieve, IE write about the socioeconomics of the systems and perhaps the morality of it.
4. The students should be able to understand what emergence is and how it surplants itself in the yard sale model which they have coded.

This task was created in a manner so that hopefully people of differing programming skills should be able to complete it. For some it will cover topics they are familiar with like array slicing, agent based modeling, emergent behavior and plotting and interpreting results. For others it will be an introductory assignment in these topics and we have tried to explain them in a way which makes them intuitive. (Professors might encourage students to google functions and documentation as this is a core part of programming.)

The assignment consists of creating the necessary functions and bool statements so that the agents can perform trading. This has been done through the following steps,

1. First the student creates a coin flip function which will be the winning/losing determiner.
2. The student creates an elector function which choses two different agents
3. They create a trading function which utilizes the coin flip function and returns post-trading values.
4. The students initialize a system then use a for loop with these functions and some added code to have the agents trade over a set of iterations
5. They plot the results and interpret it.
6. They are introduced to the Lorenz and the Gini concepts.
7. They use the concepts to analyze the first system.
8. The students write an initialization code and use it and other statements to initialize the second system.
9. The students create another for loop like the first one, but make changes to it to modify and develop the system further (wealth tax).
10. They do the same analysis of the second system as they did with the first and compare them.
11. As a final task the students will create their own system, we have given some examples and inspiration, but they chose to do their own.

As a note we encourage students to work in groups of two to three people.

**Important to know about the system:**

1. It only lets two agents trade per iteration, so the students have to use a fairly large number of iterations and not too many agents. The number of iterations needed to reach the emergent behaviour goes up drastically with the addition of many agents. However this can be easily fixed by nesting a for loop inside of the iteration for loop to let more agents trade per iteration.
2. Multiple places in the assignment we wish for the students to experiment with the variables given, like how much money traded, how many iterations and agents. The starter capital is not interesting to look at given that they trade a certain percentage of their wealth.
3. Too large systems(agents and iterations) will make the jupyter notebook run out of memory, so the students should stay within the limits stated in the task.

We also wish for different instructors to add notes on their implementation below: