**Problem Set 6B (due 11:59 pm, 25 November 2021)**

Remark: Students should submit both Excel files and executable Python programs through Canvas. Text answers in Excel files can be written in Text Box. Text answers in Python programs can be written as comments in the program or in Jupyter Notebook.

**1. Simulation of the Asset Exchange Model**

(a) Write a Python program to simulate the asset exchange model with the saving factors distributed according to

The number of agents is not less than 1,000.

The number of iterations is not less than 5,000.

The number of samples is not less than 5.

The wealth of each agent is initialized to 1.

(b) Construct a histogram of the wealth distribution in log-log scale. To facilitate the analysis of the tail distribution, the histogram should be cumulative from the higher end of the wealth. Check whether the tail distribution obeys the power law. If so, calculate the exponent.

(c) To explain your result to a lay person, you would say that “when we compare agents’ wealth with those agents whose wealth is doubled, the average number of agents will be reduced by a factor of x”. Calculate x.

(d) Construct a histogram of the wealth distribution in log-linear scale among the less wealthy agents, defined as those agents with final wealth less than the initial wealth. To facilitate the analysis of the tail distribution, the histogram should be cumulative from the higher end of the wealth. Check whether the tail distribution obeys the exponential law. If so, calculate the exponential function.

(e) To explain your result to a lay person, you would say that in this regime, “when we compare agents’ wealth with those agents whose wealth is higher by 0.5 times the average wealth of the market, the average number of agents will be reduced by a factor of y”. Calculate y.

(f) Pareto discovered the 80-20 rule when he studied the wealth distribution in Italy in the early 20th century, namely, 80% of the country’s wealth was owned by the wealthiest 20% of the population. Calculate the percentage of wealth owned by the wealthiest 20% of the agents in your model.

**2. Simulation of the Minority Game**

Write a Python program of the Minority Game. The state update is endogenous. The updates of the virtual and real scores are linear scores. Fix and

Outline of the Endogeneous Minority Game simulation:

* Generate the strategies of all agents.
* Initialize the virtual scores of all 𝑁𝑠 strategies to 0.
* Initialize the real scores of all 𝑁 agents to 0.
* Initialize the input state at 𝑡 = 0.
* The strategy choices of the agents are randomly chosen at 𝑡 = 0.
* During each iteration of the trading period, each agent chooses the strategy with the highest virtual score.
* Compute the excess demand. It is sufficient to count the number of buyers, since the other agents must be sellers.
* Update the virtual scores and real scores using the linear score.
* Update the state of the game.
* Iterate the game for not less than 1,000 steps.

Outputs for 1 sample with = 101:

* Plot the number of buyers vs time.
* Plot the real scores of the 3 best agents and 3 worst agents. For your convenience in selecting the agents, you may export the real scores of all agents to an Excel file and sort the final real scores of the agents.

Outputs for 10 samples at each value of for different :

* For each sample, compute the variance of the number of buyers. (‘np.var’ is useful.)
* Repeat the simulation for 10 samples at each value of and compute the average of the variance with its standard deviation.
* Repeat the simulation for the range of *N* between and . Then plot versus using 10 to 20 data points. Include error bars for
* If you are lucky, you may observe the minimum at but do not be surprised if you cannot observe it, as the observation requires computer resources as high as 10,000 iterations and 32 samples.
* For your convenience, the plots can be done in Excel or Python.

Additional comments on 16 November 2021:

* To generate more consistent results, it is a good idea to compute the variance only after the dynamics has reached a steady state. For example, you may compute the variance from iteration 201 to 1,000.
* The requirement of 1,000 iterations and 10 samples should produce reasonable results, but students are welcome to increase the number of iterations and samples if they wish. However, increasing the number of agents to a large value may not provide more information about the game behavior.
* Students may find that the minimum is closer to This is fine if we take into account the finite size effect of the simulations.