Econophysics Project

`ARTIFICIAL FINANCIAL MARKET : an Agent Based Model`

Abstract: Agent-based model of an artificial financial market in which two agent categories, fundamentalists and chartists, trade one single asset through a realistic trading mechanism for price formation mediated by a market maker embodied in a third passive agent category. Cash and shares are assumed to always be available in order to filter out scarcity-induced effects on price time series, hence there is no value-creation process in the model. In each period, agents make stochastic buy and sell decisions, constrained by their corresponding category's trading logic. The model wants to capture the dynamics induced in the price time series by the interaction between the two main categories of traders: fundamentalists and chartists. The model has been implemented through python-based object-oriented programming over the MESA ABM framework.

1 Introduction

The goal is been to build an artificial market that exhibits realistic trading features, takes into account the interaction between the two main trading styles: the fundamentalist and the chartist. Even the most advanced trading models developed by econometric knowledge or by quantitative finance can be traced back to these two very broad categories. It would be unpractical to implement these extremely advanced models on an artificial financial market, It would also probably be unnecessary as we wish to approximate the most important general features in order to understand their origin in real financial markets. We rely on a probabilistic modeling of these two general behaviours and implement this probabilistic behaviour inside agent's choices. We take into account a switching probability for agents to change trading styles based on its near-past performance. Agents are interacting by sharing with each other their respective profits, then this information influences their probability to change their behaviour. It is observed that when the price reaches the fundamental price fixed by fundamental traders, stability is not reached due to the destabilization induced by the presence of the trend-following chartist traders. The interaction between the stabilizing behaviour of fundamentalists and the destabilizing behaviour of chartists is studied by searching for the behaviour of the market when one of the two categories is singled out from the simulation. The next two subsections will be devoted to explain the general behaviour of Chartists and Fundamentalists. In §2 the overview of the model is explained, in §3 the Python implementation is explained, §4 shows the validation tests performed in order to see the validity of the implementation, §5 is devoted to concluding remarks and future work suggestions.

1.1 Chartists

The Chartist category models the behaviour of traders following the discipline of Technical Analysis. It is a trading technique employed to evaluate investments and identify trading opportunities by analyzing statistical trends gathered from trading activity, such as price movement. This stems from the belief that future price can be forecast by evaluating its past performance, without too many consideration on the actual business underlying the asset. This behaviour is the main one of traders in the cryptocurrencies market, which in turn shows greater volatility than the traditional stock market in which fundamental traders generate stability. The other important feature of the Chartist category underlined in the literature is the influence of the behaviour of other traders, which can generate majority opinions among an ensemble of individuals by an infection-like process.

1.2 Fundamentalists

The Fundamental trader category aims to model the behaviour of traders following the discipline of Fundamental Analysis. This kind of trader aims to compute the "fair value" of a certain asset through the analysis of company-specific economical indicators, market placement and concerning events. If the actual price of the asset is lower than the fair value, the fundamental trader will buy the asset and wait for it to reach its fair value. If the assets' price is higher than the fair value the fundamental trader will instead refuse to buy or even decide to short the asset.

2 Model

The model here presented is composed of Agent Traders and a Market Object which implements various market making functions.

2.1 Trading Agents

Our model market contains a fixed total number $N = n_f + n_c$ of traders, where $n_f = n_f(t)$ and $n_c = n_c(t)$ are respectively the the time varying number of fundamentalists and chartists. The chartist group is furthermore divided into two subgroups: optimists and pessimists, which will be denoted as $n_+(t)$ and $n_-(t)$ with $n_+(t) + n_-(t) = n_c(t)$. Optimist chartists will buy and pessimist chartists will sell. On the other hand, fundamentalists will buy if the price p is below the fixed fundamental price p_f, they will sell otherwise.

Chartist's change of opinion:
A charter agent can either be pessimist or optimist. We can measure the overall opinion of chartist agents through the opinion index:

$$x = \frac{u}{n_c} \tag{1}$$

The two extremes $x = 1$ being uniform optimist and $x = -1$ being uniform pessimism. Chartist agents can change their opinion from optimistic to pessimistic and viceversa through rules that dictate how the probability of an agent to switch opinion will fluctuate in time.

The main factors of opinion change are:

- price movement, modeled by price slope $\dot{p} = dp/dt$. It is an average price change calculated over a tunable parameter Δt defined as a certain number of simulation steps, kept fixed for each simulation.
- perceived opinion, modeled by the opinion index which is an information perfectly available to all chartists at all times.

These two factors are going to be linearly combined as follows:

$$U_o = \alpha_1 x + \alpha_2 \frac{\dot{p}}{\nu_1} \tag{2}$$

The parameter ν_1 models the frequency of opinion re-evaluation as will be seen in opinion transition probabilities. The average price change $\dot{p}$ has to be divided by ν_1 since we shall consider the mean price change during the average time elapsed between successive revisions of agent's opinions.
Factors α are weighting the importance of opinion index (α_1) against the influence of price movement (α_2) in the formation of their expectations about future price.
The future expectations of chartists about price movement will influence their transition probabilities through an exponential probability distribution with probability density distribution:

$$f(\lambda, x) = exp(\lambda, x) \begin{cases} \lambda e^{-\lambda x} & x \geq 0 \\ 0 & x < 0 \end{cases}$$

Where λ is called frequency factor. In our particular case we'll have π_{+-} for a pessimist to become optimist and π_{-+} for an optimist to become pessimist:

$$\pi_{+-} = \frac{n_c}{N} exp(\nu_1, U_o) \tag{3}$$

$$\pi_{-+} = \frac{n_c}{N} exp(\nu_1, -U_o) \tag{4}$$

These transition probabilities stem from quantitative sociological models as shown in [2].
During the simulation, the probability of opinion change will be $\pi_{-+}\Delta t$ (or $\pi_{+-}\Delta t$) and will be implemented asynchronously. It is important to note that even if the present state of the market will dominate (e.g. positive price slope and overall optimist sentiment), there will always be a minority going against the current.
The scaling factor $\frac{n_c}{N}$ in front of the exponential is a mean-field parameter for the fact that chartists change opinion when encountering other chartists, while they encounter fundamentalist traders with a probability $\frac{n_f}{N}$ and in that case they will undergo a probabilistic strategy switch decision. We say mean-field because the model doesn't have any spatial interaction implementation.
In general we have that at each step a chartist trader can meet another agent from the simulation, which can either be a fundamentalist or a chartist. Since the chartist fraction n_c/N is also the probability of encountering a chartist, we need to multiply the probability of change of opinion with the probability of encounter with a chartist. In fact, if the chartist agent will encounter a fundamentalist, with probability n_f/N, he will not consider a change of opinion but instead a change of strategy, which leads us to the next subsection of the model.

Strategy Switching:
Chartists can choose to become fundamentalists and vice-versa. The two different strategies will yield different so-called excess-profits during the simulation. They will switch strategy with a probability proportional to the difference between the excess profits of both strategies. In order to evaluate this difference, which will be used to get the strategy transition probability, we need to see how excess profits can be computed for chartists and for fundamentalists.

1) Chartists excess profits:
Considering an optimistic chartist, we know that the possession of shares will pay a certain dividend r, assumed constant. The dividend gain has to be summed with the variation of price $\dot{p}$, which if positive will increase excess profits. By normalizing this sum with price we get the relative excess profit of the strategy. This will be discounted by a factor R which represents a safer but less rewarding investment strategy. This stems from the fact that in finance, profits are computed as the gain over a standard investment class.
This can be expressed as $(r + \dot{p})/p - R$.
On the other hand, for pessimist chartists, the goal is to reduce loss by selling, this can be expressed as $R - (r + \dot{p})/p$.
In order to avoid the influence of R in the case of stable prices $\dot{p} = 0$ we keep $R = r/p_f$ with p_f the fundamental price.

2) Fundamentalists excess profits:
We can modify the expression for chartists' excess profits by taking into consideration two things: Dividends bring a term r/p in the expression and are paid infrequently to fundamentalists evaluated to being equal to R since we can take $p = p_f$ because of the fact that dividends usually get paid infrequently. We then get $r/p_f = R$ and an $R - R$ cancels out in the expression. This leaves us with $\dot{p}/p$ but since fundamentalists have their reference price p_f this can be rewritten as $s|(p - p_f)/p|$ with $s < 1$ a discounting factor for the fact that future returns are valued less than present returns in financial accounting.

Summarizing these concept in the excess demand parameters we get:

$$ep_{c+} = (r + \dot{p})/p - R \tag{5}$$

$$ep_{c-} = R - (r + \dot{p})/p \tag{6}$$

$$ep_f = s|(p - p_f)/p| \tag{7}$$

Strategy switch probabilities are denoted by $\pi_+ f$ as the probability of a fundamentalist to become an optimist chartist, the same reasoning applies to all the other combinations.

$$\pi_{+f} = \frac{n_+}{N} exp(\nu_2, U_+) \tag{8}$$

$$\pi_{-f} = \frac{n_-}{N} exp(\nu_2, U_-) \tag{9}$$

$$\pi_{f+} = \frac{n_f}{N} exp(\nu_2, -U_+) \tag{10}$$

$$\pi_{f-} = \frac{n_f}{N} exp(\nu_2, -U_-) \tag{11}$$

With α_3 weight factor on the exponential distribution arguments, measuring the difference betweene excess profits:

$$U_+ = \alpha_3(ep_{c+} - ep_f) \tag{12}$$

$$U_- = \alpha_3(ep_{c-} - ep_f) \tag{13}$$

2.2 Market maker

The market maker is needed to facilitate the trading between active trading agents. Its role is to open the market at a certain price and to introduce agents to it, to compute the opinion index and the price slope and to communicate it to all participating traders. Its most important role is updating the price at each timestep. In a real-world market, price updates are communicated to the market by the market regulator at each time-step after a very complicated bid-ask fulfilling process done in real time. In our Market maker we aim to coarsely model this behaviour. Since we know the probabilistical behaviour of our agents and we know how many of them are going to buy and how many of them are going to sell, we can compute the total excess demand generated by them.
We know that for each chartist, if it is optimist it will buy, if pessimist it will sell. On the other hand we know that a fundamentalist will buy if $P(t) < P_f$ and will sell if $P(t) > P_f$. We now want to calculate a quantity $d(t)$ total excess demand:

$$d(t) = d_c(t) + d_f(t) \tag{14}$$

$$d_c(t) = n_c(t)\, x(t)\, \gamma_c \tag{15}$$

$$d_f(t) = n_f(t)\, (P_f - P(t))\, \gamma_f \tag{16}$$

with γ_c and γ_f parameters measuring the relative influence chartists and fundamentalists have on price formation. One can think of these two parameters as how many shares of the asset each category buys on average at each time step. Since $d_c(t)$ and $d_f(t)$ are proportional to the number of traders, γ parameters are kept proportional to how many traders are initialized in the simulation by the two hyperparameters:

$$T_c = N\, \gamma_c \tag{17}$$

$$T_f = N\, \gamma_f \tag{18}$$

2.3 Parameters Recap

Here we list all the parameters needed for running our simulation:

1. Market parameters
   - $n_c(0)$: ..... number of initial chartists
   - $n_f(0)$: ..... number of initial fundamentalist
   - Tc: ..... chartists' influence on price formation
   - Tf: .... fundamentalists influence on price formation
   - P_0: ..... initial price
   - β: ..... price change frequency
   - dP: ..... basal price change
   - σ_n: ..... Noise traders variance
   - $t_{\dot{P}}$: ..... time range for price derivative
2. Chartists' opinion parameters
   - ν_1: ..... transition frequency
   - α_1: ..... opinion weight
   - α_2: ..... price slope weight
3. Strategy switch parameters
   - ν_2: ..... transition frequency
   - α_3: ..... weight of strategy change
   - R: ..... safe investment returns
   - r: ..... asset dividends
   - s: ..... discount factor
4. Fundamentalists parameters
   - P_f: ..... value price of the simulation asset
5. Computational parameters
   - Δt: ..... Time step value
   - m_t: ..... Minimum number of traders

2.4 A missing ingredient: Noise Traders

Noise Traders category models the population of uninformed/retail traders participating in a market. They are assumed to buy and sell at random, hence creating a random excess demand value. This $d_n(t)$ will be a normally distributed random variable since the actions of noise traders are assumed to be completely independent. The ultimate excess demand will then be:

$$d_{final}(t) = \beta(d(t) + \mathcal{N}(0, \sigma) \tag{19}$$

3 Model Implementation

The simulation code has been written on python language with the aid of the agent-based modeling MESA framework [1]. Mesa is a modular agent-based modeling tool composed of three main components:
Modeling: Modules used to build the models themselves: a model and agent classes, a scheduler to determine the sequence in which the agents act, and space for them to move around on.
Analysis: Tools to collect data generated from your model, or to run it multiple times with different parameter values.
Visualization: Classes to create and launch an interactive model visualization, using a server with a JavaScript interface.
The Modeling module has four main submodules: `mesa.Model, mesa.Agent, mesa.Space, mesa.Time.`
In our project mesa.Space has not been used, as there is no need for defining spatial dynamics for trading agents. The mesa.Model object serves as the main manager of the simulation, it creates agents, updates their internal state and manages the information flow to, from and in between agents. On the other hand the mesa.Agent has been used to model the behaviour of the two main agent classes: Technical traders and Fundamentalist traders.

3.1 `mesa.Model: Market.py`

In our project, the `mesa.Model` submodule has been implemented in the `Market.py` python script. In it the price of the asset exchanged by agents `class PriceSeries` is implemented as an object containing a list of floating point values representing the price of the asset in time, it contains the information about price's hystory. The `PriceSeries` is controlled by `class Mercato(mesa.Model)` which is sub-classed from the `mesa.Model` class. The main role of this class is to generate agents at the beginning of simulation, step the simulation and control the flow of time, update the price at each timestep and insert its value into the PriceSeries and update the total number of traders for each possible category. The information about price movement and the count of traders in each category is available at each time step to all traders in real time.
Price update is performed by following the procedure described in sections 2.2 and 2.4. After $d_{final}(t = \texttt{timestep})$ is computed for the current timestep, a price transition probability is computed to account for the fact that price could stay stationary if excess demand is not strong enough. The tendency of price to always move is proportional to the magnitude of parameter β (see 2.3). Then with probability $p = abs(U) = abs(\beta d_{final}(t))$ the price will be updated of a quantity $\Delta P(t) = U(t)dP$.

3.2 `mesa.Agent: Agenti.py`

An unique agent class has been implemented as `class Trader(mesa.Agent)`. Each agent belonging to this class has an internal strategy state which can take two possible states: `Strategies.Technical` or `Strategies.Fundamentalist` by sub-classing from a `class Strategies` object. If the internal state of an agent is `Strategies.Technical` then its internal state will also contain a `self.Opinion` component.
At each timestep, every agent will be activated at random by the simulation scheduler and will then perform its actions. It first will perform a buy or sell action based on its current internal logic. For example it will buy if its internal strategy state is equal to `Strategies.Fundamentalist` and the price is below the fundamental value. How much it will buy is established in how the excess demand is calculated in Market.py and is not an internal decision that each agent will take independently. The only independent decision agents have to take is if to buy or to sell, which just depends on their current state. Afterwards the agent will decide if to keep its current state or to change it by changing opinion and/or strategy. If the agents wants to change from its current step, then this decision will take place on the next time-step. Each time an agent changes its internal state, this action is communicated to the Market Maket through calling the mesa.Model wide function `Update Traders Count`.

3.3 `run.py`

The `run.py` script serves as a wrapper script to run the whole simulation at once. It is just needed to run this script in order to launch the simulation. This script also automatically performs the statistical analysis described in section 4.

3.4 `conf.py`

This script collects all the different model parameters (as described in section 2.3) in order for the user to change them at will and see how the model changes its behavior. In there some important hyper-parameters such as total simulation step and number of simulations to run in the same batch can be defined. While running, the model will automatically save its data in an user-defined folder which can be defined in `conf.py`.

4 Statistical analysis of financial stylized facts

One of the main features a good agent-based model of an artificial financial market has to reproduce are the so-called stylized facts, that is to say consistent empirical findings in financial asset data such that they are recognised as true even if no true bottom-up explanation is still given to them in the economic literature. The stylized facts we will analyze are related to statistical features of returns series, defined as:

$$r_t(\tau) = p(t) - p(t - \tau) \tag{20}$$

These are:

- **Fact 1:** Non-stationarity of asset prices
  Non-stationarity implies that by feeding time-series data into a unit-root statistical test such as the Dickey-Fuller test, one is unable to reject the null unit root hypothesis, that is to say one is unable to reject the null

hypothesis of price being just a random walk. Given:

$$p(t) = \rho p(t-1) + \epsilon(t)$$

One is unable to reject $\rho = 1$ on the Dickey-Fuller test. (note: $\epsilon(t)$ is some stationary random increment) In fact, if the process would be stationary, then one could predict its future behaviour by knowing its past behaviour through the knowledge of time-autocorrelation of prices. That is not to say that real financial markets are unpredictable but that their predictability is very short-ranged.

- **Fact 2:** Volatility Clustering
  Volatility clustering is easily understood through the quote of Benoit Mandelbrot: *large changes tend to be followed by large changes, of either sign, and small changes tend to be followed by small changes.*. This means that while returns themselves aren't correlated, the square of them r_t^2displays a positive and slowly decaying autocorrelation function $corr(r_t^2, r_{t+\tau}^2)$ with different τ values such as seconds, minutes, hours, days, weeks and months.

- **Fact 3:** Fat tail phenomena
  When searching for the probability distribution function of returns $f(r)$, one sees that it has more probability density on the tails and less on the centre of the distribution than an usual normal probability distribution.

4.1 Stylized facts check

Our code automatically analyzes stylized facts after each model run. Fact 1 is analyzed by feeding prices data to python's `statsmodels.tsa.stattools.adfuller` which performs the Dickey-Fuller test on our data and gives a p-value which is confronted with a significance value $\alpha = 0.05$. For each run in each parameter sets the null hypothesis was never refused.
Fact 2 is analyzed by feeding the square of prices data to python's `statsmodels.stats.diagnostic.acorr_ljungbox` which analizes autocorrelation and also produces a p-value. Here the null hypothesis is that data is independently distributed. In most of our runs this hypothesis has been refused, meaning that we see volatility clustering in our model. Fact 3 has been analyzed by calculating the kurtosis value of returns distribution, if this value is greater than zero we then have leptokurtosis, which means more probability density is present in the tails. This fact has been difficult to reproduce consistently, probably due to limitations in run-length or due to limitations in the used kurtosis algorithm `scipy.stats.kurtosis`. Different plots can be seen in the next section, divided per parameter parameter set.

5 Parameters sets examples

In every simulation the following parameters have been kept at the fixed values of:

Table 1: these parameters are equal for each parameter set

$n_c(0)$	number of technical traders	250
$n_f(0)$	number of fundamentalist traders	250
dP	basal price change	1e-4
$t_{\dot{p}}$	price slope time range	20
Δt	timestep value	0.01
m_t	minimum traders count	5

5.1 Parameters set 1

Table 2: Parameter set 1

T_c	technical's demand	2
T_f	fundamentalist's demand	1
P_0	initial price	10
$t_{\dot{p}}$	price slope timesteps	20
β	price change frequency	6
ν_2	strategy change frequency	3
R	safe investment return	0.04
r	asset dividends	0.12
s	discount factor	0.75
ν_1	opinion change frequency	5
α_1	opinion index weight	0.8
α_2	price slope weight	0.5
σ_n	noise traders variance	2
P_f	fundamental price	10

Figure 1: set 1 - run 1 - on top the statistical results are shown. From top to bottom we have price, strategy fraction and opinion index

Figure 2: set 1 - run 2

T_c	technical's demand	2
T_f	fundamentalist's demand	1
P_0	initial price	10
$t_{\dot{p}}$	price slope timesteps	20
β	price change frequency	**3**
ν_2	strategy change frequency	**2**
R	safe investment return	0.04
r	asset dividends	0.12
s	discount factor	0.75
ν_1	opinion change frequency	**1**
α_1	opinion index weight	0.8
α_2	price slope weight	0.5
σ_n	noise traders variance	**1**
P_f	fundamental price	10

Table 3: Parameter set 2

5.2 Parameters set 2

Figure 3: set 2 - run 1

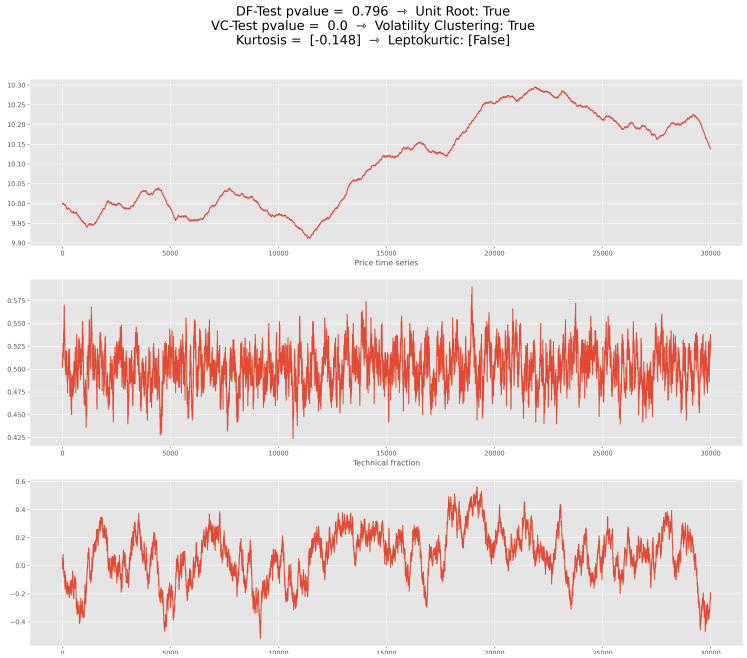

Figure 4: set 2 - run 2

With respect to parameter set 1, this set has lower frequencies β, ν_1, ν_2 and lower noise traders variance. It can be seen by confrontation that the opinion index series here is "slower" than in set 1 and also slower than technical fraction series as now $\nu_1 < \nu_2$.

5.3 Parameters set 3

Set 3 shows what happens if the initial price is not equal to the fundamental price. In order to have a clear picture of what's happening, we put technical and fundamental demand at an equal value of 3, initial price has been set to 12 while the fundamental price is 11, hence we expect the prices to decrease in the long run. All other parameters have been kept at the same value as parameter set 1.

Table 4: Parameter set 3

T_c	technical's demand	3
T_f	fundamentalist's demand	3
P_0	initial price	12
$t_{\dot{p}}$	price slope timesteps	20
β	price change frequency	6
ν_2	strategy change frequency	0.5
R	safe investment return	0.04
r	asset dividends	0.12
s	discount factor	0.75
ν_1	opinion change frequency	1
α_1	opinion index weight	0.8
α_2	price slope weight	0.5
σ_n	noise traders variance	2
P_f	fundamental price	11

Figure 5: set 3 - run 1

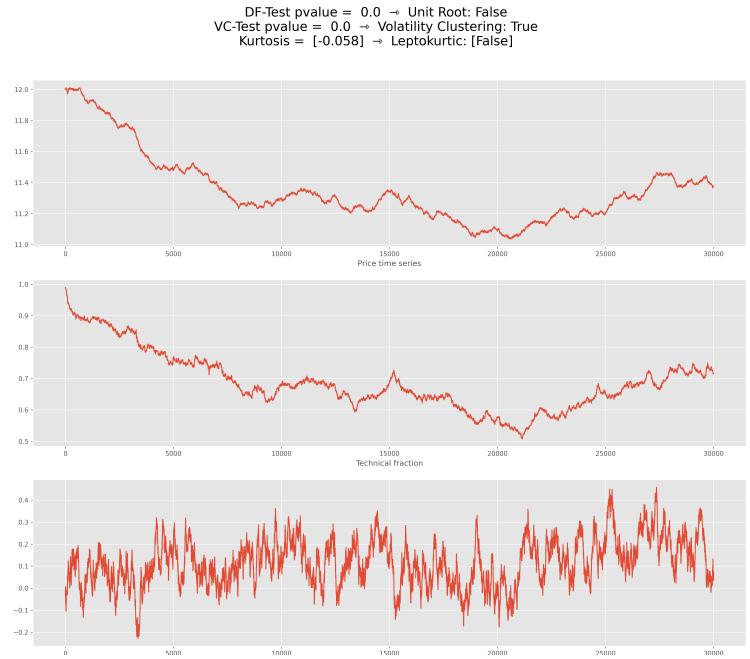

Figure 6: set 3 - run 2

6 Conclusion

Our Agent-Based Model of an artificial financial market shows the right financial statistical behavior one would expect to get from a proper financial model. Its parameter-dependant behaviour reflects the meaning associated to various parameters, although as it is usual for ABM models, it is difficult to interpret the simulation parameter values with true values associated with real-life financial markets. This model is still quite simple, for example agents in each category could have different parameters, or these could change as the simulation unfolds. Also there are no effects related to finite resources, as there is an endless amount of money and stock to be exchanged in our market. All those last considerations can be ideas to further develop the model for it to be more accurate and realistic. It is anyway interesting to see that such a reduced model can already account for the complexity of financial market and can help us understand qualitative characteristics of a specific market by fine-tuning of parameters.

References

[1] Jackie Kazil, David Masad, and Andrew Crooks. "Utilizing Python for Agent-Based Modeling: The Mesa Framework". In: (). `https://mesa.readthedocs.io/en/latest/index.html`.

[2] Thomas Lux and Michele Marchesi. "Volatility clustering in financial markets: a microsimulation of interacting agents". In: *International journal of theoretical and applied finance* 3.04 (2000), pp. 675–702.