

COMSM0140: Internet Economics and Financial Technology (IEFT)

Academic Year 2023-24.

Main Coursework: this final summative coursework is for a maximum of 10 Credit Points.

Coursework specification release date: Friday 17th November 2023.

Coursework submission deadline: Thursday 7th December, 1pm (Blackboard)

Jupyter Notebook exploring automated trading in the BSE financial-market simulator. Maximum length 2000 words.

Rather than write a static printed conference-paper, for this assessment you are required to create a fully functioning Jupyter notebook, with text, graphics, and executable code segments, all in one zip-compressed folder. It should be written as if the final intended use for the notebook is in future teaching on INFT: you don't need to explain the background history or write a literature review, but your notebook should have a definite narrative, explaining what is going on – as in the Jupyter notebook of the Vernon Smith demo experiments that is available on the BSE GitHub repository (Cliff, 2023). The maximum length of your notebook is 2000 words of Markdown. You are provided with a template notebook which contains a word count function. Do not edit this word count function.

You should submit a single zip-compressed folder containing:

- Your Jupyter notebook
- A printed PDF version of your notebook (to create a PDF, use either the “export” function of VSCode or the “Print” option in the “File” menu if using web browser or JupyterLab).

For this coursework submission you are required to work alone: you should not share code or data or text or images with any other student.

The coursework is in four parts:

Part A: This requires you to use BSE to perform a straightforward A/B comparison between two types of trader-agent. Assign appropriate values to parameters passed to BSE's `market_session` function to set up a market as follows:

- Set the supply/demand to be based on that used by Vernon Smith in Chart 4 of his landmark 1962 paper in the *Journal of Political Economy* (Smith, 1962): all sellers should have the same limit price of 310, giving a flat supply curve; buyer limit prices should range from a maximum of 490 at the high end of the demand curve, down to 250 at the low end, with equal-sized steps along the curve.
- Set the population of buyers, and also the population of sellers, to be R% traders of type SHVR and (100-R)% traders of type ZIC. Smith's Chart 4 shows 13 buyers and 13 sellers, but in your version of the experiment you should use 20 buyers and 20 sellers.

Then run $n=50$ independent and identically distributed (IID) repetitions of this market session with $R=50$ to gather a data-set of profitability figures for the two trader types. Select and apply an appropriate visualization of this data-set. Select and apply an appropriate statistical test, or sequence of tests, to determine whether, in the $n=50$ data-set you generated, traders of type SHVR are statistically more profitable than ZIC or statistically less profitable than ZIC, or whether the profitability of the two is, in your experiments, statistically indistinguishable. Then repeat this experiment again, but now using $n=500$. Write narrative text in the Jupyter notebook to explain your experiment set-up, the results, your visualization and analysis, and to state what conclusions you draw. **[10 marks]**

Part B: Generate more data sets as for Part A, but now for $R=10, 20, 30, 40, 60, 70, 80$, and 90 , for $n=50$ and $n=500$. Select and apply appropriate visualization methods for this data set, and select and apply an appropriate test or sequence of tests to determine at each R -value whether SHVR is statistically more profitable than ZIC, less profitable than ZIC, or indistinguishable from ZIC. Write narrative text in the notebook explaining what you did and what your findings are. **[15 marks]**

Part C: Add code to your notebook to run a new set of experiments, in which there are four different types of trader, SHVR/GVWY/ZIC/ZIP, and generate data from market sessions in which the ratios of these traders are $(25:25:25:25)$, all four permutations of $(40,20,20,20)$ ¹, all four permutations of $(10,30,30,30)$, and all four permutations of $(70,10,10,10)$. Again, select and apply appropriate visualizations, and select and apply appropriate statistical test(s) for each ratio's data, to determine whether there are any statistically significant differences in profitability between SHVR, GVWY, ZIC, and ZIP at any of the specified ratios. **[20 marks]**

Updated Part D (version: 21st Nov. 2023)

NOTE: To do well in Part D you will need to edit BSE.py and execute relatively large numbers of market-session simulations using your edited code, which collectively can use up quite a lot of CPU time. Once you understand what to do, making the necessary edits to the code might not take you very long but to generate enough data you may then need to set simulation experiments to run on your laptop/desktop PC for several hours or more of continuous processing (e.g. overnight), during which you can be working on other things, or sleeping. Because of this, it is important to plan ahead, and to not leave until the last minute before starting work on this part of the coursework. What is required to do well here is readily achievable on a current mid-range personal computer: you do not require access to high-performance supercomputers like the University's BlueCrystal/BluePebble facilities, but you should expect to have to wait quite a while for enough data to be generated.

Part D: A recent paper by Cliff (2023) describes the long-term co-evolutionary dynamics of markets populated by ZIP traders that are continually trying to improve their hyperparameter values, to trade more profitably, by using a simple *stochastic hillclimber* (SH) method to search for better hyperparameter values, thereby giving a variant of the ZIP adaptive trading strategy known as ZIPSH. As described in more depth in (Cliff, 2023) the SH mechanism in a single ZIPSH trader maintains a private local population of k different candidate strategies, tests the profitability of each of them in sequence, and then identifies the most profitable of the k as the 'elite' strategy; it then creates $k-1$ new "child" strategies, each child being a *mutation* of the elite strategy, and replaces the non-elite members of the old population with these new children; and then it

¹ When we say "...all four permutations of (N, M, M, M) ..." we mean the following set of permutations: (N, M, M, M) ; (M, N, M, M) ; (M, M, N, M) ; and (M, M, M, N) .

evaluates the profitability of each member of the new population to identify its elite individual, which is then mutated to create $k-1$ children; and so on, repeating this process forever.

This simple SH process is referred to as a *hillclimber* because it is often explained via the metaphor of a walker in the outdoors trying to find the highest part of a landscape, while blindfolded: the walker has a walking stick, and she finds the highest nearby piece of ground by using the stick to prod k different patches of ground around her, to determine which is the highest (here “nearby” means within prodding distance of her walking stick held from an outstretched arm); and then she takes a step to the highest point prodded; and then she starts prodding again. The SH process is *stochastic* because the mutation process introduces random changes rather than systematic/deterministic ones: returning to our walker, if her $k=4$ then on each iteration she could systematically always prod nearby patches of ground to the compass north, south, east, and west – that would be a fixed, deterministic pattern; or she could instead pick four directions at random on each iteration – that would be a stochastic choice of local sample-points.

In ZIP there are five key parameters (the trader’s *initial margin*, its learning rate *beta*, the *momentum* parameter used to smooth over noise in the learning process, and the trader’s two constants c_a and c_r which respectively determine absolute and relative perturbations when forming the trader’s *target price* used in the ZIP learning process), so the candidate strategies in ZIPSH can each be thought of as five-dimensional (5D) vectors of real numbers, and the stochastic *mutation* process involves adding random values to each of the five: in ZIPSH, each random mutation value is drawn from a zero-mean Gaussian distribution with a small variance. It’s not essential to use a Gaussian distribution, but it is important that the distribution for mutations is zero-mean, because any nonzero value of the mean would introduce a systematic bias; and it’s important that the variance is small to preserve the notion of “nearby”, i.e. to keep the k -point search *local* at each step.

A process as simple as SH can often work pretty well, but it has some definite drawbacks. It works well in situations where the *fitness landscape* (in our ZIPSH example, this is the profitability at each point in the 5D space of possible strategy vectors) is smooth or *locally correlated* – that is, where the difference in fitness/profitability of any two points in the strategy space typically grows smaller as the distance between the two points under consideration reduces. Unfortunately, many real-world fitness landscapes are not smooth, but may instead have sudden major step-changes up or down, or may have fractal-like “ruggedness”, and in practice very often the only way to know for sure what the fitness landscape is actually like is to sample it at many points, which can be a very time-consuming process.

One well-known problem is that simple search processes like SH can easily become trapped on *local maxima* – when our blindfolded walker reaches the top of a small hill, every point she prods with her stick will be lower than her current position, and so from then onwards she is stuck, even if the slopes of a much higher hill lies only a few steps away. Similarly, an individual ZIPSH trader may evolve its private population of k strategies to the point where every mutant it generates always has lower profitability than the current elite strategy, and so the improvement in profitability stops, even if the current elite strategy is not particularly good. Another problem is that the outcome of a SH process can be very sensitive to initial conditions: whatever values are in the population of candidate strategies at the start of the process can have a big effect on how the ZIPSH process unfolds over time and where it ends up, what the final elite strategy is. To revisit the metaphor: for our hillwalker, which point of high ground she finally ends up on could depend a lot on where on the map she started walking from. And, finally, if there is significant noise in the

fitness function (in ZIPSH, this is the method of evaluation of the profitability of any one candidate strategy), this can have a major destabilising effect on the efficiency and final outcome of the SH search, because as noise levels increase, so more and more of the SH “adaptive steps” will be mis-steps, taking the search in the wrong (or, at least, a suboptimal) direction.

Using BSE version 1.8 (released on Tuesday 21st November 2023), we can illustrate and explore ZIPSH’s adaptive process in action by setting up a population of traders in which there is only one ZIPSH trader – here we will make that ZIPSH trader a buyer, with all other traders in the market being the non-adaptive ZIC strategy.

Because transaction-price time series in financial markets are almost always noisy (and especially because the ZIC traders we’re using here so highly stochastic), the process of evaluating the profitability of our sole ZIPSH trader’s candidate strategies will involve monitoring the profitability of each individual strategy over an appropriately long period of time. In BSEv1.8 the function `init_stratwaittime()` within ZIP’s `__init__()` function chooses a random evaluation period of between two and three hours (i.e. 7200 to 10800 seconds). This evaluation period determines how many ‘adaptive steps’ (i.e., cycles through the loop of evaluating the current population of strategies, identifying the elite, and mutating the elite to make children that will form the next population) can be taken in any one 24-hour period. For example, if a ZIPSH trader evaluates each candidate strategy for two hours, and uses a population size of $k=4$, then evaluating the whole population takes eight hours and so in any one day of continuous round-the-clock trading that trader can take no more than three adaptive steps. Furthermore, because we don’t know in advance how many adaptive steps the trader will need to demonstrate the adaptation is working well, we should in the first instance run our experiment for a fairly large number of days, some number of weeks of continuous 24x7 trading: the graphs below show the results from BSEv1.8 market sessions lasting 30 days of continuous trading.

Figure 1 shows the profitability (measured as profit-per-second, or PPS) of the sole ZIPSH buyer in the otherwise all-ZIC market, with 10 buyers and 10 sellers, from a single 30-day market session. Here the demand curve ran from 150 down to 125, and the supply curve ran from 50 up to 75, which means that every trader should be expected to find a counterparty for a transaction on every order assigned to it. To highlight the effect of the progressive adaptation of hyperparameter values, in all the illustrative experiments discussed here the initial values of the ZIP hyperparameters `margin_buy` (the buyer’s initial margin), `beta` (its learning rate), and `momntm` (its momentum) were all deliberately set to very low values,² values so low that they result in reduced profitability: that is, these illustrative experiments have been deliberately set up to make it easy for the ZIPSH to improve over time: by increasing these initially low hyperparameter values, profitability is likely to increase.

It’s important to remember that the situation our single ZIPSH buyer faces remains highly challenging, and is unlike any real-world market scenario: for our ZIPSH, every other trader in the market is a ZIC, and ZIC quote-prices are just (appropriately bounded) random noise values which change very frequently.

² Specifically, in all these experiments `init_beta()` used `U(0.01,0.01)`; `init_momntm()` used `U(0.0,0.1)`; and `init_margin()` used `U(0.05,0.05)`, where in each case `U(min, max)` denotes a real-valued random draw from a uniform distribution over the interval $[min, max]$ – when $min=max$, the outcome of the random draw is just a constant with probability one.

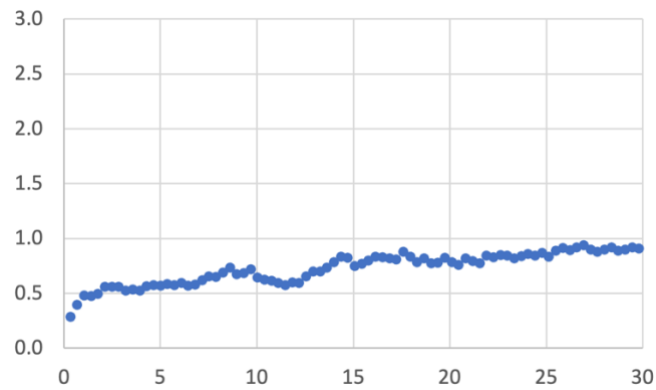


Figure 1: profitability of a sole ZIPSH buyer in a market populated by ZIC traders. Horizontal axis is time, labelled in days; vertical axis is the profit-per-second (PPS) generated by the ZIPSH trader.

Nevertheless, as can be seen in Figure 1, there is indeed a clear improvement in profitability, which rises from a PPS of roughly 0.25 at Day 0 to a PPS near 1.0 at Day 30, an increase of roughly 300% in this experiment. So, the SH mechanism has to a decent extent tailored the single ZIPSH trader's strategy hyperparameters to make it better suited to trading in the market full of ZIC traders, with these specific supply and demand curves. This shows that, in principle, SH can perform fairly well for such a primitively simple adaptation mechanism.

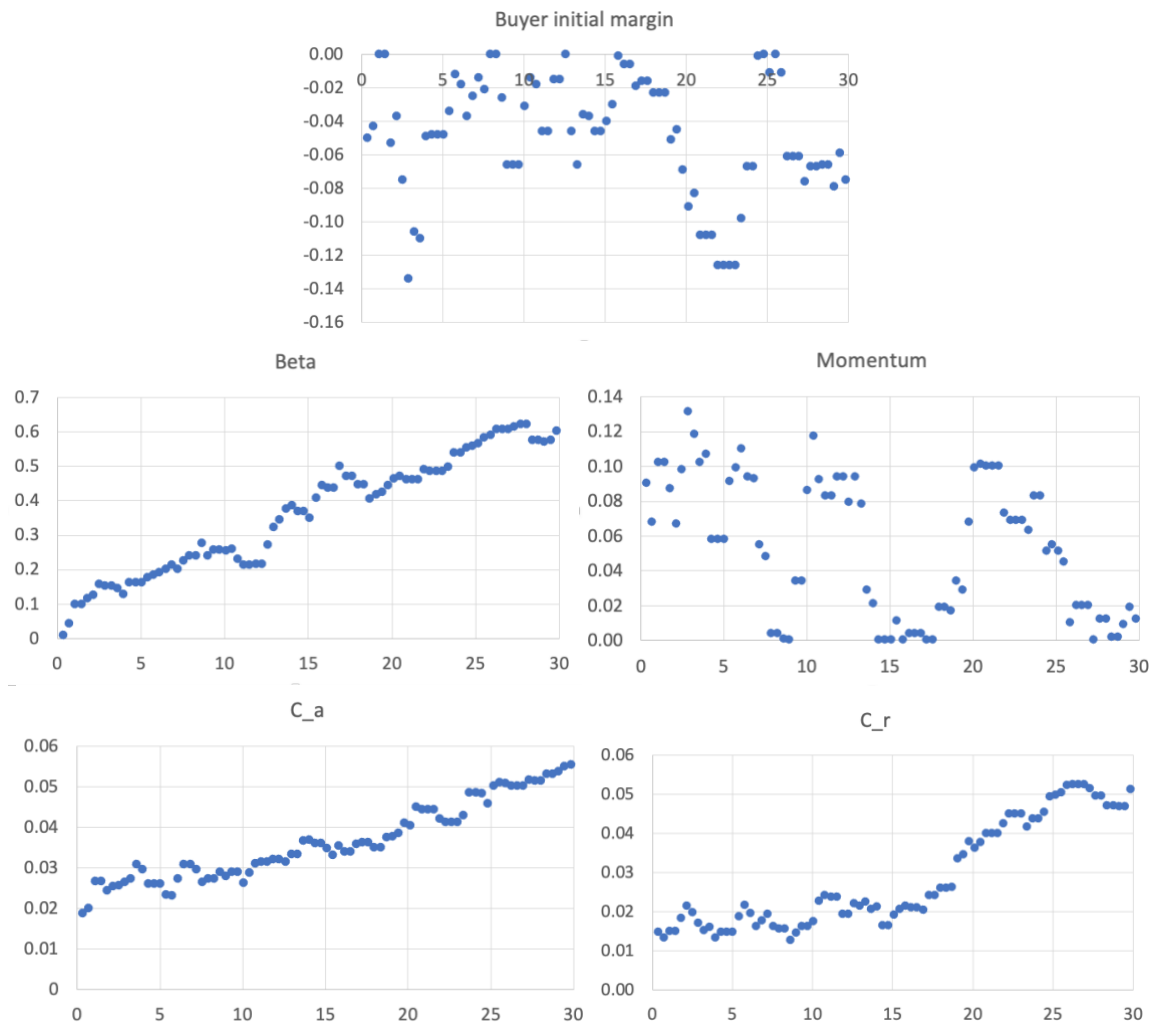


Figure 2: Time-series showing evolution of the five key ZIP hyperparameters in the ZIPSH experiment illustrated in Figure 1: in each graph the horizontal axis is time, labelled in days, and the vertical axis is the named parameter-value.

Figure 2 shows the changes in the ZIPSH trader's hyperparameter values over the duration of the same experiment as was illustrated in Figure 1. As you can see, each of the hyperparameters is gradually altered to add to the improvement and maintenance of higher profitability.

But this is an inherently stochastic system: run it again and the plots of the time-evolution of the trader's profitability and its hyperparameter values will differ from the results seen above, and so then a key question is: how can we characterise the expected or average behaviour of the system? Put another way, how representative of the usual behaviour of the system is the single set of results chosen to be illustrated in Figures 1 and 2?

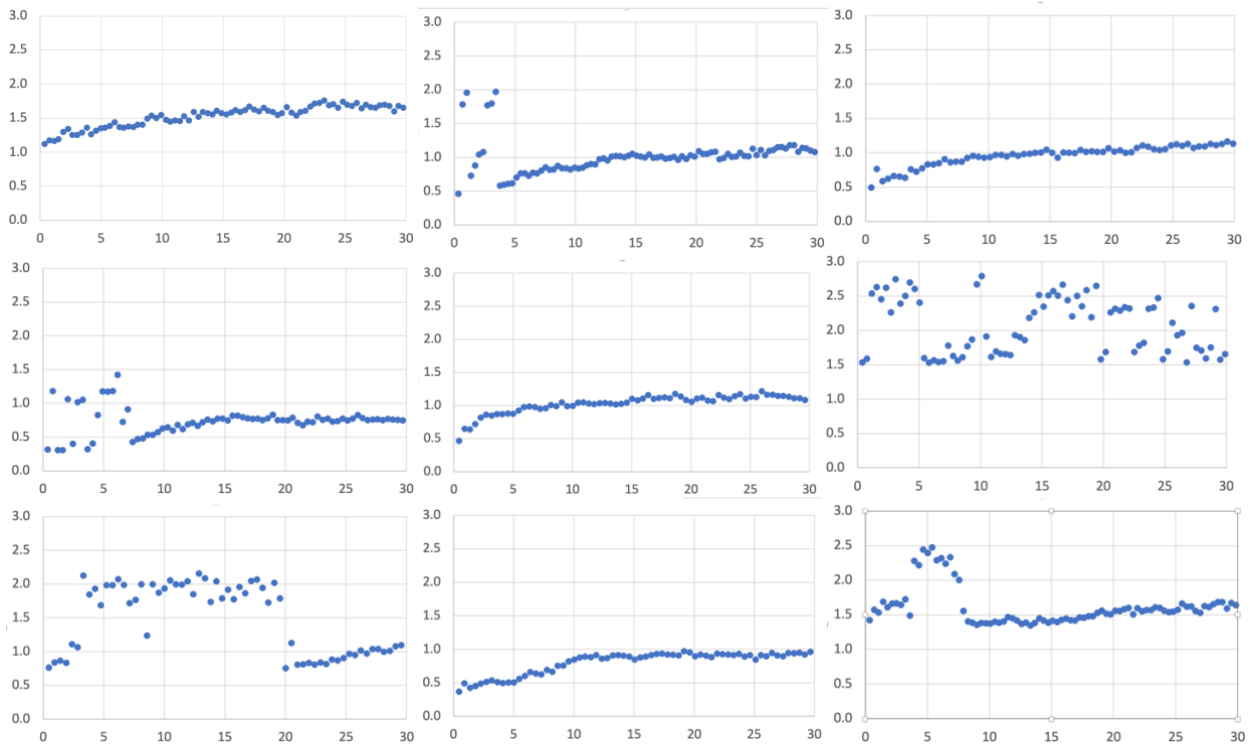


Figure 3: Profitability results from a further nine IID repetitions of the ZIPSH experiment illustrated in Figure 1: graph axes and labels are as for Figure 1; see text for further discussion.

To illustrate the variability, Figure 3 shows the profitability time-series from a further nine independent and identically distributed (IID) runs of the market session that generated Figures 1 and 2: as you can see, there is a fair amount of variation between different IID repetitions of the same single-ZIPSH session. In each of these experiments, because key hyperparameters were initialised to very low values, initial profitability is often in the range 0.25 to 1.5, and in half the runs shown here there is then a slow, steady, low-variance increase in profitability scores by 50%-100% over the course of the experiment (e.g., from PPS=1.0 at Day 0 to PPS=1.5 at Day 30; or from PPS=0.5 at Day 0 to PPS=1.0 at Day 30). In those runs that do not show a consistent low-variance improvement in profitability, markedly higher profit scores of 1.5 to almost 3.0 can be achieved for sustained periods of time, but with manifestly higher variance and often without enduring stability. These periods of higher scores often result from ZIPSH finding 'extreme' solutions, such as switching off learning (setting $\beta=0$) and instead relying on a very high `margin_buy` value, which can give an increased probability of more profitable trades but which leaves the trader no way of adjusting to a lower margin if the initial `margin_buy` value turns out to be too high -- in which case the trader will struggle to make any profit at all and then the adaptive SH process is likely to switch to strategies with nonzero β and reduced `margin_buy`, thereby reverting to the lower-variance (but lower-profit) adaptive class of strategy.

These illustrative results from ZIC markets with a single ZIPSH buyer do show that the stochastic hillclimber can indeed adapt the ZIP hyperparameters to increase profitability; but remember this is a contrived example with the initial values of the five ZIP hyperparameter set deliberately low, and we've only shown the results from 10 runs here, which is a small sample-size: for statistically rigorous analysis, we would need to generate and analyse considerably more than 10 runs.

D.1. For this part of the coursework assessment, edit your local copy of the BSEv1.8 code to give your own replication of the single-ZIPSH experiments described above, and then execute sufficiently many IID runs to be able to rigorously characterise outcomes of this system via appropriate visualizations and analyses of results, using an appropriate statistical significance test to determine whether, in your results, there is conclusive evidence that in these all-ZIC-plus-single-ZIPSH sessions, ZIPSH does show a reliable improvement in profitability. In doing this, you may choose to run your market sessions for less than 30 days, so long as you justify that choice – shorter-duration sessions will take less time to simulate. Similarly, you may choose to alter the `init_stratwaittime()` function to give each strategy less evaluation time, which will increase the number of adaptive steps per day but will increase the noise level in the profitability scores, and again if you do that then you should explain and justify your choice. You are free to make further edits to BSEv1.8 if you wish, so long as they also are explained and justified. **[20 marks]**

D.2. Next, introduce your own variations or extensions to explore the extent to which the single-ZIP-vs-multiple-ZIC results are affected by changes in the design of the experiment, and/or changes in the ZIP hyperparameter adaptation technique. For example, you might want to explore the effects of changes to the shapes of the supply and demand curves, and/or changes in the number of traders in the markets, and/or changes in the type of evolutionary hyperparameter optimization that the ZIP traders use (i.e., something other than SH). For any such change, you will need to select and apply appropriate visualizations, and you will need to select and apply appropriate statistical tests to see whether the change you introduced does lead to a statistically significant alteration/improvement in profitability. **[35 marks]**

Appendix A: On plagiarism, use of AI, etc.

For this coursework submission you are required to work alone: you should not share code or data or text or images with any other student. You should also make sure that your notebook, code, etc are not viewable or copyable by other students. Working collaboratively with other students, or not preventing other students from copying your work, would be viewed as *unauthorised group-work* which is an offence under the University's regulations on plagiarism and cheating,

Please be reminded that use of ChatGPT or any similar generative-AI system to create text which you submit for this coursework as if it was your own original work is a contravention of the University's regulations on cheating, and will be dealt with as plagiarism. If you are found to have committed plagiarism on this coursework, the most likely result is that your grade for INFT would be set to 0% for this assessment, resulting in a fail. You would then have to re-sit this unit over the summer of 2024, meaning you cannot start work on your individual project until September 2024, thereby delaying your completion of the MSc until December 2024, and the award of your final degree transcript etc would not take place until February or March 2025.

In academic writing, plagiarism is the inclusion of any idea or any language from someone else without giving due credit by citing and referencing that source in your work. This applies if the source is print or electronic, published or unpublished, another student's work, or any other person, or a machine like ChatGPT.

The University's Examination Regulations state that "Any thesis, dissertation, essay, or other course work must be the student's own work and must not contain plagiarised material. Any instance of plagiarism in such coursework will be treated as an offence under these regulations." (Section 3.1).

Section 4 of the University's [Examination Regulations](#) give information on the University's procedures for dealing with cases of plagiarism. More information about plagiarism, and how to avoid it is available from the [Library website](#).

The University's guidance on use of generative AI tools such as ChatGPT is available here: https://www.ole.bris.ac.uk/bbcswebdav/courses/Study_Skills/using-ai-at-university/index.html

Appendix B: Referencing

When you reference papers in your notebook text, you should reference them using a consistent referencing system, such as the [APA referencing style](#), and you should normally cite sources in the text. As a general rule, you should avoid using footnotes to reference. If you include a quote, it should be in quotation marks, and a page number included in the in-text reference. Whilst you should normally avoid larger quotes, if you include them, you should also indent the text by increasing the left-hand margin width for the quoted passage of text. If you cite a paper, book, article, or any other source in your essay, you should also include a full reference to the paper in the reference list at the end of the notebook. Please **do not** list sources in your reference list that you have not cited in the notebook.

References

D. Cliff, (2023), *Recurrence-Plot Visualization and Quantitative Analysis of Long-Term Co-Evolutionary Dynamics in a Simulated Financial Market with ZIP Traders*. Available at SSRN: <https://ssrn.com/abstract=4495631>

D. Cliff, (2023), *Bristol Stock Exchange*. Open-source Python code repository available at <https://github.com/davecliff/BristolStockExchange>

V. Smith (1962), An Experimental Study of Competitive Market Behavior. *The Journal of Political Economy*, 70(2): 111-137. <https://www.jstor.org/stable/1861810>

University marking criteria (postgraduate)

	Fail (<50%)	Pass (50%-59%)	(60%-69%)	(70%+)
Attainment of Learning Outcomes	An inadequate level of knowledge and understanding of the basic concepts addressed by the question.	A reasonable knowledge and understanding of the topic, although more basic than at merit level, and at least of essential reading and material covered in lectures.	A good knowledge and understanding of the topic with clear indication of some reading beyond essential recommendations or material covered in lectures. A 'good' knowledge implies largely an absence of outright mistakes with clear evidence that you have understood the material and are not merely repeating it.	An authoritative knowledge and understanding of the topic gained from wide-ranging reading that goes beyond essential recommendations or material covered in lectures.
Application of Methods	Able to use a proportion of the basic methods and techniques taught.	Able to use most of the methods and techniques taught.	Able to use well a range of methods and techniques to come to conclusions.	Mastery of a wide range of methods and techniques.
Analysis, Comprehension and Synthesis	Inappropriate or irrelevant selection of content.	An ability to use relevant material which is directed to the question.	An ability to use relevant material which is directed to the question.	An ability to manipulate material intelligently and to direct it to the question set.
Technical Mastery	Limited technical competence with major shortcomings in significant areas of the subject.	Some technical competence but with shortcomings in significant areas of the subject.	A good general level of technical competence perhaps with some shortcomings.	A high level of technical competence with very few mistakes of any kind.
Evaluation/Critical Analysis	Little evidence of a critical perspective or of analysis so that the whole remains largely descriptive.	Some evidence of the ability to engage critically with content, although not fully developed.	A good ability to evaluate critically relevant theory and research.	A well-developed ability to engage critically with that material.
Quality of presentation	Poor presentation	Adequate presentation	Very good presentation	Excellent presentation

Note: Because the marking criteria consider a number of dimensions, it is unlikely that a single piece of work fits nicely into all of the descriptions above. For example, a piece of work may have excellent presentation, but due to significant errors, and major deficiencies, the piece of work may still be awarded a fail mark.