1. Agent-Based Modelling

Questions to Answer about ABM

1. Why does ABM provide us with a unique and powerful insight into complex systems?
2. What is ABM and how is it used?
3. What are some simple ABM that we can create?
4. How do I extend an ABM that was created by someone else?
5. How do I create my own ABM?
6. What are the basic components of ABM?
7. How do I analyse the results of an ABM?

8. **How can I tell whether the implemented ABM corresponds to the concept of the model that I developed in words? (Verification.)**

   How can I tell whether the results of my ABM tell me anything about the real world? (Validation)

   How can I make sure that someone else can repeat my results? (Repetition/Replication.)

9. **What are some advanced ways of including data and using output from ABM? What are some of the open research questions in ABM?**

10. **From what computational scientific roots did ABM arise?**

See Wilensky & Rand (2015), available in soft copy from the MIT Press, at `https://mitpress.mit.edu/books/introduction-agent-based-modeling`

What is an ABM?

An agent: an autonomous computational individual or object with particular properties or actions.

An ABM: a form of computational modelling whereby a phenomenon is modelled in terms of agents and their interactions

ABM can allow us to explore, make sense of, and analyse phenomena and scenarios across a wide range of contexts and content domains.

— physical, biological, and social

Possible with cheap. powerful computation and connectivity and large data sets (often at a micro level) → ABM to be constructed.

Advantages of ABM

ABM easier to understand than mathematical representations, because ABM constructed from individual objects with simple rules for their behaviour.

As mentioned earlier, with different levels (micro, meso, macro) ABM allow the “emergence” of macro (or meso) phenomena from the interaction of agents at the micro level.

W&B: emergence is the arising of novel and coherent structures, patterns, and properties through the interactions of multiple distributed elements — with no top-down or centralised control.

ABM enables the modelling of *complex systems*.

Two Sorts of Challenges:

1. **What will the aggregate pattern be when we know how the individual elements behave? — "integrative understanding" (W&B)**
2. **What is the behaviour of the elements that could generate the known or observed aggregate pattern? — "differential understanding" (W&B)**

For the first: simulate the model.

For the second, more difficult: in a sense we seek a model sufficient to generate known aggregate behaviour, or seek an "existence proof."

Sometimes would like to know all such sufficient models, or the necessary set of such models.

Often people find it hard to accept that bottom-up behaviour, and not a top-down controller, can result in the emergent patterns.

< >

2. The NetLogo Platform

All of the following authors have written books which use NetLogo: Uri Wilensky & William Rand, Steven F. Railsback & Volker Grimm, Nigel Gilbert & Klaus G. Troitzsch.

Other ABM platforms exist: RePast, Ascape, MASON, but NetLogo is definitely the best for the beginner and even for proficient modellers: free, well supported, well documented (as part of the package), and with many way so linking to other packages, such as the statistical packages Excel, R, SAS, Mathematica, and Matlab.

Example: 1. Wolf Sheep Predation

Find it in the NetLogo Models Library/Sample Models/Biology

Elements: move, eat, breed, die, grow

Gives insights into the two population numbers and dynamics. Compared with the Lotka-Volterra differential equations.

Example: 2. Fire

Cells given rules about the fire spreading to (usually) adjoining cells.

**Gives insights into fire dynamics.
Compared to the thermal diffusivity equation and the Reynolds fluid flow equation.**

Find it in the NetLogo Models Library/Sample Models/Earth Science

What is ABM?

Many (if not most) phenomena in the world can be effectively modelled with agents, an environment, and a description of agent-agent and agent-environment interactions.

The environment is the landscape on which the agents interact and can be geometric, network-based, or drawn from real-world data.

The interactions between the agents or with the environment can be quite complex, and the interaction behaviours can change with time.

Moreover, the strategies used to decide what action to employ at a particular time or in a particular state can also change.

As a result of these interactions, agents can update their internal state or take additional actions.

Randomness v. Determinism

With ABM it is easy to incorporate randomness into models.

Decisions can be made stochastically, rather than deterministically.

Another advantage of ABM is that agents need not be uniform: the economist's "representative agent" precludes asymmetries: of tastes, or information, of possible actions.

And yet many phenomena (think market exchange) would not occur between identical agents.

Look back at the Boys and Girls model introduces earlier.

When are ABM Best?

Some problems with large numbers of homogeneous agents are better modelled using an aggregate solution such as mean field theory or System Dynamics modelling.

ABM are more useful with heterogenous agents, which allows quite complex. With a very large number of possible agent types, we can specify just a few simple rules of interaction, getting a large repertoire of interactions.

Agents can condition their behaviours, and even their strategies, on past events.

This can hold too in ABM of interactions between agents and the environment.

3. Conway’s Game of Life

See it in the IABM Textbook folder in the NetLogo models library.

In 1970 John Horton Conway created a cellular automaton he called the “Game of Life.”

Played on a large grid of cells, each of which can be “dead” or “alive.”

4 Rules: Every tick, if a cell has fewer than 2 neighbours alive, it dies; if it has more than 3 neighbours alive, it also dies; if it has exactly 2 living neighbours, it remains dead or alive; if it has exactly 3 living neighbours, the cell becomes alive if dead, or dies if alive.

(See this on pages 47 and 48 of W&B, which details building Life using NetLogo.)

Life (cont.)

It has been proved that Conway's Game of Life is universal and can function as a Turing machine.

Cellular automata have been studied in 1, 2, and 3 dimensions.

Evidence of CA-like mechanisms has been found in nature, including the shapes of shells and flowers, the colouraisation and striations of animals, and the structure of other organisms.

Stephen Wolfram has investigated a basic class of 1-dimensional CA, and each cell could only consider the states of its two neighbours. Still, remarkable complexity.

Case 1: Using a GA in NetLogo to Search for Risk Profile

What is the best risk profile for an agent to have in a risky world?

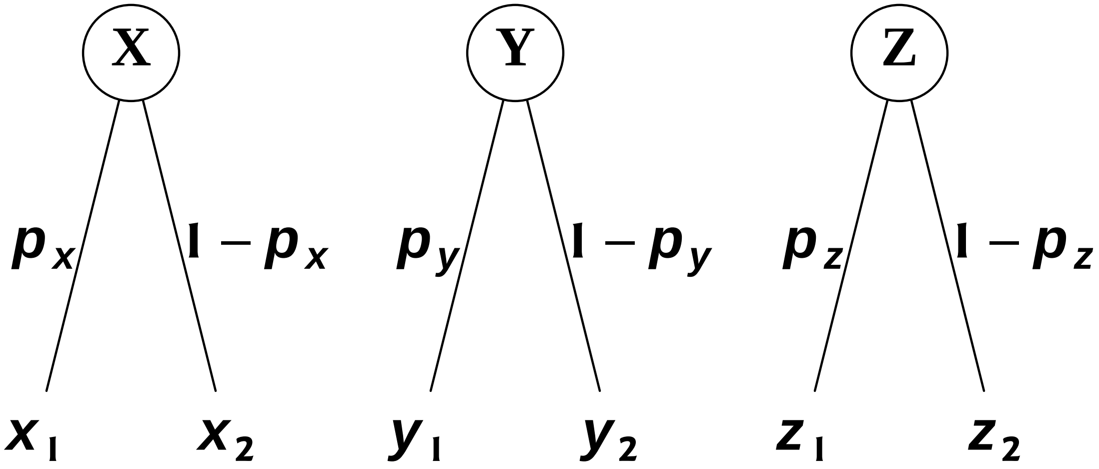
“Best” means that the agent’s average net winnings from choosing successive lotteries is highest: this is risky decision making.

By “risky” is meant that both the possible outcomes and probabilities (which might be subjective) are known.

Informally, it is widely held that in a risky world, with the possibility of the discontinuity of bankruptcy, the most prudent risk profile is *risk aversion.*

A Risky Decision

Three two-prize lotteries with random prizes and probability:



Calculate the three expected utilities:

$$U(X) = p_x U(x_1) + (1 - p_x) U(x_2)$$
$$U(Y) = p_y U(y_1) + (1 - p_y) U(y_2)$$
$$U(Z) = p_z U(z_1) + (1 - p_z) U(z_2)$$

Choose the lottery I with the highest expected utility. Win whichever prize (i_1 or i_2) is realised in that lottery, based in the lottery's probability p_i.

Methodology

We use three kinds of utility function:

1. the wealth-independent exponential function, or CARA;
2. a wealth-dependent function, the CRRA; and
3. the DRP from Prospect Theory.

These reflect actual human decision making.

Then run computer experiments in which each agent successively chooses among three lotteries, and is then awarded with the outcome of the chosen lottery k.

Repetition of this choice by many agents allows us use the Genetic Algorithm (Holland 1992) to search for the best risk profile, where “best” means the highest average payoff when choosing among lotteries.

CARA Utility Functions

The exponential CARA utility function is

$$U(x) = 1 - e^{-\gamma x}, \tag{4}$$

where $U(0) = 0$ and $U(\infty) = 1$, and where γ is the *risk aversion coefficient*:

$$\gamma \equiv -\frac{U''(x)}{U'(x)}. \tag{5}$$

Sign of γ	Risk profile	Curvature
$\gamma = 0$	risk neutral	$U''(x) = 0$
$\gamma > 0$	risk averse	$U''(x) < 0$
$\gamma < 0$	risk preferring	$U''(x) > 0$

CRRA Utility Functions

The Constant Elasticity of Substitution (CES) CRRA utility function:

$$U(w) = \frac{w^{1-\rho}}{1-\rho}, \quad w > 0, \tag{6}$$

where w is agent's wealth, and ρ is the Arrow-Pratt measure of relative risk aversion (RRA):

$$\rho(w) = -w \frac{U''(w)}{U'(w)} = w\gamma \tag{7}$$

This introduces wealth w into the agent's risk preferences, so that lower wealth can be associated with higher risk aversion. The coefficient γ is as in (5).

As $\rho \rightarrow 1$, (6) becomes logarithmic: $u(w) = \ln(w)$, risk averse.

With $w > 0$, $\rho > 0$ is equivalent to *risk averse*, while $\rho < 0$ is equivalent to *risk preferring*; $\rho = 0$: *risk neutral*.

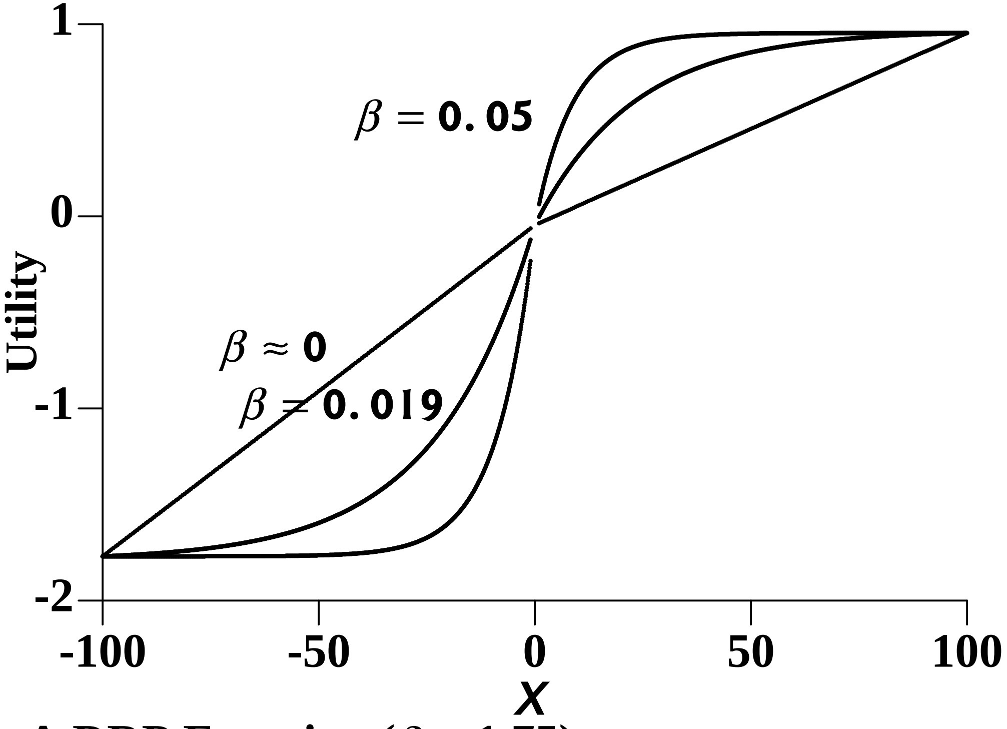
The DRP function

This function is inspired by Prospect Theory (Kahnemann & Tversky 1979):

$$U(X) = \frac{1 - e^{-\beta X}}{1 - e^{-100\beta}}, \qquad 0 \le X \le 100 \tag{8}$$

$$U(X) = -\delta \frac{1 - e^{\beta X}}{1 - e^{-100\beta}}, \qquad -100 \le X \le 0. \tag{9}$$

$\beta > 0$ models the curvature of the function, and $\delta \ge 1$, the asymmetry associated with losses. The DRP function is not wealth-independent. DRP exhibits the S-shaped asymmetric function of Prospect Theory. It exhibits risk seeking (loss aversion) when X is negative with respect to the reference point, $X = 0$, and risk aversion when X is positive.



A DRP Function (δ = 1.75).

As $\delta \to 1$ and $\beta \to 0$, the value function asymptotes to a linear, risk-neutral function.

4. The Simulations

Each lottery is randomly constructed: the two payoffs (“prizes”) are randomly chosen in the interval [−$100, +$100], and the probability is also chosen randomly.

Each agent calculates the expected utility of each of the three lotteries, using its utility function (a function of its γ or ρ/W or (β, δ)), and chooses the lottery k with the highest expected utility. To do this, agents know the prizes and probabilities of all three lotteries.

Then the actual (simulated) outcome of the chosen lottery k is randomly realised, using its probability. The winnings of agent is incremented accordingly. Each agent chooses 1000 lotteries.

Use an implementation of the GA (Gilbert 2005) in NetLogo to search for the best risk profile.

`http://www.agsm.edu.au/bobm/teaching/SimSS/NetLogo4-models/DRA-CRRA-EU-GM-revH-3l2p.html`

4. Monte Carlo — Probabilistic Sampling

1. **Run two simulation models for n runs ($n > 50$).**
2. **Save the relevant output data from each MC run. This provides a distribution for each (stochastic) model, A and B.**
3. **Other models generate (via MC) other distributions, or we could have historical data to compare with.
e.g. cumulative winnings for agents with degrees of risk preference choosing lotteries.**
4. **Null hypothesis: the two means of the outputs come from the same distribution: H_0: $\mu_A - \mu_B = 0$.**
5. **Statistical testing for the null hypothesis. Use *confidence intervals*: the probability that the difference between the two means is *not* random.**

The Results

Mean results across n Monte Carlo runs:

Model	Variable/s	Mean	p-value	Fitness	n MC runs
CARA	γ	0.0064	≈ 0	$37.65	55
CRRA	ρ	19.21	0.30	$29.40	109
DRP(a)	δ	1.466	≈ 0	$37.67	55
DRP(b)	β	0.0047	≈ 0	$38.81	102
DRP(c)	δ &	1.256	≈ 0	$37.72	54
	β	0.0072	≈ 0		

In DRP(a) $\beta = 0$ and search for δ.
In DRP(b) $\delta = 1$ and search for β.
In DRP(c) search for β and δ jointly.

In all cases the null hypothesis is risk neutrality ($\gamma = \rho = \beta = 0$ and $\delta = 1$).

CRRA is the only case here where the null is not rejected.

But the Fitness of DRP(b) > DRP(c) ($38.81 > $37.72): the GA has not converged!

This suggests two further models:

4. **Linear, risk-neutral**
5. **Clairvoyant**

Model DRP(c) is unconstrained, and so can find $\delta = 1$ (with the higher Fitness) but has not yet done so: the GA has not (yet) converged.

Strongly suggests that risk neutral (with $\delta = 1$ and $\beta = 0$) will have a higher Fitness than $38.81.

Short cut: introduce the Linear, risk-neutral model.

The Clairvoyant model has the highest possible Fitness, but requires clairvoyance or perfect foresight.

All Models Compared

Model	Variable/s	Mean	*p*-value	Fitness	*n* MC runs
CARA	γ	0.0064	≈ 0	\$37.65	55
CRRA	ρ	19.21	0.30	\$29.40	109
DRP(a)	δ	1.466	≈ 0	\$37.67	55
DRP(b)	β	0.0047	≈ 0	\$38.81	102
DRP(c)	δ &	1.256	≈ 0		
	β	0.0072	≈ 0	\$37.72	54
Linear				\$39.19	102
Clairvoyant				\$50.25	50

So the hunch from the GA results is correct: Linear has the highest Fitness.

To confirm: the data from DRP(b) and Linear are statistically distinct, using a Wilcoxen Two-Sample Test.

Case 2: Comparing Sets of Time-Series

Q: how can we measure the degree of similarity of two sets of time-series?

***One*: the historical record of the rivalrous dance among the sellers in an oligopoly, while**

***The other*: the output from a (agent-based) simulation model of the market, where each seller agent prices this week as a function of the state of the market last week (or earlier).**

Q: how can we *output validate* our model against history?

Or: how can we derive a *degree of confidence* in the model output?

The Issue: Heterogenous Agents and Time-series Prices

Two reasons to compare such model output against history:

1. **To choose better parameter values, to "calibrate" or (more formally) "estimate" the model against the historical record.**
2. **To measure how closely the output reflects history, to *validate* the model.**

We are interested in the second, having used machine learning (the Genetic Algorithm) to derive the model parameters in order to improve each agent's weekly profits (instead of fitting to history) in our agent-based model.

Figure 1 shows historical data from a U.S. supermarket chain's sales of (heterogeneous) brands of sealed, ground coffee, by week in one city (Midgley et al. 1997).

This simulation (which also uses the Genetic Algorithm) was written in C, not NetLogo, because we started this research before NetLogo was available.

Historical Data: Market Prices and Volumes

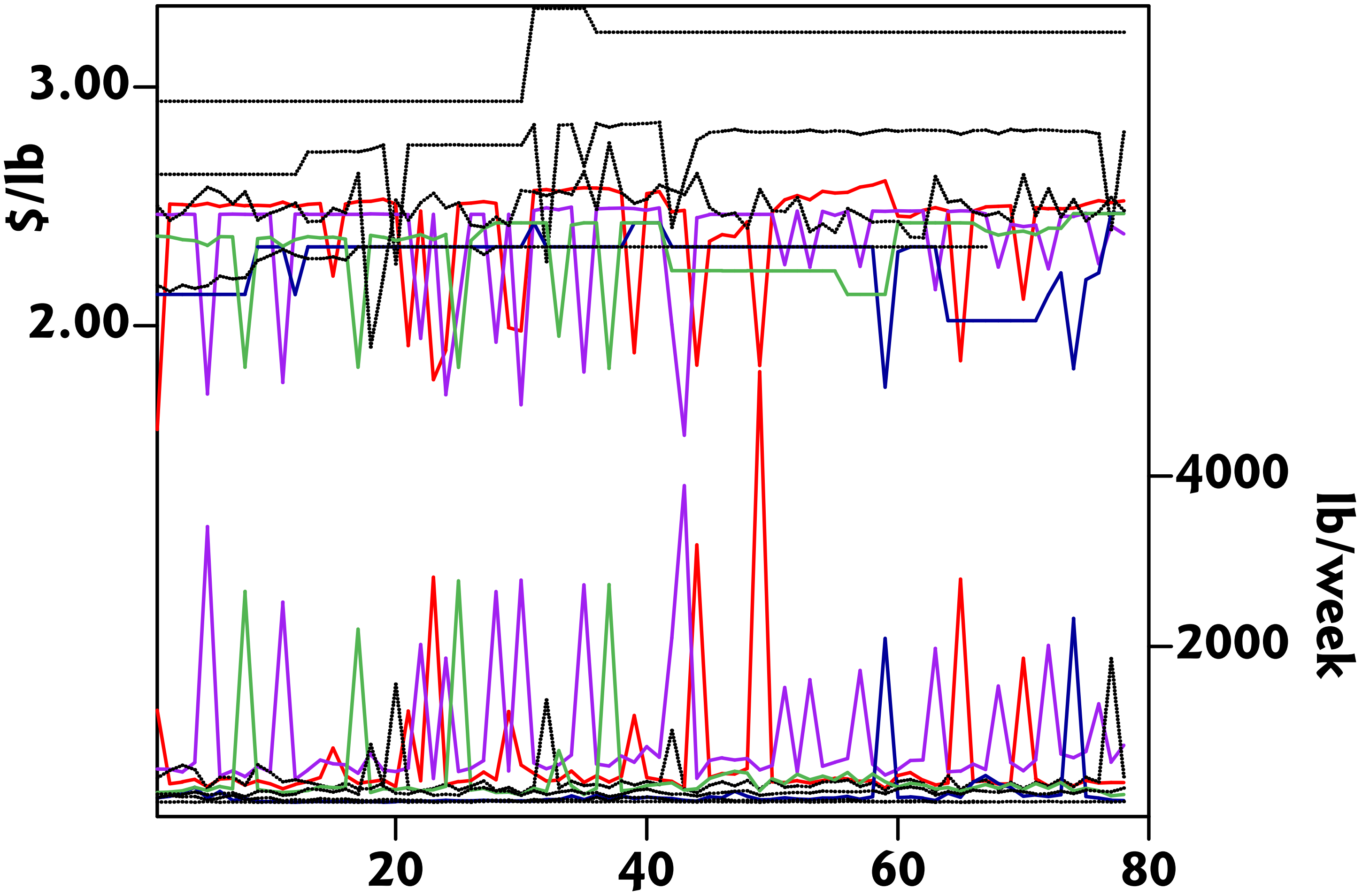


Figure 1: **Weekly Sales and Prices (Source: Midgley et al. 1997)**

Stylised Facts of the Market Behaviour

- Much movement in prices and quantities of four brands — a rivalrous dance.
- Pattern: high price (and low quantity) punctuated by low price (and high quantity).
- Another four brands: stable prices and quantities

Questions:
What is the cause of these patterns?

— shifts in brand demand?

— reactions by brands?

— actions by the supermarket chain?

— unobserved marketing actions?

Explanations?

Interactions of profit-maximising agents, plus external or internal factors → via a model → behaviour

Similar (qualitatively or quantitatively) to the brands' behaviours of pricing and sales.

Note: assuming profit-maximising (or purposeful) agents means that we are not simply curve-fitting or description using D.E.s. Going beyond the rivalrous dance.

Dichotomous Price Partitioning of the Historical Data

To handle the curse of dimensionality.

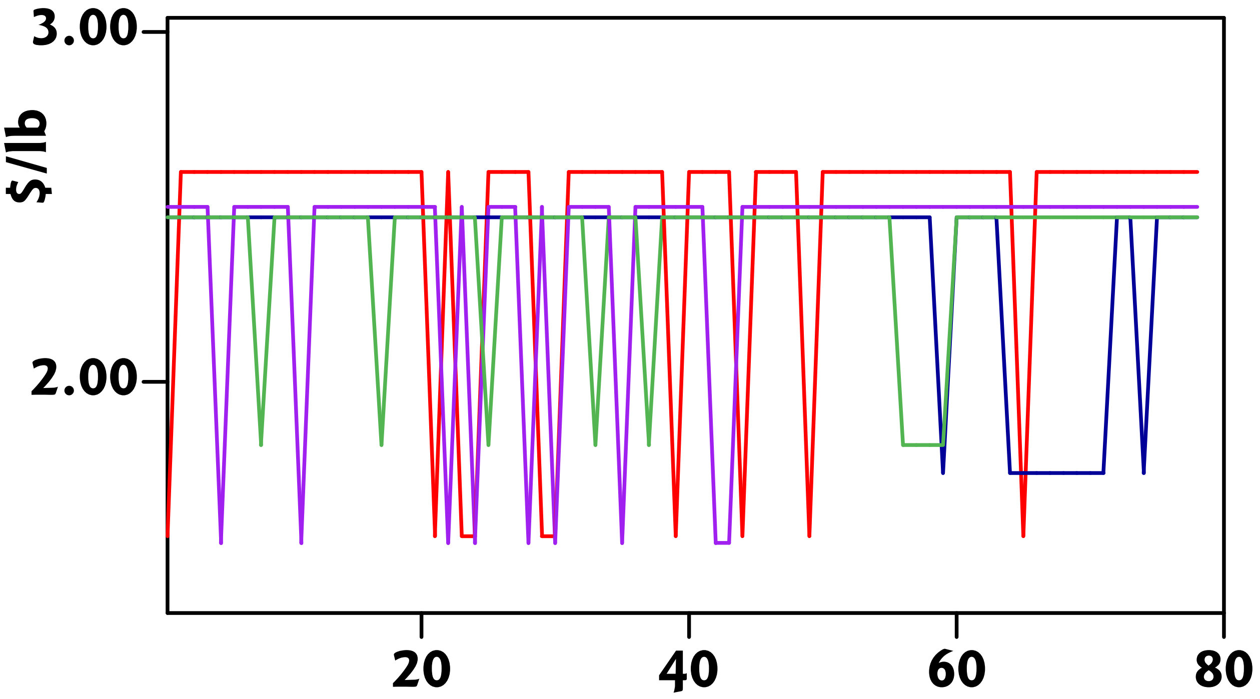


Figure 2: **Partitioned Historical Weekly Prices of the Four Brands**

A Model of Strategic Interaction

We assume that the price P_{bw} of brand b in week w is a function of the state of the market M_w at week w, where M_w in turn is the product of the weekly prices S_w of all brands over several weeks:

$$P_{bw} = f_b(M_w) = f_b(S_w \times S_{w-1} \times S_{w-2} \cdots)$$

Earlier in the research program undertaken with David Midgley et al., we used the Genetic Algorithm to search for "better" (i.e. more profitable) brand-specific mappings, f_b, from market state to pricing action.

And derived the parameters of the model, and derived its simulated behaviour, as time-series patterns (below).

The State Similarity Measure (SSM)

The SSM (Marks 2013) derives the distance between two sets of time-series, by calculating the sum of absolute differences in observed window states between the two set, so what?

First, the greater the sum, the more distant the two sets of time-series.

Second, we can calculate the maximum size of the summed difference: zero intersection between the two sets (no states in common) implies a measure of $2 \times S$ where S is the number of possible window states, from the data.

Third, we can derive some statistics to show that any pair of sets in not likely to include random series (below).

Example of a Simulated Oligopoly (Marks et al. 1995)

Simulating rivalry between the three asymmetric brands: 1, 2, and 5, Folgers, Maxwell House, and Chock Full O Nuts.

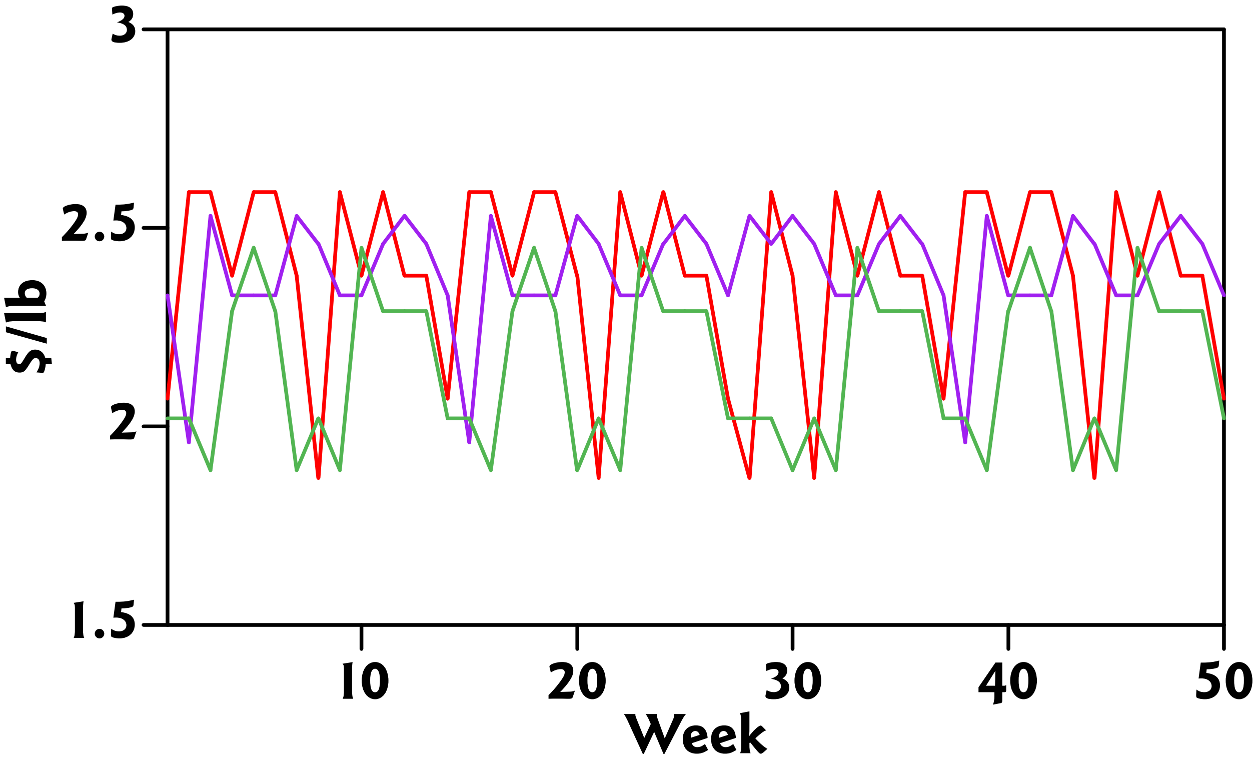


Figure 3: **Example of a Simulated Oligopoly (Marks et al. 1995)**

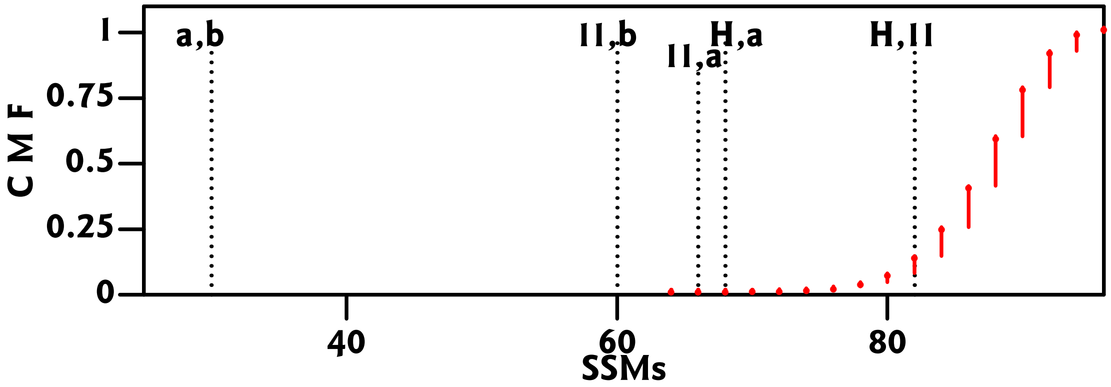
Distances Between History and Three Runs (Brands 1, 2, 5)

	History	Run 11	Run 26a	Run 26b
History	0	82*	68	68
Run 11	82*	0	66	60
Run 26a	68	66	0	30
Run 26b	68	60	30	0

Table: Distances Between History and Three Runs (Brands 1, 2, 5)
(* : cannot reject the null at the 5% level)

Here, S, the maximum number of states = 48, so the maximum distance apart is 96. The three Runs are closer to each other than to the Historical Data; Runs 26a and 26b are very close, only 30/96 = 31.25% apart.

Testing for Randomness



The red lines are the CMF of pairs of sets of random series (3 series, 48 observations) from 100,000 Monte Carlo parameter bootstraps.

The one-sided confidence interval at 1% corresponds to a SSM of 76, and at 5% 80.

Cannot reject the null hypothesis (random sets) for Historical data and Run 11; reject the null (random) hypothesis for all other pairs.

Conclusions — the State Similarity Measure

This measure, *the State Similarity Measure* (SSM), is sufficient to allow us to put a number on the degree of similarity between two sets of time-series which embody dynamic responses.

Such a metric is necessary for scoring the distance between any two such sets, which previously was unavailable.

Here, the SSM has been developed to allow us to measure the extent to which a simulation model that has been chosen on some other criterion (e.g. weekly profitability) is similar to historical sets of time-series.

The SSM will also allow us to measure the distance between any two sets of time-series and so to estimate the parameters, or to help calibrate a model against history.

Conclusion

In a single morning it is not possible to train you in programming ABM, even using NetLogo.

Nonetheless, I hope I have given you some motivation and some pointers in how to start playing with ABM using NetLogo.

Useful References

[1] **Robert Axelrod: Advancing the art of simulation in the social sciences (revised 2005) reprinted in the *Handbook of Research on Nature-Inspired Computing for Economics and Management,* Jean-Philippe Rennard (Ed.), Hersey, PA: Idea Group, 2006.**
http://www-personal.umich.edu/~axe/research/AdvancingArtSim2005.pdf

[2] **Robert Axelrod and Leigh Tesfatsion, On-Line Guide for Newcomers to Agent-Based Modeling in the Social Sciences**
http://www2.econ.iastate.edu/tesfatsi/abmread.htm

[3] **Nigel Gilbert, *Agent-Based Models (Quantitative Applications in the Social Sciences)* Sage, 2008.**

[4] **Nigel Gilbert & Klaus G. Troitzsch, *Simulation for the Social Scientist,* Open Univ. Press, Second Ed., 2005.**
http://cress.soc.surrey.ac.uk/s4ss/

[5] **John H. Miller & Scott E. Page, *Complex Adaptive Systems: An Introduction to Computational Models of Social Life,* Princeton Uni. Press, 2007**
http://press.princeton.edu/TOCs/c8429.html

[6] **Steven F. Railsback & Volker Grimm, *Agent-Based and Individual-Based Modeling,* Princeton U.P., 2012.**

[7] **Uri Wilensky & William Rand, *An Introduction to Agent-Based Modeling: Modeling Natural, Social, and Engineered Complex Systems with NetLogo* MIT Press, 2015**
https://ccl.northwestern.edu/netlogo/
https://mitpress.mit.edu/books/introduction-agent-based-modeling