Spatial microsimulation: a practical introduction

Lovelace, Robin
r.lovelace@leeds.ac.uk

June 17, 2014

Contents

Chapter 1

Preface

Chapter 2

Introduction

Spatial microsimulation is shrouded by an unnecessary mystery, and this is worsened by much of the literature on the subject. Most academic papers and even some textbooks about spatial microsimulation lack reproducible examples. In today's age of fast Internet connections, open access datasets and free software, things need not be this way. One could argue that the lack of reproducibility in spatial microsimulation is damaging to the growth and credibility of the field.

To those doubting the value of this practical approach, I ask the following questions: If only a small community of researchers hold the majority of the code needed to perform spatial microsimulation, how can the technique spread rapidly to other applications? If every PhD student undertaking spatial microsimulation must start from scratch, how are the methods going to be refined and improved in a systematic fashion? Most importantly, if the results of most spatial microsimulation research is not reproducible — as is currently the case — how can we trust them?

This final point is critical not only to spatial microsimulation but the disciplines of which it is part. Reproducibility is a prerequisite of falsifiability — if the method underlying a result cannot be replicated by others, how can the finding possibly be falsified? Moreover, the argument continues, any knowledge or theory can only claim to enter the realm of 'science' if it cannot be falsified. Because it is impossible to prove any proposition in every and all cases, the only reliable test we can apply is whether or not it can be *disproved*: falsified via a contradictory finding. This is how scientific progress is made (Popper, 1959). By writing non-reproducible research, researchers may inadvertently damage the disciplines in which they work.

Despite and because of these philosophical antecedents, the course is unashamedly practical. The aim is simple: to provide an accessible yet deep foundation in spatial microsimulation. This involves both *understanding* and *implementation* of the technique. Following the 'learning by doing' ethic, the former can best be attained through the latter: spatial microsimulation need not be an abstract process that one simply reads about. It can be practical tool used by anyone with the know-how. As Kabacoff (2011, xxii) put it regarding R, "the best way to learn is to experiment" and the same applies to spatial microsimulation.

Software decisions have a major impact on the model's flexibility, efficiency, reproducibility and ease of coding. Clarke and Holm (1987, p. 153) observed that "little attention is paid to the choice of programming language used." This appears to be as true now as it was then: software is rarely discussed in spatial microsimulation papers. In my own spatial microsimulation research, a conscious decision was made early on to use R, with impacts on model features, analysis and even design. It is thus worth understanding the tool a little before we begin to use it. The theory is discussed in chapter 5

The world is awash with computer programming languages and many of these are general purpose and 'Turing complete', meaning they could, with sufficient effort, perform spatial microsimulation. So why would one chose R? The most important criteria of evaluation include flexibility, speed of processing and, most importantly, ease and speed of writing code. R excels in each of these areas, especially the final one: it is possible to say a lot in R in few lines of code. Further are provided by Matloff (2011):

- "a public-domain implementation of the widely-regarded S statistical language; R/S is the de facto standard among professional statisticians
- comparable, and often superior, in power to commercial products in most senses
- available for Windows, Macs, Linux
- in addition to enabling statistical operations, it's a general programming language, so that you can automate your analyses and create new functions
- object-oriented and functional programming structure
- your data sets are saved between sessions, so you don't have to reload each time
- open-software nature means its easy to get help from the user community"

2.1 Learning the R language

Having learned a little about *why* R is a good tool for the job, it is time to think about *how* R should be used. The most useful advice I received on the subject is to think of R not as a series of isolated commands, but as an interconnected *language*, in the fullest sense of the word. R is not only a system by which instructions can be sent to a computer to process data; R provides a way of expressing oneself and explaining ideas to other human beings. Critical to its role as a language is R's unique *syntax*, which allows it to express relatively complex expressions efficently, with a small number of keystrokes.

The unusually concise nature of R code is not an accident. It was planned to be this way from the outset by its instigators, Robert Gentleman and Ross Ihaka, who thought carefully about syntax from the outset: "the syntax of a language is important because it determines the way that users of the language express themselves" (Ihaka and Gentleman, 2014, p. 300).

2.2 Typographic conventions

2.3 An overview of the book

Chapter 3

What is spatial microsimulation?

3.1 What spatial microsimulation is not

3.2 A method for generating spatial microdata

3.3 An approach to modelling ecological processes

3.4 A tool for combining insights from multiple scales

Chapter 4

Applications

4.1 Updating cross-tabulated census data

4.2 Economic forecasting

4.3 Small area estimation

4.4 Transport modelling

4.5 Dynamic spatial microsimulation

4.6 An input into agent based models

Chapter 5

Spatial microsimulation in theory

5.1 Iterative Proportional Fitting

5.2 Combinatorial optimisation

5.3 Multilevell modelling

Chapter 6

Spatial microsimulation with R

6.1 Loading and cleaning input data

6.2 Comparing individual and aggregate data

6.3 Reweighting using IPF

6.4 Combinatorial optimisation

6.5 Integerisation

Chapter 7

Customising code: a worked example with CakeMap

7.1 Preparing the input data

7.2 Performing IPF on CakeMap data

7.3 Integerisation

7.4 Validation

7.5 Visualisations

7.6 Analysis and interpretation

Chapter 8

Additional tools and techniques

8.1 The Flexible Modelling Framework (FMF)

8.2 Allocation of home-work locations

8.3 A spatial interaction model with individual agents

8.4 Spatial microdata: an input into agent based models

Chapter 9

Conclusions and a peak into the future

Chapter 10

Bibliography

Bibliography

Martin Clarke and Einar Holm. Microsimulation methods in spatial analysis and planning. *Geografiska Annaler. Series B. Human Geography*, 69(2):145–164, 1987. URL `http://www.jstor.org/stable/10.2307/490448`.

Ross Ihaka and Robert Gentleman. R: A Language for Data Analysis and Graphics. *Journal of Computational and Graphical Statistics*, 5(3):299–314, 2014.

Robert Kabacoff. *R in Action*. Manning Publications Co., 2011.

N Matloff. *The Art of R Programming*. No Starch Press, 2011. URL `http://books.google.com/books?hl=en&lr=&id=4jL3sfmP1l8C&oi=fnd&pg=PR5&dq=The+art+of+R+programming&ots=0SldF2sJGD&sig=oEsmrxEf66htv11kpoRRCFenDgY`.

Karl Raimund Popper. *The Logic of scientific discovery: Karl R. Popper*. Hutchinson, 1959. URL `http://books.google.com/books?id=MdvaSAAACAAJ&pgis=1`.