Dynamic Optimisation Microsimulation Using FMF: Practical

Dr Andy Evans

1) First, download the latest Flexible Modelling Framework (FMF) software from:

<https://github.com/MassAtLeeds/software/releases>

and unzip it somewhere.

2) In the newly unzipped files and directories you should find a handbooks-and-practicals directory. In this there’s a Microsimulation directory. This contains the software handbook: microsimulationHandbook.pdf plus a set of test-data directories.

Work though up to (but not including) Part 4 of the handbook.

3) For this practical, we’ll use the larger test-data dataset. If you look in the test-data directory, and then the “large” directory, you should find the following files:

age.csv

sex.csv

crosstab.csv

microdata.csv

Plus the data licence details: DataLicenceReadMe.txt

Here, we’ll assume these are in a directory on your machine c:\FMF\handbooks-and-practicals\Microsimulation\test-data\large\ -- but it will depend where you unzipped the files.

The first three files are census data for Wards in West Yorkshire. The first contains ward IDs, plus the population numbers broken down by age. The second is the same, but by sex. The third is by age and sex cross-tabulated.

The microdata file is individual-level data, crowdsourced and anonymised. It includes data on the individuals’ age and sex, along with the number of cars they own and their NSSEC8 sociodemographic grouping (<http://www.ons.gov.uk/ons/guide-method/classifications/current-standard-classifications/soc2010/soc2010-volume-3-ns-sec--rebased-on-soc2010--user-manual/index.html#7>).

4) We’ll start by running through the system once, with c:\FMF\handbooks-and-practicals\Microsimulation\test-data\large\ as our Data Source, and age.csv, sex.csv, and microdata.csv as our registered files. Our “Population Data” is the microdata.csv, and we’ll link it to sex.csv and age.csv as our linked data files. We’ll use the “Age” and “Sex” columns in our microdata.csv file, but not the “SexAge” column (yet).

The system will build us a file, each line of which will be a person ID and the ward ID they live in (the ward ID will actually be the first column, and the person ID second). The overall statistical distribution of the people will match the numbers in each ward in the age and sex files.

Run through the rest of the user guide, utilising these files. First, though, some hints:

a) To make a new microsimulation (or validation), click on the Micorsimulation menu, and choose “New microsimulation model” (or “New validation setup”).

b) Remember when registering the data files, to check the “Headers in first row” checkbox.

c) Remember on the linked table to select the top row and right-click on it to fix it as the zone IDs.

d) Remember to save the links after you’ve set them up – the system will freeze until you do.

e) When you set up a validation setup, you need to drag in and link the same files as before. However, you also need to drag your results population into the box labelled “Population to evaluate <<results table>>”. Once you’ve done this, click on the table in the Data Sources tree to expand the table in the tree (not open it as a window), and then drag the “ZoneID” column from the tree into the box labelled “Zone ID Field <<not set>>” and the “ID” column into “Person ID Field <<not set>>”. If you don’t set up this population to test, the system will freeze waiting for you to do it.

5) Once you’ve run through the user guide, you should find that it generates a very good population list. Validation should suggest a 100% match with the statistics. This is in part because we are matching our two statistics separately. There’s nothing to stop you matching against cross-tabulated data, so the population has to not only match the age and sex statistics of an area, but also make sure that the right aged people are the right sex. You can try this by linking the “SexAge” column in the microdata.csv against the crosstab.csv file.

6) In actual fact, as there is a wide variety of people in the microdata, the crosstabulated data should still produce a perfect match. Adding more and more single and cross-tabulated datasets will eventually reduce the match as you start to get combinations that can’t be found in the microdata but which are found in the statistics for an area. For example, if we included sociodemographic groupings we might have a slot in a ward that could only be filled by a 28 year old female unemployed person, but we might not have such a person in our microdata. The system would pick either a 28 year old female manager or a 28 year old male unemployed person, and accept a level of error. As a worse example, our microdata includes “Cars” as a variable, and all our microdata individuals have cars. However, there are plenty of people in our wards that don’t have cars. Any microsimulation using our car ownership data as a variable will be wrong to an error level corresponding at least to the level of no-car people in the population.

7) Finally, if we combine our synthetic population with the appropriate microdata by linking each microdata ID in the synthetic population with its data in the microdata.csv, we can predict variables in the real population. This can be done with standard database software, creating a table link, or with this script:

While the synthetic individuals will be a good match on the variables used to constrain the model (the linked tables above), other additional variables attached to each microdata individual may also be a good representation of the real population, however, this will depend on the correlation between the constraining variables and the variables carried over with the microdata. This needs some thought. It may be, for example, that if one has age, socio-economic group, and education, one might make a good stab at predicting car ownership, however, it is equally possible that it would make a poor prediction of marital status. Equally, including ages 20-70 might strengthen the predictability of car ownership, whereas including more data but in different classes, ages 0-30 and 60-100 for example, might weaken the relationship.

Microsimulation with non-crosstabulated tables essentially gives the same kind of quality of result as regression for carried-over variables, with the single (but important) advantage that the data is in a disaggregated form which can be reaggregated to different areas. However, with cross-tabulation, it leverages the correlation between the two cross-tabulated variables to improve the accuracy of predicting the third variable. If there is no relationship between sex and newspaper readership and between age and newspaper readership, but every 23 year old women reads the Guardian and every 23 year old man reads the Express, the model will accurately predict the newspaper readership for these groups. In a sense, then, it gives a prediction based on very localised relationships in variable space. This means it gives a fairly good ‘best guess’, but it is equally hard to quantify the quality of the relationship for any given variable where cross-tabulated tables are used.

It seems unlikely, for example, that given age and sex there will be a strong prediction of either car ownership or socioeconomic groups across the board. However, if we added education, employment status, and long term illness, it might be more likely we could reasonably predict both. At the end of the day, a linear regression of all variables against a predicted variable will give you the worse possible performance that the synthetic system will match, and it will, hopefully, do better if tables are crosstabulated.