A DEMOGRAPHIC MICROSIMULATION MODEL USING NETLOGO

Tony Lawson tlawso@essex.ac.uk

This paper has been submitted to the International Journal of Microsimulation.

Department of Sociology
University of Essex

Abstract:

Dynamic microsimulation modelling has become a well-established method of forecasting demographic change. This has led to important applications in areas such as estimating future tax revenues and anticipating pension and health costs. These models are complex and costly to implement and this may be one factor that has hampered the more widespread application of microsimulation modelling. This paper reports a study of the feasibility of using a high level scripting language (Netlogo) to implement a dynamic microsimulation model using an initial population drawn from the 1991 British Household Panel survey. The model was validated by running it for a number of simulated years (1991-2005) and then comparing the predicted population against the actual population observed in the BHPS. It was found that NetLogo made a suitable platform for developing a dynamic microsimulation model and the in-built features for manipulating the collection of agents and the graphical user interface have the potential for simplifying some aspects of developing a dynamic microsimulation model.

About CRESI:

Based in the UK's leading Sociology Department, the Centre for Research in Economic Sociology and Innovation (CRESI) is the first UK centre for research in economic sociology. With a clear focus on innovation, our research programmes highlight contemporary and historical processes of socio-economic transformation.

Table of Contents

1 INTRODUCTION

Microsimulation is a powerful and increasingly popular method of representing the development of a population through time. Micro-units, which typically represent individuals, households or firms, can be in one of a number of states. They change state by means of a transition probability from one time period to the next. By simulating each entity individually, the results can be aggregated in a number of ways that do not have to be specified in advance. Multiple levels of organisation can be represented in the same model and this gives microsimulation the capability of modelling the emergence of novel or unexpected phenomena between levels. In addition, microsimulation is well suited to representing the non-linear relationships that are such a feature of the social world.

In a history going back to the late 1950s, microsimulation models have been developed to forecast pensions (Hancock et al. 1992), healthcare requirements (Hancock, Juarez-Garcia et al. 2006) and tax revenue (Ettlinger & O'Hare 1996), decades into the future. As such, microsimulation modelling already has had an important impact on policy formulation at a high level.

It is possible to draw a distinction between static and dynamic microsimulation. In a static model, the effect of a policy change such as a variation in the rate of tax is calculated for each unit. After this, the results may be aggregated to determine the effects of the change on different sectors of the population. In a dynamic model, the initial population is repeatedly advanced over time according to events such as births, deaths, partnership formation and dissolution. This 'ageing' of the population can take into account changes in the structure of the population as well as the behavioural response of members of the population to a policy intervention. This paper is written in the context of dynamic microsimulation.

Despite its advantages, microsimulation modelling is beset by a number of problems. Harding (2007) describes a range of difficulties that present themselves in the construction and use of microsimulation models. These include budgeting and project management, collaboration between modellers, obtaining high quality data, user friendliness and adding behaviour to the microsimulation units. One factor that underlies some of these difficulties is the complexity of the software necessary to implement the model. There is now a range of software toolkits available that were developed in the related discipline of agent based modelling. This paper describes the construction of a demographic microsimulation model using one of these toolkits, NetLogo. The paper begins with an overview of microsimulation, as background to a discussion of the similarities with agent based modelling. It then reviews some of the currently available agent based modelling approaches. These range from native language implementations in C++, through frameworks of classes written in Java to scripting languages like NetLogo. Following this, the paper describes the design, construction and validation of a microsimulation model using NetLogo that projects the demographic characteristics of the UK population over time. The paper concludes with an appraisal of the potential for agent based modelling toolkits to form the basis of other large-scale microsimulation models.

2 MICROSIMULATION

Dynamic microsimulation originates from a 1957 paper by Guy Orcutt with the intriguing title 'A new type of socio-economic system' (Orcutt 1957). The problem, as he sees it, in using models based on aggregated data is that inevitably, information is lost in the process of aggregation. In particular, this is information about the distribution of the population being studied. For example, an indefinite number of age distributions could give rise to the same mean population age. The solution he suggests is to create a model 'of various sorts of interacting units which receive inputs and generate outputs'. The inputs he defines as 'anything which enters into, acts upon, or is taken account of, by the unit'. Inputs can be things like age, income or rate of inflation. Outputs are 'anything which stems from, or is generated by, the unit'. Examples of outputs include expressions of opinion, marriage, birth of a child and death. The heart of the unit that generates outputs from inputs is what Orcutt calls the 'operating characteristics'. These can be implemented in a variety of ways including

equations, graphs or tables. Typically, the operating characteristic works in a probabilistic manner and this has the effect that identical units with the same inputs may produce different outputs. This can be implemented by selecting the output from a probability distribution. The strength of this method is that although the behaviours of individual units cannot be predicted, the relative proportions of responses mirror the target population as long as the probabilities are correctly estimated. Orcutt proposed an illustrative model that included birth and death as well as marriage and dissolution. In 1961, he implemented the model with the help of some of his students (Orcutt 1961) and in the 1970s, developed a more sophisticated version called DYNASIM2 (Orcutt et al. 1976) that was used for several years to model the financial situation of retired people in the United States. Dynasim3 represents an updated version where the transition probabilities were re-modelled using new data sources.

The idea of microsimulation modelling spread internationally and in Norway, MOSART was developed to model public expenditure (Fredriksen 1998) and in France, DESTINIE was created to project pension requirements (INSEE 1999). In the UK, a range of models has been developed to assist in policy formation. These include PENSIM, which has been used in pension forecasting (Hancock et al. 1992), CARESIM for anticipating demand for healthcare (Zaidi & Rake 2001) and SAGE (Simulating Social Policy in an Ageing Society) (Cheesbrough & Scott 2003) which provides an excellent illustration of the complexity entailed in developing a large dynamic microsimulation model.

3 IMPLEMENTING A MICROSIMULATION MODEL – SAGE AS AN EXAMPLAR

SAGE was initiated in 1999 at the London School of Economics in collaboration with Kings College London. Its base dataset comprises a 1% Sample of Anonymised Records (SAR) from the 1991 UK Census. This includes details of 216,000 households and over 500,000 individuals. Updates to the simulation are made based on annual transition probabilities. The modular design is run by processing mortality first. The fertility module is processed next, then partnership dissolution. Following this, the cohabitation and marriage modules are run.

The probabilities for mortality were obtained from the Office for National Statistics (ONS) life tables according to age, sex and social class.

Factors predicting fertility were obtained from analysis of the British Household Panel Survey (BHPS). This is a large-scale longitudinal survey of the British population covering over 5000 households and involving more than 10,000 individual participants. It is conducted annually and has been in operation since 1991. The longitudinal nature allows researchers to follow individuals between years to determine the probability of various outcomes. Separate logistic regression models were developed for partnered and unpartnered women. For married or cohabiting women, the duration of the current partnership was taken into account as well as previous partnerships, age and number of children. For unpartnered women, age and present children were the significant factors. Once the probability of birth has been calculated, the probability of that birth being multiple is assigned based on the age of the mother. The probability of a child being male is 0.512 according to data derived from the ONS.

Probabilities of separation are modelled for married and cohabiting couples. The process is female dominated in the sense that transition rates are calculated for the female and this determines whether the associated male remains in a partnership. Logistic models were used to calculate transition probabilities taking into account the woman's age at the formation of the cohabitation, the duration of the cohabitation, whether the woman has previously been married and whether there was a birth in the following year (known from past data). For cohabiting couples, whether either partner is a full time student was taken into consideration. For married couples, the age difference between the partners was significant. On separation, 90% of children under 12 years old accompany the mother while 84% of those aged 12 and over do so.

Partnership formation is again female dominated and applies to women aged 16 to 69 who were not widowed in the current year's mortality module. Separate logistic regression models select which women will marry or cohabit. The cohabitation module runs first. Age of the woman, recent marital separation, participation in full time education and birth in the next year determine the probability of the woman beginning a period of cohabitation during the current year. In the marriage module, age, birth in the next year, current cohabitation, any

previous marriage and participation in full time education are taken into account.

Once the woman has been selected for marriage or cohabitation, an appropriate male must be found. This is done by first creating a list of potential partners by random selection. Each male in this list is assigned a probability of becoming the woman's partner according to the age difference relative to the woman, both partner's educational qualifications, previous marriage and the combination of social classes.

Continuing into higher education is modelled by an annual probability of leaving education after the age of 16. By the age of 22, everyone remaining is assumed to gain a higher qualification and at that point, their education is completed. SAGE also contains sophisticated modules to simulate the labour market and earnings.

This description goes some way to explaining why, as Harding (2007: 4) notes, the construction of a microsimulation model of sufficient quality to underpin government policy can be a multimillion pound exercise occupying a team of researchers and taking several years to complete.

4 SOFTWARE CONSIDERSATIONS

SAGE was written in C++ and as an object oriented language, C++ is a good choice for implementing a microsimulation model. This is because there is a natural correspondence between the software object and the microsimulation unit. Objects are defined in the program and contain all the information about the unit such as age, marital status and income as well as what it can do such as form partnerships, give birth and eventually die. One of the difficulties with this approach is that the model developer must write the code to manipulate the collection of units. This includes adding and deleting objects, stepping through the list, sorting it into order and selecting a particular unit from the list. One way this can be done is by arranging the objects in a linked list[1]. Here, each object is connected to the next by holding its memory address. This is quite advanced software engineering and it takes considerable skill to implement successfully. Another area of difficulty is in developing the user interface. A high quality graphical interface that makes the model easy to use requires specialist skills on the part of the developer. Quite often, microsimulation implementations make use of third party software to facilitate user interaction with the program. EUROMOD for example uses spreadsheets to enter model parameters and format the output into charts. This paper will however focus on interfaces that are integral to the microsimulation environment.

One attempt to alleviate the problem of software complexity is ModGen produced by Statistics Canada. This general-purpose microsimulation language hides the underlying mechanisms for controlling the units. It also provides a visual interface, documentation and a range of tools for tabulating the output. This frees the modeller from a good deal of low level programming and allows them to concentrate on setting up the parameters, units and events. Despite this, ModGen as a superset of C++, requires the modeller to know something of this language and 'have some understanding of structured programming principles' (Statistics Canada 2008; 5).

Another approach is to hold the units in a database and access them through a database management system (DBMS). This is the method used in UMDS (Universal Micro DataBase System) (Sauerbier 2002). Here, the microsimulation language MISTRAL (Micro Simulation Transformation and Analysis Language) is based on a cut down version of Pascal and includes a graphical user interface.

While generic microsimulation tools like ModGen and UMDS are quite rare, there are numerous examples of general-purpose Agent Based Modelling toolkits, which, this paper argues, share many features in common with microsimulation modelling. The next part of this paper considers the feasibility of using these as the basis of a microsimulation model.

---

[1]   See for example 'SOCSIM Documentation' http://www.demog.berkeley.edu/~socsim/c_doc.html

5 AGENT BASED MODELLING

The origins of agent based modelling (ABM) can be traced back to the idea of a self-replicating machine, now known as the von Neumann machine (von Neumann 1966). This was itself derived from a kind of universal computing machine or Turing machine (Turing 1937). In the 1970s, Conway implemented the 'game of life' (Gardner 1970), which became widely known to computer enthusiasts at the time. The basis of this game was a grid of cells, which could be 'alive' or 'dead' represented by being black or white on the computer screen. Each cell was in a sense aware of its environment and altered its state according to simple rules. If a cell is dead and three neighbouring cells are alive then it becomes alive. If more than three neighbouring cells are alive, the cell dies of overcrowding. If less than two neighbouring cells are alive, the cell dies of isolation.

These cells had a fixed location. One of the first implementations of a model implementing mobile agents appeared in a simulation of the flocking behaviour of birds called Boids (Reynolds 2008). The simulated birds operated under rules for separation, steer to avoid crowding local flockmates; alignment, steer towards the average heading of local flockmates; cohesion, steer to move toward the average position of local flockmates. By varying the rules, it became possible to simulate a range of other creatures such as herding cattle and schools of fish as well as human behaviour like co-operating in a social network or forming alliances between nations.

Due to the similarities among agent-based simulations, it became apparent that much of the software could be implemented in the form of a toolkit. This leaves the modeller free to concentrate on specifying the rules for the agents to follow. Swarm[2] was the first widely available agent based modelling toolkit. This provided a set of classes to model a hierarchy of agent collections or swarms. It also included the code for an 'observer' to monitor agent behaviour as well as tools for the user interface. Swarm was originally written in Objective C and subsequently, a version in Java has become available which uses Java classes to call the underlying Objective C routines. Since the late 1990s, a range of 'Swarm clones' has been developed written entirely in Java. These include Repast[3], MASON[4] and JAS[5] (Java Agent Simulator). Recently, agent based modelling toolkits that enable simulations to be created using scripting languages have become available. These make use of a high level language to access Java libraries. Repast allows models to be constructed in Java or a scripting language known as Python. Others such as NetLogo (Wilensky 1999) provide their own scripting language.

There are some clear parallels between the structure of agents and microsimulation units. In both cases, they have inputs in the sense that they can be affected by features of their environment. Output consists of a response to a combination of their inputs and current state. They both have operating characteristics to control the way inputs are transformed into outputs. These similarities between agents and microsimulation units seem to invite consideration of using an agent based modelling toolkit as the basis of a microsimulation model.

Chattoe-Brown (2009) points out some important differences between microsimulation and agent based modelling that seem largely to be centred around the quantitative, data driven approach associated with microsimulation modelling and the more behavioural emphasis of many agent based models. This interesting debate is not discussed in detail here as the focus of this paper is to address the more technical aspects of using agent based modelling toolkits to accelerate the development of dynamic microsimulation models.

One current example of a microsimulation model based on an agent based modelling toolkit is the IFSIM model (Baroni, Zamac & Oberg 2009). This is an agent-based model written in Java, based on the JAS framework. It models and forecasts demographic changes taking place in the Swedish population and includes modules covering fertility, mortality, matching and leaving

---

2   Swarm Development Group http://www.swarm.org/index.php?title=Main_Page
3   Repast, Recursive Porous Agent Simulation Toolkit, http://repast.sourceforge.net/
4   MASON, Multi Agent Simulator of Networks http://cs.gmu.edu/~eclab/projects/mason/
5   Java Agent-based Simulation Library http://jaslibrary.sourceforge.net/index.html

home. There are also modules that simulate the labour market, tax, benefit and pension systems as well as human capital and education. This allows recursive modelling of the effect of demographic changes on the economy as well as the reaction of economic changes on demography through fertility rates.

JAS assists the model developer by providing a library of pre written Java classes that are modified for a particular application. One of the central ones is buildModel() which calls buildObjects() and buildActions(). BuildObjects() creates the model objects and calls initPopulation() which loads in the external data. BuildActions() controls what events and actions are to take place in the simulation. Other important classes are setParameters() which initialises the model and buildDisplay which includes the graphical output methods. In addition to the standard JAS classes, it is possible to load in any external library that is written in Java. This places a vast array of functionality into the hands of the developer but it also makes it necessary to have a high level of competence in Java as well as learning about the JAS framework and how to use the libraries.

Agent based modelling toolkits like JAS have the potential for making the development process easier and faster compared to writing the simulation from scratch. In addition, there is no loss of functionality because the developer can access the underlying Java code when necessary, However using this type of toolkit still requires a high level of skill in the software engineering. A modeller who is interested in studying a social process must first acquire this skill by learning it themselves or alternatively by obtaining the services of a programmer and then communicating to them in detail what the program is required to do.

The toolkits that implement a high level scripting language are much less demanding from a software-engineering point of view. NetLogo, for example was derived from a language created to allow schoolchildren to learn about agent based modelling. Here, software agents called 'turtles' move around on a grid of 'patches' controlled by simple commands such as 'forward 1' to advance to the adjacent patch and 'right 5' to change course 5 degrees to the right. It would seem unlikely that this could be extended to the point of implementing a complex microsimulation application. However, there are examples of quite sophisticated models from a wide range of application areas in the NetLogo models library[6]. In addition, Railsback, Lytinen and Jackson (2006) found that all of their test models in the 'stupid model' series could be implemented in NetLogo. Perhaps the only way to demonstrate whether a scripting language like NetLogo has sufficient functionality to implement a microsimulation model is to create one in practice. The next section of this paper describes the development a demographic microsimulation model of the UK population, which as far as I am aware, represents the first attempt to implement a dynamic microsimulation application using a high level agent based modelling toolkit.

6 A NETLOGO BASED DEMOGRAPHIC MODEL

From DYNASIM onwards, dynamic microsimulation models that wish to project a population forwards in time have included modules to predict births, deaths, partnership formation and dissolution. If NetLogo is to be considered as a tool that is suitable for microsimulation modelling then it should, as a minimum, demonstrate that it can implement these functions. The model was intended to achieve a reasonable level of sophistication in that the transition probabilities should be calculated with sufficient accuracy so that the modelled aggregate population accurately predicts the evolution of the target population. This was assessed by comparing model projections for previous known years against past survey data from the BHPS. A further objective was to establish that NetLogo had the capacity to model a population of several thousand units at an acceptable execution speed which was defined as a few minutes to complete each simulated year. The exercise also allowed an assessment of the user interface as well as the environment for programming and model development.

---

[6] NetLogo Models Library http://ccl.northwestern.edu/netlogo/models

6.1 Design

The first part of the development process was to identify the states that the microsimulation units, representing individuals can occupy. From this, the number and type of the state transitions could be mapped.

The life history of an individual can be thought of as a trajectory through a number of states. This is represented in Figure 1.
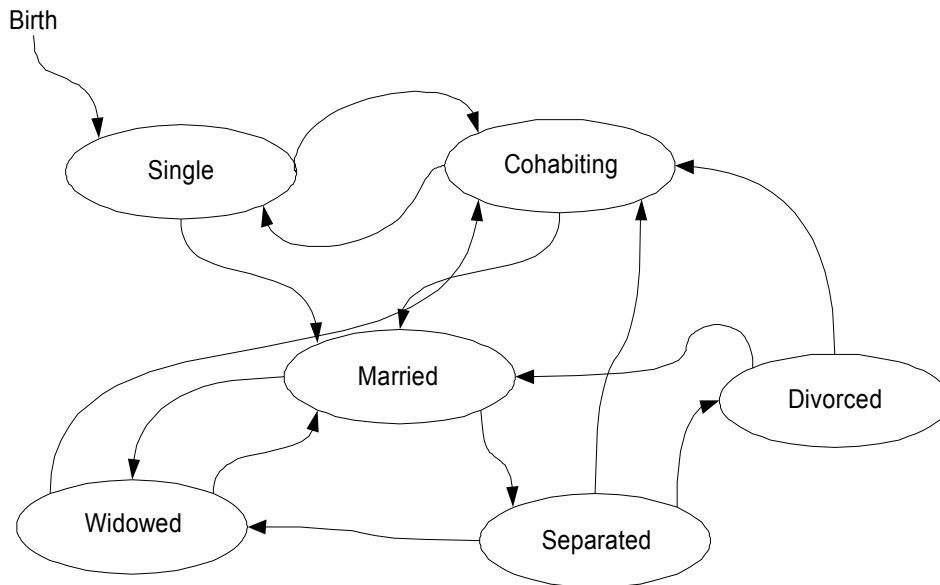
Figure 1: States in the Life History

The event of birth leads to the state of being single. From there, two possibilities exist. The individual can move to be cohabiting or they can marry. Assuming the new state is to cohabit, the individual can either move back to being single again or change to being married. This process continues as each individual moves through the life stages. Death can occur from any state and can be thought of as moving outside the plane of the diagram.
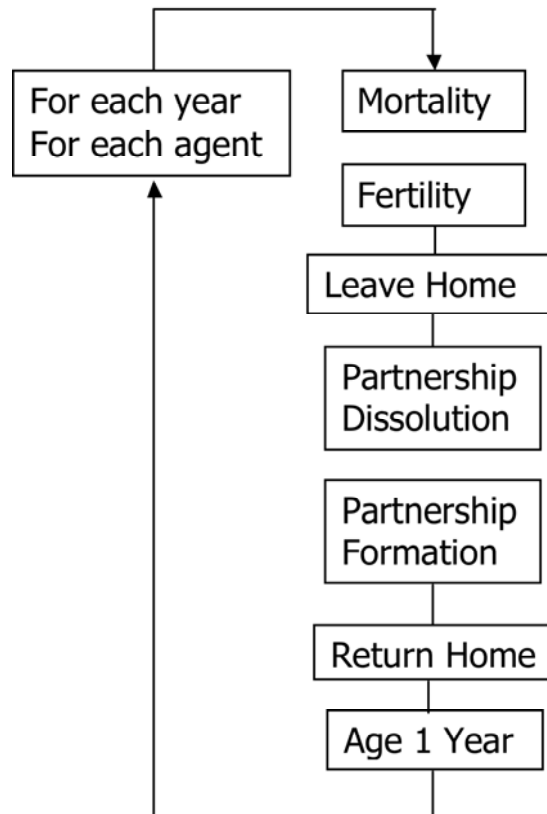
Figure 2: Microsimulation Modules

Each arrow between states corresponds to a transition for which a probability is to be calculated. To ease debugging and maintenance of the program, this was implemented in a series of modules (Figure 2). For example the partnership formation module will select a number of individuals from the population each year to either marry or cohabit. A mortality module will select which individuals are to be removed from the population. On top of this basic structure, modules were added to represent single people leaving and returning to the parental home. The modules are run sequentially each year.

Each transition between states takes place with a probability that depends on a number of factors. The determination of what these factors are is not a trivial exercise. Fortunately, the developers of some existing microsimulation models have made their findings available and these were used in the creation of this model. Zaidi and Rake (2001) provide a useful summary of the design choices made in some microsimulation models. In MOSART for example, mortality depends on age, sex, marital status, educational attainment and disability. In DYNASIM2, marriage depends on factors such as age, race, sex and previous marital status. The design choices underpinning SAGE are described in detail in the SAGE Technical Notes (Scott 2003) and were a key resource for the NetLogo based model.

6.2 Base Survey Data

Microsimulation modellers in the UK are fortunate to have access to the British Household Panel Survey (BHPS). This was used to provide the initial population for the model, the dataset used for calculating the annual transition probabilities and as the means of validation for the population forecasts produced by the model.

The transition probabilities were calculated by constructing a set of logistic regression equations for each transition to be modelled and as will be discussed below, additional probabilities were taken from the SAGE Technical Notes produced by the SAGE project team (Scot 2003).

Some of the annual transitions are relatively rare. For example, the number of males between the ages of 20 and 30 in the BHPS who marry each year is typically in single figures. To obtain a sufficient number of these events for the derivation of an accurate annual transition probability, data from the first 15 years were pooled and weighted as needed using the

provided BHPS longitudinal weight (wLRWGHT) to allow for differential attrition throughout the course of the survey. (Taylor et al 2009)

The process used to calculate the coefficients for the regression equations was first to create a data file that contains information on each individual for pairs of years over the 15-year sample. Each row in this table holds items such as marital status in the first year and what it was in the following year; age of the youngest child in the first year and age of the youngest child in the following year. From this, it is possible to tell whether a transition occurred during each particular year and so estimate the probability of a transition occurring given the individual's parameters in the first year. There can be up to 14 pairs of years for each person depending on how many of the years of the BHPS they participated in. The next step was to run the binomial logistic regression. Some of the independent variables are divided into levels for example age might be categorised as being in the 'twenties', 'thirties' etc. In a few cases, some of the levels were found not to be significant at the 5% level of confidence. This implies that the probability of the event for this level is about the same as what it would be for the reference level. When this occurred, to keep the model consistent, the levels were not changed. The coefficients that were obtained are given in the series of tables included as 0 along with the standard error and 5% significance level.

6.3 Description of the NetLogo Demographic Model

As with all NetLogo models, the 'model' itself is a textual script file that is loaded and executed by the NetLogo modelling engine. This can be done using the full NetLogo toolkit or using the NetLogo Java applet.

6.3.1 Initial Data Loading

The initial population is stored in two files, both derived in this case from the BHPS. One of these contains data at the household level. The other has data about individuals. The household file is read first. This sets up households identified by a unique case number ready to be occupied by the agents. Next the individual (agent) file is read. Each agent moves to the household indicated by the respective case number.

The BHPS provides a cross sectional household weight (wHHWGHT in wave A and wXHWGHT for subsequent waves) to adjust for non-response and unequal selection probabilities at the household level. This is used so that the model can be interpreted as a representation of the UK population. A difficulty arises here because the weights are decimals but it is not possible to have a fraction of a household. This was circumvented by creating or deleting an entire household with a probability that depended on the weight. For example, if the weight for a particular household was 0.5, there would be 50% chance that the household would be deleted; otherwise, it would be unchanged. If the household weight was 1.1, there would be a 10% chance that the household and its occupants would be duplicated. In this way, the now partly synthetic population was balanced to correspond with the real UK population and statistical analysis suggested no significant differences between this population and the original weighted sample.

Children are assigned to the parent they will be dependent upon in the event of the couple separating. In 9 out of 10 cases, following the SAGE model, this is the female. In the birth module, all offspring depend on their mother.

6.3.2 Mortality

The annual probability of death was taken from the Office for National Statistics (ONS) Life Tables 2004 - 2006.

For those aged under 40, this was approximated by a simple average of the annual probability of mortality for this group. For those over 40, the annual probability of mortality was fitted with an exponential function.

To determine whether the agent dies before the end of the year, the annual probabilities calculated from the life tables are; for males aged 40 and younger the probability of not surviving p = 0.000736. For males over 40, $p = 0.00003 \, e^{0.0954 \, age}$. For females aged 40 and younger, p = 0.000411. For females over 40 $p = 0.00001 \, e^{0.1017 \, age}$.

6.3.3 Birth

This module applies to females aged between 15 and 60. The probability of a birth in any year is calculated using three logistic equations derived from a pooled sample of the BHPS. Pairs of years from waves A (1991) to wave O (2005) are combined. Separate logistic equations are obtained; one for married women, one for cohabitors and one for singles and others. They are all based on age, number of children and age of the youngest child. Probabilities of twins or triplets are modelled using data from SAGE Technical Note no. 4, which was in turn derived from (Office for National Statistics 1996).

6.3.4 Leaving home

Individuals who are not living alone, are aged 18 and over, are single and have no dependants move to an unoccupied patch with a probability of 0.1. This is a nominal value estimated from Aassve et al. (2005). Sensitivity analysis showed that the household composition after 14 years varied by about 0.5% when this parameter was set at values between 0 to 0.2.

6.3.5 Marriage dissolution

Separate logistic models for males and females determine which marriages end in the current year. The model that operates for the current individual is randomly selected to be male or female dominated with a probability of 0.5. If the current agent in a particular instance happens to be male and the model for this agent is selected to be female dominated, then the male avoids being considered for marital dissolution. This effectively halves the probability of dissolution. This is compensated for when his partner is processed through this module as he is subject to the full risk of her dissolution probability if the model is selected to be female dominated.

When an agent is selected for separation, their status is set to 'separated', and a 'newlyseparated' flag is set to prevent these individuals from re-marrying in the current year. It is reset at the start of the next year's cycle. Next, the status of their partner is set to 'separated'. With a probability of 0.9, the male moves to a vacant household space. In the remaining 10% of cases, the female moves out. In all cases, any children present remain at the current location.

6.3.6 Cohabitation dissolution

This module operates in the same format as the marriage dissolution module although the actual probabilities are calculated using separate logistic equations.

6.3.7 Cohabitation formation

Separate probabilities are calculated for males and females with age and current status taken into account. In the female model, birth in the current year's fertility module (which has already been run at this point) is also allowed for. If selected, the first step is to check whether there is a potential partner of appropriate age available in the population. If not the process is aborted and a warning message appears in the output window. If a suitable match is found, the individual's status is set to cohabiting and it moves to a vacant household space. Next a search is made for a partner with an age difference chosen using a probability derived from Bhrolchain (2005). If there is one, its status is set to cohabiting and it moves to the household space of the current agent. If not, the process is repeated with a wider age range. The maximum age difference is 20 years.

6.3.8 Marriage formation

Couples who are already cohabiting are considered for marriage provided the cohabitation did not start in the current year. As the cohabitation module runs before the marriage module, new cohabitors must remain in this state until the current year's processing is completed to avoid depleting the proportion of cohabiting couples observed in the current year. This is determined by the 'newcohabitation' flag. If selected, the resident couple set their status to 'married'. They do not move house.

Individuals living separately are considered for marriage in a way similar to the cohabitation module. A separate set of logistic regression equations is applied.

6.3.9 Age 1 Year

Age is incremented by 1 year ready for the next annual cycle. The current year is incremented.

6.3.10 Returning home after separation

Individuals whose cohabitation or marriage dissolved in the current year return to their parents or original home if they are aged under 30 and have no dependants.

6.3.11 Returning home for all agents

With a probability of 0.01, individuals aged under 30, living alone or with one other return to their original address. In many cases, they will already be at this address. This is a small nominal estimate meant to reflect the movement of younger individuals sharing accommodation for example. As members of this group are quite likely to move out again in a relatively short time due to the leaving home module or the partnership formation module, the model is not highly sensitive to this parameter.

6.3.12 Locating children

As agents change location as a result of partnership formation and dissolution, their offspring require a method of moving with them. This is mediated by a unique identification number that all agents have called the 'who number'. Offspring record the who number of the parent it is designated as being dependent upon. This is recorded at the offspring's birth or in the 'findparents' module, which is run once after loading the initial data files. At the end each the year, all offspring check the location of this parent and move to the household space which the parent currently occupies.

6.4 Validation

To find out whether the model gives an accurate reflection of annual transition probabilities, the BHPS population as at 1991, weighted by household weight (AHHWGHT), was loaded into the model. The model was then run forward to 2005 and the forecast population was compared against the weighted observed population in subsequent waves of the BHPS. If the model forecasts are compared against individual level data from the BHPS then the individual cross-sectional weight (wXEWGHT for the whole population and wXRWGHT for adults) is used to re-weight the BHPS data to the UK population. When comparing household level variables, the cross-sectional household weight would be used (wXHWGHT). The model forecasts do not need to be re-weighted because weights were applied to the initial population and the transition probabilities.

The results for partnership status and age are shown in Figures 3 to 5. The solid line represents observed BHPS data. The dotted line is the model forecast. These data represent the mean values from 10 consecutive simulations. There is some variation between each run due to the probabilistic nature of the model, which decreases with increasing number of agents. In this case the weighted BHPS population consisted of approximately 5,500 households and 13,600 individuals. There is a small variation in the initial population due to the weighting process. Each simulation took approximately thirteen minutes to run on a computer with a 3GHz processor.
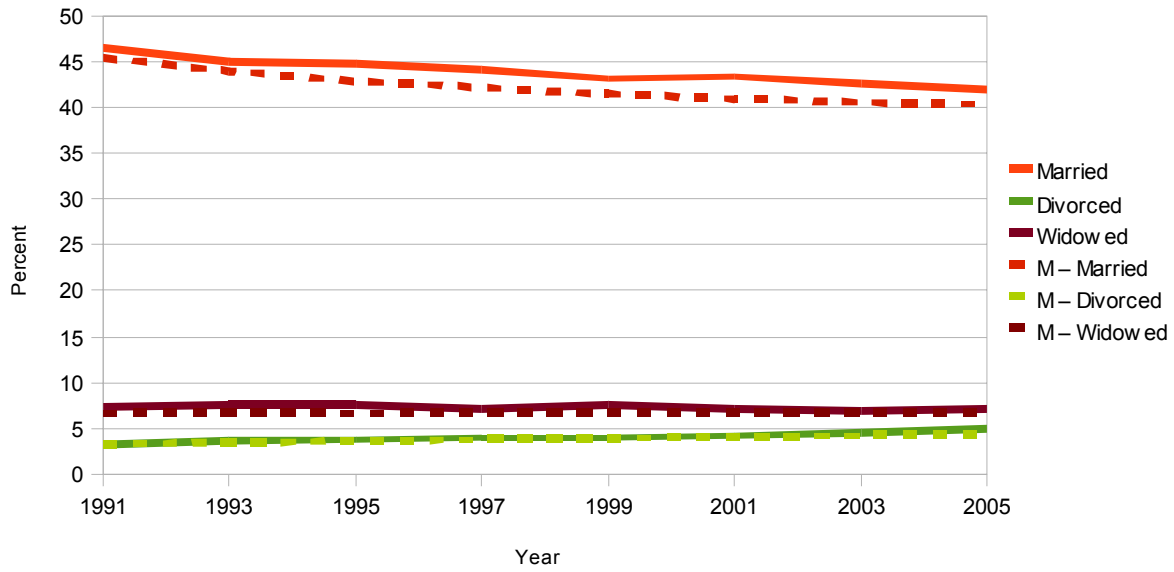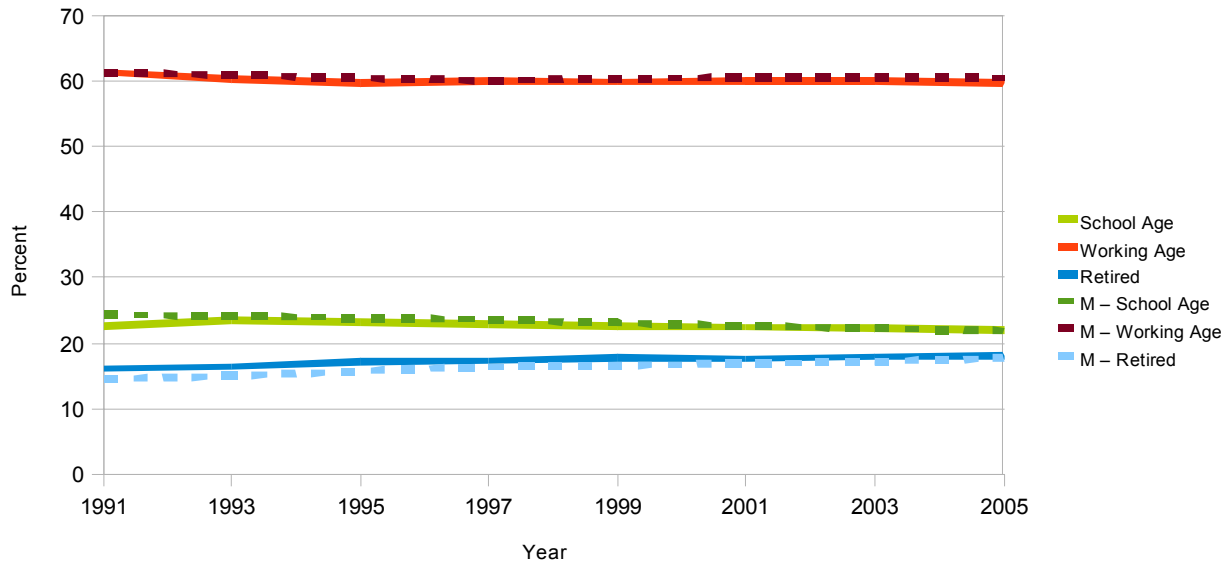
Figure 3: Model Projections (married divorced and widowed)
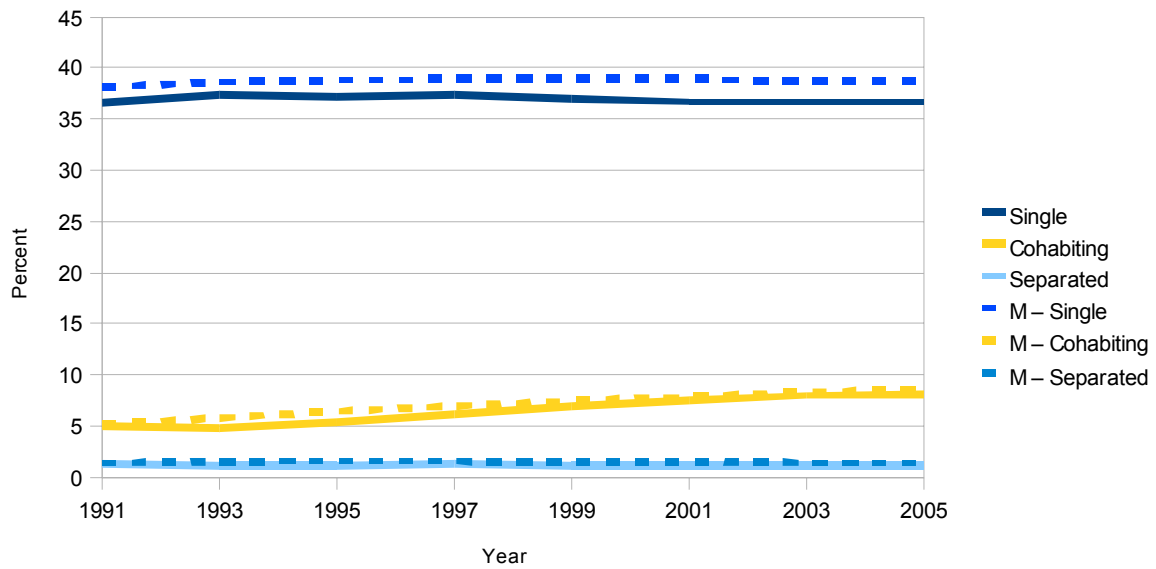
Figure 4: Model Projections (single, co-habiting and separated)

Figure 5: Model Projections (Age groups)

Although some of the parameters do not change greatly, the simulation has identified the decline in the proportion of the population who are married and the corresponding increase in cohabitation that is indicated in the BHPS. The model also correctly identifies an increase in the proportion of people who are retired; defined as those aged 65 and over.

On the whole, the model forecasts at the end of the 14-year simulation are within about 2% of their actual value. This is not a true 'out of sample' test however. Although the initial population was taken from 1991 data, the transition probabilities are in effect an average for the period 1991 to 2006. Despite this, the simulated projections parallel the actual data over the whole of the time period. This gives some confidence that the model could give reliable indications of population trends into the future, contingent on the underlying transition probabilities not changing significantly over time. The long-term stability of these probabilities could form an area for further investigation. In addition to this, there are a number of features of the model that could be improved or extended. The accuracy of the forecasts could be increased by taking more factors into account in calculating the transition probabilities and replacing the nominal estimates of some of the transition probabilities with more sophisticated logistic models. There is also scope to add other modules such as employment and income.

7 EXPERIENCES OF USING NETLOGO

This exercise has demonstrated the feasibility of developing a dynamic microsimulation model using NetLogo. It can be run as an applet from the project web page sited at the University of Essex[7].

NetLogo is well documented and there were numerous examples of models from a wide range of areas included with the software, which is free to download from the NetLogo web site. Downloading and installation were straightforward and it was possible to begin writing simple models in the same day. On opening, NetLogo presents the user with a screen containing the world grid, which is the area where the agents are displayed. By default this is a toroidal arrangement of static squares over which the mobile agents can move. This format was ideal for the demographic simulator because the squares naturally corresponded to households and the mobile agents were used to represent the occupants. Although it had no application in the final model, the world grid was useful in debugging the program. Clicking on an agent makes it possible to view the current state of its internal variables while the simulation is running and

---

[7]   Project web page: _____ http://cresi.essex.ac.uk/students/tlawso/model1

so to check the values are as expected. Also on this screen are the menus for developing a model and access to a set of tools for setting up the user interface. These include buttons, sliders, input and output boxes. There are also charting facilities that make it easy to plot a model variable as it changes over time or statically as a histogram.

In total, the development of the model took up the first year of a Ph.D project. This included reviewing the agent based modelling software, learning to program using NetLogo, researching existing microsimulation models then writing and testing the software. Although I had some previous programming experience, I had no knowledge of NetLogo. The production of the software itself took about six months. A simple prototype was set up in one month and the remaining five months were taken up writing and testing the final version. The major task here was the calculation of the annual transition probabilities. Thirteen logistic regression models were developed and this was complicated by the need to pool pairs of years over the fifteen-year longitudinal sample in the BHPS. This process was accelerated significantly by the availability of descriptions of existing microsimulation models indicating the important variables to include in the logistic regression equations. In some cases, actual probabilities were available as in the probability of multiple births from the SAGE model, age differences at marriage from Population trends and the mortality life tables from the ONS.

While this exercise has demonstrated that NetLogo is a suitable platform for developing a microsimulation model, it does not necessarily imply that it is the best one. NetLogo has an efficient language and performed well in the context of a single person developing all the code. Is does however have some limitations. The user interface is easy to use but could be restrictive in some applications. It seems that there is always one world grid visible. It is difficult to remove if not needed (apart from covering it with another item) and there seems to be no way to run more than one grid simultaneously. This precludes running multiple views of a model. The controls like graphs and buttons have a standard format that while simple to implement, cannot easily be changed. This could become a problem if the appearance of the simulation is required to mesh with other applications.

The NetLogo Frequently Asked Questions[8] indicates that the authors are working on releasing the source code under an open source licence. In the meantime it is possible for keen programmers to write Java extensions to augment the functionality of NetLogo.

8 CONCLUSION

Although microsimulation modelling is a powerful simulation method, it can also be complex, time consuming and therefore expensive. This is arguably one of the reasons why its uptake has been limited to a relatively small number of high profile projects. If the benefits of microsimulation modelling are to be brought to a wider community of users, one element that might help in this process would be the availability of a set of tools that alleviate the complexity of the task of developing the underlying software. While NetLogo is known to be a much simpler programming environment than, for example, C++ or Java, it was not clear at the beginning whether it contained sufficient functionality to the enable the development of a dynamic population microsimulation. This was investigated by creating a model of the UK population using NetLogo. The resulting model demonstrates the feasibility of implementing a dynamic microsimulation model in an environment with quite sparse resources. While NetLogo performed well in this exercise, this does not necessarily imply that it is the optimum platform. However it does offer the possibility that this multilevel non-linear modelling technique will become much more widely used and thus bring us closer to Orcutt's original vision.

---

[8] NetLogo Frequently Asked Questions http://ccl.northwestern.edu/netlogo/docs/faq.html#cite

9 ACKNOWLEDGEMENTS

10 REFERENCES

Aassve A, Davia M, Iacovou M and Mazzuco S (2005) 'Leaving home and poverty among youth: A cross European analysis', Working Paper of Institute for Social and Economic Research, paper 2005-24. Colchester: University of Essex.

Baroni Zamac and Oberg (2009) 'IFSIM Handbook', Working Paper, Institute for Futures Studies, Stockholm 2009

Bhrolchain M (2005) Population Trends 120, National Statistics, London, HMSO.

Chattoe-Brown (2009) Agent-based modelling and microsimulation: ne'er the twain shall meet?, ESRC/BSPS Seminar Series <http://pcwww.liv.ac.uk/~william/ESRC-BSPS%20MSM%20Seminar%20Series_files/Chattoe-Brown%20edited%20transcript.pdf>

Cheesbrough S and Scott A (2003) 'Simulating Demographic Events in the SAGE Model', London, ESRC-SAGE Technical Note no. 4, London School of Economics.

Ettlinger M and O'Hare J (1996) 'Revenue and Incidence Analysis of State and Local Tax Systems: Tools, Technique and Tradition', Paper presented at the National Tax Association's 89th Annual Conference, 1996.

Fredriksen D (1998) 'Projections of Population, Education, Labour Supply and Public Pension Benefits: Analysis with the Dynamic Microsimulation Model MOSART'. Social and Economic Studies, Statistics Norway, Oslo.

Gardner M (1970) 'Mathematical Games: The fantastic combinations of John Conway's new solitaire game "Life"', Scientific American 223: 120–123

Gilbert, N and Troitzsch K (2005) 'Simulation for the Social Scientist – second edition', England, Open University Press.

Hancock R Juarez-Garcia, et al. (2006) 'Winners and Losers: Assessing the Distributionel Effects of Long-term Care Funding Regimes', ISER Working Paper 2006-43, Colchester, University of Essex.

Hancock R, Mallender and Pudney S (1992) 'Constructing a Computer Model for Simulating the Future Distribution of Pensioner's Incomes for Great Britain' in Hancock and Sutherland (eds.) Microsimulation Models for Public Policy Analysis: New Fromteirs, STICERD Occasional Paper no. 17, London School of Economics, London.

Harding A (2007) 'Challenges and Opportunities of Dynamic Microsimulation Modelling', NATSEM, University of Canberra.

INSEE (Institut National de la Statistique et des Etudes Economiques) (1999) Le Modele de Microsimulation Dynamique, DESTINIE. Document de travail, G9913. Paris: INSEE.

Nikolai and Madey (2009) 'Tools of the Trade: A Survey of Various Agent Based Modeling Platforms'. Journal of Artificial Societies and Social Simulation 12(2)2 <http://jasss.soc.surrey.ac.uk/12/2/2.html>.

Orcutt G (1957) 'A new type of socio-economic system', Review of Economics and Statistics, 39(2), 116-123. Reprinted in the International Journal of Microsimulation (2006) 1(1), 2-8.

Orcutt G H, Greenberger M, Korbel J et al. (1961) Microanalysis of Socioeconomic Systems: a

simulation study, New York: Harper and Row

Orcutt G H, Caldwell S and WertheimerII R (1976) Policy Exploration through Microanalytic Simulation, Washington D.C.: The Urban Institute.

Office for National Statistics, (1996) 'Birth Statistics, Review of the Registrar General on births and patterns of family building in England and Wales', 1994, London: The Stationary Office.

Railsback S, Lytinen S, Jackson S (2006) Agent-based Simulation Platforms: Review and Development Recommendations, Simulation, Vol. 82, No. 9, 609-623.

Reynolds C (2008) Boids: Background and Update. <http://www.red3d.com/cwr/boids/>

Sauerbier T (2002) UMDBS – 'A new tool for dynamic microsimulation', Journal of Artificial Societies and Social Simulation,5(2). <http://jasss.soc.surrey.ac.uk/5/2/5.html>

Scott A (2003) 'Implementation of demographic transitions in the SAGE Model', London, ESRC-SAGE Technical Note no. 5, London School of Economics.

Statistics Canada (2008) 'Modgen Developer's Guide' v9.0.10

Tobias R and Hoffmann C (2004) Evaluation of free Java-libraries for social-scientific agent based simulation, Journal of Artificial Societies and Social Simulation, 2004;7. <http://jasss.soc.surrey.ac.uk/7/1/6.html>

Taylor M (ed) with Brice J, Buck N and Prentice-Lane E (2009) 'British Household Panel Survey User Manual Volume A: Introduction, Technical Report and Appendices', Colchester: University of Essex.

Turing A (1937) 'On Computable Numbers, With an Application to the Entscheidungsproblem', Proceedings of the London Mathematical Society, Series 2, Volume 42 (1936-37) pp230-265.

University of Essex, Institute for Social and Economic Research, British Household Panel Survey: Waves 1-17, 1991 – 2008 [computer file]. 5th Edition. Colchester, Essex: UK Data Archive [distributor], March 2009. SN 5151.

von Neumann J (1966) in Burks A: The Theory of Self-reproducing Automata, Urbana, IL: Univ. of Illinois Press.

Wilensky U (1999) NetLogo <http://ccl.northwestern.edu/netlogo/>

Zaidi A and Rake K (2001) 'Dynamic Microsimulation Models: A Review and Some Lessons for SAGE', SAGE Discussion Paper no. 2, SAGEDP/02. <http://www.lse.ac.uk/collections/SAGE/pdf/SAGE_DP2.pdf>

Annex A PARAMETER ESTIMATES

Table 1: Parameter estimates of whether a married woman gives birth in the next year

	β	S.E.	Sig.
Age	-0.55	0.0005	0.000
Number of previous children	-0.274	0.46	0.000
Age of the youngest child	-0.01	0.008	0.886
Constant	2.897	0.177	0.000

Table 2: Parameter estimates of whether a cohabiting woman gives birth in the next year

	β	S.E.	Sig.
Age	-0.078	0.010	0.000
Number of previous children	0.158	0.086	0.065
Age of the youngest child	0.011	0.017	0.525
Constant	-0.442	0.301	0.142

Table 3: Parameter estimates of whether an unpartnered woman gives birth in the next year

	β	S.E.	Sig.
Age	-0.031	0.006	0.000
Number of previous children	0.463	0.055	0.000
Age of the youngest child	-0.020	0.009	0.023
Constant	-3.219	0.173	0.000

Table 4: Parameter estimates of whether a non-cohabiting male will marry in the next year

	β	S.E.	Sig.
Age	-0.001	0.006	0.833
Current marital status			
Widowed	0.147	0.474	0.756
Divorced	0.723	0.226	0.001
Separated	1.739	0.220	0.000
Constant	-4.247	0.185	0.000

Table 5: Parameter estimates of whether a cohabiting male will marry his cohabitee in the next year

	β	S.E.	Sig.
Age			
Thirties	0.016	0.118	0.889
Forties	-0.547	0.177	0.002
Fifties	-1.007	0.278	0.000
Sixties	-0.829	0.342	0.015
Constant	-1.796	0.078	0.000

Table 6: Parameter estimates of whether a non-cohabiting female will marry her cohabitee in the next year

	β	S.E.	Sig.
Age	-0.027	0.006	0.000
Current marital status			
Widowed	0.139	0.358	0.697
Divorced	0.717	0.211	0.001
Separated	2.111	0.179	0.000
Birth in the following year	0.684	0.309	0.027
Constant	-3.479	0.178	0.000

Table 7: Parameter estimates of whether a cohabiting female will marry their cohabitee in the next year

	β	S.E.	Sig.
Age			
Thirties	-0.140	0.116	0.226
Forties	-0.131	0.160	0.415
Fifties	-0.728	0.240	0.002
Sixties	-1.083	0.618	0.080
Constant	-1.849	0.068	0.000

Table 8: Parameter estimates of a male beginning cohabiting in the next year

	β	S.E.	Sig.
Age	-0.021	0.004	0.000
Current marital status			
Widowed	-0.009	0.331	0.979
Divorced	0.676	0.136	0.000
Separated	1.221	0.155	0.000
Constant	-2.2240	0.102	0.000

Table 9: Parameter estimates of a female beginning cohabiting in the next year

	β	S.E.	Sig.
Age	-0.045	0.004	0.000
Current marital status			
Widowed	-0.657	0.280	0.019
Divorced	0.532	0.126	0.000
Separated	0.499	0.156	0.001
Birth in the following year	0.750	0.172	0.000
Constant	-1.398	0.106	0.000

Table 10: Parameter estimates of marriage dissolution for males

	β	S.E.	Sig.
Age			
Thirties	-0.612	0.164	0.000
Forties	-0.939	0.159	0.000
Fifties	-1.813	0.213	0.000
Sixties	-3.032	0.353	0.000
Constant	-3.155	0.133	0.000

Table 11: Parameter estimates of marriage dissolution for females

	β	S.E.	Sig.
Age			
Thirties	-0.492	0.127	0.000
Forties	-1.024	0.141	0.000
Fifties	-1.920	0.192	0.000
Sixties	-3.340	0.378	0.000
Birth in the following year	-1.114	0.301	0.000
Constant	-2.992	0.101	0.000

Table 12: Parameter estimates of cohabitation dissolution for males

	β	S.E.	Sig.
Age			
Thirties	-0.668	0.150	0.000
Forties	-1.405	0.265	0.000
Fifties	-0.599	0.256	0.019
Sixties	-2.068	0.648	0.001
Constant	-1.978	0.084	0.000

Table 13: Parameter estimates of cohabitation dissolution for females

	β	**S.E.**	**Sig.**
Age			
Thirties	-0.500	0.136	0.000
Forties	-0.868	0.219	0.000
Fifties	-1.077	0.292	0.000
Sixties	-1.178	0.679	0.083
Birth in the following year	-0.462	0.258	0.073
Constant	-1.959	0.074	0.000