ABM – Week 3 – Seminar

Purpose

This task is intended to deepen your understanding of the relationship between the ODD protocol and the implementation of ABMs in NetLogo. It will also serve to further develop your knowledge of the software and give you experience of building complete models without the same level of direction that you have been given in the tasks so far.

Task

Download the Marriage Model PDF from Moodle. This file contains an ODD description for a simplified version of a published social science model. Use the description to create a NetLogo model from scratch. Use the following steps as a guide:

- Start by creating a skeleton of the model, using the 'Overview' section of the ODD description. Create the necessary procedures (without any details), define the necessary variables and put the processes in the right order.
- Write each procedure, one-by-one, testing the model after each step to see whether it behaves as you expect. Think carefully about the best (easiest?) order in which to add these procedures.
- Create a line plot and a histogram to monitor key information about the simulation as it runs.

Questions to Consider

- What do you think about the ODD description given here?
- Does it conform to your expectations?
- Do you think it could be improved?
- Are there any ambiguities that you had to deal with when programming the model?

Programming Hints

We completed this task ourselves and the whole model will be provided on Moodle later in the week. Here is a complete list of the 43 NetLogo primitives we used (not including the plotting commands or standard mathematical operators like +). If you understand how all of them work (see the NetLogo Dictionary), you should be able to build the model:

and, ask, ca, carefully, color, count, crt, die, end, exp, globals, hatch, heading, if, ifelse, let, list, max, mean, min, myself, n-of, nobody, of, one-of, other, patches, pcolor, random, random-float, report, reset-ticks, set, setxy, size, stop, tick, ticks, to, to-report, turtles, turtles-own, with

[Of course, there are lots of ways to write a program, so you may not need all of these primitives, or you may use others to achieve the same goals. This is just to give you an idea of primitives that might be useful.]