

Instructions for Lab 4

Modelling of Systems for Sustainability

17th October 2024

1 Introduction

1.1 Aim

The aims of this Lab session are to become familiar with the BehaviorSpace functionality in NetLogo, which supports exploration of model behaviour, parameter sensitivity, and obtaining results in a stochastic setting.

1.2 Overview

You should work through all the instructions listed below. While the labs are not assessed, they are an integral part of the learning experience and will help you in getting the most out of this course. The labs are intended to be executed on DICE computers within the School of Informatics.

1.3 Prerequisites

You need to have NetLogo installed, and it would help to have finished Labs 1 and 2 and 3, so that you've gained some confidence with NetLogo.

2 Instructions for Lab 4

As before, pair up to work together on the lab, and we advise that at least one person in each pair should be familiar with Python, and ideally also with the Noteable service for running it.

2.1 NetLogo BehaviorSpace tutorial

1. Look through the BehaviorSpace Guide, which is under the "Features" heading of the NetLogo User Manual, available from the NetLogo Help dropdown.
2. Read through Railsback and Grimm section 8.2 and follow its instructions to learn about the Simple Birth Rates model and experiment with it.
3. Follow the steps in [Railsback and Grimm \(2019\)](#) section 8.3, to obtain both spreadsheet and table output from BehaviorSpace (page 109). Read the Programming notes on page 109 if you want to understand better what is going on under the hood in BehaviorSpace.
4. Read through the rest of the material in section 8.3. You'll see that you are advised to import the BehaviorSpace output into software of your choice in order to visualise and analyse it. Many of you will be familiar with Python and its analysis/visualisation libraries, others may be more familiar with spreadsheet software such as Excel or Numbers, and their analysis/visualisation tools. In this lab we'll be using Python, so if your pair does not include someone familiar with Python, join up with another pair for what follows.
5. The simplest way to use Python is the University's Noteable service, instructions follow. If you prefer to use your own Python setup, you can do the equivalent.
6. Log in to Learn, navigate to the MOSS course page (INFR10088), and click the "View Course and Institution tools" link on the right. You'll see on the right-hand section that appears that there is a link to Noteable about 4 items down. Click on this link to start Noteable, and then click on "Start" or "Reconnect" under "Standard Python 3".

7. In Noteable create a folder for this Lab using the button with a folder icon with a "+" on it. Upload the .csv files you created, both table and spreadsheet, to this folder (Upload button).
8. Download the SimpleBirth.zip file from the [MOSS schedule page](#), unzip it and upload the Fig8.3 Notebook (Fig8.3.ipynb) to the folder on Noteable you've just created. Once uploaded, double-click the Fig8.3 notebook to start it running.
9. Once it is visible, look in the second cell at the pd.read.csv statement. Check that it has the same filename as your uploaded table data. If not, then change the statement to match your file name.
10. Now click in the first cell (containing all the "import" statements) and type Shift-Return (the keys on your keyboard, not as a string!). This executes the statements in the cell. It will not produce any output, so your screen will not change, except now the second cell will be selected.
11. The second cell has python code to load the data and visualize it. Once it is selected, click Shift-Return again. You should see a graph appear that is similar to the one in Railsback and Grimm, Figure 8.3, but for the data you generated.
12. Look at the python code in the second cell and see if it makes sense to you. You will need to use code like this, or some other system like Excel, to do visualisations of your model output data in your project. If you haven't used Matplotlib and/or Seaborn in Python before, you might want to do some simple tutorials on these libraries to get an idea of what they can do for you in the way of visualising results.

2.2 Running Experiments and Analysing and Visualising Results

This section of the tutorial is designed to show you how to gather results and visualise them for a model that has multiple (two in this case) parameters that you want to see the effect of. It is based on the Wild Dogs model introduced in Chapter 16 of Railsback and Grimm, and the contour visualisation introduced in Chapter 23. These were both introduced in [Lecture 7, Monday of Week 4](#), which you might want to review.

1. Download WildDog.zip from the [MOSS schedule page](#), and unzip it into a folder. You should see two files:
 - african-wild-dogs.nlogo - the NetLogo model file
 - Wild Dog analysis.ipynb - a Python Notebook which will show you how to process the BehaviorSpace data
2. Open the african-wild-dogs.nlogo model in NetLogo. You can try the **Setup** and **Run** buttons to see it running, but we'll use BehaviorSpace to get the data we need.
3. In the NetLogo Tools menu, select **BehaviorSpace**. In the BehaviorSpace dialogue, you'll see three experiments - Basic, Sensitivity Analysis, and Two Parameter. Select each one, Click **Edit** to see what variables and values BehaviorSpace will run through in the experiment. For example, in Basic you will see

```
["disperser-meeting-rate" [0 0.5 2]]
```

which means run the experiment with disperser-meeting-rate starting at 0, and then increasing by 0.5 until 2 is reached (including 2).

4. Run the Basic experiment twice with two different numbers of repetitions, 100 and 400. Append -100 to the table output file name for the 100 repetition experiment, and -400 for the 400 repetition experiment. When you run the experiments, you'll notice that the **Update view** and **Update plots and monitors** checkboxes are unchecked - this is so the experiments run faster.
5. Notice that you now have two .csv files in your folder, one from each of the two runs you did. [Note, if you have difficulty creating the .csv files, you can download pre-created versions from the [MOSS schedule page](#) - WildDogData.zip]
6. Run the Sensitivity and Two Parameter experiments to generate their table .csv files. Two Parameter may take a few minutes as it is 6000 runs. You'll now have two more .csv files in your folder.

7. What these experiments are looking at is when, if at all, the wild dogs go extinct (population drops to zero), with an interest in what fraction of runs go extinct within 100 ticks. The spreadsheets have collected that data
8. Now it is time to analyse and visualise this data. Upload the `.csv` files and `Wild Dog analysis.ipynb` to Noteable. Then in Noteable, double-click the `Wild Dog analysis.ipynb` file to start it up.
9. Look at the structure of the Notebook. The first cell loads the libraries you need. The second cell defines some functions you'll execute later. Execute these two cells. The remaining sections have several cells each, which you can execute as you read through the bullet points below:
 - **Plot the fractions that are extinct** – this shows bar plots with error bars for the Basic runs with 100 repetitions and 400 repetitions. Look at the axes. You can see that the fraction extinct within 100 years goes down as the `disperser-meeting-rate` goes up, although at the highest `disperser-meeting-rate` the fraction extinct is increasing again. Why might that be? How do the error bars in the 100 and 400 repetition plots seem different to you? Why do you think that is? Notice the last code line in the cell – it saves the 400 plot as a `.png` file that could be included in a report.
 - **Sensitivity analysis** The cells load the data, turn it into the form needed, and plot it as a scatter plot. You'll see that the data is much more fine-grained, with a data point every 0.05 in the `disperser-meeting-rate`, and it is all centred around a value of 1.0 (from 0.5 to 1.5). This is exactly what BehaviorSpace was told to do. The subsequent cells in this section use a statistics library to fit a linear model (a straight line) to these data points, which is then plotted on top of the data points.
 - **Two parameter** The cells extract the data from the CSV and use the Seaborn contour plot function to make a plot of extinction contours for the two parameters `disperser-meeting-rate` and `initial-num-packs`, similar to the plots in Figure 23.2 in Railsback and Grimm. These show, consistent with the analyses above, that at very low `disperser-meeting-rates` there is a high probability of extinction, and this decreases as `disperser-meeting-rate` increases. But, and this is new, while `initial-num-packs` has little influence at low `disperser-meeting-rates`, the roles are almost reversed at `disperser-meeting-rates` above 1.0, where probability of extinction is higher for low number of packs, decreasing as the number increases, and then once again increasing a little as the initial number of packs gets into the high teens

These two exercises are intended to have given you an idea of how you can use BehaviorSpace to produce data about how a model behaves at various parameter settings, and how you can use Python to analyse and visualise that data. These are two things you will need to use quite a bit in your project work later in the semester.

Useful Links

- | | |
|----------------|---|
| Book | https://read.kortext.com/reader/pdf/653121/Cover |
| Book Resources | https://www.railsback-grimm-abm-book.com/downloads-errata-2nd-edition/ |
| NetLogo | https://ccl.northwestern.edu/NetLogo/ |

References

Railsback, S. F. and Grimm, V. (2019). *Agent-Based and Individual-Based Modeling: A Practical Introduction*. Princeton University Press, New Jersey, US, 2nd edition.