ABM – Week 9 – Seminar – LVL EXTRA

Purpose

This practical will guide you through analysing the output of a model to investigate how different parameter combinations can be explored for better verification, calibration, and validation.

Model

The practical will make use of the Hoopoebird model from Lecture 9 – please check on Moodle for a copy of the model. The model and its workings have been introduced during the lecture. Please recall that the quality of the calibration of the model was assessed based on three criteria:

1. **Mean abundance**: the long-term mean number of woodhoopoes is in the range of 115 to 135.
2. **Variation**: the standard deviation among years in the annual number of birds is in the range of 10 to 15 birds.
3. **Vacancy**: the average percentage of territories that lack one or both alphas is in the range of 15% to 30%.

Tasks

1. Run the model to create a dataset to analyse. The model comes with an experiment already created within BehaviorSpace.
2. Take the dataset and, using a tool of your choice, create a graph to show whether this parameter combination meets the criteria described above.

Step-by-Step Guide

*Remember: after **each change** you make, check that **the code runs** before you move on to the next step.*

EXERCISE ONE

<u>Generating the dataset</u>

Within BehaviorSpace, an experiment has already been set up for the parameters shown in the in-class run. You can use this as your template to further explore the data.

Do remember that, as shown in Figure 1, you can select between outputting the data as a spreadsheet and as a table. For this exercise, we recommend outputting the data as a table – it will be easy to read with other tools.

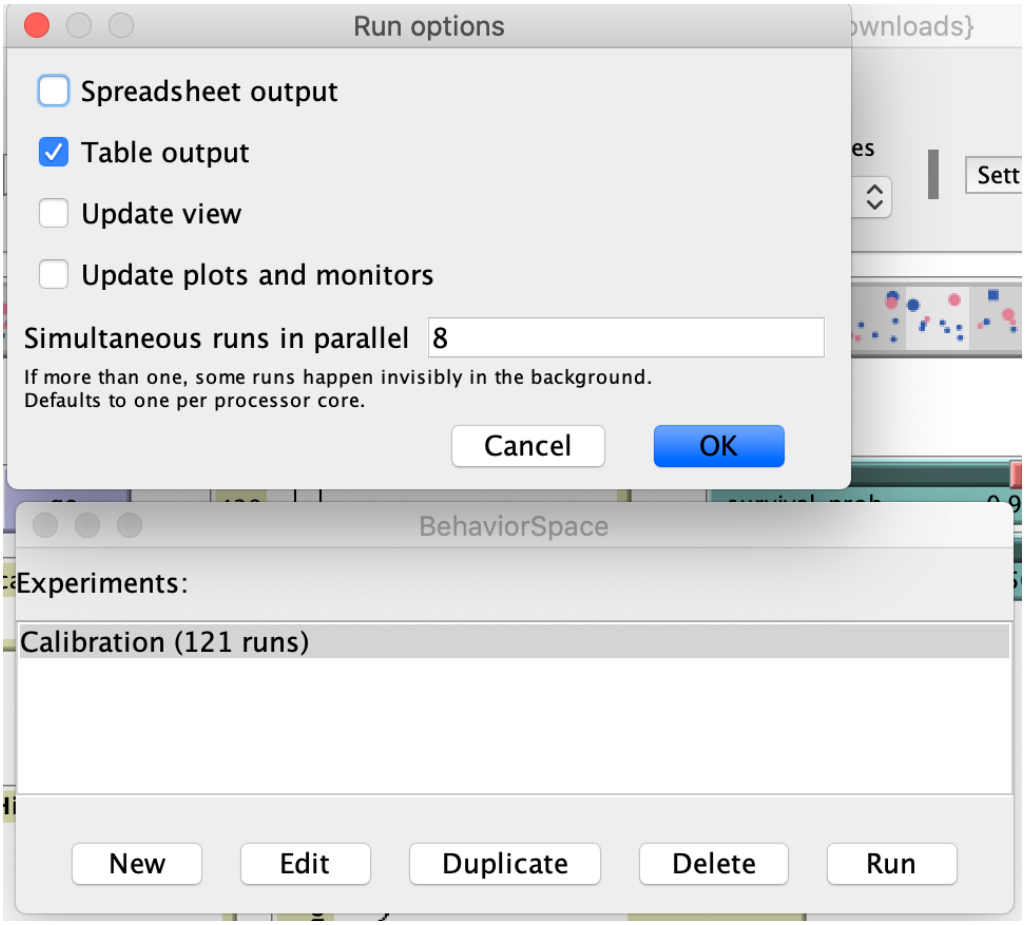

Figure 1 Running the BehaviorSpace experiment.

EXERCISE TWO

Please note: you are free to use *any tool you like* to accomplish this. This guidance will make use of Python, specifically the Pandas and Matplotlib libraries, but this is by no means the only way to do this.

Reading in the data

You can use the Pandas library within Python to read in the data. Keep in mind a few things:

- BehaviorSpace outputs its table with a few extra lines of information at the top of the file. This may confuse some tools – but it is possible to tell many programs to skip the first few lines of a file.
- Some tools will want you to tell them what is the "separator" between different records – that is, whether the different columns are separated from one another by a comma, a tab, or some other special character. You can check how your data is formatted by opening it in a basic text file tool – e.g. Notepad on Windows, TextEdit on a Mac, SublimeText.

[Useful functions: pandas.read_table, which accepts parameters 'header' and 'sep']
Further documentation:
https://pandas.pydata.org/docs/reference/api/pandas.read_table.html?highlight=read_table#pandas.read_table

Calculating the metrics

Having read in the raw data, now you have to translate the criteria described above into mathematical terms. To take an example:

Mean abundance: the long-term mean number of woodhoopoes is in the range of 115 to 135.

How can you extract this information from the dataset you have generated? Think about the following:

- What does "long-term" mean in this context?
- Which column of data can you use to calculate this?

[Useful functions: df.groupby, agg, reset_index,]
Please note: a MultiIndex can make it hard to work with your data! See an example of how to flatten your column names here: https://www.pauldesalvo.com/how-to-flatten-multiindex-columns-into-a-single-index-dataframe-in-pandas/

Assessing fit to the criteria

PLEASE NOTE: **the experiment in BehaviorSpace measures the *number* of patches which lack at least one alpha, NOT the *percentage* of patches. What does that mean for your workflow?**

For each parameter combination, you can now assess whether the metric you've calculated (e.g. mean abundance) fits within the criteria.

[Useful concepts: Python comprehensions to create new columns within a dataframe]
See for example https://medium.com/@masonrchildress/how-to-perform-a-list-comprehension-on-your-dataframe-python-8548d3de7bd8

Plotting the results

Now, for the parameter combinations which meet each of the criteria, you can plot the combination of scout-prob and survival-prob, creating a point if the criteria is met (hint: is True?). You may want to plot each of the different criteria on the same plot, using different shapes and colours to distinguish among the criteria.

[Useful functions: plt.figure(), plt.show(), fig.add_subplot(111), ax.scatter(), ax.set_xlim, ax.set_ylim]