CMEE Masters: Miniproject Assessment

April 13, 2022

Assignment Objectives: To address on a model-fitting problem using computational methods, and produce a written report, all in a coherent, reproducible, modular workflow under version control.

Student's Name: Cherie Yu

Overall Miniproject Mark: 77%

Overall Project Organization

All your directories are in place, though you have a graphs subdirectory containing model fitting plots in the code directory rather than in the results directory.

You have included a very comprehensive `readme` file describing the project, listing the languages (with version numbers!) and all of the required packages, and outlining the project structure and the name and role of each file. This is an exemplary readme, and very good programming practise.

You could have put the writeup LaTeX source files and pdf in a separate directory – this is what you should aim to do for your final dissertation.

Overall an excellently organised project (with the sole exception of the stray code/graphs subdir), and excellently documented. Well done!

The Code

Your choice of coding tools is appropriate and it is good to see you using a combination of R and Python for different individual components of your workflow. Your use of packages is minimal which is also good to see – over-reliance on packages can hinder your development as a programmer and can cause issues for reproducibility.

Your R and Python code are sensibly commented, and it is generally straightforward to see what each part of each script is doing. However at several points in sample.py you define functions within the main body of the code, rather than at the beginning. It is generally encourged to define functions at the beginning of each file so that users can see all of them before starting to look through the logic of the scripts themselves. Even better practise is to write separate scripts for containing functions/modules and for executing your code.

The main body of your code ran almost without error, with one exception: In AIC_summary.R you load only the dplyr package, but also call the pivot_longer() function, for which you need to load tidyr as well. Once we added the line "library(tidyr)" to this file, the remainder of your project compiled without errors. Your file sample.py generated a large number of warnings (though no errors) about the number of figure that were opened. These warnings cluttered the terminal output and made it rather hard to see how the workflow was progressing. When plotting many figures with a loop, consider using commands like plt.clf() at the end of each

iteration to avoid large numbers of figures building up in memory and triggering warnings. You successfully fit 3 models (quadratic, cubic and Gompertz) to your data, and compare them using AIC and BIC. However, we note that you chose to log all the population sizes even for the polynomial models. This is an unusual choice, and technically means you have chosen to investigate whether the log of the population follows a polynomial relationship w.r.t time, rather than the population itself. A better option might have been to fit the polynomial models to non-logged data, and the Gompertz model to logged data, and to manually calculate non-logged residuals for these so that you can still perform model comparison using AIC/BIC.

Recall that you should write into your workflow commands that will delete all existing output files every time the workflow is run (they should be re-generated afresh). As it stands you only do this for some of the Latex-generated temp files (.aux, .log etc...).

Your project took some time to run (12.5 mins), with the initial parameter sampling for your Gompertz model fitting taking up most of this time. In particular, a small number of your subsets seemed to take a particularly long time to generate initial Gompertz parameters. Consider how you might cut this down slightly by limiting the number of iterations per subset in your optimisation function (e.g. using Minimizer's `max_nfev` argument).

Your workflow would have benefitted from adding progress updates to the terminal, particularly during the long initial parameter sampling, when it was not especially clear from the terminal output how quickly the code was progressing. A (reliable!) time estimate would be especially good so that users can see how long remains for each specific task.

Overall a good project. Your code is well organised and runs nearly without error. You were ambitious with regard to optimising initial parameters for the Gompertz model, which is good, even if it increased the computational cost of the project. Take care in future not to miss out any required packages and be aware of the consequences/technical interpretation of data transformations like logging.

Marks for the project and computational workflow: 68%

The Report

You demonstrate a solid understanding of the methods, particularly the model comparison tools, and write a clear and well structured study with concrete goals, robust analysis and consistent reference to the wider literature. A pleasure to read!

Title: Concise, descriptive, presents main finding.

Abstract: Pretty excellent. Concise background, clear statement of objectives, methods and results. Conclusion/take-home is directly linked to the results and is confidently and concisely stated. (95%)

Intro: Solid background and motivation, even if the Silby paper is a little over-cited in the first paragraph. Clear description of the mechanistic-phenomenological distinction and justification of the choice to focus the paper on relative performances of those two model types. Research question, study objectives and an explicit hypothesis all present. (80%)

Methods: Each section of the workflow is described in appropriate detail, with attention to potential statistical issues such as sample size. Extra credit stuff includes random sampling of

init parameters and model comparison beyond simple AIC. Formatting is a little dense, but this is a minor aesthetic concern. Computing tools section is present. (87%)

Results: Impressive number of fits for Gompertz model. Clear and concise description of results with informative plots. Table 2 formatting slightly off. Fig 3 and fits relative to distribution/number of datapoints could have been presented in results, leaving the interpretation to the discussion. (76%)

Discussion: Genuinely excellent, hits every point in the guideline with reasoned analysis and consistent links to the wider literature. (100%)

(Some specific feedback is in the attached pdf, and we can also discuss more aspects of your write-up in our 1:1 feedback meeting)

Marks for the Report: 86%

Signed: Samraat Pawar & Alexander Kier Christensen

April 13, 2022

Notes on Assessment :

- This written feedback will be discussed in a 1:1 session scheduled after this assessment has been been given to you.
- The coursework marking criteria (included in this feedback at bottom) were used for both the computing and report components of the Miniproject Assessment. *In contrast*, Your final dissertation project marks are going to be based pretty much exclusively on the written report and viva (not code). Expect your final dissertation report to be marked more stringently, using the dissertation marking criteria (also included in this report).
- In the written feedback, the markers may have contrasted what you have done with what you should do in your actual dissertation. *This does not mean that you were penalized* — one of the main goals of the miniproject is to provide feedback useful for your main dissertation. However, there may be cases where what you have done is just really bad practise (for example missing line numbers or abstract), irrespective of whether it is a mini- or main- project report – you will be penalized in that case.
- The markers for this assessment are playing the role of somebody trying to understand and use your project organization and workflow from scratch. So it will seem like the feedback is particularly pedantic in places — please take it in the right spirit!