CMEE Masters: Computing Coursework Assessment

Assignment Objectives: To work on a series of computing/programming exercises and problems in a coherent, modular, reproducible workflow under version control.

Note that:

- *The overall assessment will typically have significantly lesser marks than a simple weighted average of each week's points because the overall assessment is based on not just the "Computing Coursework Assessment Criteria", but also the the "Marking Criteria for Exams, Essays and Coursework". Both sets of marking criteria are in the Assessment Appendix of the online TheMulQuaBio notes and git repository.*
- *In your 1:1 post-assessment feedback session, we will discuss where you gained or lost marks, and what you could have improved further. To the extent possible, please come with questions about specific scripts based upon the overall and weekly feedback you have received. This may require you to compare your code with the solution code in many cases.*

Student's Name: Amelia Wake

1 Specific feedback

1.1 The Good (what you did well!)

1. Found all the core CMEE weekly directories in your parent directory, but Week3 had an (unexpected) extraneous subdirectory (see below).
2. Your organisation and code are generally neat, and logically structured.
3. Your Git repo size when we checked week 7 was about 4 MB – nicely compact! This suggests you correctly suppressed unnecessary files from version control, and did not commit excessively. It could also mean that you did not commit enough, and/or somehow along the the way lost parts of your git history – but we don't check these possibilities!
4. You had an overall readme file, (but this could have included more detail!). You also had one within each week. The weekly Readmes were quite comprehensive.
5. Your coding is generally capable, but in some places suffered from syntax errors or from producing code that, although it runs without error, does not really achieve the intended outcome (see `lc1.py` comments below). Take care that you understand what a given coding tool does and why you might be using it in a script.
6. Your Python code is often reasonably modular, which is good Pythonic practise.
7. You had a .gitignore throughout, with meaningful exclusions specific to certain weeks. Good.
8. Your commenting is consistently detailed, though not obtrusive. This is good practise, and makes your code generally quite readable. Well done.
9. Your Groupwork practicals were all in order, and your group did well in collaborating on it. More feedback on this in the 1:1 sessions.

1.2 The Bad (errors, missing files, etc)

1. In Week2, your `lc1.py` doesn't really carry out the task as intended. Your list comprehensions do not build up the required lists from the info in 'birds'. Rather, you have hard-coded the required lists yourself and then written list comprehensions to re-generate them. Similarly, your loops do not build up the required lists from 'birds', but rather you hard-code the lists and then replace them with themselves within the loops. It is hopefully clear that this hard-coding approach is: (i) pointless, since there is no need to use a list comp or a loop to build a list that you have already generated! (ii) not scalable to larger problems – if the required lists had consisted of 1000 species, rather than 5, then clearly typing out the lists yourself would not have been a feasible solution to the problem. Your approach to `lc2.py` is similarly flawed. You have gotten it right in other scripts (e.g. `oaks.py`), which is encouraging.

2. Your Week3 directory contained all of the necessary material, but it was stored within an additional subdirectory 'Week3/MyRCoursework', which served no clear purpose and was not alluded to in the readme(s). Ultimately this is a relatively minor problem, but it would have been good to keep the weekly directory structures consistent and in keeping with the requested format!

3. Week3 syntax errors. Your wrangling scripts `DataWrang.R` and `DataWrangTidy.R` both threw syntax errors when you try to call functions on the un-defined dataframe 'MyWrangledData'

1.3 The Ugly (niggling issues like commenting, cosmetics, complexity of code, etc)

1. Your overall readme didn't contain any useful information.

2. In your readmes you included some note to the language and dependencies requirements, but could stand to include versions used as well. Also check out this resource: `https://github.com/jehna/readme-best-practices`. As you become a seasoned programmer, you will learn to make the readme file descriptions even more informative yet succinct.

3. Your docstrings are occasionally missing at the top of a script (e.g. scope.py, oaks.py).

2 Overall Assessment

You did a solid job overall, though with some mistakes here and there. Take care to avoid careless syntax errors, and to be clear about the underlying logic of your coding choices. These are easily correctable mistakes however, which should fall away naturally with practise.

Provisional Mark: 65%

Signed: Alexander Kier Christensen & Samraat Pawar

March 23, 2022