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: Chalita Chomkatekaew

1 Specific feedback

1.1 The Good (what you did well!)

1. Found all the core CMEE weekly directories in your parent directory.
2. Your organisation and code are generally neat.
3. Your Git repo size when I checked week 7 was about 2 MB – very 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. Very good job with the coding overall. Good attention to detail, with only minor errors, and minimal warnings.
5. Your Python is generally reasonably modular – this is excellent as this is how Python is meant to be! Good job remembering all the docstrings as well!
6. 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. `DataWrangTidy.R` threw an error because you appear to have forgotten to load the relevant library for the command `%>%` at the top of the script.

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

1. You had an overall readme file, as well as one within each week. The Readmes were clear, and generally to the point. Curiously, although you chose to include dependencies (e.g. required packages) in your Week3 readme, you did not do so for the other weeks, which would have been a nice touch both for good practise and for consistency. 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.

2. You had a project-wide .gitignore, which is good, but you can fine tune the exclusions further, for example with exclusions specific to certain weeks. You will likely find this useful: `https://www.gitignore.io`.

3. Commenting and docstrings could be improved – when you do comment a file thoroughly it is generally well done. However occasionally you miss out an explanatory docstring at the very top of a file (e.g. `apply2.R)`, `browse.R`), and you are inconsistent in whether you do or don't record, for example, the __author__ or __version__ in your scripts. This and commenting will improve with experience, as you will begin to get a feel of what is "common-knowledge" among programmers, and what stylistic idioms are your own and require explanation.

2 Overall Assessment

Overall a very good job. Minimal errors and aside from some minor inconsistencies, your documentation is solid. You have developed a good foundation on which to build with practise from here on out.

Provisional Mark: 75%

Signed: Alexander Kier Christensen & Samraat Pawar

March 23, 2022