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: Junyue Zhang

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 code and file structure is neatly and logically organised.
3. Your Git repo size when I checked week 7 was about 8 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, as well as one within each week. The Readmes were clear, and comprehensive, even including info like dependencies. Good work!
5. Excellent job with the coding overall. Good attention to detail, reasonable commenting and minimal warnings. Good job remembering all the docstrings and having only one error in all of your Python code.
6. You have made efforts where appropriate to write modular Python code – this is great Pythonic practise.
7. 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. `get_TreeHeight.py` throws a IndexError if no command line argument is supplied. This could have been caught and treated with a more informative error message.

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

1. 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.

2. You had a .gitignore throughout, to help you control which files were tracked with VC, which is good. Nonetheless, you could have added specific rules for different weeks (e.g. R-related exclusions for the R week, python for the python weeks etc). You will likely find this useful: `https://www.gitignore.io`.

2 Overall Assessment

A very good job overall. Essentially error-free code, reasonably well documented and only minor cosmetic criticisms beyond that. You appear to have built a solid foundation from which to progress further as a programmer.

Provisional Mark: 90%

Signed: Alexander Kier Christensen & Samraat Pawar

March 23, 2022