CMEE Coursework Feedback 2023/24

Note:

> - All script/code files, errors and other info mentioned below are in the weekly log/feedback files.
> - 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: *Zhongbin Hu*

The Good

- Excellent READMEs, well done!
- Repo of a good size, though you didn't need to push your HPC data files.
- *Most* of the required weekly folders were present.
- Your repo-level `.gitignore` file was sensible
- Really nice tuple unpacking solution for the list comprehension part of `dictionary.py`.
- You handled vectorisation (particularly the tricky `Vectorise2.R` task) very well.
- I think generally the code you write is good and fairly standard in terms of form, you have done well!

The Bad

- You entirely missed the Week 7 submission! This was a real shame. I cannot mark what is not there…
- You were also missing some files from Weeks 2 & 3, particularly `using_name.py` and some of the statistics examples.

- You lacked the `FirstBiblio.bib` file from your Week 1 directory. The `FirstExample.tex` file will not compile without it.
- `Ricker.R` was lacking all the code within it. I assume this was just an accidental deletion but it should have been checked.
- There were a few typos in some Week 3 scripts that caused errors (e.g. `reutrn` in `apply2.R`). You should have picked these up when you ran the scripts prior to submitting them.

The Ugly

(pedantic code style bits and other notes)

- It is good practice to check to see whether files are present before deleting them, for example `[ -e *.aux ] && rm *.aux`
- You also should include shebang lines at the top of *every* standalone script.
- Whilst you did have some docstrings, these should be present for every function and every script. They are invaluable when writing larger projects and coming to use them later.
- Some more commenting would be nice to aid in navigation of your code. I sometimes got lost when trying to make my way through your logic.

Mark

Provisional mark: *75*