

CMEE Masters: Computing Coursework Assessment

Note that:

All script/code errors and other info mentioned below are in the weekly log files

In the weekly feedback/assessments, please compare with the solutions whenever needed to see why I might have taken off points for a particular exercise/script or code file. We can then discuss these in your 1:1 post-assessment feedback session.

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

Student's Name: Katherine Bickerton

Overall Project workflow

Found all the expected weekly directories in your parent directory.

You had a .gitignore throughout, with meaningful exclusions specific to certain weeks – great. Later on you could have included more pattern exclusions. You will likely find this useful: <https://github.com/github/gitignore>.

You had an overall readme file, and then within each week, a readme with a detailed directory/file tree, presumably using the nifty `tree` command. OK, but you may not always want/need to do it in this detail. For example, no need to list files in Sandbox or Data. After all, the Code files would presumably contain sufficient info about what sorts of data are expected as input.

You had a nicely formatted readme file with a list of the weeks' content. Within each week you had a readme for each week – good. Glad you did not list files in `Data`. After all, each Code file would presumably contain sufficient info about what sorts of data are expected as input. Of equal or more importance is a description of what the overall project structure is and what the language and dependencies requirements are, which would be equally useful for a new user trying to understand and run your workflow. As you become a seasoned programmer, you will learn to make the readme file more informative yet succinct. Check out this resource: <https://github.com/jehna/readme-best-practices> Your Git repo size when I checked week 7 was about 4.24 MB — a good size. This suggests you did not keep unnecessary binary files under VC, and that you did not commit excessively. It could also mean that you did not commit enough, and/or somehow along the way lost parts of your git history — but I won't check these possibilities!

WEEK 1

Found directories Data, Results Sandbox, Code

Found 12 code files: ConcatenateTwoFiles.sh, CountLines.sh, variables.sh, CompileLaTeX.sh, tiff2png.sh, csvtospace.sh, FirstExample.tex, MyExampleScript.sh, FirstBiblio.bib, UnixPrac1.txt, tabtocsv.sh, boilerplate.sh

UnixPrac1.txt was fine. Each solution was described in a comment, great. You could have broken the description down into the key components of the unix command, but that's OK. Compare with the solutions, some of which you may find are more simple/compact, especially the last one!

csvtospace.sh was fine, but one addition you could have made to the script was to throw an error (with a message) if no input csv file was provided. In general, it is a good idea to add some input checks and return a meaningful message with error for utility files like this, especially in case somebody else uses it. Similar comment for ConcatenateTwoFiles.sh (running without two input files will not work), tabtocsv.sh and CompileLaTeX.sh, CountLines.sh, Variables.sh. But it's OK. No points deleted for this.

Points for this week: 100

WEEK 2

Found the Code, Sandbox, Data, Results directories

Found 19 code files: lc2.py, boilerplate.py, basic_csv.py, cfexercises2.py, dictionary.py, debugme.py, scope.py, cfexercises1.py, tuple.py, basic_io.py, lc1.py, oaks_debugme.py, oaks.py, loops.py, using_name.py, align_seqs.py, sysargv.py, control_flow.py, test_control_flow.py

Found no extra files; great!

lc1.py, lc2.py, dictionary.py, tuple.py were all fine. They could have given an somewhat better formatted output – Compare with the solutions on the repo; -0.5 pts each.

align_seqs.py was nicely implemented. You could have written it as a self-sufficient script that could also take external inputs optionally (though I did not ask for it). Compare with the solution.

All other scripts were fine.

Points for this week: 98

WEEK 3

Found directories Practicals, Code, Data, Results

Found 27 code files: browse.R, PP_Regress.R, maps.R, apply1.R, sample.R, run_get_TreeHeight.sh, Preallocate.R, boilerplate.R, TreeHeight.R, PP_Lattice.R, next.R, Ricker.R, Girko.R, Vectorize1.R, break.R, plotLin.R, basic_io.R, try.R, apply2.R, get_TreeHeight.R, TAutoCorr.R, Vectorize2.R, DataWrangTidy.R, DataWrang.R, TAutoCorr.tex, MyBars.R, control.R

Found the following extra files: TAutoCorr.pdf -0.5pt (put it in Results)

Vectorize1.R was fine.

Vectorize2.R was fine, nice job — also compare with the solution.

PP_Regress.R: good — also have a look at my solutions.

TAutoCorr.R was OK, but you should have shuffled the entire time series instead of the two equal sets separately – compare with the solution; does your solution permute every possible pair of successive years? -3pts

The report: nicely implemented. You could have plotted the data and histogram of the permuted correlation coefficients as well.

You did the Mapping extra credit: +2.5 pts.

Points for this week: 97 pts

WEEKS 4, 5 & 6

Not assessed, but happy you kept everything organized as much as possible.

WEEK 7

Found directories Code, Data, and Results

Found a README

Found 14 code files: regexs.py, profileme2.py, timeitme.py, blackbirds.py, profileme.py, fmr.R, profile_LV.py, using_os.py, LV1.py, DrawFW.py, Nets.R, run_fmr_R.py, LV2.py, run_LV.sh

MyFirstJupyterNb.ipynb was missing: -10 pts

using_os.py worked, and provided some meaningful output to screen – great. Compare with the solution as well.

blackbirds.py was nicely implemented. Do look at the solution as well.

You did the two LV* scripts with profiling. Good. However, profile_LV.py gave an error because `%run` does not work within a python script. Look at the solutions, which also cover the other LV challenges.

Points for this week: 90 pts

Overall Assessment

You did an very good job overall, including some extra credit Qs. You could have gone the extra mile in a few cases, maybe attempting a couple more extra credit questions.

But overall, tidy workflow and scripts, with very few errors. You showed a good eye for detail, including the formatting of outputs. I was impressed by your efforts to understand as many details of the programming languages and coding as possible. You clearly like coding!

Overall, given that this is the first time you have done programming in a mix UNIX, Python, & R, you did very well! In particular, you seem to on your way to becoming equally comfortable with Python and R.

It was a tough set of weeks, but I believe your hard work in them has given you a great start towards further training, a quantitative masters dissertation, and ultimately a career in quantitative biology!

Provisional Mark: 75

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.

We will discuss where you gained or lost marks, and what you could have improved further in your 1:1 post-assessment feedback session. To the extent possible, please come with questions about specific scripts based upon the feedback you have received. This may require you to compare your code with the solution code in many cases.

Signed: Samraat Pawar

January 18, 2019