

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: Rachel Bates

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://www.gitignore.io>.

You had a readme file with a list of the weeks' content, and then within each week, a readme for each week. The weekly Readmes were succinct and clear, including a description of what the overall project structure is and what the language and dependencies requirements are - great job. 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.

Your Git repo size when I checked week 7 was about 19.33 MB — a good size, suggesting 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, Sandbox, Code

Found 12 code files: ConcatenateTwoFiles.sh, Boilerplate.sh, CountLines.sh, variables.sh, CompileLaTeX.sh, tiff2png.sh, csvtospace.sh, FirstExample.tex, MyExampleScript.sh, FirstBiblio.bib, UnixPrac1.txt, tabtocsv.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, 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 21 code files: lc2.py, boilerplate.py, basic_csv.py, cfexercises2.py, align_seqs_better.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, align_seqs_fasta.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 better formatted output – Compare with the solutions on the repo; -1 pt each.

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

You did align_seqs_fasta.py and align_seqs_better.py correctly — so +5 extra credit pts.

All other scripts were fine – great job.

Points for this week: 100

WEEK 3

Found directories Practicals, Code, Data, Results

Found 30 code files: browse.R, PP_Regress.R, Vectorize2.py, apply1.R, sample.R, run_get_TreeHeight.sh, get_TreeHeight.py, Mapping.R, Vectorize.sh, boilerplate.R, TreeHeight.R, PP_Lattice.R, next.R, Girko.R, Vectorize1.R, break.R, plotLin.R, basic_io.R, Vectorize1.py, try.R, apply2.R, get_TreeHeight.R, TAutoCorr.R, Vectorize2.R, DataWrangTidy.R, preallocate.R, DataWrang.R, TAutoCorr.tex, MyBars.R, control.R

Vectorize1.R was fine.

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

PP_Regress.R: fine, apart from some commented out code lines (why leave them in?). Also have a look at my solution.

TAutoCorr.R was fine, again, except for commented out experimental code. However, compare with the solution for a different approach — does your solution capture every pair of successive years? -3pts

The report: Nicely done, including keeping it in a separate directory. Glad you plotted the histogram of the permuted correlation coefficients as well. You could also have plotted the correlation pattern itself. Some more interpretation of the results would have been nice.

You did the Mapping extra credit (good description of the data and potential biases) – +2.5 pts.

Points for this week: 99.5 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 19 code files: TestR.py, regexs.py, LV3.py, profileme2.py, timeitme.py, blackbirds.py, Nets.py, TestR.R, profileme.py, LV4.py, fmr.R, MyFirstJupyterNb.ipynb, using_os.py, LV1.py, DrawFW.py, Nets.R, run_fmr_R.py, run_LV.py, LV2.py

You had an Output directory in addition to Results, which made the project it slightly more confusing to run and inspect: -3pts.

run_LV.py gave an error: -5pts. Compare with solution. Otherwise the two LV* scripts with profiling were fine. You also did the LV3-4 extra credits, but compare with the solutions : +4pts

Also look at the solution, which also covers the other LV challenges.

using_os.py is OK, but compare with the solution. Also, the script could have provided some meaningful output to screen. -2.5pts

Nets.py: Good job, but avoid using additional packages unless absolutely necessary: +2.5pts

blackbirds.py was fine. Do look at the solution as well.

Points for this week: 96 pts

Overall Assessment

You did an excellent job overall, including many extra credit Qs.

Neat, clean project organization and code, and practically no errors. I was impressed by your efforts to understand as many details of the programming languages and coding as possible. You clearly like coding!

Overall, as this is the first time you have done programming in a heady mix of UNIX, Python, & R with a sprinkling of L^AT_EX and git, you did very well! In particular, you seem to have become quite comfortable with both Python and R – that’s great!

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: 86

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