

CMEE Masters: Computing Coursework Assessment

Note that:

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

In the weekly feedback/assessments, please compare with the solution 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: Oliver Tarrant

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. You could have included more pattern exclusions – You will likely find this useful: <https://github.com/github/gitignore>

You had a readme file with a list of the weeks' content within each week - using a nifty unix command ;) . Good, but you may not always want/need to do it in this detail (it was too much in week 7, for example). 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.

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 useful for a new user trying to understand and run your workflow. As you become a (more!) seasoned programmer, you will learn to make the readme file descriptions more informative yet succinct.

Your Git repo size when I checked week 7 was a decent 76.23 MB, 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, Results Sandbox, Code

Found 11 code files: ConcatenateTwoFiles.sh, CompileLatex.sh, CountLines.sh, variables.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, especially for the last problem.

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 26 code files: `lc2.py`, `boilerplate.py`, `LV3.py`, `basic_csv.py`, `timeitme.py`, `profileme.py`, `LV4.py`, `test_oaks.py`, `dictionary.py`, `LV1.py`, `debugme.py`, `scope.py`, `tuple.py`, `basic_io.py`, `lc1.py`, `oaks.py`, `run_LV.py`, `LV2.py`, `loops.py`, `using_name.py`, `cfexercises.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, well documented too. They could have given a slightly better-formatted output – compare with the solution on the repo.

`align_seqs.py` was fine. You could have written it as a self-sufficient script that could take external inputs optionally. Do compare with the solution.

You did `align_seqs_fasta.py` — so +2.5 extra credit points.

All other scripts were fine, with docstrings included.

Points for this week: 100

WEEK 3

Found directories Practicals, Code, Data, Results

Found 28 code files: browse.R, PP_Regress.R, Vectorize2.py, apply1.R, sample.R, TAutoCorr_Analysis.tex, run_get_TreeHeight.sh, get_TreeHeight.py, Mapping.R, boilerplate.R, TreeHeight.R, run_vectorize.sh, PP_Lattice.R, next.R, Ricker.R, Vectorize1.R, SQLinR.R, break.R, basic_io.R, TAutoCorrCode.R, Vectorize1.py, try.R, apply2.R, get_TreeHeight.R, TAutoCorr.R, Vectorize2.R, DataWrang.R, control.R

Found the following extra files: .Rhistory, TAutoCorr_Analysis.Rnw: -1 pt (.gitignore candidates)

The “extra” Rplots.pdf file is a R quirk — I am not actually deleting points for it, as it only happens when a R script is run externally from unix using **RScript**, and there is no easy solution for this.

Vectorize1.R was fine.

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

PP_Regress.R: good — also have a look at my solution for a different approach.

TAutoCorr.R was nicely done, knitr and all. Also compare with the solution for a slightly different approach for the simulation. The report: Nicely done. You could have plotted the histogram of the permuted correlation coefficients as well. The interpretation was good!

You did the DataWrangTidy and Mapping extra credit – +5 pts.

Plus you did Vectorize*.py scripts – +5 extra credit points.

Points for this week: 100 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 21 code files: MyFirstJupyterNB.ipynb, using_os.py, Nets.py, regexs.py, timeitme.py, LV4.py, DrawFW.py, TestR.R, LV3.py, Nets.R, LV2A.py, LV1.py, fmr.R, re4.py, run_LV.py, TestR.py, blackbirds.py, run_fmr.R.py, profileme.py, LV5.py, LV2.py.

Found the following extra file: MyFirstJupyterNb-checkpoint.ipynb (could have .gitignore-d it) ; -0.5pt

Nets.py, blackbirds.py gave syntax errors: -10pts

using_os.py worked, but compare with the solution. The code could have provided some meaningful output to screen. -1pt

Nets.py was fine other than the error.

blackbirds.py was also fine other than the error, but compare with the solution.

You did ALL the LV* scripts with profiling. Excellent! +10 extra credit points.

Points for this week: 100 pts

Overall Assessment

You did an excellent job overall, including almost all extra credit Qs!

Very few errors, but some silly ones at the end in week 7. You went just that extra mile in many cases. I also liked the fact that you annotated your scripts meaningfully. I was impressed by your efforts to understand as many details of the programming languages and programming as possible.

Overall, You delivered on most fronts, and if 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 hope it gave you an inkling of why and how you would/could use Python, R, UNIX, etc together or as and when required.

Provisional Grade

A*	88
A	
B	
C	
F	

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 Appendix of the SilBioComp document.

We will discuss where 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.

Signed: Samraat Pawar

January 31, 2018